University of Bedfordshire

Title: A Methodology For Contextual Recommendation Using Artificial Neural Networks

Name: Ghulam Mustafa

This is a digitised version of a dissertation submitted to the University of Bedfordshire.

It is available to view only.

This item is subject to copyright.

A Methodology For Contextual Recommendation Using Artificial Neural Networks

By

Ghulam Mustafa

Institute of Research in Applicable Computing

University of Bedfordshire

This dissertation is submitted for the degree of

Doctor of Philosophy

IRAC February 2018

I would like to dedicate this thesis to my loving dad Dr. Allah Yar for being an inspiration of my life and my mom Mrs. Ayesha Bibi for her unconditional, immeasurable and endless love. I also dedicate this work to my brothers, Dr. Muhammad Irfan, Dr. Arshad Mahmood and Muhammad Zubair, who support me on every step of my life and never left my side. I want to extend my dedications to my younger sisters Dr. Rakhshinda Iram and Dr. Salma Batool for their encouragement and believe on me. I also want to dedicate this work to my wife, Sadaf Hameed, whose selfless devotion and love give me this courage for my quest. The work is also dedicated to my little princess, Rehab, whose smile always motivates me to achieve my goals.

Declaration

I hereby declare that this thesis is my own work, except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

By
Ghulam Mustafa
February 2018

Acknowledgements

First, I would like to express my gratitude to my supervisor, Dr. Ingo Frommholz for his guidance at every step of my research. I will further extend my gratitude to my second supervisor Dr. Vitaly Schetinin for his suggestions on different stages of my research. I highly appreciate the suggestions and counseling throughout the degree, at every step, provided by Research Graduate School. I am also thankful to the management of IRAC for providing me different opportunities to participate in research activities.

I am thankful to the management of Autumn School for Information Retrieval and Information Foraging (ASIRF) for providing me the learning opportunity at Schloss Dagstuhl. I am also grateful to the management of Hasso Plattner institute, Germany for the privilege to present my work at Lernen Wissen, Daten, Analysen, (LWDA 2016).

Finally, I want to thank my family, especially my parents for their entire support. I want to thank my friend Dr. Adnan Nabeel Qureshi, for his support and friendship. I am also thankful to my lab fellows and friends for their entire support and friendship.

Abstract

Recommender systems are an advanced form of software applications, more specifically decision-support systems, that efficiently assist the users in finding items of their interest. Recommender systems have been applied to many domains from music to e-commerce, movies to software services delivery and tourism to news by exploiting available information to predict and provide recommendations to end user. The suggestions generated by recommender systems tend to narrow down the list of items which a user may overlook due to the huge variety of similar items or users' lack of experience in the particular domain of interest. While the performance of traditional recommender systems, which rely on relatively simpler information such as content and users' filters, is widely accepted, their predictive capability perfomrs poorly when local context of the user and situated actions have significant role in the final decision. Therefore, acceptance and incorporation of context of the user as a significant feature and development of recommender systems utilising the premise becomes an active area of research requiring further investigation of the underlying algorithms and methodology.

This thesis focuses on categorisation of contextual and non-contextual features within the domain of context-aware recommender system and their respective evaluation. Further, application of the Multilayer Perceptron Model (MLP) for generating predictions and ratings from the contextual and non-contextual features for contextual recommendations is presented with support from relevant literature and empirical evaluation. An evaluation of specifically employing artificial neural networks (ANNs) in the proposed methodology is also presented. The work emphasizes on both algorithms and methodology with three points of consideration: contextual features and ratings of particular items/movies are exploited in several representations to improve the accuracy of recommendation process using artificial neural networks (ANNs), context features are combined with user-features to further improve the accuracy of a context-aware recommender system and lastly, a combination of the item/movie features are investigated within the recommendation process. The proposed approach is evaluated on the LDOS-CoMoDa dataset and the results are compared with state-of-the-art approaches from relevant published literature.

The results show that the proposed approach can assist in dealing with context either as a standalone input to the recommendation process as well as in the form of categories that are formulated based on the dynamic and static nature of features to predict recommendations. However, we observed that combinations of contextual features perform better than the standalone contextual features. The LDOS-CoMoDa dataset provides a rich set of dynamic contextual features which can potentially dominanate the predictions during context-aware recommendation process. The contextual features are incorporated into the recommendation process after pre-processing and normalisation which is helpful to move geometrical biases towards the dimensions of the data vectors. Considering the LDOS-CoMoDa dataset, the impact of movies' features in recommendation process shows a performance accuracy of 66.5%, 74.1%, 66.5% and 75.1% with Principal Components Analysis (PCA), Support Vector Machines (SVM), Logistic Regression and ANNs, respectively. Similarly, the impact of users' features shows that ANNs have an average accuracy of >76% which is bettar than that of the SVM (<75%) and with Combination 2 (Actors, Genre, Director, Country), the ANNs accuracy is 77%.

Based on the results, we have shown that ANNs based approach outperforms SVM and PCA in improving the accuracy of contextual recommendation processes and, therefore, can play a vital role for context-aware recommender systems.

Table of contents

List of figures x

List of tables xiii

1 Introduction 1

1.1 Recommender Systems . . . 2

1.1.1 Value of Recommender systems in industry . . . 3

1.2 Motivation . . . 3

1.3 Aims and Objectives . . . 4

1.4 Contributions . . . 6

1.5 Publications . . . 6

1.6 Structure of the thesis . . . 7

2 Recommender Systems: State of the art 8

2.1 Introduction . . . 8

2.1.1 Recommender System Algorithms . . . 10

2.1.2 Collaborative Filtering . . . 11

2.1.3 Content Based Recommendations . . . 20

2.1.4 Hybrid Recommender Systems . . . 22

2.1.5 Comparison of different recommendation algorithms . . . 22

2.2 Context-Aware Recommender Systems . . . 25

2.2.1 Contextual Pre-Filtering . . . 30

2.2.2 Contextual Post Filtering . . . 30

2.2.3 Contextual Modeling . . . 31

2.2.4 Context-Aware Recommender System in Mobiles . . . 32

2.2.5 Multiverse recommendation and attribute aware recommender system 32

2.3 Evaluation Methodologies . . . 33

2.3.1 Offline Evaluation . . . 34

2.3.2 Online Evaluation 36
2.3.3 The role of Artificial Neural Network in Context-Aware Recommender System 36
2.4 Summary and Conclusion 40

3 Proposed Methodology and Experimental Set-up **41**
3.1 Proposed System Architecture 41
3.1.1 Data pre-processing, Normalisation and transformation 41
3.1.2 An Artificial Neural Networks Phase 44
3.2 Experimental Set-up 45
3.2.1 Dataset 46
3.2.2 Software and Hardware Resources 52
3.2.3 Evaluation Measures 54
3.2.4 Significance Test(Paired t Test) 55
3.2.5 Cross Validation 56
3.3 Summary and Conclusion 56

4 A Feature Selection Approach to Compare the Contextual, User's and Movie's Features **57**
4.1 Introduction 58
4.2 Comparison of different categories of features using proposed approach . . 61
4.3 Evaluation 65
4.3.1 Dataset 65
4.3.2 Results 65
4.3.3 Discussion 69
4.4 Summary and Conclusion 71

5 Contextual Recommendation using an ANN based approach **72**
5.1 Introduction 73
5.2 Overview of User Features, Movie Features and Contextual Features 73
5.3 Comparison of Proposed Methodology with different Machine Learning Models 75
5.4 Experiments 76
5.4.1 Impact of Movie's features in Contextual Recommendations 77
5.4.2 Impact of User features in Contextual Movie Recommendations using the proposed approach 84
5.4.3 The role of Location in contextual Recommendations 89

5.4.4 The role of Emotions in Contextual Recommendations 92
5.4.5 The Role of Mood in Contextual Recommendations 96
5.4.6 The role of Contextual Feature "Physical" in Contextual Recommendations 98
5.4.7 The role of Contextual Feature "Decision" in Contextual Recommendations 101
5.4.8 The Role of Interaction in Contextual Recommendations 103
5.4.9 The role of day-type in Contextual Recommendations 106
5.5 Summary and Conclusion 109

6 Conclusion and Future Work 112
6.1 Contribution Revisited 114
6.2 Limitations of proposed ANN based approach 116
6.3 Future Work 117

References 120

Appendix A ANNs Confusion Matrices 131

List of figures

2.1 A Traditional Recommendation Approach 10
2.2 Breakdown of recommendation algorithms used for recommender systems . 11
2.3 User-based CF versus Item-based CF: The two users are identified who share the interests in user based CF method by calculating the similarity between the users and recommendations made, while in the item-based CF method, similarity is calculated between the items to recommend an item to the user [91]. 13
2.4 Systematic view of Restricted Boltzmann Machine 19
2.5 Results for MovieLens 100K Dataset 25
2.6 Comparison of context-aware recommender system techniques 31
2.7 Evaluation scheme for recommender system using confusion matrix 35
2.8 Basic structure of Artificial Neural Network 37
2.9 Architecture of Artificial Neural Network. The inputs i_1–i_n to ANNs are multiplied to respective weights W. A weighted sum J is calculated and passed to the transfer function which produce output as y as classification results such as y = {0,1} 38

3.1 Proposed System Architecture 45

4.1 Architecture of the proposed ANN approach for rating prediction 62
4.2 Distribution of different features from LDOS-CoMoDa Dataset 63
4.3 ROC Curve of Dynamic Representation 67

5.1 A general Overview of the contextual recommendation process 76

A.1 ANNs results for Location with combination C2 in table 5.9 with accuracy of 67.7% and precision 98.4%. 131
A.2 Confusion matrix for Combination number 1 of contextual features given in table 5.23 132

A.3 Confusion matrix for Combination number 2 of contextual features given in table 5.23 132
A.4 Confusion matrix for Combination number 3 of contextual features given in table 5.23 133
A.5 Confusion matrix for Combination number 4 of contextual features given in table 5.23 133
A.6 Confusion matrix for Combination number 5 of contextual features given in table 5.23 134
A.7 Confusion matrix for Combination number 6 of contextual features given in table 5.23 134
A.8 Confusion matrix for Combination number 7 of contextual features given in table 5.23 135
A.9 Confusion matrix for Combination number 8 of contextual features given in table 5.23 135
A.10 Confusion matrix for Combination number 9 of contextual features given in table 5.23 136
A.11 Confusion matrix for Combination number 10 of contextual features given in table 5.23 136
A.12 Confusion matrix for Combination number 11 of contextual features given in table 5.23 137
A.13 Confusion matrix for Combination number 12 of contextual features given in table 5.23 137
A.14 Confusion matrix for Combination number 13 of contextual features given in table 5.23 138
A.15 Confusion matrix for Combination number 14 of contextual features given in table 5.23 138
A.16 Confusion matrix for Combination number 15 of contextual features given in table 5.23 139
A.17 Confusion matrix for Combination number 16 of contextual features given in table 5.23 139
A.18 Confusion matrix for Combination number 17 of contextual features given in table 5.23 140
A.19 Confusion matrix for Combination number 18 of contextual features given in table 5.23 140
A.20 Confusion matrix for Combination number 19 of contextual features given in table 5.23 141

A.21 Confusion matrix for Combination number 20 of contextual features given in table 5.23 141
A.22 Confusion matrix for Combination number 21 of contextual features given in table 5.23 142
A.23 Confusion matrix for Combination number 22 of contextual features given in table 5.23 142
A.24 Confusion matrix for Combination number 23 of contextual features given in table 5.23 143
A.25 Confusion matrix for Combination number 24 of contextual features given in table 5.23 143
A.26 Confusion matrix for Combination number 25 of contextual features given in table 5.23 144
A.27 Confusion matrix for Combination number 26 of contextual features given in table 5.23 144
A.28 Confusion matrix for Combination number 28 of contextual features given in table 5.23 145
A.29 Confusion matrix for Combination number 29 of contextual features given in table 5.23 145
A.30 Confusion matrix for Combination number 30 of contextual features given in table 5.23 146
A.31 Confusion matrix for Combination number 31 of contextual features given in table 5.23 146
A.32 Confusion matrix for Combination number 32 of contextual features given in table 5.23 147
A.33 Confusion matrix for Combination number 33 of contextual features given in table 5.23 147
A.34 Confusion matrix for Combination number 34 of contextual features given in table 5.23 148

List of tables

2.1 Clustering Example 16
2.2 Features of items that can be used for content based approach 21
2.3 Different methods used for hybrid recommender systems 23
2.4 Overview of the techniques used for the recommender systems 24
2.5 An overview of some available recommender systems 29
2.6 Some Context-aware Recommender Systems for mobile devices 33

3.1 Example dataset for data preprocessing and transformation 42
3.2 Example dataset for data preprocessing and transformation 43
3.3 Rating Transformation into binary 43
3.4 Example data after preprocessing and transformation 43
3.5 Abstract View of User Features in LDOS-CoMoDa Dataset 47
3.6 Abstract View of Movie Features in LDOS-CoMoDa Dataset 48
3.7 Abstract View of contextual Features in LDOS-CoMoDa Dataset 49
3.8 List of Features in abstract form from LDOS-CoMoDa dataset 50
3.9 Contextual Features from LDOS-CoMoDa and their assigned numeric values 51
3.10 Binary Transformation of ratings in LDOS-CoMoDa dataset 53
3.11 Proposed Evaluation Method 55

4.1 Representations and Features from LDOS-CoMoDa dataset 64
4.2 Basic Statistics of LDOS-CoMoDa 65
4.3 Sampling from ANN 66
4.4 Performance of different features using ANN and MF (where reported in the literature) 67
4.5 Dynamic Contextual variables from LDOS-CoMoDa 69
4.6 Performance of contextual dynamic attributes 70

5.1 List of Features from LDOS-CoMoDa dataset 75

5.2 The comparison of different machine learning algorithms for contextual recommendations from Movie features . . . 78
5.3 The comparison of different machine learning algorithms for contextual recommendations from Movie features . . . 79
5.4 The comparison of different movie features for contextual recommendations using the elimination technique. . . . 81
5.5 The evaluation and comparison of the different Movie features when incorporating with the context to generate recommendations . . . 83
5.6 The comparison of Impact of overall User Features for contextual Recommendations . . . 85
5.7 The comparison of Impact of individual User Features for contextual Recommendations . . . 87
5.8 The evaluation of the different combinations of user features . . . 88
5.9 The evaluation of the different combinations of user features . . . 90
5.10 The evaluation of the different combinations of user features with location . 91
5.11 Results of contextual feature Emotions . . . 94
5.12 The evaluation of the different combinations of user features with Emotions 95
5.13 Accuracy of contextual feature mood in contextual recommendation process using SVM and ANNs . . . 97
5.14 The accuracy of different combinations of user features with contextual feature mood and a selected conditional combination of movie features . . . 98
5.15 Accuracy of contextual feature *Physical* in contextual recommendation process using SVM and ANNs . . . 99
5.16 The accuracy of different combinations of user features with contextual feature *physical* and a selected conditional combination of movie features . 100
5.17 Accuracy of contextual feature *Decision* in contextual recommendation process using SVM and ANNs . . . 102
5.18 The accuracy of different combinations of user features with contextual feature *Decision* and a selected conditional combination of movie features . 103
5.19 Accuracy of contextual feature *Interaction* in contextual recommendation process using SVM and ANNs . . . 104
5.20 The accuracy of different combinations of user features with contextual feature *Interaction* and a selected conditional combination of movie features 105
5.21 Accuracy of contextual feature $day-type$ in contextual recommendation process using SVM and ANNs . . . 106

5.22 The accuracy of different combinations of user features with contextual feature $day-type$ and a selected conditional combination of movie features 108
5.23 Summary of individual contextual features with all user features and selected combination (Comb-2) of the movie fearures 108
5.24 The evaluation and comparison of the different contextual features when incorporated into the recommendation process 110

Abbreviations

ANNs	Artificial Neural Networks
CARs	Context-aware recommender system
CF	Collaborative Filtering
DBN	Deep Belief Networks
DMO	Destination Marketing Organizations
DNN	Deep Neural Network
FN	False Negative
FP	False Positive
GMDH	Group Method for Data Handling
GFF	General Factorization Framework
HCI	Human Computer Interaction
MF	Matrix Factorization
MLP	Multilayer Perceptron
NB	Naive Bayes
PCA	Principle Component Analysis
PoP	Principle of Polyrepresentation
RBM	Restricted Boltzmann Machine
RMSE	Root Mean Square Error
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency
TN	True Negative
TP	True Positive

List of variables

A	User Matrix in Matrix Factorization
A^T	Transpose of User Matrix in Matrix Factorization
B	Item Matrix in Matrix Factorization
C	Contribution
D	Set of Contexts
d	context from set of contexts
$e^{-\Sigma}$	errors
F	Utility function
f	Utility function of a particular item
g	contents
H	Hypothesis
I	Set of items
i´	Chosen item from the list of items
J	Weighted sum§
j	Weighted sum to transfer function in RBM
K	Content based profile
M	Rating Matrix in Matrix Factorization
N	Total sample value
n	Total number of observations in t-test
net_j	Net input in RBM
O	Objectives
o_j	activation in RBM
ϕ	Activation Function in RBM
θ_j	Threshold in RBM
P	Prediction function in matrix factorization
q	Item in the neighbourhood of item i
R	Recommended Items
r̂	Actual rating in RMSE
$\bar{r}$	Rating for a particulkar item by the user in Matrix Factorization
r̂	Predicted rating in RMSE
s	Standard deviation in t-test
T	Set of contents of documents
t	T-test
U	Set of users

u	user belong to the set of users
v	User in the neighbourhood of user u
W	Allocated weights in restricted boltzmann machine
x	Inputs to restricted boltzmann machine
Y	Rating matrix
y	Particular rating by user for an item
ŷ	Average rating by user
y^*	Predicted rating
z	Sample in T-test

Chapter 1

Introduction

The massive increase of structured and unstructured data available on the internet introduced the concept of Big Data that is difficult to process using traditional data processing techniques [87]. This abundance of uncategorized data on the internet makes it difficult to find useful information and creates the problem of information overload [142]. In order to handle such problems two major internet technologies, information search retrieval, and recommendations, have been developed over the last decade [28]. Recommendations are a growing paradigm which automatically presents and assist the user with what he/she is looking to search for from a huge amount of information.This paradigm is known as a recommender system. Recommender systems emerged as an independent research area during the 1990s, when researchers and practitioners started focusing on the problems of recommendations that explicitly rely on the ratings provided by the user as a way to capture user's preferences for different items [4]. These user preferences further help to specify the initial ratings for items. These ratings can be further used to recommend items to different users.

The main focus of this thesis is on context-aware recommender systems. We study two major issues of context-aware recommender systems: the role of non-contextual features and the incorporation of relevant contextual features into context-aware recommender systems. We cross compare the different contextual and non-contextual features that remain static and dynamic in nature and incorporate these features into context-aware recommender systems based on detailed analysis.

In this chapter, we present a general overview of recommender systems, then we discuss the value of recommender systems in the industry. We further describe the motivation of our proposed research followed by the aims and objectives of the proposed research. Then we present our research contribution, list of publications and structure of the thesis.

1.1 Recommender Systems

Recommender systems are software applications that are used to recommend a product or service with the aim of optimizing some user-oriented objectives in the light of inherent uncertainty concerning users and content [15, 90]. Traditionally, recommender systems have been useful in assisting users when they search in large information spaces such as collections of products (movies, books, music CDs), documents (news articles, medical texts, Wikipedia articles), or users for matchmaking (dating services, online game players/teams, consumer-to-consumer marketplaces) [3]. A recommender system can be defined as a decision-making approach for users in complex information environments [108]. It is an advanced application of software that assists users to search through records of knowledge according to their interests and preferences that users expressed in the form of ratings. The recommendations use these ratings to predict what a user will like in the future.

The ratings can be collected explicitly in the form of feedback based on the experience of something about the user or implicitly by observing the search behaviour of the user. For example, in explicit feedback on Amazon, a user explicitly rates an item after buying it based on his/her experience. In comparison, the implicit feedback clicked or search behavior of the user is observed to associate the preferences of the user with its profile. The focus of the proposed research is more on the explicit feedback than the implicit feedback due to the reason the user can provide explicitly after experiencing something, for example, a user can rate a movie after watching it. Explicit feedback provides more reliable data as it does not involve extracting preferences from actions of the users as implicit feedback does. Explicit feedback also brings transparency to the recommendation process that results in a better recommendation quality [27].

Traditional recommender systems have performed well in many cases, however, many recommender systems do not consider some useful information such as location and time [129]. This additional information is known as contextual information in recommender systems. Nowadays, the context has great importance in the domain of recommender systems. For example, the music that a user would like to listen to at a party will be completely different from the music that a user will prefer on a date. The recommender systems that use such contextual information in traditional recommendation processes are called context-aware recommender systems [6].

1.1.1 Value of Recommender systems in industry

Recommender systems have become very well known in recent years. Companies such as Amazon and eBay have developed a large number of products to meet different needs of customers. An increasing number of options are available to customers in the era of E-commerce. Thus, in this new level of customization, in order to find what they really need, customers must process a large amount of information provided by businesses [46].

The market value of recommendations in the industry is vital in the domain of e-commerce, news, services delivery and many others. According to research on Netflix[1], 2/3 of the movies watched on their platform are recommended to the users. Similarly, 35% of sales at Amazon[2] comes from recommendations that are recommended to the users and users have shown a great interest. At the same time, recommendation generates 38% more clicks reported by Google News[3]. Similarly, at Choice stream[4], a very famous music platform, 28% people would like to buy more music they liked from the recommendations [66].

1.2 Motivation

Traditionally recommender systems focus on recommending the most relevant items to the users or the most appropriate users to the items [10]. While traditional recommendation approaches have performed well in many applications [80], in a number of other applications and contexts, such as location and time-based service recommender systems and travel recommendations, it may not be sufficient to consider only users and items [141]. It is also important to incorporate additional contextual information into the recommendation process [80].

A context can be defined as a "dynamic set of factors that further describe the state of a user at the moment of user's experience" [51]. Nowadays, context-aware recommender systems have emerged as a very popular field in many applications such as movie, music and

[1] www.netflix.com
[2] www.amazon.com
[3] https://news.google.com/
[4] http://www.choicestream.com/

mobile recommendations, services for learning, travel, and tourism, shopping assistance and multimedia [48, 82]. Most of the context-aware recommender system approaches assume that the contextual information does not change significantly and remains static (for example, the cast of the movie is static in nature and does not change with the time), but some dynamic contextualization approaches have also been proposed.

Although recommender systems are gaining more and more popularity these days, there is still a long way to go. In a recommender system, each user must rate a certain number of items so that the system can learn the preferences of the user and predict the reliable recommendations for the user [32]. Typically, users show their preferences by rating only a small number of items that are available which make the dataset sparse. In context-aware recommender systems, the issue is usually resolved by applying latent factor methods such as matrix factorization. However, in the domain of context-aware recommender systems, for the different type of datasets, Matrix Factorization methods are not always that effective. Different approaches of context-aware recommender systems can be categorized by the contextual factors they take into consideration [109]. Many approaches assume that the contextual information does not change significantly and remains static. This assumption is made in most of the cases, while some recent research has been proposed for dynamic contextualization [70].

Over time, different approaches to dealing with the challenges of context-aware recommender systems have been developed [150], and machine learning algorithms are used to develop models and find patterns based on training data. Most models are based on using a clustering technique for identification of a user based on a test set. Some well-known model-based techniques are Clustering, Association Rules, Matrix Factorization, Restricted Boltzmann Machines and others [65, 83, 89, 148]. In our approach, we use Artificial Neural Networks (ANNs), which have not been used in detail for contextual recommendations and have the potential to outperform most techniques used for traditional and contextual recommendations.

1.3 Aims and Objectives

The aim of the study is to improve the effectiveness of a recommender system by incorporating the additional contextual information using Artificial Neural Networks (ANNs). This will be further done by adopting a multilayer perceptron model for rating prediction and contextual recommendations. In order to achieve the aim of the study my objectives are as follows:

O1 Investigate and analyze the existing recommendation techniques for different recommender systems from the literature (Chapter 2).

O2 Analyze and evaluate different feature sets available for contextual recommendations (Chapter 4).

O3 Analyze and evaluate performance by defining a minimum contextual attribute subset which can generate more accurate contextual recommendations (Chapter 5).

O4 Develop and evaluate a novel conceptual methodology to include and integrate both contextual and non-contextual features into the contextual recommendation process (Chapter 3, Chapter 4, Chapter 5).

The hypothesis, thus formulated for this research, can be defined as:

H0 There will be no improvement in accuracy between the traditional contextual recommendation approach and contextual recommendations through proposed ANNs based methododology (Section 5.4.4).

H1 The accuracy of the recommendation process can be further improved by incorporating the combinations of contextual and non-contextual features in the recommendation process using proposed ANNs methodology (Section 5.4.9).

Since it is an open problem to incorporate the contextual data with a traditional recommender system and improve the performance of the system by generating more accurate recommendations, the proposed research is highly motivated in this domain. In order to achieve the objectives of this research, we further emphasize on comparing contextual and non-contextual features so that the relevant contextual features could be identified to incorporate into recommender systems. For this purpose, we will cross compare the different contextual and non-contextual features that remain static and dynamic in nature and incorporate these features into a context-aware recommender system based on detail analysis. Once we have identified the relevant features that can play a vital role in the development of a context-aware recommender system, we will assume them to generate recommendations following the proposed framework. The results will be evaluated as we mentioned in research objectives.

1.4 Contributions

The key contribution of my research can be summarized as follows:

C1 Categorisation of contextual and non-contextual features in context-aware recommender system and evaluation thereof (Chapter 4).

C2 Applying and evaluating the Multilayer Perceptron Model (MLP) for rating prediction. MLP can predict ratings from the set of initial ratings while provided with the contextual and non-contextual features for contextual recommendations (Chapter 4, Chapter 5).

C3 Specifying and evaluating the applications of ANNs in contextual recommendation with the help of proposed methodology. Evaluating and integrating the contextual features along with non-contextual features in recommendation process (Chapter 3, Chapter 4, Chapter 5).

1.5 Publications

- Ghulam Mustafa and Ingo Frommholz. Comparing contextual and non-contextual features in ANNs for movie rating prediction. Lernen, Wissen, Daten, Analysen (LWDA 2016), pages 361-372, 2016.

- Ghulam Mustafa and Ingo Frommholz. Performance Comparison of top N Recommendation Algorithms.The IEEE Fourth International Conference on Future Generation

Communication Technologies (IEEE FGCT 2015), pages 100-105, 2015.

1.6 Structure of the thesis

The structure of the thesis is organized as follows:

Chapter 1: We introduce traditional recommender systems and describe our motivation followed by the aims and objectives. We present our contribution and provide a list of publications.

Chapter 2: We provide a detailed review of techniques and algorithms used for the recommender system and context-aware recommender systems. We also present evaluation measures for context-aware recommender systems. We also provide our motivation towards the use of artificial neural networks for contextual recommendations.

Chapter 3: We introduce our proposed methodology and our test collections. We describe different phases of our proposed methodology and explained our experimental setup.

Chapter 4: We present an ANN-based approach to compare the different contextual, user's and item's features. We explained how we can identify the relevant features using ANNs.

Chapter 5: We present our experiment on our test collection in detail. We also describe the role of different user-features, movie-features, and contextual features. We also present the comparison of our results with some machine learning techniques as well as with the results reported in the literature.

Chapter 6: We present overall conclusion and findings of our research. We also present our contribution as well as discussed future work.

Chapter 2

Recommender Systems: State of the art

In this chapter, we have conducted a review of the state of the art in the domain of recommender systems. Particularly, we have revised the related work on recommender systems and described the motivation towards a context-aware recommender system. We study different algorithms and techniques used for recommender system and their advantages as well as shortcomings to find new perspectives related to proposed research. In this chapter, we further present a review of context-aware recommender systems. We describe the concepts and methods of context-aware recommender system as well as explain how a context-aware recommender system is different from the traditional recommendation approaches. We also present the different methods that can be used to evaluate the accuracy and performance of context-aware recommender systems. We further explain the Artificial Neural Network (ANNs) based approaches that can be used for machine learning problems and their potential role in context-aware recommender systems. A detailed description of different context-aware recommendation techniques and an introduction to ANNs can be found in this chapter. This chapter referred to the objective O1 that focus on investigation and analysis of recommendation techniques for the domain of recommender system.

2.1 Introduction

Recommender systems are the applications of software that can provide suggestions for users about particular lists of products such as movie, books or services in a personalized manner (in the case of e-commerce recommender systems) [11, 79]. Recommender systems have been very useful in assisting users in the scenarios of information overload that makes the exploration and selection of items from large information space a difficult task. Recommender systems offer a personalized suggestive assistance in form of discovery as an effective way of improving revenue on many e-commerce and media streaming platforms such as Amazon, Netflix, Youtube [1] and many others by increasing user satisfaction [122].

Basically, recommender systems are built on the theories, algorithms, and technologies from different domains such as Information Retrieval (IR), Artificial Intelligence (AI), Machine Learning (ML), Human-Computer Interaction (HCI), E-commerce Marketing [7]. Recommender systems remain an emerging and active research topic that has attracted more attention in the last decade.

A recommender system attempts to estimate/predict a rating function R on the basis of an initial set of ratings. The rating function R can be specified with the help of following Equation 2.1.

$$R = User \times Item \rightarrow Rating \tag{2.1}$$

Rating, in above formula, can be defined as an order set to capture preferences of the users, while users and items are the domains of users and items. A recommender system recommends the items that are highest rated, for the user once the rating function R is defined for the user and item space [73]. In the following Figure 2.1, an approach following the traditional method of a two-dimensional recommender system is given.

The basic recommender system problem is to estimate or predict a utility function that can help to predict how a user will like an item. In the utility function, U is represented as set of users $U := \{users\}$, I is represented as a set of recommendable items $I := \{recommendableitems\}$ where $F := utilityfunction$, then

$$F = U \times I \rightarrow R \tag{2.2}$$

where $R := \{recommendeditems\}$ and for each user u, we want to choose the items $\acute{i}$ that maximize f, as shown in the following Equation 2.3.

$$u \in U \qquad \acute{i}_u = argmax_f F(u,i) \tag{2.3}$$

The traditional paradigm of recommender system has three main aspects: user, item, and rating. Rating, in this case, represents the feedback that a user gives for a specific item.

[1] www.youtube.com

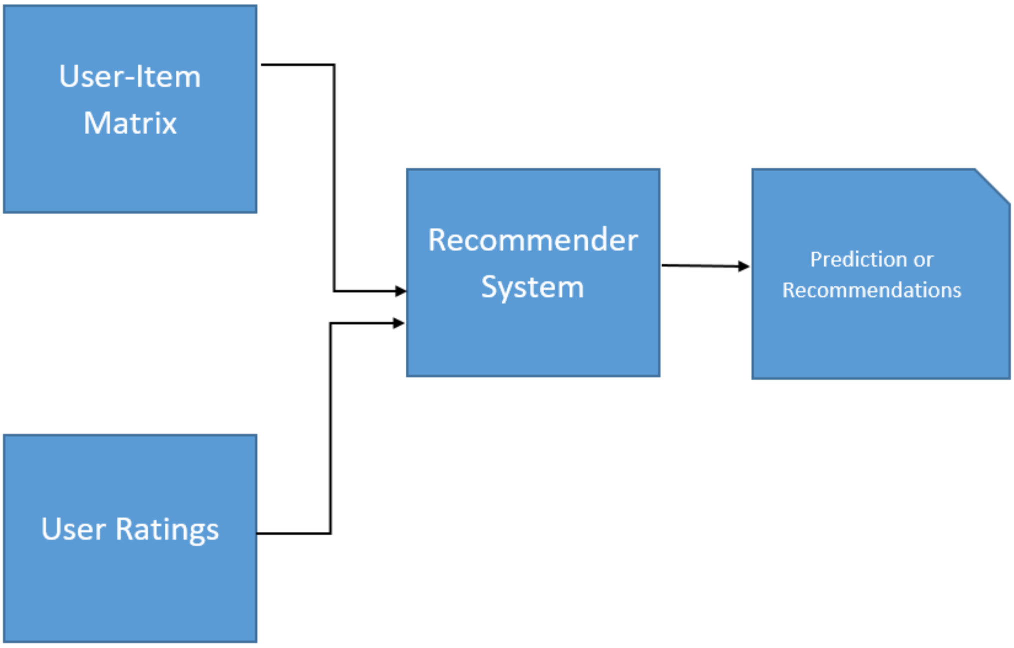

Fig. 2.1 A Traditional Recommendation Approach

Ratings can be in different forms; implicit and explicit. Ratings can be defined as explicit when a user rates an item to share shopping experience. Ratings can be defined as implicit when a user buys an item and the information comes in the form of user logs. Rating can be in the form of a numeric value on a multi-point scale e.g. 1 to 5 or can be in a binary form such as yes/no format [118].

One of the most studied cases in the domain of recommender systems is movie recommendation for Netflix in which the recommendation task is to suggest the movies to the users from a large list of movies [19]. The suggestions or recommendations are made based on the preferences that users provide in form of initial ratings of the movies that they have watched. However, there are many other examples in which the typical conditions of movie recommendation do not apply such as book recommendations where the contents of the books are used for recommendation process.

2.1.1 Recommender System Algorithms

Recommender system algorithms can be categorized based on the recommendation task and the nature of feedback provided by the users. In the literature, a common criterion based on how the different algorithms use the user preferences with respect to the items is used to divide the algorithms into three main categories which are known as Collaborative Filtering, Content-Based Recommendations, and Hybrid Recommender Systems. Each of the algorithms is used based on the recommendation task and the user preferences for the particular items.

In Figure 2.2, the categorization of different recommendation algorithms is presented in the form of a flow diagram.

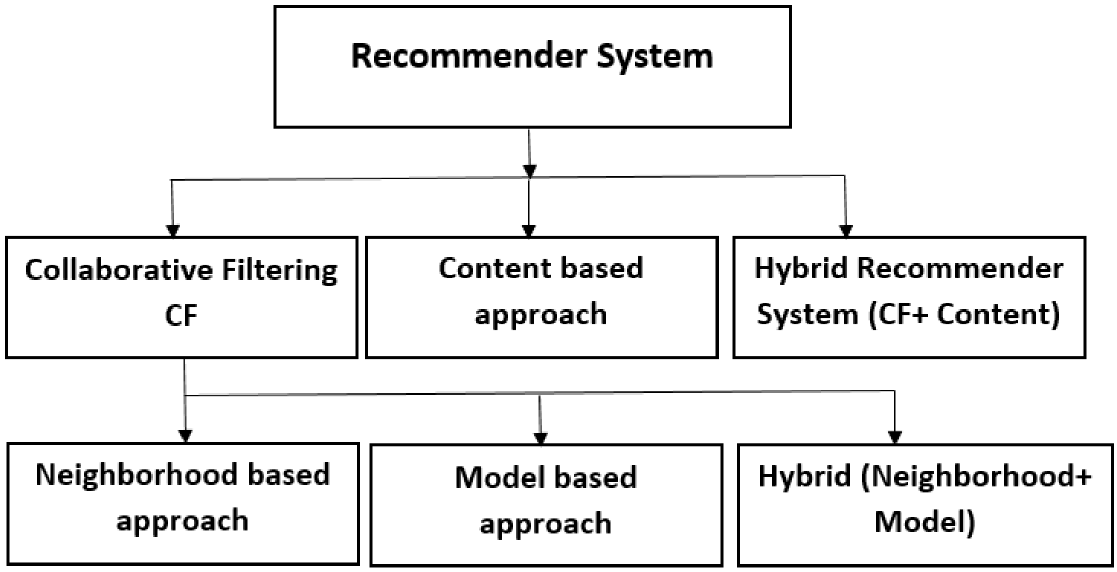

Fig. 2.2 Breakdown of recommendation algorithms used for recommender systems

We further proceed to review above mentioned different categories of recommendation algorithms and discuss the purpose, advantages, and weakness of each category.

2.1.2 Collaborative Filtering

Collaborative Filtering (CF) algorithms are mostly used to design a recommender system. In Collaborative Filtering (CF), large amounts of information on the behaviour, activities, and preferences of the user are collected and analysed to further predict what a user likes based on the similarity to the other users [109, 146]. The users are identified from the list of users with similar preferences by matching the ratings of the users, that they provide initially, against other users. The system recommends the items to similar users that are rated high within the domain of similar users but not rated by the specific user [140].

The basic methods of collaborative filtering involve the process of identification of users with similar interests and calculate the ratings from the same users in order to calculate the predictions for an active user [12]. Collaborative filtering uses the initial rating data that comes from different users for many items, as a base to predict the missing ratings and create a top-N recommendation of the most popular items as a list for a given user which is also known as an active user. In collaborative filtering, ratings are stored in a matrix R for users U and items I [55]. In typical CF a very small fraction of ratings is known while the values are missing in many cells of the matrix R. At this stage, a number of different algorithms operate on ratings within a specific scale. Collaborative Filtering algorithms are classified into two types, known as Memory-Based Collaborative Filtering and Model-Based Collaborative Filtering.

Memory-Based Collaborative Filtering:

In memory-based CF, initial rating data of the user is used for computing the similarity between the users and items, which can be further used for the recommendation process [135]. Since there is no learning phase involved in this method, memory-based CF algorithms are regarded as easy to implement and incorporation of new data is possible. The most well known memory-based CF is the neighborhood methods which are further divided into two types [49] *user-based collaborative filtering* and *item-based collaborative filtering*.

In ***user-based CF***, each user is identified based on the opinion that is expressed for some items where the opinion of the user provides the rating for a target user. A similarity function is used to find the similarity between the identified user and the other users in the neighborhood in order to predict the missing ratings. Once the missing ratings are specified, these are passed to the recommender system to recommend the items [115]. In ***item-based CF***, set of users who rated the target item i and identify the other items (neighbors) were rated by the users set A similarity function is used to find the similarity between the items in the neighborhood items to predict the missing ratings that can be used in the recommendation process.

A comparison between user-based CF and item-based CF is presented with the help of Figure 2.3, in which four different fruits are recommended to three users with the help of both aforementioned techniques. The two users are identified who share the interests in user-based CF method by calculating the similarity between the users and recommendations made, while in the item-based CF method, the similarity is calculated between the items to recommend an item to the user.

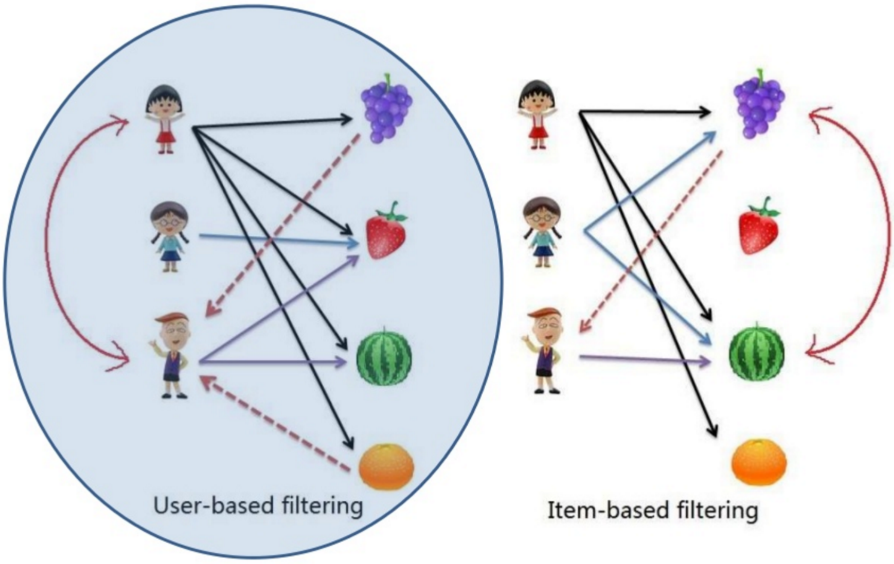

Fig. 2.3 User-based CF versus Item-based CF: The two users are identified who share the interests in user based CF method by calculating the similarity between the users and recommendations made, while in the item-based CF method, similarity is calculated between the items to recommend an item to the user [91].

In neighborhood-based algorithms [104], similarity is calculated between two users as target user U, rating matrix Y (initial rating matrix) where $y_{v,i}$ is the rating by user v for item i, $\hat{y}_u$ is average rating by user u and I_{uv} is the set of items rated by user u and v. A similarity Pearson's r correlation $sim(u,v)$ between users u and v can be represented as:

$$sim(u,v) = \frac{\sum_{i \in I_{uv}} (y_{u,i} - \hat{y}_u)(y_{u,i} - \hat{y}_u)}{\sqrt{\sum_{i \in I_{uv}} (y_{u,i} - \hat{y}_u)^2 \sum_{i \in I_{uv}} (y_{u,i} - \hat{y}_u)^2}}. \tag{2.4}$$

where the i in I sums over the items that are rated by both user u and user v.

In this case the predicted rating can be described as $y^*(u,i)$

$$y^*(u,i) = \hat{y}_u + \frac{\sum_{q \in I_{y*q \neq 0}} sim(v_q, u)(y_{v_q,i} - \hat{y}_{v_q})}{\sum_{q \in I_{y*q \neq 0}} |sim(v_q, u)|} \tag{2.5}$$

Item-based collaborative filtering can be further modified into the *popular items method* and the *random item method* to get better accuracy in the recommendation process. In the popular items method recommendations are made based on the popularity of the items in the rating matrix [94]. In the random item, method recommendations are made randomly from the real rating data. In item-based CF, the similarity is calculated between two items for target item i and q using the following equation.

$$sim(i,q) = \frac{\sum_{u \in I_{iq}} (y_{u,i} - \hat{y}_i)(y_{u,q} - \hat{y}_q)}{\sqrt{\sum_{u \in I_{iq}} (y_{u,i} - \hat{y}_i)^2 \sum_{u \in I_{iq}} (y_{u,q} - \hat{y}_q)^2}} \tag{2.6}$$

The predicted rating $y^*(u,i)$ can be described by the following equation.

$$y^*(u,i) = \hat{y}_i + \frac{\sum_{v \in I_{y*u \neq 0}} sim(i,q_v)(y_{u,q_v} - \hat{y}_{q_v})}{\sum_{v \in I_{y*u \neq 0}} |sim(i,q_u)|}. \tag{2.7}$$

The neighborhood-based algorithm in collaborative filtering is used to calculate the similarity between two users and items that can be further used to produce the prediction for the user or items. In this approach, computing similarity is important very important that can be computed between items or users where multiple methods such as Pearson correlation between users and items and vector cosine-based similarity are used for this purpose [92]. The similarity computing technique, cosine similarity for item-based collaborative filtering can be represented with following Equation 2.8:

$$cos(i,j) = \frac{\sum_{u \in I_{ij}} y_{u,i} y_{u,j}}{\sqrt{\sum_{u \in I_{ij}} y_{u,i}^2 \sum_{u \in I_{ij}} y_{u,j}^2}} \tag{2.8}$$

The Movielens recommender system is an example to test the performance of the different user-based, item-based, and model-based recommendation algorithms. The Movielens[2] dataset is available publicly for research purposes.

[2] www.http://grouplens.org/datasets/

Model-Based Collaborative Filtering:

In model-based Collaborative Filtering methods, machine learning algorithms are used to develop models and find patterns based on training data. Machine learning algorithms are generally based upon the notion of finding trends, patterns or anomalies in a set of data, and could be beneficially used for analyzing large and complex datasets [139]. This technology could either be used for grouping certain data points together, finding anomalies in data or for predicting an attribute out of several known parameters, based on trends and patterns found in the training data [125]. Predicting an unknown attribute of an object based on other known attributes and information about it is what is generally known as supervised machine learning. Supervised learning is mainly used to either classify an object into one of several predefined categories, such as vehicle type or a binary yes/no answer or in estimating a continuous attribute not bound to any predefined classes, for example, the upcoming petrol price, in what is generally known as regression analysis [26].

In model-based methods, algorithms depend on the learning phase, unlike in memory-based CF. This technique is commonly used to predict the real data with the help of many model-based CF algorithms. These methods are highly inspired by the machine learning techniques such as ANNs. In this method, most of the models are based on using a clustering technique for identification of a user based on a test set. Model-based CF algorithms include techniques such as Clustering [121], Association Rules [114], Restricted Boltzmann Machines [134], Principle Component Analysis (PCA) [8], Matrix Factorization [72], and many others.

One simple model-based CF modes to make recommendations is **Naive Bayes** (NB) [119]. In NB, we assume that features are independent of each other. When we apply NB to recommender systems, a similar assumption is used. Several researchers conduct experiments to exploit the potential of NB in recommender systems. In the work of [143], NB is used to solve the recommendation problem in a binary rating matrix. They show that better recommendations than neighbor-based CF systems can be achieved by a simple NB based recommender systems. In addition, Bayesian Network Classifier is also effective for the recommendation problem.

In ***clustering,*** the users/customers are divided into different clusters that are formed based on different categories and preferences that may come from past purchases of the users. The user preferences are used to calculate recommendations at the level of each cluster where all customers in a cluster receive the same recommendations. Clustering technique is very

Table 2.1 Clustering Example

	Book 1	Book 2	Book 3	Book 4	Book 5	Book 6
Customer A	x			x		
Customer B		x	x		x	
Customer C		x	x			
Customer D		x				x
Customer E	x				x	

helpful to apply for selecting K most relevant neighbors in an algorithm based on CF [22]. Clustering is the faster technique of recommendations however recommendations generated from this technique are less personalized. An example of clustering technique is given in Table 2.1, where clusters are formed based on the interests that five customers have taken six books. Clustering algorithms are also useful in recommender systems. Clustering algorithms are unsupervised learning algorithms and designed to cluster objects into different categories without label information. In CF, clustering usually can be used as an intermediate step. First, one clustering algorithm, such as K-Means, is used to cluster users or items into different groups. Then, the conditional probability of ratings for an item can be calculated based on their group information [60].

In contrast, the association rule method uses past purchases to find the relationship among most common purchases. The association rule model is fast to implement and execute requiring very little storage space and is very successful too in large population applications such as retail stores [117]. However, it is not suitable in cases in which a user's preferences change rapidly and false associations can arise.

The **Matrix factorization** method is used in recommender systems to characterize items and users by vectors of factors that are inferred from item rating patterns [74]. Matrix factorization models are used to analyse the user-item matrix with the purpose of finding latent factors [62]. These latent factors can further characterize the relationship between user and items. For example, in the domain of movie recommendations, latent factors can map the genres of the movies as well as the user characteristics such as age. The latent factor is further computed through matrix factorization techniques in which matrix of the user and items is further decomposed into two smaller matrices. Recommendations are made to the users based on the latent factors space of items in the way those items are recommended to the users which are closed in latent factors space.

The most successful model-based CF technique is Matrix Factorization (MF) [72]. It finds common factors that can be the underlying reasons for the ratings given by users. For example, in a movie recommender system, these factors can be genre, actors, or director of movies that may affect the rating behavior of users. Matrix factorization techniques not only find these hidden factors but also give the importance of them for each user and how each item satisfies each factor. Matrix factorization techniques get the matrix containing all the available ratings and find a feature set for each user and item as the result of the factorization process. Then, a rating that each user assigns to each item can get estimated by the scalar product of the two feature vectors corresponding to that user and item. In this way, users with similar preference will have similar latent features, and items which are favored by similar users will share similar latent features.

As shown in equation 2.9, MF approximates the rating matrix M with two matrices: A and B, which can be viewed as latent factors of users and items respectively. Each row of matrix A and each column of matrix B are the latent factors of a user or item respectively. By multiplying latent factors of a user with latent factors of an item, we get an estimation of the corresponding rating.

$$M = A \times B \tag{2.9}$$

The matrix of user-items in recommender system is usually sparse in nature, the method works to minimize the loss function. The ratings in this method can be predicted for the users u and item i, by using the following Equation 2.10.

$$P(u,i) = \overline{r} \times A_u^T + B_i \tag{2.10}$$

Recommendations, in matrix factorization method, are generated based on high correspondence between the factors of items and users. Matrix factorization has become the most dominant method within CF recommendations and experiences with the Netflix[3] dataset technique have delivered more accurately than other nearest neighbor techniques [75]. However, matrix factorization like other model-based CF approaches also lose useful information for dimensionality reduction techniques. The idea of matrix factorization based on the technique of latent factors is highly influenced by the domain of Information Retrieval in which singular value decomposition (SVD) is used to find the latent factors in documents. Later on, SVD and principal component analysis (PCA) [17] was applied to the domain of recommender

[3] 1http://www.netflix.com/

systems. Following the idea of SVD in information retrieval, Matrix Factorization is used to overcome the data sparsity issues of traditional collaborative filtering methods.

Principle Component Analysis (PCA) is a dimensionality reduction technique that supports the faster computations for recommendations. It is multivariate in nature which transforms a set of correlated variables into a new set which is called uncorrelated variables. PCA is marked as one of the most effective model-based recommendation technique for recommender system used by many researchers [37, 97] for recommendation problems in many domains. In this work, we select PCA as a strong baseline from a family of recommender system algorithms, to compare our results from experiments. Another standard used for recommendation problem is Support Vector Machine (SVM), which assist in classifying different features and recommend by predicting the missing ratings from rating matrix. The SVM based algorithms have also been proposed in a number of scenarios in recommender systems. Due to wide applications of SVM [76, 137] in recommendations, we also select it as baseline method to compare the performance of ANNs.

Restricted Boltzmann Machines (RBM) are generative models that start from the initial visible state and the model can calculate the hidden state where the hidden state is used to reconstruct the visible state [2, 113]. Restricted Boltzmann Machines are powerful models and work efficiently when probability distributions of all the training states converge. The goal of learning is to make the Restricted Boltzmann Machine more likely to reconstruct data that is similar to the provided training data. The model attempts to find a weight configuration which extracts the right set of features by learning the probability distribution over its inputs as shown in the following picture. Each input of a unit is associated with a weight and a transfer function is used to calculate for each unit a score based on the weighted sum of the inputs. Once the weighted sum is calculated it is further passed to an activation function which calculates at the end the probability that the unit state is active [45].

The weighted sum can represented as:

$$\sum_{i=1}^{n} W_i j x_i$$

Where $\sum_{i=1}^{n}$, represent the sum of inputs from 1 to n and W represents the allocated weight to the inputs. while the transfer function that passes the weighted sum to activation function can be represented as following Equation 2.11:

$$\phi = \frac{1}{1+e^{-\Sigma}} \tag{2.11}$$

The probability distribution over the inputs can be represented as following Equation 2.12.

Although RBM can be successfully applied to the recommendation problem with 2 layers to model temporal data. RBM does not make use of content-based information and contextual information where plenty of context features are available as inputs to the system. RBM is typically used in two-dimensional recommender systems with missing values in rating data and take advantage of this fact to compute tractability. Thus, the Restricted Boltzmann Machine is not suitable for the analysis of contextual features and contextual recommendation and in comparison, returns less accurate predictions than the more advanced method of Matrix Factorization. A systematic view of RBM is given in Figure 2.4.

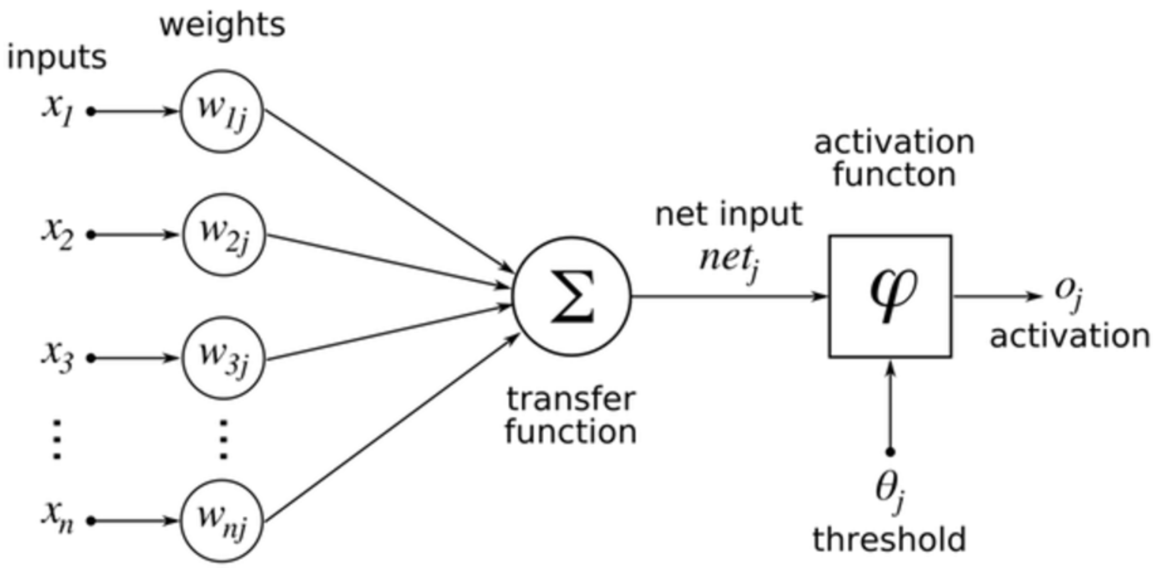

Fig. 2.4 Systematic view of Restricted Boltzmann Machine

Advantages and Limitations of Collaborative Filtering Methods

Collaborative filtering methods do not require information about the contents of an item, the method remains optimal for recommendations [127]. Collaborative filtering exploits only ratings of items given by the users and does not involve any information about users or items. Collaborative filtering systems can generate more personalized recommendations, by considering the experience of the people and can generate recommendations based on that experience. Another main advantage of a collaborative filtering based approach is that the recommender systems can recommend items by observing similar shopping behavior of the users.

However, Collaborative filtering methods cannot generate recommendations if users have not provided any ratings for some items or items are not rated by the users. Collaborative filtering methods determine the poor accuracy of recommendations when fewer user ratings are available which means that a new user must provide enough ratings for the items before the system can recommend items to the users. The problem is known as the cold start problem in traditional CF based methods. Initially, most of the research focused on developing more effective methods for recommender systems that could be helpful to overcome the limitations of traditional recommender systems. particularly, plenty of work in this domain has been done to reduce sparsity and the cold-start problem of collaborative filtering based recommender systems.

2.1.3 Content Based Recommendations

In *content based recommendations*, the information comes directly from the contents of the items rather than the opinions of the user. In order to model the preference data of a user from the examples, a machine learning algorithm is used based on the description of the contents where the contents of items are the explicit attributes or characteristics, for example, the genre of a movie, feature and release year. In content-based recommendations, the items based on the profile are recommended for a particular user based on analysing the contents of the items that a user have liked in the past [24, 36]. For example, if a user liked a science book in the past, the user will be recommended other books of the same type. The challenging task in the content-based recommendation is to extract the features that can help to recommend the items to the users. The following Table 2.2 provides the example of the features of books that can be helpful for the recommendation process.

A content-based recommendation technique is most suitable for text based products, for example, books and web pages, where items can be described by their features as well as users can be described by the keywords in the items that a user bought. The common technique in the content-based recommendation is to build the user profile based on the list of keywords that can describe the items [23].

Table 2.2 Features of items that can be used for content based approach

Title	Author	Genre	Nationality	Keywords
Book 1	Alice	Science	British	Science, distinct, battle death
Book 2	Bob	Fantasy	American	fantasy, magic, dreams
Book 3	John	Adventure	British	adventure, surprise, mesmerism
Book 4	Alice	Science	British	Science, distinct, battle death
Book 5	Alice	Science	British	Science, distinct, battle death

Recommendations are generated on the basis of a match between the user keywords and contents of items. In content-based recommender system, the utility function F can be calculated with the help of Equation 2.13:

$$F(K,T) = score(ContentBasedProfile(k), Content(g)) \tag{2.12}$$

The Content-Based Profile (k) belongs to user and Content(t) of document T, are represented as term frequency-inverse document frequency (TF-IDF) vectors of keyword weights. A utility function F(K, T), in this case, is usually represented by the cosine similarity measure. The Content-Based Profile (k) belongs to user and Content(t) of document T, are represented as TF-IDF vectors of keyword weights.

The user model can also be a classifier in this case for the neural network, SVM, and others. Although in content-based recommendation sparsity problems and cold start problems are resolved by recommending a user with unique taste and the ability to recommend new and popular items, however, the problem with this type of recommendation is that it works with only contents that can be extracted as meaningful features. Where some other type of items like movies and music etc. are not suitable to feature extraction methods. The main drawback of this approach is that it is hard to predict for the items with the similar features.

In this type of recommendation, it is also hard to exploit the quality judgments of other users. Filtering based on the content was the most popular recommender system until the appearance of collaborative filtering [50]. However, the main problem that this system faced was overspecialization. Overspecialization takes place when the contents of recommendations are very similar and do not consider the interests of the users. Another problem of the recommender systems based on the content is that they only offer partial information (usually textual information), whereas the contextual, visual or semantic information is more difficult to know and, therefore, connections between similar objects get lost in a less obvious way.

2.1.4 Hybrid Recommender Systems

Hybrid recommender systems are more advanced that combines the two or more recommendation techniques for achieving better performance in the recommender systems. In hybrid recommender system, most of the times, a memory-based CF algorithm is combined with the model-based CF algorithms to overcome the limitations and drawbacks of the typical approach of collaborative filtering methods [25].

The combination of different recommendation techniques in hybrid recommender system depends on the nature of the final application. However, some combinations of recommendation techniques may not be able to use the models or the features generated by the other technique. Table 2.3 shows some of the combined methods that have been used for hybrid recommender systems. Hybrid models have been proposed by Konstas [71], to combine collaborative method for contents and social information. Another similar approach has been proposed for tag recommendation given in [85].

The approach of hybrid recommender systems helps to overcome the typical problems of collaborative filtering methods such as sparsity and information loss. However, these types of systems have increased the complexity of the recommender systems and are very expensive to implement. An overview of the collaborative filtering techniques [126] with respect to advantages and shortcomings are described in the Table 2.4.

2.1.5 Comparison of different recommendation algorithms

In Figure 2.5, the performance comparison of chosen techniques is given where the user based collaborative filtering performs better than the item-based collaborative filtering, random, popular, PCA and SVD. The comparison shows that the memory based UBCF and popular movie selection in top N recommendations are good in performance than other model-based PCA and SVD. whereas model based PCA and SVD performed better than memory based item-based CF and random items selection techniques for movielens 100K dataset [95]. A comparison of the selected recommendation techniques is given with ROC curve and precision and recall in Figure 2.5.

Table 2.3 Different methods used for hybrid recommender systems

Hybrid Method	Description
Weighted	In weighted method, scores/output of different recommendation techniques such as memory based technique and model based technique are combined to produce a single recommendation.
Switching	In this method, recommender system assume the current situation of the process and switch between two different recommendation techniques.
Mixed	In mixed method, recommendations generated from several different recommender systems are further presented at the same time.
Feature combination	In Features combination method, different features from different recommendation data sources are provided together into the process of a single recommendation algorithm.
Cascade	In this method, one recommender system further refines the recommendations generated by another.
Feature augmentation	In this method, Output generated by one technique is further used as an input feature to another technique.
Meta-level	In this method, the model learned by one recommendation process is further used as an input to another.

Table 2.4 Overview of the techniques used for the recommender systems

Type	Techniques	Advantages and Disadvantages
Memory based CF	Neighbour-based CF User-Item based Top N Recommendations	+Easy to implement. +New data can be added. + Do not have to consider contents of items. + Scale well the items that are co-rated, -Human rating dependency, - Can not recommend for for new users unless enough ratings provided, -In large datasets Scalability is limited
Model based CF	Bayesian belief nets Clustering MDP- based CF Latent semantic Sparse factor analysis SVD PCA	+Better deal with sparsity and scalability +Improved prediction performance -Expensive Models -Loss of useful information for dimensionality reduction techniques
Hybrid RS	Content based CF Recommender Content boosted CF Hybric CF	+ Overcomes limitations of traditional CF approach. + Improved prediction performance +Overcomes problems such as sparsity and grey sheep - complexity and cost is increased -Requires external information that is usually not available

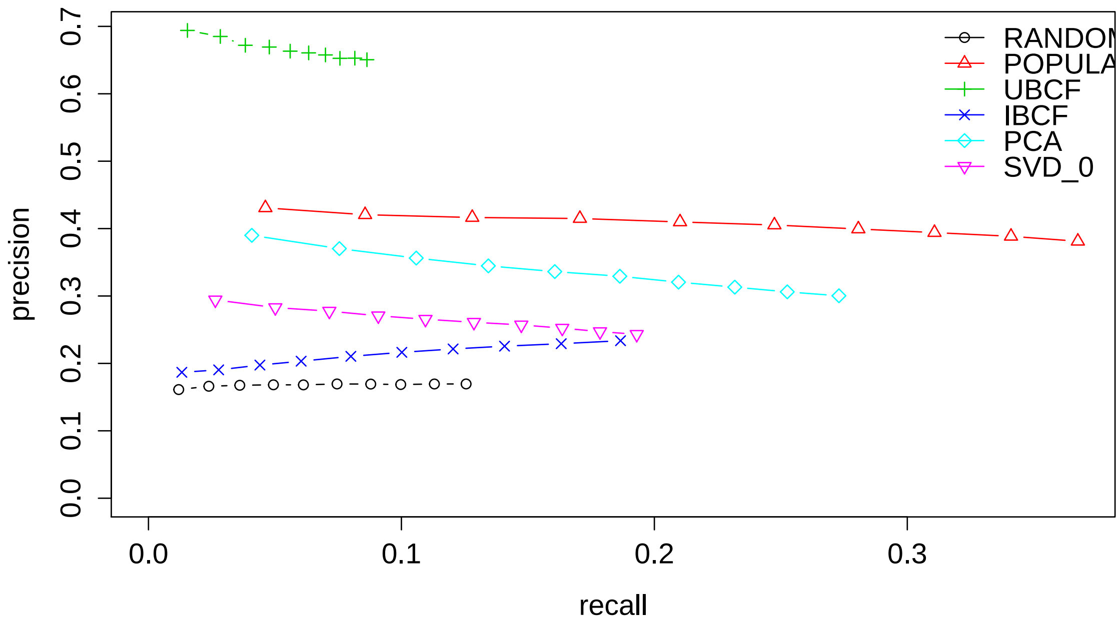

Fig. 2.5 Results for MovieLens 100K Dataset

2.2 Context-Aware Recommender Systems

Traditional methods of recommender systems are two dimensional (users x items) in nature and do not take contextual features into account. The information about the current situation of the users in which items could be recommended comes from the contextual feature, which is very important in many applications. In many cases, the preferences of the users depend on the runtime circumstances that can be used as contexts. These contexts can be the location, time and other factors such as mood of the user [129]. This further leads to the third dimension of context in the recommender system.

A context can be any relevant information that could further be used to characterize the situation. A context can be defined as a dynamic set of factors that further describe the state of a user at the moment of user's experience. A context in recommender system can also be defined as:

"Any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

In this work, a context can be defined as any relevant information that can be used to characterize the situation of an entity. The contexts can be divided among the static contexts and dynamic contexts. Static contexts are the contextual information that is fully observed and remains the same over time, while dynamic contexts are the information which is partially observed and can change over time. The addition of context to the traditional recommendation approach opens a new area of research in this domain which is more challenging than the traditional two-dimensional recommender system. These long-term user preferences are usually expressed in the form of ratings provided by the users that are further modeled as the function of the items, users, and the context.

In CARs some additional contextual information is available to influence the rating behaviour. A context in this case can be defined as a set D where $d \in D$ and contexts consist of values such as $d_1 = \{happy, sad, ..\}$ and $d_2 = \{Morning, Afternoon, \cdots\}$. We can claculate the utility function F with the help of Equation 2.14. In this case multiple contexts $D_1, .., D_m$ are available besides users U and items I, so the utility function y to estimate the rating R can be expressed as:

$$F = U \times I \times D_1 \times \ldots \times D_m \rightarrow R \tag{2.13}$$

Therefore, the rating function R in the above equation is defined by the three aspects including the user, item and multiple dimensions of context which consist of multiple dimensions. The index of context variable starts with 3 here as technical users and items are placed at D1 and D2. In context-aware systems, the main obstacle is to adapt contextual user preference to the dynamic changing contexts. The importance and incorporation of the contextual information into recommender systems have been discussed by researchers in many domains such as movies, trips etc. The contextual recommendations can help to suggest the products more accurately than the traditional two-dimensional recommender systems [14].

One of the interesting questions in the domain of recommender systems is to define what is a good recommendation. Research has reported that a good recommendation is always relevant to the end user and presented in a well-personalized way. It is diverse in nature, presenting the user with all possible suggestions in form of recommendations from different categories. The recommendations should be in the form of discovery and suggestions made to the user should be unsought. The user should not be recommended the items that have already been seen or found [132].

Context-aware recommender system incorporates additional contextual information into the recommendation process. The area of context-aware recommender system has emerged as one of the hottest research topics in the domain of recommender system. Traditional recommender systems focus on recommending the most relevant products/items to the users or the most appropriate users to the items [102]. Whereas the traditional recommender systems have performed well in many applications, in a number of other applications and contexts, such as location and time-based service recommender systems and travel recommendations may not be sufficient to consider only users and items. It is also important to incorporate additional contextual information into the recommendation process [129].

User and Item are specified as the domains of users and items individually where the rating is the domain of ratings, and Context specifies the contextual information associated with the application. The traditional method of recommender system can assume the preferences of the users in different contexts. For example, whether a song is suitable to play for a party or not. In context-aware recommender system, some additional contextual information is available to influence the rating behavior. However, the lack of contextual data remains a fundamental issue for the domain of context-aware recommender systems [29]. The issue has been identified by many researchers. Adomavicius and Tuzhilin [129] have argued that the lack of real-time contextual data pushed the focus of context-aware recommender systems towards the development of conceptual methods where a limited amount of contextual data could be tested.

Since the early work on context-aware recommender systems [109], there have been many efforts made in this domain. Most of the work on context-aware recommender systems is described with directions of future work. The applications of CARs have become very popular in many fields such as movies [20], music [48], mobile recommendations [111], services for learning [136], travel and tourism [82], shopping assistance [112] and multimedia [41]. Different techniques and models of context-aware recommender systems can be categorized using the base of contextual factors. In context-aware recommender systems, contextual information is mostly static in the natures and does not change significantly. However, some recent research has been proposed as dynamic contextualization [106].

Recent work on CARS [93] has focused on developing the models by integrating the contextual information with the user/item relations and models the user, item as well as context interactions. The work presented in [98], provide a new model for representing the preferences of the users into context-aware movie recommender system. To date, different approaches have been proposed under different categories of context-aware recommender systems including Tensor factorization [57] and factorization machine (FM) [110] however both of these approaches have been proposed using explicit feedback. In contextual recommendations researchers [105] also suggest to incorporate the metadata, for example, the attributes of the users and item, into the recommendations, however, the drawback of using metadata in predictions normally yields a small improvement for the prediction of rating.

One of the recent works [130] on CARs suggested a method for dimensionality reduction in space by extracting the latent context from the data acquired sensors of smartphones. In this method, latent contexts consist of unsupervised hidden context patterns which are formed as numeric vectors and extracted from the raw sensor data. Then, the unsupervised deep learning techniques, as well as PCA, is applied to the collected data to learn from the latent contexts. The author further suggested a method to utilize these latent contexts into the recommendation process by describing a hybrid recommendation approach that can utilize these contexts and improve the accuracy of the recommendation process. However, the proposed method considered only two contexts; the geographic location of the user and weather conditions into the contextual recommendation process.

The challenging task in context-aware recommender systems is the lack of contextual data that can be used to recommend the items/products to the user [64]. In traditional recommender system, the user-item rating cannot indicate the preference of the users in different contexts [69]. In order to make recommendations based on the contexts, the user is required to rate the item in some particular context which can be used to define the preferences of the user and generate contextual recommendations. For large systems and data-sets, it is more challenging to divide the user preferences based on the contexts which make the context-aware recommendations an open challenging problem.

On the other hand, Hidasi [52] proposed a General Factorization Framework (GFF), which is based on a single flexible algorithm. The proposed framework takes the preference model as an input to the system and computes latent feature matrices for the dimensions of the input. The benefit of the GFF based approach is that it eases the process of performing experiments with the different linear models for the contextual recommendations tasks

Table 2.5 An overview of some available recommender systems

Name	Type	Research-oriented	Status
BibTip	Document recommendations	Yes	Active
Refseer	Citation recommendations	Yes	Active
Lumi/Last.fm	Music recommendations	Yes	Active
BeerRecommender	Beer recommendations	Yes	Active
Yelp	Food recommendations	Yes	Active
TripMatcher	Trip recommendations	Yes	Active

for both; the implicit or explicit feedback. The scaling properties of GFF make it usable under the different circumstances however the model is tested with few contextual features and leave an open question to expend it in the future to use it with the multiple sets of features.

One of the recent attempts [34] in the domain of contextual recommendation is Postvia 360, which is an advanced architecture in tourism industry domain. The architecture aimed to provide support for tourists and Destination Marketing Organizations (DMO). The system is built on cutting edge of technologies, including artificial immune systems, geo-representation, feature-based opinion mining, semantic technologies. The system was designed specifically to collect data from the initial visit of the users by means of pervasive approaches. The system, then, analysed the data to regenerate the accurate visit data which is further used to offer the relevant recommendations based on the position of the user as a bio-inspired recommender system. The recommender system is particularly tested and validated as per the requirements of the tourism industry.

Some of the recent works [21] [54] [86] [133] focus on the job recommendation on the social networks such as XING where job recommendation [84] task is very close to traditional prediction tasks such as ad click prediction or app download prediction, with a common key component is used to estimate the probability that a user will click on the target item. An overview of some available recommender systems [16] from different domains is presented in the following Table 2.5.

Most of the previous work on contextual recommendation deals with the contexts that are pre-assumed to result in better accuracy of recommendations. Also, the work done on contextual recommendation deals with the limited (few contexts) number of contexts along with the user and item in the recommendation process. In proposed ANNs methodology, the basic motive of the study is to develop an approach that can be used to incorporate the multiple contexts with the user and item data into recommendation process. The methodology, then, can be used to compare the contextual features along with non-contextual features to find a set of features or standalone feature that can be used in the recommendation process to generate recommendations with the better accuracy. The proposed approach is used to predict the ratings that are provided with the set of both contextual and non-contextual features by considering the features (both; contextual and non-contextual) one and in the form of groups (categories/representations). The proposed research is concerned with the explicit feedback which is collected from the users in the form of ratings as opposed to the inference of contexts from implicit information.

A number of approaches have been proposed to deal with contextual data into recommender system, however, most of the researchers [101] agreed on three main types of context-aware recommender systems. The type of CARs are known as contextual pre-filtering, contextual post filtering, and contextual modeling.

2.2.1 Contextual Pre-Filtering

Context-aware recommender systems based on pre-filtering prune the preference data that comes from the user according to the target recommendation context before applying an algorithm for recommendations. In this method, input data is contextualized for the recommender system. The contextual pre-filtering method applies a data processing filter on traditional user-item recommendation method and then a contextualization filter is applied to contextualize the recommendations [129] as shown in the Figure 2.6.

2.2.2 Contextual Post Filtering

Context-aware recommender systems based on the post-filtering methods apply recommendation algorithms on the preference data that comes from the user and then adjust the generated recommendations according to the context. Similar to contextual pre-processing, contextual post-processing also ignores the contexts at input stage and generates recommendations. Once the recommendations are generated, these recommendations are adjusted to the context by removing the irrelevant recommendations.

The post-filtering contextualization method can provide the simplest approach in which recommendations are made through the traditional two-dimensional recommender system approach. further, the contextual recommendations can be generated when context-aware recommendations are required by contextualizing the results from typical traditional recommender systems as shown in the Figure 2.6.

2.2.3 Contextual Modeling

Context-aware recommender systems based on the technique contextual modeling, incorporate the additional contextual information into the model at the same time when it is used to generate recommendations. The input is provided to apply a multi-dimensional recommendation technique and the results are contextualized at the same time the model/algorithm is applied to generate recommendations. A comparison of the type of context-aware recommender system is presented in the Figure 2.6.

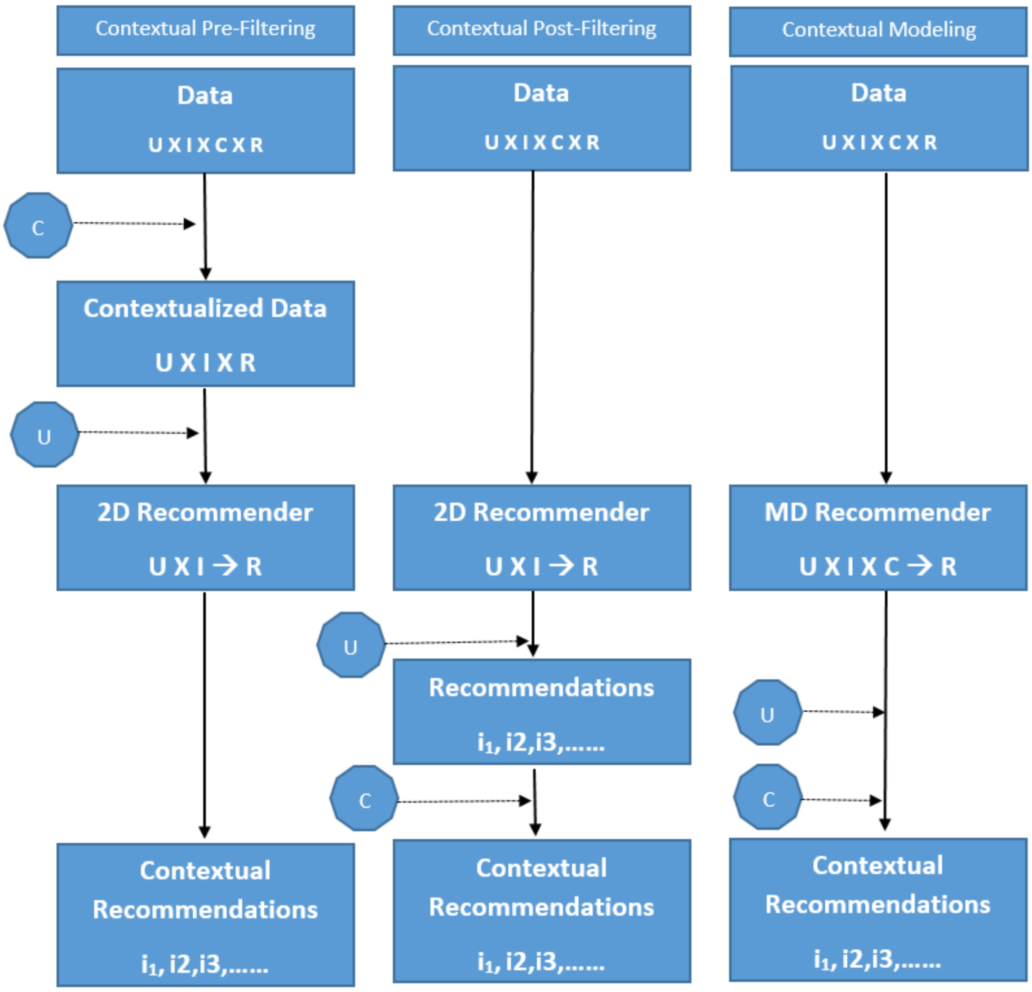

Fig. 2.6 Comparison of context-aware recommender system techniques

In the comparison of CARs techniques, the contextual pre-filtering method is marked as very simple and could work well with the large number of data [5]. However, the method also increases sparseness and does not scale well with many context variables. While post filtering technique, in comparison, is based on a single model that takes interactions into account but it is computationally expensive and also increases data sparseness as well and does not model the context directly.

2.2.4 Context-Aware Recommender System in Mobiles

Recently, the popularity of smart-phones is increasing day by day and most smart-phones have the ability to locate users with the help of GPS. Most users prefer the suggestions on their mobiles, therefore, context-aware recommender systems for mobile phones have a great scope.

The applications of context-aware recommender systems in the domain of mobiles includes restaurant recommendations [13] and tour guide recommendations [30] which works with the time, user ID and location as contextual information. In this case, location is described as the present geographical location of the mobile user where time is described as the time of the request for the service and ID relate to the user of that particular device. However, the limitation of mobile devices such as latency means more personalized recommendations are provided to a mobile user [103].

Some mobile-based recommender systems are presented in the following Table 2.6. In this Table 2.6, different recommender systems designed for the purpose of research and use under different academic institutions are presented with the information about the type of recommender system, type of contexts that are used for recommendations, adopted method and the list of devices on which the designed context-aware recommender system can be further used.

2.2.5 Multiverse recommendation and attribute aware recommender system

In multiverse recommendations technique, a recommender system is proposed to factorize the tensor over the users, items and all the categories of the contexts [67]. The technique shows a better accuracy and performance in predicting items than item splitting, however, the computational complexity is the main drawback of this model. In this scenario, the computing

Table 2.6 Some Context-aware Recommender Systems for mobile devices

	MIT Lab	**Tsing University**	**Telematica**	**FuJen University**
Type	Restaurant	Restaurant	Travel Information	Commercial
Context	Location	Location, Time, Weather	Location, Time, Weather,Schedule, Shopping	Location, Time, Contents
Method	Interaction between 2 agents	Search after request	Variable prediction strategy	NN Learning
Device	GPS	PC, GPS	Mobile,GPS	Mobiles

complexity on the context-aware rating from the users is exponential in the number of modes and polynomial in the number of factors which further lead both towards the poor learning and poor prediction as well as runtime the number of factors grow-up [110].

Attribute aware recommender systems, in comparison, consider some additional information about the user or item individually where contextual data is also attached to rating mode simultaneously [124]. However, this technique is proposed and used for the specific problem only and cannot examine or extend the general problems of context-aware recommender systems.

Although plenty of techniques and methodologies have been proposed for the problems in the domain of context-aware recommender systems for different scenarios, however, lack of contextual data and method to deal with the multiple contextual information is a quite challenging process.

2.3 Evaluation Methodologies

Evaluation of recommender systems is a challenging task because recommender systems have to provide useful recommendations which are relevant to the end user. Traditional methods to evaluate the recommender system can be divided into two categories: offline evaluation and online evaluation [59, 18]. Both methods have their advantages and drawbacks which are discussed one by one in this section.

2.3.1 Offline Evaluation

Offline evaluation method in the domain of recommender systems is a more convenient way to evaluate the recommender system. In this method the dataset which contains the information of users, items and ratings are divided into training and validation sets to train a model on the training set and test the model on the validation set. While the performance of the system is further evaluated on the validation set with different evaluation techniques. The offline evaluation method is most convenient to use as it gives an opportunity to cross-compare the different recommendation algorithms.

Since the method works on the data and behavior of the users collected in the past, the offline evaluation method does not deal with the current users. However, the method is useful when experiments are required to repeat plenty of times, especially in a research environment.

Accuracy of rating prediction with Root Mean Squared Error

Root Mean Error Square is used to compute the accuracy of recommendations based on the prediction of ratings in the validation set from the chosen dataset [102]. It measures the average magnitude of errors in the predicted values. That is the average distance of a data point from the fitted line. Being a quadratic measure Root Mean Square Error (RMSE) is most affected by large errors, thus, it is useful for when large errors are especially undesirable. Recommender system produces the predicted ratings for the user and items and given ratings. The Root Mean Square between the predicted and given ratings can be calculated by the following equation 3.2.

$$RMSE = \sqrt{\frac{1}{N}\sum(\hat{r_{ui}} - r_{ui})^2} \tag{2.14}$$

In recommender systems, most of the times, we do not want to find how accurately the rating prediction is performed, rather, we are more interested to know whether the items that are recommended by the recommender system are relevant or not. In such case, evaluation metrics such as Precision and Recall are very helpful for the evaluation of recommender systems.

Precision and Recall Metrics

In recommender systems, precision and recall are used for the accuracy and performance evaluation [44]. The probability that a selected item is relevant is represented by precision. Precision can be defined as the ratio of selected relevant items to the total number of selected items. Recall, at this stage, can be defined as the ratio of selected relevant items to the number of relevant items available.

We can observe through this technique whether an item which is recommended is relevant or not based on the predicted rating of the user for some given items. The large predicted value for an item indicates the item is relevant, while a small value represents that the item is not relevant. Therefore, we will have four possible outcomes for the recommendation based on relevance and irrelevance of an item.

	Recommended	Not Recommended
Relevant	True Positive (TP)	False Negative (FN)
Irrelevant	False Positive (FP)	True Negative (TN)

Fig. 2.7 Evaluation scheme for recommender system using confusion matrix

Recommender systems recommend an item which is relevant that comes through true positive values and irrelevant from the false positive value from relevancy prediction. Recommender systems do not recommend items which are relevant and irrelevant to the user from false negative and true negative values to evaluate the accuracy of recommendations. The relationship between relevant and irrelevant items as well as recommended and not recommended is presented in Figure 2.7.

By using the precision and recall method, we can generate a test set consisting of both relevant and irrelevant items to measure the proportion of recommended items that are relevant. Precision can be calculated with the help of following equation 3.5.

$$Precision = \frac{TP}{TP+FP} \tag{2.15}$$

Precision is highly suitable for the evaluation of context-aware recommender systems, thus, as part of our evaluation methodology, we assumed to calculate confusion matrix as well as Precision-Recall. The Recall at this stage can be calculated with the help of the following equation.

$$Recall = \frac{TP}{TP+FN} \tag{2.16}$$

2.3.2 Online Evaluation

Online evaluation performs evaluation by interacting with the real users online. In this type of evaluation, variables are divided among the dependent such as the satisfaction of the user and independents such as demographic user features. User studies and questionnaires are presented to the user to evaluate the accuracy and performance of the recommender system.

The technique is very useful as the real-time users interact with the system and provide feedback based on their experience. However, these techniques require plenty of efforts in gathering the response and feedback from the user. Although online experiments overcome the typical limitations of offline experiments by dealing with real human beings that have to evaluate the system, the main disadvantage of this method is the difficulty to perform online experiments several times.

2.3.3 The role of Artificial Neural Network in Context-Aware Recommender System

Artificial Neural Networks (ANNs) consist of artificial neurons which are interconnected and a computational model for processing on the inputs. Typically, ANNs consist of three layers known as an input layer, hidden layer, and output layer. Artificial neurons which are called perceptron from input layer are used to provide the input to the artificial neural network while the hidden layer further assigns weights to the input. The output layer calculates the output based on the weights assigned as part of the hidden layer to provide final results based on the techniques chosen for ANNs [1]. In the following Figure 2.8, a simple structure of an artificial neural network is given.

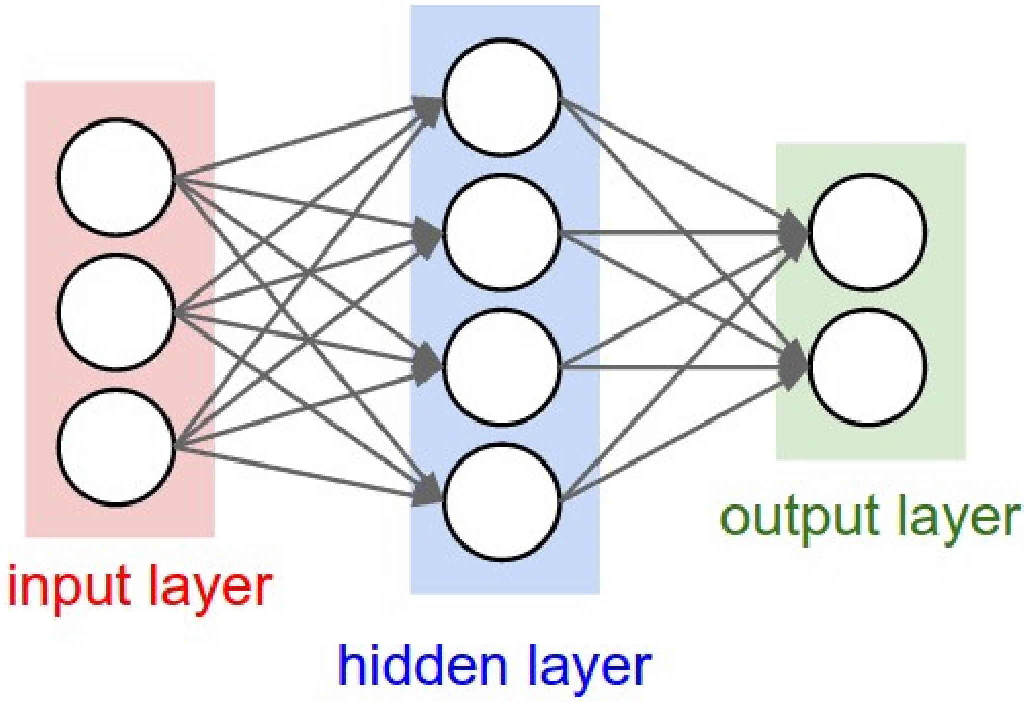

Fig. 2.8 Basic structure of Artificial Neural Network

Artificial Neural Networks can learn, memorize, and establish a fine relationship between the input data of different types. Artificial Neural Networks are also capable of modeling non-linear dependencies. ANN-based algorithms are generally based upon the notion of finding trends, patterns or anomalies in a set of data, and could be beneficially used for analyzing large and complex datasets. This technology could either be used for grouping certain data points together, finding anomalies in data or for predicting an attribute out of several known parameters, based on trends and patterns found in the training data. Predicting an unknown attribute of an object based on other known attributes and information about it is what is generally known as supervised machine learning. Supervised learning is mainly used to either classify an object into one of several predefined categories, such as vehicle type or a binary yes/no answer or in estimating a continuous attribute not bound to any predefined classes, for example, the upcoming petrol price, in what is generally known as regression analysis [107].

ANNs are marked as promising modeling techniques, are very useful when the data sets have non-linear relationships when given as an input [138]. In addition, ANNs can combine and incorporate experimental data to solve problems such as prediction and pattern finding. Different applications of ANNs can be used for the problems such as classification, pattern recognition, prediction, and modeling. A simple ANNs based perceptron model is given in figure 2.9.

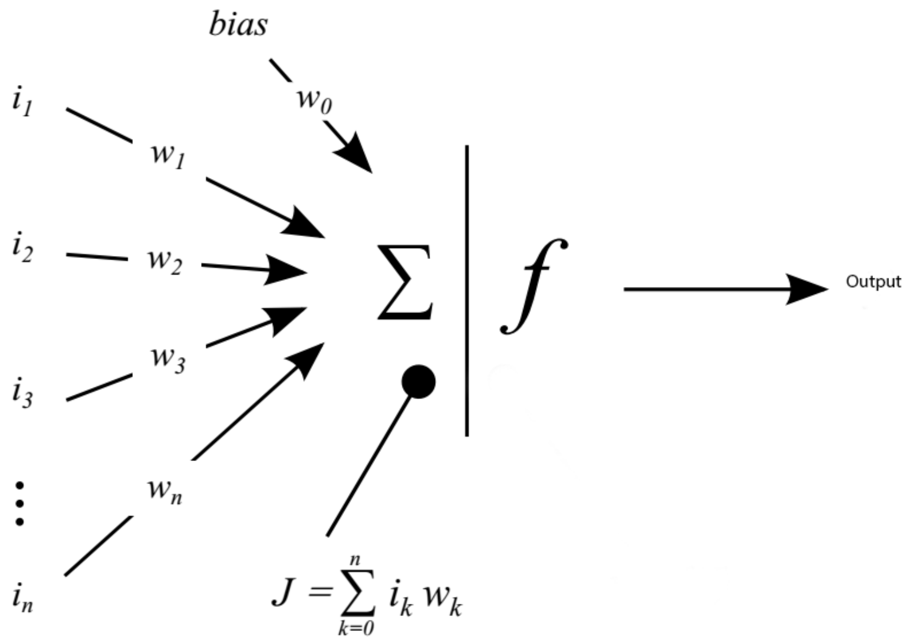

Fig. 2.9 Architecture of Artificial Neural Network. The inputs i_1–i_n to ANNs are multiplied to respective weights W. A weighted sum J is calculated and passed to the transfer function which produce output as y as classification results such as y = {0,1}

A number of techniques based on Artificial Neural Networks have been used for different machine learning problems including Back Propagation Neural Networks (BPNN), Feedforward networks, Group Method for Data Handling (GMDH) are the few ANNs based techniques [61]. Since ANNs have been widely adopted for machine learning and optimization problems, there are a few examples when ANNs are applied to recommender systems. A TV programme recommender system is presented in [77], in which a Feedforward ANN is applied to recommend a TV programme to the user. The approach is tested with one hidden neuron and received an accuracy of 92% which proves that ANNs are highly capable of predicting accurately.

Another ANN based technique is proposed by Chou et al. [32], to make a personalized recommendation based on navigational behaviors and similar interests of the users. A Back Propagation Neural Networks (BPNN) technique is used to train a model that can classify the users into different groups based on the navigational behavior of the user. The user in the same group is recommended the same items. Another example of ANN-based recommender systems is a content recommendation tool [31] developed for Web personalization in which the online shopping behaviors of the users is analyzed to predict what a user will like in future.

Recently experimental results have suggested that, in order to train better AI models, a deep architecture is needed. Before that, models with two or three layers at most perform better than deep models. Deep models tend to give worse results and become harder to train. Research shows that with a layer-wise training strategy, a Deep Belief Network (DBN) can be successfully trained to predict handwritten digits [53]. This is the first attempt and success to training a deep model. Before that, researchers have not seriously exploited deep models due to lack of data and computational power. Generally, deep architecture models consist of multiple layers and can learn a hierarchy of features from low-level features to high-level ones. A DBN is formed with a stack of Restricted Boltzmann Machines (RBM). In its first two layers, two-layer RBM with one visible layer and one hidden layer is trained. Then, the activation probabilities of the hidden layer form a visible layer to learn another hidden layer.

Restricted Boltzmann Machine can be stacked to learn a multi-layer DBN. Another type of deep model is Deep Neural Networks (DNN). DNN is a Multi-Layer Perceptron (MLP) with many hidden layers. Back-propagation (BP) is employed to learn DNN. The success of DNN is due to two techniques: a larger number of hidden units and better parameter initialization techniques. A DNN with a large number of hidden units can have better modeling power. Even when the learned parameters of the DNN is locally optimal, the DNN can perform much better than those with fewer hidden units. However, in order to converge to a local optimum, a DNN with a large number of units also requires more training data and more computational power. This also explains why DNN has become popular until recently. Learning a DNN is a highly non-convex problem, there is no doubt that better parameter initialization can lead to better performance. Researcher [131] found that parameters of DNN can be initialized with the learned parameters of a DBN with the same architecture.

Artificial neural networks can be applied for the purpose of predictions and recommendations. However, we can find only a few examples where ANNs are used for recommendation purposes. The great tendency of using ANNs in many machine learning areas make artificial neural networks highly emerging candidate to introduce a methodology that can be applied to context-aware recommender systems.

2.4 Summary and Conclusion

In this chapter, we defined the basic recommendation problem and explained how traditional recommender systems work. We reviewed different algorithms and techniques used for recommender systems and discussed the motivation about context-aware recommender systems. We also discuss the advantages and shortcomings of different recommendations techniques and models. In this chapter, we further described the concepts and challenges that context-aware systems are facing in the domain of recommender systems. We introduced the different methods for context-aware recommender systems and described the context factor in contextual recommendations. We also described the limitations of context-aware recommendation methods and explained the role of artificial neural networks in the domain of context-aware recommender systems. We also outlined the evaluation methods that can be used to evaluate a context-aware recommender system.

Chapter 3

Proposed Methodology and Experimental Set-up

In this chapter, we describe the research methodology which is proposed for the contextual recommendation as well as experimental setup and evaluation measures that will be used as part of the experiments. The proposed methodology combines three different stages which are data pre-processing (Data Transformation), Artificial Neural Networks (ANNs) based model and output which refer to the final recommendations. In this chapter, we will discuss the proposed methodology and its key phases that can be used for contextual recommendations, the dataset that we will use for our experiments, experimental setup and evaluation measure that will be used to evaluate our results. This chapter referred to the objective O4 that focus on the proposed methodology that is used to integrate the contextual and non-contextual features into the recommendation process. Later on, the proposed methodology as part of chapter 4 and chapter 5 is evaluated.

3.1 Proposed System Architecture

The architecture of the system as the proposed methodology is given in Figure 3.1. The proposed system consists of three main phases including data pre-processing and transformation phase, Multilayer Perceptron (MLP) model from ANNs and the contextual recommendation phase or output phase. Each phase of proposed system architecture is described as following

3.1.1 Data pre-processing, Normalisation and transformation

The first phase of the proposed architecture is data pre-processing and transformation or normalisation of an input to ANNs. If the input variables are combined linearly, as in an MLP,

Table 3.1 Example dataset for data preprocessing and transformation

User ID	Item ID	Ratings	Location	Time
U1	P1	2	Home	Morning
U2	P2	3	Work	Afternoon
U3	P1	4	Home	Evening
U2	P2	3	Work	Afternoon

then it is rarely strictly necessary to standardize the inputs, at least in theory. The reason is that any rescaling of an input vector can be effectively undone by changing the corresponding weights and biases, leaving you with the exact same outputs as you had before. However, there are a variety of practical reasons why standardizing the inputs can make training faster and reduce the chances of getting stuck in local optima.

Since the contextual data and ratings comes in the abstract form of the data which requires pre-processing before it can be used for the recommendations, we normalise the input values to make the training phase faster and accurate. Traditional recommender systems work with the user-item matrix only, while working in context-aware recommender system an additional input comes in the form of context. Since the second phase of proposed architecture MLP model is an ANNs based model, it requires a specific data format that could be used for the further process. ANNs based approaches deal with the normalized form of data for the input of features (Contexts) and binary format for the target data (ratings). Data pre-processing phase in proposed system architecture helps the data to convert it into the format that can be used in next phase. In order to describe the data transformation stage, we report an example of data in following Table 3.1.

The example dataset in this table consist of users, items, and rating alongside two contextual attributes as Location and Time. We have ratings on the scale from 1 to 5, and we have two contextual attributes for the location which can be denoted as Home = 1 and Work = 2. Similarly, for contextual attribute time we can denote Morning = 1, Afternoon = 2, and Evening = 3. Now, at first stage, we pre-process the table which can be written in the form given in Table 3.2.

At next stage, the rating data is transformed into binary representations which could be done by binary transformation of the numeric data. The rule of binary transformation is simple for rating data as the number of bits depends on the rating scale. In example data-set, we have ratings on the scale from 1 to 5, so we need five bits (Number of columns) for binary

Table 3.2 Example dataset for data preprocessing and transformation

User ID	Item ID	Ratings	Location	Time
U1	P1	2	1	1
U2	P2	3	2	2
U3	P1	4	1	3
U2	P2	3	2	2

Table 3.3 Rating Transformation into binary

Ratings	Binary	Format			
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
3	0	0	1	0	0

representation. All we will switch on the relevant bit (If the rating is 3 out of five the third column will be 1 and remaining will denoted with 0) to transform ratings into binary format. In Table 3.3, ratings are transformed into binary format from Table 3.1.

Now the Table 3.1, after pre-processing and transformation will look like the Table 3.4. The data is in processed form, now, and could be passed to the next phase which is multilayer Artificial Neural network perceptron model. Normally, ANNs predict better on aforementioned data format, however, in order to improve the prediction and classification task, the data can be further normalized by dividing with the number of attributes in a specific data column. For example, contextual attribute *Time* have total three attributes in Table 3.1, which are {Morning, afternoon, Evening} . After denoting {Morning = 1, Afternoon = 2, and Evening = 3 }, we can divide the numeric values by 3. we will have the normalized numeric number as {Morning = 0.33 , Afternoon = 0.66, and Evening = 1}. Once, the data is preprocessed and ratings are transformed into binary transformation, now the data is ready to pass to the next phase of proposed Artificial Neural Networks based approach.

Table 3.4 Example data after preprocessing and transformation

User ID	Item ID	Ratings					Location	Time
U1	P1	0	1	0	0	0	1	1
U2	P2	0	0	1	0	0	2	2
U3	P1	0	0	0	1	0	1	3
U2	P2	0	0	1	0	0	2	2

3.1.2 An Artificial Neural Networks Phase

Once the data is processed, it is passed to the next phase of the proposed approach. The next phase is Multilayer Perceptron Model: An ANN-based model which can be used to predict the missing ratings and generate recommendations from input data. The MLP phase consists of three further stages, in fact, three layers known as the input layer, hidden layer, and output layer as shown in the Figure 3.1. An MLP model is technique based on ANNs. It is a feedforward artificial neural network model that has ability to map the sets of input data on a set of appropriate target outputs.

Multilayer Perceptron Model consists of multiple layers, where each layer is fully connected to the next layer with the help of nodes except the input nodes. Each node in multilayer perceptron is a neuron, which is also known as a processing unit as well as a nonlinear activation function. multilayer perceptron-based approach uses a back-propagation technique for the purpose of training in artificial neural networks. The three-layer architecture of MLP model consists of the input layer, hidden layer, and output layer. The input layer is used to provide inputs to the model that can be the input data from user-item tables in the context of recommender system while an additional input can be provided as the list of the contexts that can be modeled. The rating data is provided in the form of binary format to map the user-items alongside the contexts.

Hidden layer in multilayer perceptron model consists of hidden neurons that contain activation function. The hidden neuron uses to assign the weights to the input data while the activation function in each neuron is used map the weighted inputs to the output. In this phase, learning occurs through backpropagation in perceptron by changing the assigned weights after each piece of data that is provided as input is further processed on the base of errors in the output compared to the expected output result. The output layer consists of the final results generated by the model in form of recommendations, that could be evaluated using ANNs. The structure of the multilayer perceptron model is given in the second phase of following Figure 3.1.

Once the predictive results are generated through multilayer perceptron model, the results can be stored in the form of contextual recommendations based on the predictions made by ANNs. The last phase of proposed approach given in Figure 3.1 is the final output in form of recommendations based on the contexts that are provided as input to the system. The context-aware recommendation is generated through proposed multilayer perceptron model as the more refined approach which has potential to produce more accurate and relevant recommendations to end user.

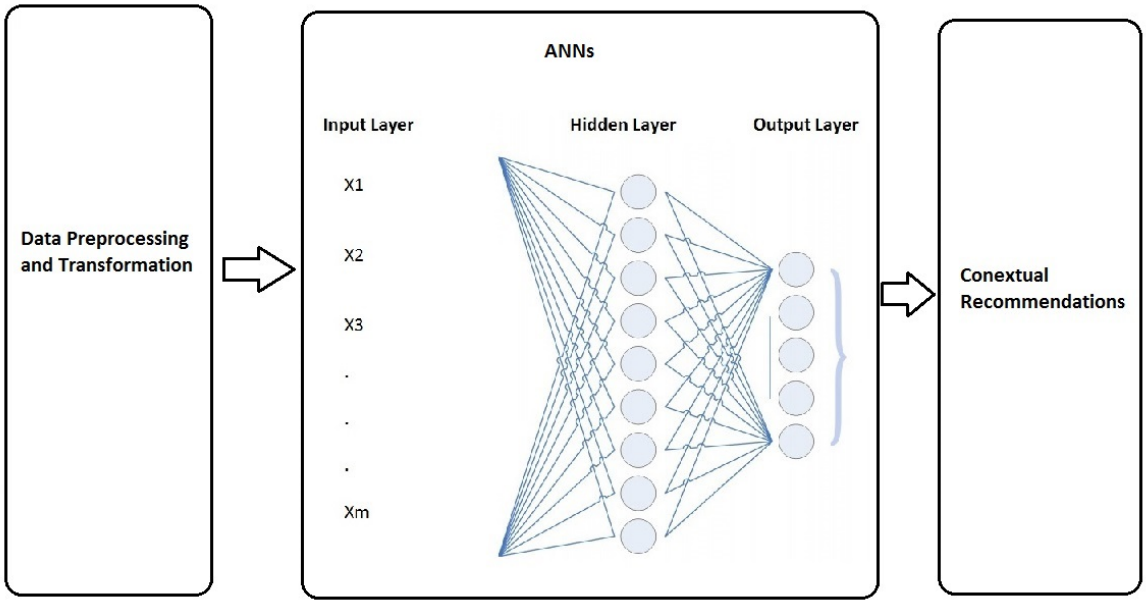

Fig. 3.1 Proposed System Architecture

On the basis of proposed approach, a setup for the experiments is established and described in the next section. The experiment setup will assume the proposed methodology to run further experiments.

3.2 Experimental Set-up

In order to perform experiments and validate the results, following the methodology proposed in Section 3.1, the experimental ground is prepared by selecting the appropriate dataset, tools, and technologies along with the hardware resources. For this purpose, we reviewed the literature [63], to search open datasets that can be used for the problem of the context-aware recommender system. In order to make maximum utilization of our proposed context-aware recommendation methodology, we searched out UCI[1] machine learning repository of the datasets which contains 360 data sets as a service to the machine learning community. However, in particular domain of context-aware recommender system, we can only find few of them relevant including Opinrank hotel recommendation dataset [42], restaurant recommen-

[1]www.http://mlr.cs.umass.edu/ml/datasets.html

dation dataset [103]. Most of these open datasets come with few contextual variables which provide no reason why we should use that particular contexts. As well as both datasets come with pre-assumptions that the given contextual variables will yield good recommendation accuracy. While, one of our main objective O2, of this research, is to analyze different contextual and non-contextual attributes to describe which contextual variables/features can provide the better recommendations, in comparison to the others.

For this purpose, we discovered an interesting dataset, LDOS-CoMoDa[2] [100], a context-based movie recommender dataset which is acquired for conducting research in the domain of context-aware recommender systems. The dataset contains ratings for the movies as well as the contextual information describing the situation in which the movies were watched alongside the non-contextual information such as user features and movie features. The features of the chosen datasets are described in the following section.

3.2.1 Dataset

LDOS-CoMoDa [100] is rich in terms of contextual attributes in which ratings and contextual information are collected explicitly from the users after a user watched a movie. In this dataset, contextual information is the situation in which a user watched that movie. The ratings and the context information are gathered from the real users immediately after user watch a movie based on the interaction of user-item relationship. However, user's past experiences based on the memories are not considered. Users are also asked to rate the same movie more than one if they have watched a movie multiple times. The dataset is available on request for the research purpose and provides rich contextual attributes with almost 12 contextual features which make it ideal to use for contextual feature analysis and contextual recommendations.

The abstract view of the data from LDOS-CoMoDa that is divided into user features,movie features and item features is given in Table 3.5, 3.6 and 3.7. In Table 3.5, a brief portion of user features is taken to give an overview of the structure of user features in LDOS-CoMoDa dataset. Similarly, in Table 3.6, a brief portion of movie's features are presented from the selected dataset follow by the Table 3.7, in which a brief portion of contextual features are given from LDOS-CoMoDa.

[2]http://www.ldos.si/comoda.html

Table 3.5 Abstract View of User Features in LDOS-CoMoDa Dataset

User ID	Item ID	Rating	Age	Sex	City	Country
23	14	5	33	1	20	2
21	5	3	28	1	10	3
21	6	4	28	1	10	3
22	13	4	28	1	20	2
21	7	3	28	1	10	3
20	12	5	30	1	10	3
21	15	3	28	1	10	3
15	11	4	30	1	10	3
23	18	4	33	1	20	2
26	17	5	-1	2	3	5
26	18	4	-1	2	3	5
26	16	1	-1	2	3	5
27	19	2	25	1	10	3
28	19	2	26	2	10	3
29	20	3	30	1	20	2
30	21	4	29	1	10	3
30	22	1	29	1	10	3
31	3	3	30	2	20	2
31	9	4	30	2	20	2
33	5	3	26	2	10	3
33	6	4	26	2	10	3
34	4	4	26	1	20	2
35	23	1	30	1	10	3
35	24	3	30	1	10	3
35	25	4	30	1	10	3
27	18	4	25	1	10	3

Table 3.6 Abstract View of Movie Features in LDOS-CoMoDa Dataset

User ID	Item ID	M-L	M-Y	M-C	Dir	Budget	Genres	Actors
23	14	9	2007	36	234	9000000	7, -1, -1	1303, 1524, 1656
21	5	9	1998	36	234	90000000	7, 6, 10	1636, 1539, 1402
21	6	9	2008	36	234	24000000	7, 10, 18	1373, 1510, 1691
22	13	9	2010	36	234	60000000	1, 14, 19	1382, 57, 1109
21	7	9	2003	36	234	35000000	3, 10, 18	98, 385, 1827
20	12	9	2003	36	234	-1	10, -1, -1	482, 1190, 393
21	15	9	2071	36	234	4000000	1, 8, 21	384, 93, 715
15	11	9	2010	36	234	20000000	7, 18, 18	862, 61, 1378
23	18	9	2010	36	234	82000000	1, 3, 21	1803, 251, 1376
26	17	9	2010	36	234	8000000	1, 7, 8	190, 558, 1657
26	18	9	2010	36	234	82000000	1, 3, 21	1803, 251, 1376
26	16	9	2010	36	234	110000000	1, 17, 21	98, 1181, 328

LDOS-CoMoDa consists of 4381 movies which are rated by 121 users. The number of ratings available in this dataset is 2296 and the maximum number of ratings provided by a single user is 220; the minimum number of ratings is 1. The dataset consists of 12 contextual variables in addition to static information of movies and users. LDOS-CoMoDa dataset consists of an abstract form of the features which can be divided among the User features, Movie features and Contextual features as given in given in the following Table 3.8.

Once we have the features in three different categories i.e. user features, movie features, and contextual features, we further look into the numeric representation of the features so that we can use our proposed methodology which is described in Section 3.1. The quality of the LDOS-CoMoDa, representation of the features with the numeric value make it an ideal dataset for our proposed methodology that we described in Section 3.1. We need the data in the numeric form which can be further processed to exploit our proposed approach. We are more interested in the processed form of the data that can result in good recommendation results.

An overview of the contextual features with respect to associated numeric value is given in the Table 3.9, where each contextual attribute in every contextual feature is represented by a number. Once, we have the numeric data which correspond to the contextual attributes in contextual features, now, the contextual features data is ready to use as an input to the user-item data and ratings. In order to achieve good prediction accuracy, this data can be further normalized by dividing the numeric number by value of a total number of attributes

Table 3.7 Abstract View of contextual Features in LDOS-CoMoDa Dataset

Time	day	Season	Loc	Weather	Soc	E-Em	D-Em	Mood	Phy	Dec	Int
4	1	3	3	1	3	4	1	3	1	1	1
4	1	2	2	2	3	7	7	1	1	1	1
2	1	2	1	2	1	7	7	3	2	1	1
4	2	2	1	2	1	1	2	1	1	2	1
2	2	2	1	1	2	7	7	1	1	2	2
3	2	2	1	1	2	2	2	1	1	2	1
3	1	2	1	3	5	4	7	1	2	2	1
3	1	3	1	1	1	6	6	2	2	1	1
4	1	3	1	1	1	2	4	1	2	1	1
3	2	2	1	1	5	2	2	1	1	2	2
3	3	2	1	2	7	7	2	1	2	1	1
4	3	2	1	2	7	2	4	1	2	1	1
2	2	2	1	1	1	2	2	1	1	2	2
2	2	2	1	2	1	2	2	1	1	2	2
2	2	2	1	2	1	4	2	2	1	1	1
2	2	2	1	2	1	2	4	2	1	2	2
3	2	2	1	1	1	5	7	2	1	1	1
3	2	2	1	5	1	1	6	1	1	2	2
2	2	2	3	2	2	7	7	2	1	1	1
4	2	3	1	2	2	7	7	2	1	1	2
3	2	2	1	1	2	7	2	2	1	2	1
3	2	3	1	2	1	1	3	2	1	2	2
4	2	3	1	2	1	4	3	2	1	2	2
3	1	2	2	2	6	2	2	2	1	2	1
2	2	2	1	2	3	2	2	2	1	2	1
4	2	3	1	2	1	4	2	3	1	1	1
3	2	2	1	3	1	7	6	2	1	1	2
2	2	2	1	2	2	7	7	2	1	2	1
3	2	2	1	2	2	2	2	1	1	1	1
1	3	2	3	1	1	2	4	1	1	1	1
3	1	3	1	5	1	2	7	1	1	1	1
4	1	3	1	5	1	2	2	1	1	1	1
2	1	3	1	2	1	2	4	1	1	1	1
3	1	3	1	2	1	7	4	1	1	1	1
3	1	3	2	5	6	7	7	1	1	2	1
3	1	2	1	1	1	2	2	2	1	2	2
4	1	2	1	1	1	1	2	2	1	2	2

Table 3.8 List of Features in abstract form from LDOS-CoMoDa dataset

Features	Attributes
Movie Features	Item ID, movie language, movie year, movie country, dir, budget, a1, a2, a3 (Actors of the movie), g1, g2, g3 (Genres of the movie)
User Features	User ID, age, gender, city, country
Contextual Features	time, day-type, season, location, weather, season, dominant emotions, end emotion, mood, physical, decision, interaction

in a single contextual feature. For example, after replacing {Morning = 1, Afternoon = 2, and Evening = 3 } in contextual feature Time, we can divide the numeric values by a total number of contextual attributes 3 as defined in section 4.1. we will have the normalized numeric number as {Morning = 0.33 , Afternoon = 0.66, and Evening = 1} which is ideal to use for the artificial neural networks based approach.

The input data consist of contextual data alongside user item data and ratings. In LDOS-CoMoDa, we have rating data on scale 1-5. In order to use this rating data as an input to the proposed system, we need the binary transformation of the data which could be done by the following method described in section 3.1. The data transformation phase of the proposed approach will help to convert the rating data into binary that which is suitable for the ANNs based approach to process the input.

The rating data from LDOS-CoMoDa dataset is transformed into a binary representation of the data as per the requirements of the proposed system. The transformation of the first 40 rating entries from LDOS-CoMoDa dataset is given in the following Table 3.10. The table contains the entries of user-items, ratings and the binary form of ratings, after transformation process.

Once the data from chosen dataset LDOS-CoMoDa is pre-processed and user ratings are transformed to the binary format, the data is ready to go through the proposed ANNs based model. Since the LDOS-CoMoDa dataset contains plenty of contexts, user and movie feature, one of the most interesting finding could be the optimal combination o user features and movie features with contextual features that can predict recommendation with higher accuracy. This analysis will not only help to identify the relevant features that could be incorporated into context-aware recommender system, but it will also help to refine the data by

Table 3.9 Contextual Features from LDOS-CoMoDa and their assigned numeric values

Contextual Features	Corresponding Numeric Value	Detail
Time	1-4	Morning = 1, Afternoon = 2, Evening = 3, Night = 4
Season	1-4	Spring = 1, Summer = 2, Autumn = 3, Winter = 4
Location	1-3	Home = 1, Public Place = 2, Friend's House = 3
Weather	1-5	Sunny = 1, rainy = 2, stormy = 3, snowy = 4, cloudy = 5
Mood	1-3	Positive =1, Natural=2, Negative = 3
Physical	1-2	Healthy = 1, ill = 2
Decision	1-2	User's Choice = 1, Given by others = 2
Interaction	1-2	First = 1, nth = 2
daytype	1-3	Working day = 1, Weekend = 2, Holiday = 3
Social	1-7	alone = 1, partner = 2, friends = 3, colleagues = 4, parents = 5, public = 6, family = 7
dominantEmo	1- 7	sad = 1, happy = 2, scared = 3, surprised = 4, angry = 5, disgusted = 6, neutral = 7
endEmo	1-7	sad = 1, happy = 2, scared = 3, surprised = 4, angry = 5, disgusted = 6, neutral = 7

extracting and refining the irrelevant features that normally do not result in good/better results.

This could be done by mapping the input features with user item data over the binary rating using proposed Multilayer Perceptron Model. The feature analysis could also be performed by predicting the user ratings with the help of contextual data, however, dealing with the too many contextual, user and movie features from the LDOS-CoMoDa is another big challenge.

Once the data is in proper processed format, it can be used in proposed methodology. The next step is to identify the evaluation methodology that can be used to evaluate the performance and accuracy of recommendations, generated from the proposed approach. We have discussed some common methodologies used for the context-aware recommender system in Chapter 2, However, in next section, we will briefly discuss the evaluation methods that will be used to evaluate the results as part of our experiments.

3.2.2 Software and Hardware Resources

The experiments are performed using Matlab R2016b on Windows 10 and Mac OSX platforms with 16GB RAM and Intel Core i7 processor. Since machine learning is quickly becoming a powerful tool for solving complex modeling problems across a broad range of domains, several machine learning applications and toolbox are available on Matlab platform to provides functions and applications that can be used to describe, analyse, and model the data. For multidimensional data analysis on Matlab platform, machine learning toolbox provides the feature selection, stepwise regression, principal component analysis (PCA) and many other dimensionality reduction methods to identify the variables or features that have an impact on the model.

The toolbox facilitates both; the supervised and unsupervised learning algorithms such as support vector machines (SVMs), decision trees, k-nearest neighbor, k-means, clustering, Gaussian models, and hidden Markov models to compute on data sets that are too big to be stored in memory using the applications of Matlab. In particular, we use classification learner applications to train the models on the dataset and classify data using supervised learning. Using Classification Learner, we perform classification of the data to compare the multiple contextual and non-contextual features as well as generate predictions from the given data. We also compare the different data models from classification learner applications such as SVM, PCA, Logistic Regression to compare with our proposed ANNs based methodology to

Table 3.10 Binary Transformation of ratings in LDOS-CoMoDa dataset

User ID	Item ID	Rating	Binary Transformation of ratings				
23	14	5	0	0	0	0	1
21	5	3	0	0	1	0	0
21	6	4	0	0	0	1	0
22	13	4	0	0	0	1	0
21	7	3	0	0	1	0	0
20	12	5	0	0	0	0	1
21	15	3	0	0	1	0	0
15	11	4	0	0	0	1	0
23	18	4	0	0	0	1	0
26	17	5	0	0	0	0	1
26	18	4	0	0	0	1	0
26	16	1	1	0	0	0	0
27	19	2	0	1	0	0	0
28	19	2	0	1	0	0	0
29	20	3	0	0	1	0	0
30	21	4	0	0	0	1	0
30	22	1	1	0	0	0	0
31	3	3	0	0	1	0	0
31	9	4	0	0	0	1	0
33	5	3	0	0	1	0	0
33	6	4	0	0	0	1	0
34	4	4	0	0	0	1	0
35	23	1	1	0	0	0	0
35	24	3	0	0	1	0	0
35	25	4	0	0	0	1	0
27	18	4	0	0	0	1	0
27	26	5	0	0	0	0	1
31	27	4	0	0	0	1	0
31	28	3	0	0	1	0	0
23	31	4	0	0	0	1	0
21	29	4	0	0	0	1	0
23	32	3	0	0	1	0	0
21	30	4	0	0	0	1	0
28	33	2	0	1	0	0	0
26	34	3	0	0	1	0	0
28	35	4	0	0	0	1	0
35	30	4	0	0	0	1	0
35	36	3	0	0	1	0	0
33	37	5	0	0	0	0	1
21	38	5	0	0	0	0	1

compare and evaluate based on the accuracy of Matlab environment.

3.2.3 Evaluation Measures

Once, we have the data in the processed form, our experimental setup is ready to run the experiments with all basic elements. In this section, we present a brief overview of the evaluation methods that we will use to evaluate our final results/output.

Accuracy

Artificial Neural Networks provide evaluation schemes by dividing the data into three phases, Training, Selection and Validation. Validation data is used to evaluate the accuracy of the results that are produced using ANNs based approach. Accuracy can be used for the predictions of the ratings using the proposed approach which actually tells how accurately the system has classified the inputs and generate the predictive results. The evaluation can be shown into confusion matrix in which we will have the values for True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). The accuracy of the predictions will be used to evaluate the performance of different contextual inputs alongside the initial ratings and movie features from the proposed methodology. The calculation of the accuracy can be shown with the help of the following Equation.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{3.1}$$

Precision and Recall metrics are more practical in the context of the recommender system. We also put some of the confusion matrices from our experiments into appendices section to show that we have calculated different measures such as precision, AUC, accuracy, and others. We find accuracy is more relevant and reported by the most of literature [151], so we assumed it as main values to put into tables of our results. Precision and Recall help to evaluate the relevance prediction accuracy by calculating the relevance of items to the users. In recommender systems, precision and recall are used for the accuracy and performance evaluation. The probability that a selected item is relevant is represented by the precision.

We can observe through this technique whether an item which is recommended is relevant or not based on the predicted rating of the user for some given items. The large predicted value for an item indicates the item is relevant, while a small value represents that the item is not relevant. Therefore, we will have four possible outcomes for the recommendation

Table 3.11 Proposed Evaluation Method

	Recommended	Not Recommended
Relevant	True Positive (TP)	False Negative (FN)
Irrelevant	False Positive (FP)	True Negative (TN)

based on relevancy and irrelevancy of an item. True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). The accuracy of the predictions will be used to evaluate the performance of different contextual inputs alongside the initial ratings and movie features from the proposed methodology. The detail description of Evaluation measure is given in Chapter 2.

Recommender system recommends an item which is relevant that comes through True positive values and Irrelevant from the False Positive value from relevancy prediction. Recommender system does not recommend items which are relevant and irrelevant to the user from False Negative and True Negative values to evaluate the accuracy of recommendations. The relationship between relevant and irrelevant items as well as recommended and not recommended is presented in table 3.11.

3.2.4 Significance Test(Paired t Test)

The basic objective of the significance test or paired t-test is to observe if there is a statistical evidence that can confirm that the mean difference between paired observed values on a specific outcome is significantly different from zero [123]. It is a statistical hypothesis test to confirm if the null hypothesis is supported following the student's t distribution.

For example; the null hypothesis suggests the mean as μ_0, and sample mean as $\bar{z}$, then we can calculate one sample t-value with the help of following Formula 3.2, where s is the standard deviation and n represent the total number of observations.

$$t = \frac{\bar{z} - \mu_0}{\frac{s}{\sqrt{n}}} \tag{3.2}$$

The paired t-test considers the difference between paired values in observations from the algorithm and expert, estimates the variation of values within each and produces a single number known as a t-value. It represents if two sets of observations are significantly different from each other resulting in the acceptance or rejection of the null hypothesis.

3.2.5 Cross Validation

Algorithms that deal with recommendations and classification problems potentially face the problem of over-fitting of data. The problem can be resolved by validating data in the manners of splitting the sample data further into two subsets training and testing. The over-fitting issue can be circumvented by splitting the sample data into subsets where one subset is used for the training algorithm and other subsets act as independent data. This method of splitting data into several subsets and use one for the training purpose and remaining as independent test data is called cross-validation. The method is efficient for ANNs based approaches.

In this K folds cross-validation method, the whole set of inputs from LDOS-CoMoDa dataset is divided into K folds where we use K-1 subsets as independent test data and one subset as the training set. As part of this work, we use 10 fold cross validation using ANNs based approach.

3.3 Summary and Conclusion

In this chapter, we presented the description of proposed methodology. We have introduced, an Artificial Neural Network based multilayer perceptron model for context-aware recommender system. We described our proposed approach and presented the test collection that will be used to perform our experiments. We discussed the key features of our chosen dataset and argued why these collections are suitable to perform our experiments. We also discussed how we can pre-process the data available in LDOS-CoMoDa dataset and transform the rating data into binary ratings using proposed approach. We further discussed, how we can evaluate our results in this chapter by selecting evaluation methods that we discussed in chapter 2. We have selected the accuracy measure as evaluation measure to calculate the accuracy of correctly classified relevant items. We also concluded that ANNs use data as an input in the specific format to predict better results. So we proposed an approach to transform data into the form as per the requirements of ANNs.

Chapter 4

A Feature Selection Approach to Compare the Contextual, User's and Movie's Features

In context-aware recommender system, one of the basic problems is to select an appropriate context that can provide good accuracy when it is combined with the rating data to generate recommendations. The LDOS-CoMoDa dataset provides different contextual, movies and user's features which are large in number to consider for the context-aware recommender system given in Section 3.2.1. In context-aware recommender system, all contextual information or the other user or movie features may not be possibly relevant to the recommendation process. In this chapter, we aim to investigate the role of these features by dividing them into different groups/representations by using ANNs approach. We further aim at identification of the relevant group of features that can be incorporated into context-aware recommender system for recommending movies.

In this chapter, we briefly discuss the problem space to compare different contextual and noncontextual features and explain how the test collection is divided into different representations (Categories). Since we have a variety of features which can be used to predict ratings, it is very important to analyse the features and identify the relevant feature set. We further, explain how we divided these features among the six categories as well as explain on what basis we formed categories. We also describe the transformation of the rating data into binary data which can be used as target data in our proposed approach. We also explain in this chapter, how the proposed ANNs based approach is used to compare different categories of features using ANNs. We, then, perform our experiments and present our results in the evaluation section. Later on, We discuss our results and summarize our key findings. This chapter refers to the objective O4 that focus on evaluation and analysis of the different feature sets available for contextual recommendation process where the proposed methodology is used to integrate the features into the recommendation process as per O4. This chapter also refers to contribution C1 which focus on the categorisation of contextual and non-contextual features in context-aware recommender system and evaluation thereof.

4.1 Introduction

Feature selection is the curse of dimensionality, that can be used to reduce the number of features available in a selected dataset, especially, when a large number of features are available. There are plenty of techniques that can be used for the selection of an appropriate set of features, however, complex structures of the many machine learning algorithms result in the selection of irrelevant features [144]. In this work, we propose a simple ANNs based methodology that can be helpful to compare the features and check the performance of the features when mapping input our output. This can further help to identify the list of features with the better performance. Generally, ANNs can be helpful to map an input over an output to classify the input data and find the patterns. However, ANNs can also be applied to quality of an individual feature as well as a group of features that can support the feature selection process. So in our proposed approach, we are following a typical method of mapping input and classification. Using concept of classification in ANNs, an input feature set consist of $X = \{x_1, x_2, , x_n\}$ which can identify the class these inputs using ANNs [78]. A feature selection process can be performed based on the computation of ANNs that compute a new set of outputs from provided inputs X which can be denoted as output Y. The selection of the features is made by considering only those features contains useful information.

Context-aware recommender systems go beyond traditional personalized recommendation models as they correspond not only to the user's preference profile but also consider the given situation and context. However, the selection and incorporation of relevant contextual features in context-aware recommender systems are always challenging. In the domain of context-aware recommender systems, it is assumed that not all the features from test collections are relevant. Most of the time, users show their interests for some items/movies in different circumstances or contexts. There are several methods to identify the relevance of a context in context-aware recommender systems. However. mostly, the relevance of the contextual data is determined manually using the advice of an expert with the domain knowledge. However, there are many machine learning and data mining techniques that can also be used to determine the relevancy of contextual features. These techniques work with the rating data that is initially collected by the users during the data preprocessing phase. As part of our work, we investigated the role of different contextual, user and movie features by using Artificial Neural Networks as feature selection approach.

We evaluated different representations (feature sets) from the given dataset (LDOS-CoMoDa) for contextual recommendations, by comparing the different features sets. We form these representations (Feature sets) from our test collections assuming the nature of the data. The available contextual and non-contextual features are divided based on the dynamic and static nature of data. The non-contextual features are further divided into sub-categories/representation by separating the features that can be normalised using the data preprocessing and normalising technique described in Section 3.1.1. For example, the dataset contains movie's features, user's and contextual features. These features in the test collections can be divided among the *dynmaic* and *static* in nature. The contextual features are more dynamic in nature than the user's features and the movie's features. So we assume all contextual features as dynamic features and represent them as category/representation *Dynamic* in our test collection. Now we have the users' features which are static in nature so we put them into the category/representation of the user. The movie features are also identified as more static in nature, however, by analysing the attributes, we decided to form four representations out of movie features so that we can have more insight into the classification of ratings when and ANNs based approach can deeply and more correctly provide the accuracy on these representations. The four representations from movie features are named as *Movie* which contains the movie information as shown in Table 4.1, *Maker* contains the features about director and budget, *Stars* which contains the features of cast of the movie and representation with the title *Category* which contains the features related to the genre of the movies.

We further cross-compare these representations to select the useful and relevant category of the features from different representations and their combination as well as the individual features from the dominant representation. Another contribution of this work is the comparison of the performance of standard matrix factorization to Artificial Neural Networks (ANNs) in CARs. Since relevancy of the contextual features can be determined by the different machine learning and statistical techniques. One of the dominant technique in this field is Matrix Factorization which is used by the author in [99] for the chosen dataset LDOS-CoMoDa. Our evaluation show and confirm that dynamic, contextual features are highly dominant compared to non-contextual ones in the given data set [96]. We also show that ANNs slightly outperform matrix factorization approaches typically used in CARs for the detecting of the relevant category/representation of the features. Finally, in our discussion, we briefly outline the usage of ANNs as an alternative approach to detect the relevant representation of the features and how ANN-based approach can potentially mean to guide the feature selection process in the context-aware recommender system.

In this chapter, our main focus is on an important sub-problem in the domain of recommender system; the selection of the relevant features from all types of movie features, contextual features, and user features using initial set of the ratings (e.g., 1 to 5 stars) that a user might give a certain items/movies. In a later step, this prediction can be used for recommendation, for instance by recommending items with a predicted rating of 5 stars.

To predict ratings, machine learning algorithms are reported in the literature to develop models and find patterns based on training data [128]. In context-aware recommender systems, the selection of the appropriate context feature remains a persisting challenge [100]. In CARs, using too many context features may result in low accuracy and high dimensionality in the process of recommendation. Recommendation algorithms usually depend on the assumption that the features selected in advance will result in better accuracy [147]. However, this is not the case all the times. Although, in context-aware recommender systems, some features such as location have performed well in many cases. However, it is hard to decide on some of the features whether those are relevant or irrelevant such as the value of the stock market is less relevant than the purpose of buying a book while dealing with contextual information for book recommender systems. Similarly, the currency value/rate may be less relevant while booking a trip than the leisure or time information. Our test collections from chosen dataset LDOS-CoMoDa, which is rich in contextual features as well as user and movie features, we have space to identify, discuss and describe the role of features. That is the reason why we are more interested to look at the comparison of the features from our test collection using our proposed methodology as Artificial Neural Network which has not been much used in particular domain of context-aware recommender systems.

The aim of this chapter is to gain more insights to aid the feature selection process. We investigate different feature sets (which we call *representations*) and their performance either as a single representation or combination of features. To conduct our studies we use LDOS-CoMoDa, which is the most prominent collection of contextual movie recommendation [150]. This is a very specific collection for the evaluation of CARs as it contains *dynamic contextual features* like location, mood, etc as well as static features for the users and the movies. Previous work on the similar domain has shown that applying dynamic features leads to highly accurate results. However, previous work has only considered dynamic contextual features and find the relevancy using statistical matrix factorization method but did not look at other available non-contextual ones (like gender, movie type, etc). As well as the previous method does not describe the role of other machine learning techniques that can also be used to predict the relevancy or over a baseline method. Hence one aim of this study is to check the performance of non-contextual features, either alone or combined with contextual ones. Furthermore, we show that utilising Artificial Neural Networks (ANNs) instead of matrix factorization, which is prominent in CARs, improves the performance of the rating categorization.

4.2 Comparison of different categories of features using proposed approach

In order to compare the different contextual features, we introduce our ANN-based approach which is composed of a three layers architecture as illustrated in Figure 4.1, consisting of an input, hidden and output layers. The input layer is composed of 6 *representations*, which are provided as input to ANNs to predict the output y that represents the ratings from 1 to 5 from an initial set of ratings. These representations are manually formed based on the nature of the different contextual attributes and explained at the end of this section. An overview of the nature of different available features in LDOS-CoMoDa dataset are shown across the contextual vs non-contextual features as well as dynamic vs static features in given Figure 4.2 The different representations are also combined as input, for example, the Dynamic representation is combined with the Category and User to find a better match in terms of accuracy which is computed from the accurate classifications of the given inputs. The rating data is made as the target data in the proposed ANN based approach to model the input over the target data. Each of the representations and their combinations is evaluated against the target data, which is the rating data that comes from the users. The user rating (1-5) is transformed into binary rating as explained in the data preprocessing part of the proposed approach so that the ANNs can be trained on the collections. A good set of representations of the context features will be identified at the end and will be recommended based on the accuracy that comes from the ANNs for each input. The accuracy of the classified ratings will be used to describe the relevancy of the input categories/representations, in our proposed approach A brief description of the different representations with respect to the list of features is given in the Table 4.1.

In order to train ANNs on the different manually formed representations (formed based on the nature of the data that is static versus dynamic and contextual versus non-contextual), we then normalize the contextual features as per the approach described in our proposed approach. Since ANNs work better on the normalized data for the predictions and classification, we, therefore, described the approach in Section 3.2 of the third chapter, how to normalize the data for ANNs. This results in better accuracy for different features and their combinations. The hidden layer, a feed-forward multi-layer perceptron neural network, is used to map the input into the output binary classes y. The detail systematic view is shown in Figure 4.1.

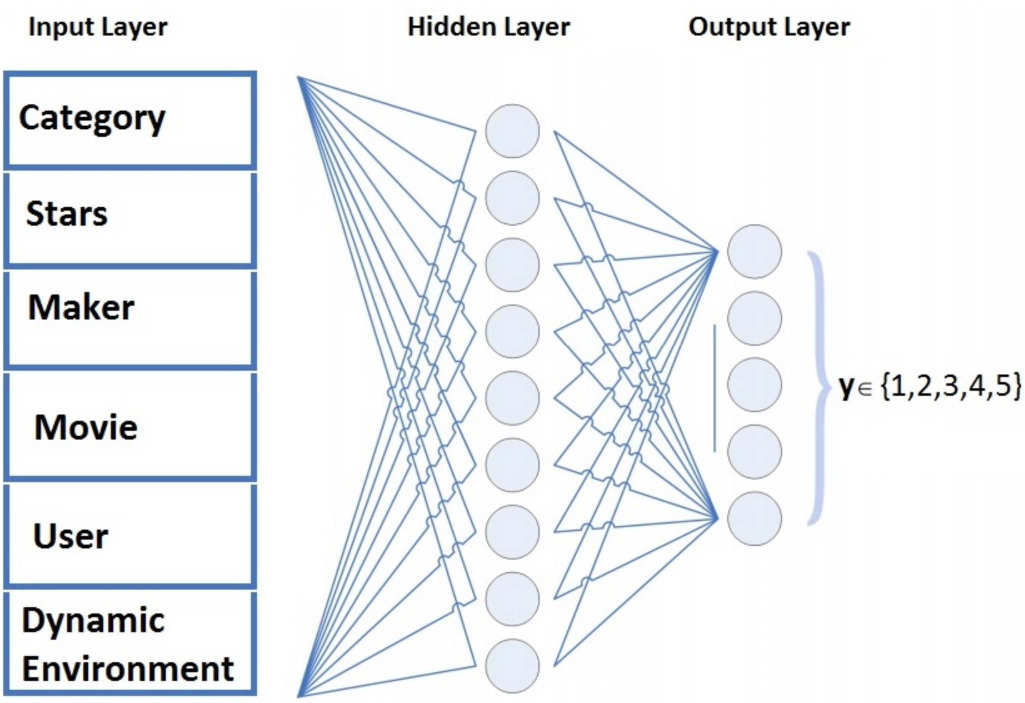

Fig. 4.1 Architecture of the proposed ANN approach for rating prediction

We pre-process the data to train a model using neural networks. The features available in the dataset are a dynamic set of features and static features. The dynamic set of features are assumed as contextual features which incorporate the additional contextual information such as time, season, location etc. The static features are static in nature assumed as the non-contextual features which incorporate the additional static information such as user features as well as details of actors and genres of the movies. The different types of contextual and

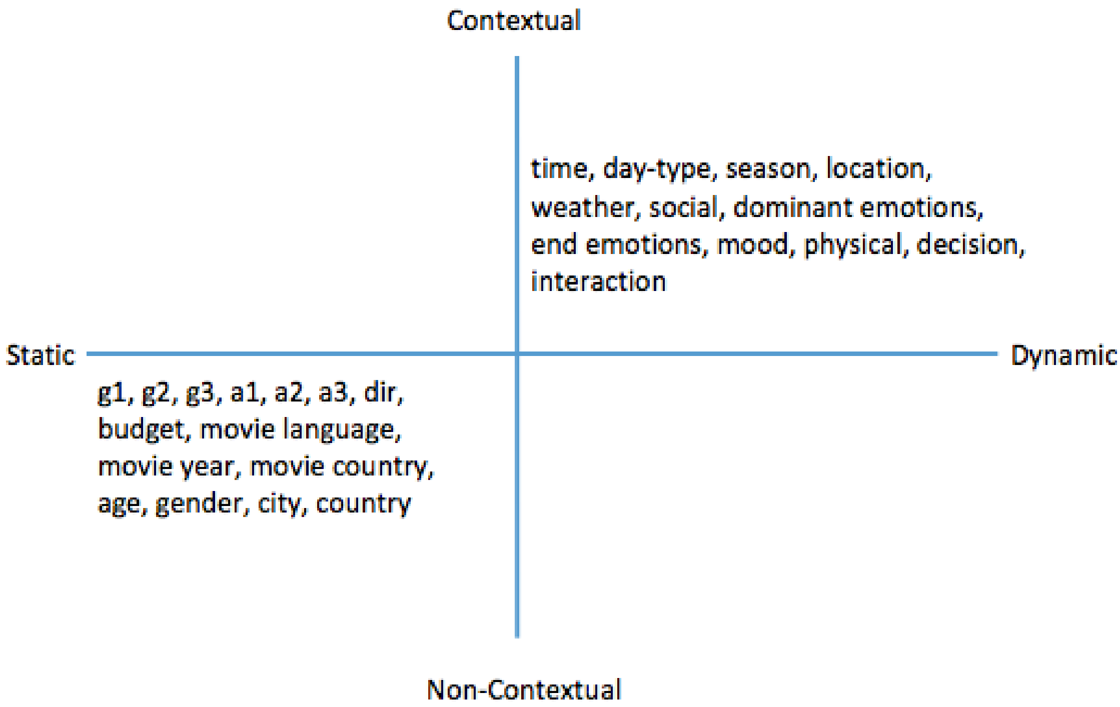

Fig. 4.2 Distribution of different features from LDOS-CoMoDa Dataset

non-contextual features in the LDOS-CoMoDa dataset are illustrated in Fig. 4.2.

The different features available in the dataset are manually categorized into 6 representations based on the type of contextual attributes as described in Table 4.1. Each of the representations is evaluated against the target data (user ratings) in our experiments. Since ANNs are binary classifiers, the target data is converted into binary representations for comparison and evaluation. To turn each of the 5 classes into a binary classification decision, each of the 5 possible ratings is compared to the rest as a yes/no decision (e.g. "Class 1 / not Class 1" to decide if the rating was 1 or not), resulting in 5 classes Class 1 to Class 5. The detailed methodology of transformation of rating data into binary rating data is described in section 3.2 of chapter 3. In comparison, matrix factorization considers the ratings provided by the users for the items to map the users and the items in a joint latent feature space [38].

Different representations are also combined to find a better match in terms of accuracy, with less error rate. After evaluating the different representations and their combinations, the optimal feature set with the highest accuracy will be considered for the recommendation process. Note that an item might be classified into more than one of the above classes (e.g.,

Table 4.1 Representations and Features from LDOS-CoMoDa dataset

Representation	**List of Features**
Category	g1, g2, g3 (Genres of the movie)
Stars	a1, a2, a3 (Actors of the movie)
Maker	dir, budget
Movie	movie language, movie year, movie country
User	age, gender, city, country
Dynamic Environment	time, day-type, season, location, weather, season, dominant emotions, end emotion, mood, physical, decision, interaction

the ANN may predict 1 star and 4 stars based on the single binary decisions). In this case, our policy is to select the highest rating prediction. The prediction of the ratings for items will also allow to rank items, that have not been rated, to improve the recommendation process. Since other approaches such as probabilistic neural networks are slower than multilayer perceptron networks and require more memory space to store the model, they are not better options at this stage [39].

Following the representations that are given in the Table 4.1, the contextual features are distributed among 6 representations. The *Category* representation consist of movie genres which show each movie is presented in three genres. The representation of *Stars* consists of the cast of movies, whereas the *Maker* representation contains information about the director of the movies as well as the budget. The representation *Movie* consist of information about the movie country, movie language and movie year. The representation *User* consists of the static information of users including age, gender, city, and country of the user. The *Dynamic Environment* representation contains dynamic variables such as time, day type, season, location, weather, social, dominant emotions, end emotions, mood, physical, decision, and interaction. Different representations with the associated contextual information from the LDOS Comoda dataset are shown in Table 4.2.

Once the different representations are identified, a multilayer perceptron model based neural network is trained to compare every single representation and combinations thereof with the target data to evaluate the performance and accuracy of the different context features. ANNs classify based on the rating data so here we called at this point prediction of the ratings. The better set of representations of the context features will be identified and recommended based on the experiments. Further details are provided in the next section.

Table 4.2 Basic Statistics of LDOS-CoMoDa

Users/Items	121/4381	**Ratings**	2296
Rating scales	1–5	**Context factors**	12
User attributes	4	**Item attributes**	7

4.3 Evaluation

In this section, we briefly describe the dataset and the methodology used for our experiments. First of all, we examine the dataset to find which information can be used as a potential context. Based on the structure of the dataset we define an approach how different representation can be formed based on the nature of the contextual features. In order to evaluate our results for movie rating prediction, we refer to the evaluation measure briefly explained in Chapter 4, in which we described evaluation measures. Since we are using an ANNs based approach, the results we are having in the form of confusion matrix represented by True Positive, False Positive, True Negative and False Negative form. We observed these values and calculated that how accurately the given input representation can be classified with the initial rating data.

4.3.1 Dataset

The chosen dataset LDOS-CoMoDa[1] consists of 4381 movies which are rated by 121 users. The number of ratings available in this dataset is 2296 and the maximum number of ratings provided by a single user is 220; the minimum number of ratings is 1. The dataset consists of 12 contextual variables in addition to static user information. The basic statistics are given in Table 4.2.

In order to evaluate the performance of different representations using binary classification, the true positive rate versus false positive rate is used as part of accuracy matrices [120].

4.3.2 Results

In this section, the different representations derived from the given contextual variables in the LDOS-CoMoDa dataset are evaluated, presented and discussed. The work presented in [100] on detecting the relevant context in movie recommender systems provides the relevance and irrelevance of contextual variables. However, we can categorize the contextual variables into the 6 different representations discussed above and cross-compare the representations as well

[1] http://www.ldos.si/comoda.html

Table 4.3 Sampling from ANN

Entire dataset size	No. of Samples	Cross-Entropy Measure	% Error
Training dataset size	1608	1.4213	3.17
Selection dataset size	344	3.85	2.03
Validation dataset size	344	3.87	3.19

as their combinations to find successful sets of contextual representations. In order to train the neural network on the chosen dataset, the data is preprocessed and normalized in the first stage. The rating data is transformed into a binary form as the neural network performs better using binary classifications. Different features available in the dataset are then normalized with respect to the number of available context features. Then, the ANN is trained using the methodology described above and the samples are divided among the training data, selection data, and the validation data.

The statistics from the ANN samples division are giving in the Table 4.3. The number of the samples used by the Neural Network for training is 1608 (70%), 344 for the selection purpose and 344 for the validation. The cross-entropy during the training stage of ANN was measured as 1.4213, which shows a small fraction of the error occurring during the training stage. The error percentage in the training stage is 3.17% which shows a small fraction of samples is misclassified during the training stage. Similarly, the selection stage of the Neural Network utilizes 244 (15%) samples with cross-entropy 3.85 and error percentage 2.03. The validation stage also utilizes 344 samples (15%) with cross-entropy 3.87 and the percentage of error at 3.19. We have also tried the combinations of all representations and observed the higher error rate of 62.20% which shows that it is not an ideal condition to use the features from all representations. A full intersection of the all six representations is not better matched, however, a combination given in Table 4.4 performed at the rate of 80% which shows the intersection of the Category, User and Dynamic can perform better in the scenario.

Using the approach described above, we cross-compared the different representations with the target data to find the relevant representation (i.e. set of features). Features are cross-compared one by one by training the neural network which learns over 2296 samples (70% for training, 15% for testing and 15% for validation).

We use the results reported in [99], using matrix factorization (MF) as a baseline to compare the performance of our ANN approach since it utilizes the contextual attributes

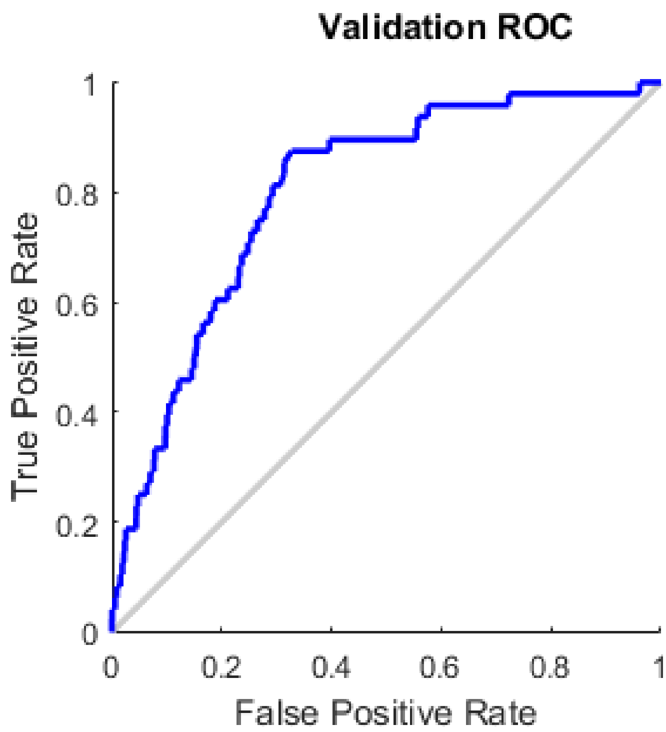

Fig. 4.3 ROC Curve of Dynamic Representation

Table 4.4 Performance of different features using ANN and MF (where reported in the literature)

Representation	Performance (Accuracy)	MF
Dynamic	**97.12**	**96.9**
Category	80.68	Not Reported
Makers	65.8	Not Reported
Stars	66.71	Not Reported
Where	66.58	Not Reported
User	64.9	Not Reported
Category + User + Dynamic	80.81	Not Reported

which are part of the Dynamic Environment in our approach. The results comparison in Table 4.4 between the contextual attributes in the baseline method on the one hand and the ANNs, on the other hand, shows that contextual attributes performed better with ANN than with MF.

After comparing ANNs to MF, we looked at the performance of the different representations or feature sets. As we can see in the Table 4.4, the context features available in the Dynamic Environment representation performed better while the other representations struggle with respect to the performance and errors. So the Dynamic Environment representation is picked as the single optimal set of features. We also tried combinations of Dynamic Environment with other representations such as Category, Makers, Stars and User Statics to study combinations of representations. The comparison of combinations given in Table 4.4 shows that the good performance of the Dynamic Environment is not further improved when combining this representation with others; the representations do not seem to complement each other. This means Dynamic Environment is indeed the dominant representation in the LDOS-CoMoDa collection.

We confirmed the role of dynamic contextual variables in the aforementioned comparison, we now have a closer look at the dynamic representation. The different dynamic features are tried one by one as per the approach defined above to investigate the role of each variable. In this case, the input layer of the proposed methodology consists of one contextual variable out of 12 contextual variables from the Dynamic variables/features. A detailed description of the contextual variables in the dynamic representation is given in Table 4.5. The number of attributes, at this stage, helps to normalize the contextual variables before training ANN on the data.

Each contextual attribute in the dynamic category is chosen as input to the proposed methodology. The ANN is trained on each contextual feature to observe and discuss its role in the Dynamic category as a step towards the design of an interactive recommender system. Once again the sampling ratio remains same as mentioned in Table 4.3 for training, selection and validation processes. The accuracy is calculated for each contextual feature and reported in table 4.6. Our evaluation shows the dominant role of the Location feature with the accuracy of 97.8%. The mood is another prominent contextual feature which shows the accuracy of 97.6%. Physical and Interaction features contribute equally with 97.4% while Dominant emotions and end emotions show the accuracy of 97.3% and 97.2% respectively. The role of Decision and day-type features have also shown the good accuracy of 97.3%

Table 4.5 Dynamic Contextual variables from LDOS-CoMoDa

Dynamic Variable	No. of Attributes	**Description**
Time	4	Morning, Afternoon, Evening, Night
Season	4	Spring, Summer, Autumn, Winter
Location	3	Home, Public Place, Friend's House
Weather	5	Sunny, rainy, stormy, snowy, cloudy
Mood	3	Positive, Natural, Negative
Physical	2	Healthy, ill
Decision	2	User's Choice, Given by others
Interaction	2	First, nth
Day-type	3	Working day, Weekend, Holiday
Social	7	alone, partner, friends, colleagues, parents, public, family
DominantEmo	7	sad, happy, scared, surprised, angry, disgusted, neutral
EndEmo	7	sad, happy, scared, surprised, angry, disgusted, neutral

and 97.2%. However, the role of time, season weather and social features from Dynamic category have shown less level of accuracy in comparison with other features in the same category with 96.3% for the time feature, 96.9% for season feature, 96.5% for weather and 96.7% for the social feature. The detailed results are presented in Table 4.6. We take top eight contextual features with higher accuracy from representation for further experiments from aforementioned results in Table 4.6.

The performance of the ANN is also evaluated by computing the Cross-Entropy which helps to evaluate the performance of three different stages of ANN (Train, Validation, and Test) against the best performance. The results presented in the Table 4.3 shows that the performance of ANN remains better for all three stages when the ANN is trained for 22 epochs. In ANNs, an epoch is used to present the set of training vectors to the network for the calculation of new weights.

4.3.3 Discussion

The results have shown that contextual Dynamic Environment features by far outperform the static non-contextual features in the chosen LDOS-CoMoDa collection when it comes to rating prediction. The results also show that applying ANNs instead of matrix factorization

Table 4.6 Performance of contextual dynamic attributes

Dynamic Variable	**Accuracy %**
Location	**97.8**
Mood	97.6
Physical	97.4
Interaction	97.4
Decision	97.3
DominantEmo	97.3
Daytype	97.2
EndEmo	97.2
Season	96.9
Social	96.7
Weather	96.5
Time	96.3

improves the rating prediction accuracy even further when using the Dynamic Environment features. It confirms the important role of contextual features for CARs and the rather inferior role non-contextual features play, at least in the given dataset. ANNs are indeed a very effective method for rating prediction, which is crucial for the context-based recommendation.

The further experiments on the Dynamic category have confirmed the important role of contextual features such as location, mood, physical and interaction, which can play a key role in the development of an interactive context-aware recommender system. The location feature has been used in most of the context-aware recommender systems without discussing or comparing it with the other contextual features. Whereas the domain specialists believe that location is a key feature in most of the context-aware recommender system, in the case of LDOS-CoMoDa dataset, the proposed ANNs based methodology also confirmed the role of location. ANNs based approach for the comparison of the contextual features plays a handy role to determine the relevant features. In our proposed methodology, we not only compare the location feature as part of a category/representation but also an individual feature for finding its relevancy through classification of the provided ratings. In this section, we introduced ANNs based approach for comparing the contextual features, user features and movie features as an alternative approach which can be helpful to select a feature, while dealing with a standalone feature or a category/representation of features set. We presented how to form different representations from a chosen dataset LDOS-CoMoDa. Different representations are cross-compared and we concluded that the Dynamic Environment context features performed best when applied alone, also outperforming the chosen matrix factorization baseline method. We further cross-compared combinations of the Dynamic Environment

with other representations and observed that they do not perform well and are even not able to further complement the dynamic features. We also investigated the role of different features within the Dynamic category and the contribution of each contextual feature within Dynamic representation is reported.
We further include these contextual features with the user-item matrix, user features, and movie features to generate contextual recommendations using rating data and present in next Chapter 5.

4.4 Summary and Conclusion

In order to determine the relevancy of the different contextual, user and movie features, there are a different number of approaches have been used. In particular domain of contextual recommendations, machine learning algorithms and manual selection from the domain specialists are two more common and well-known techniques to find the relevant context. We used our proposed ANNs based methodology to identify the relevant and irrelevant contextual, movie and user's features over a strong machine learning technique. We divide our collections into different representations that are used as input with the rating data and further used them in our proposed methodology. We, then, select ratings as target data to model input over ratings so that we can observe the accuracy of correctly classified input data. We trained the neural network and store output y to evaluate the results of the approach. In order to evaluate the output results, we use evaluation measure described in this chapter. We, then, reported our results and compared with the baseline method from the literature. We also confirmed the importance of contextual features which are dynamic in nature to those non-contextual features which are static in nature. The top-performing contextual features with higher accuracy are selected for the further experiments that we discuss in next chapter.

Chapter 5

Contextual Recommendation using an ANN based approach

In contextual recommendations, the main aim is to suggest to users which item they might be interested in. The interest of the user is highly influenced by the context in which the item was rated to further predict what a user will do in some particular context. In this chapter, we explore the linkage between the context and other features of movies and users from our chosen test collection LDOS-CoMoDa. In the previous chapter, we cross compare the different contextual, user and movie features. In this chapter, we use different user features, movie features, and contextual features to generate contextual recommendations that refer to our contribution C2. For this purpose, we use our proposed Artificial Neural Networks based approach described in Section 3.1. We also use some most developed machine learning approach such as Support Vector Machine (SVM), and Principal Component Analysis (PCA) to compare our results. We also compare our results with some user-based studies in the same domain. We also evaluate our results with widely used evaluation techniques in context-aware recommender systems that further help to compare our results with other machine learning techniques as well as with results from the other researchers in the same domain.This chapter referred to the objective O3 that focus on evaluation and analysis performance by defining a minimum contextual attribute subset which can generate more accurate contextual recommendations where the proposed methodology is used to integrate the features into the recommendation process as per O4. We present our experiments and results on the LDOS-CoMoDa dataset and describe each experiment in this chapter. We discuss our results and make recommendations.

5.1 Introduction

Recommender systems aim to recommend the item to users in which a user can be interested in looking into the interaction of the users with the system. In order to use contextual information in the recommendation process, a more recent trend in context-aware recommender systems is known as preferences elicitation and estimation. The method attempts to model the user preferences by learning on observations and interactions of the user with the system. The technique obtains feedback or preferences from the end users on different items that are recommended to the user in different contexts. In order to model context into recommender systems and generate recommendations, this technique adopts the traditional methods of collaborative filtering and/or applies intelligent data analysis techniques from the domain of machine learning or data mining such as support vector machine (SVM) and Bayesian Classifiers. The interest of the user in a particular item is highly influenced by the context variables. While a number of machine learning and data mining techniques have been used for context-aware recommender systems so far, we propose an Artificial Neural Network based approach which is a binary classifier to generate contextual recommendations using the LDOS-CoMoDa dataset. This context variable defines the situation in which a user likes the item. At this stage, we know the fact that the interest of the user is not only influenced by the contents of the items but also by the context in which the user has made choices. In this chapter, we present our experiments based on the link between the user features, item features and the context variables that can be used to recommend items to the users.

For this purpose, we used the LDOS-CoMoDa dataset which is rich in terms of contextual features, user features, and movie features. We further showed that the kind of features from all three categories of user features, movie features, and contextual features could be used to make recommendations with better accuracy. We used our proposed approach which is defined in Chapter 4 and present our results using some widely used machine learning algorithms. The remainder of the chapter includes the brief overview of the methodology, experiments on different feature sets, results, and discussion.

5.2 Overview of User Features, Movie Features and Contextual Features

In Chapter 4, we confirmed the importance of Dynamic features that consist of contextual features from the LDOS-CoMoDa dataset. Although user features and movie features result in less accuracy while predicting the ratings, however, the role of user features and movie features in contextual recommendations is vital. In this section, we will have another look at the user features, movie features and contextual features from our test collections before we use these features in contextual recommendations.

In contextual recommendations, it is assumed that the interests of the users are strongly influenced by the circumstances (Contexts), in which decisions are taken. The role of the user features and movie features play a key role in the interests of the users such as age group matters while watching a movie or genre of the movie or for some users cast of the movies has a great impact on their decision about watching movies. So instead of ignoring the user features and movie features, we conducted our experiments on the dataset to recommend movies, by assuming four conditions.

1. Contextual recommendations with user features
2. Contextual recommendations without user features
3. Contextual recommendations with movie features
4. Contextual recommendations without movie features

Although we use different contextual variables for the prediction of the ratings which is a sub-problem of context-aware recommender systems, in which a *Location* attribute presents better results than the other contextual variables, however, the role of remaining contextual variables is important. It is one of the widely researched topics too in which researchers are trying to justify the role of each contextual variable for contextual recommendations. An example work presented in [149] describes the role of emotions in contextual movie recommendations in which the author has described how the user interests can be influenced by their emotions and how emotions can be incorporated to recommend movies to the users.

At this level, an interesting question could arise of which context we will use. Each contextual feature from the LDOS-CoMoDa dataset can have a great impact on the interests of the user. The dataset is one of the richest in terms of contextual features and provides the best scenario to analyze the role of different contextual features for generating movie recommendations. In the following Table 5.1, we give an overview of the user features, movie features and contextual features from test collections.

As part of our work, we use all contextual attributes one by one to recommend the movies from test collection. We set four conditions to record and compare our results so that we can

Table 5.1 List of Features from LDOS-CoMoDa dataset

Movie Features	movie language, movie year, movie country , dir, budget, a1, a2, a3 (Actors of the movie), g1, g2, g3 (Genres of the movie)
User Features	age, gender, city, country
Contextual Features	time, day-type, season, location, weather, season, dominant emotions, end emotion, mood, physical, decision, interaction

find a better way and better match of features to recommend the user the movies. In the next section, we describe how we will compare our results with the proposed approach with other machine learning models.

5.3 Comparison of Proposed Methodology with different Machine Learning Models

In order to generate contextual recommendations, we have separated each feature from the test collection so that we can use the different mix and match of features. The proposed methodology explained in chapter 3 (Section 3.1.1), will be followed where we have the test data in processed and transformed form. The transformed data is further imported into our workplace one by one to make use of each individual feature and select the better one with good recommendation accuracy. A generic overview of the recommendation schema is given in the following Figure 5.1 in which user features, movie features and contextual features will be used as input to the proposed ANN based approach. The final recommendations will be evaluated using the evaluation scheme described in Chapter 3 (Section 3.2.3).

Once we have the recommendation accuracy of different feature sets (Contextual, user, movies), we will also run some reference machine learning algorithms on the same test collection. We have selected two well-established machine learning algorithms: Support Vector Machine (SVM) and Principle Component Analysis (PCA) to compare our results. SVM and PCA are two well-established models used for recommender systems. We will compare the accuracy of recommendations of our ANN based approach with PCA and SVM while using them on the same test collection at the same time.

Once we have discussed the conceptual and theoretical concepts, in the next section we discuss our experiments on the test collection, results and discussions.

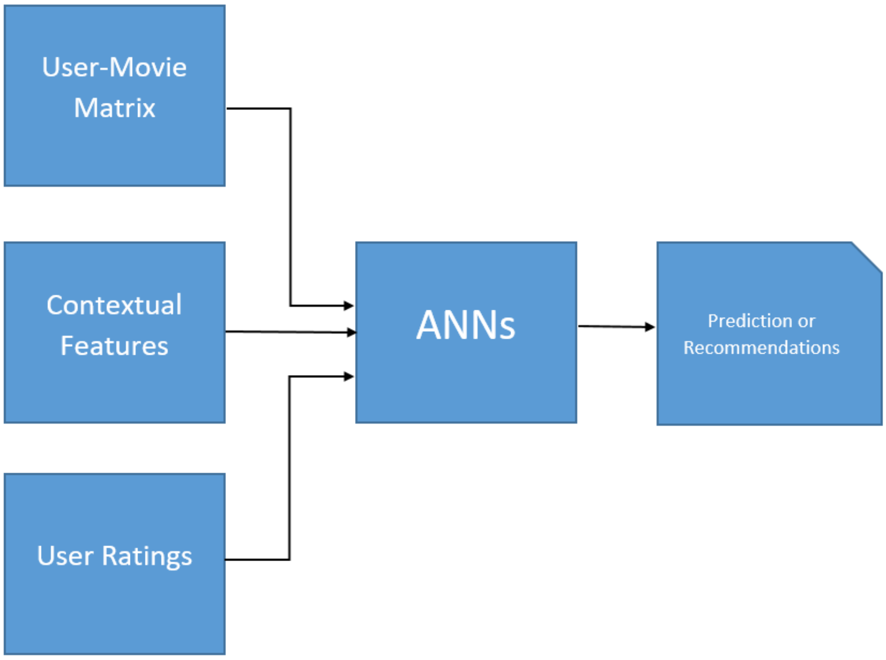

Fig. 5.1 A general Overview of the contextual recommendation process

5.4 Experiments

In context-aware recommender systems, user interests have key importance. User interests are more related to the circumstances in which decisions are made. These circumstances are the contexts that are used to generate contextual recommendations. In this work, we connected the preferences of the users on movies using our test collections as LDOS-CoMoDa with the contexts in which those movies were watched. We performed our experiments using ANNs in which we trained a multilayer perceptron model on the user data, ratings, contexts and movie features. ANNs, by default, divide the input data into three phases: training, validation, and evaluation where we selected 10-folds cross-validation.

For evaluation purposes, we have divided movie ratings from our test collections into two categories zeros (0) and one (1) as described in Chapter 3 (Section 3.1.1). The category with zeros represents the movies that are not recommended while rating category with one represents the movies that are recommended for the user. Based on the classification, being classified as one is considered as True Positive (Recommended Movies). While zero is classified as zero which is a negative condition True Negative (Not Recommended Movies).

5.4.1 Impact of Movie's features in Contextual Recommendations

In this section, we explored the link between features of movies and the context when the movie is selected or watched to better understand how contextual information from the chosen dataset LDOS-CoMoDa, such as location, weather and time of the day can influence the movie selection. We showed with experiments that by using preferences in particular contexts, we are able to recommend to the user in a given context based on the preferences of the user. In this section, we further discuss the influence of movie features, when these are selected, to compute recommendation accuracy from algorithm generated results.

In order to generate recommendations, we further divide our dataset into two different parts. We assumed that the ratings 4 and 5 are the positive ratings that can be used for generating recommendations while ratings below 4 are the negative ratings that can be considered as not recommended. We further divided our dataset into all contextual variables and all movie features as individual inputs to the system. We have tried all the movie features together and individually to compare the accuracy of the recommendations by considering eight contextual features. We would like to refer our results from the previous chapter given in Table 4.6, in which we have shown the analysis of contextual features through ANNs as a binary classifier. We do not take $time, season, weather, social$ into account because of their low classification performance in comparison with the other eight contextual features. A detailed feature analysis and selection approach is described in Chapter 4 (Section 4.2). We follow our proposed approach, an ANN based system to generate recommendations and evaluate our results in terms of recommendation accuracy. We also compare our approach with other machine learning algorithms such as SVM, Logistic Regression, and PCA as well as presenting our results in this section. The details of the results from our proposed technique and other machine learning algorithms used are described in the following section of results.

Results

Considering this point that some of the features of movies will result in better recommendation accuracy, we use them as input with all 12 contextual attributes (Contextual Features) of the LDOS-CoMoDa dataset. First of all, we use all movie features with all contextual features to see an overall accuracy of the recommendations. We divide our dataset into three different samples training, validation, and testing while we evaluate the accuracy of the testing phase which contains test data. The default test data percentage is 15% in ANNs based approach so we consider this ratio while evaluating our results. When we combine all

Table 5.2 The comparison of different machine learning algorithms for contextual recommendations from Movie features

PCA Accuracy	SVM Accuracy	L-Regression Accuracy	ANNs Accuracy
66.5	74.1	66.5	75.1

Movie features and all contextual features as an input to the proposed approach to generate recommendations, we observe an accuracy of 75.1%. The recommendation accuracy is determined and evaluated based on the criteria that we explained in section 3.2.3 of chapter 4.

In order to compare our proposed approach, which is based on ANNs, we also used the well-known machine learning algorithms such as PCA, SVM and Logistic Regression on the same test set. We observe different values while PCA provides us an accuracy of recommendation 66.5%, SVM hit the highest value at 74.1%, while, in comparison, Logistic Regression provide 66.5% recommendation accuracy. In this case, PCA and logistic regression model provide the same level of accuracy on test collections. The results are observed and presented in Table 5.2.

Once we have the recommendation accuracy of all movie features as one combination, we observed that the role of Logistic Regression does not give accurate results in comparison to the remaining three approaches. Although ANN based approaches have outperformed the well-established SVM and PCA algorithms, still their results are comparatively better than Logistic Regression. So at this stage, we drop our idea to further compare our results with the Logistic Regression and take SVM and PCA as baselines to compare the results.

As we mentioned earlier, we divided our data into a features wise approach (Each feature as separate input), now, we further cross-compared the role of each movie feature into contextual recommendations for movies. For this purpose, we incorporate eight contextual features and one movie feature as an input to the system with a user-item matrix. We observed the role of each feature by following the same pattern of evaluation. We observed the accuracy of movie feature Budget as 75.9% using proposed ANNs based approach while using the Budget feature with SVM and PCA, the recommendation accuracy is 75.2% for SVM and 63.8% for PCA. Similarly, movie feature Year recommends movies at the accuracy of 66.5% with PCA, 74.4% with SVM with positive improvements and 75.8% with ANN based approach. The ANNs based approach outperformed so far by a fair margin to both PCA and SVM for budget

Table 5.3 The comparison of different machine learning algorithms for contextual recommendations from Movie features

	PCA	SVM	ANN
Movie Feature	Accuracy	Accuracy	Accuracy
Budget	63.8	75.2	75.9
Year	66.5	75.4	75.8
Country	66.5	75.4	76.0
Language	66.5	75.4	76.2
Director	66.5	75.3	76.5
Genre	66.5	75.5	76.6
Actor	66.5	75.0	76.6

and year features of the movie. We further look into the role of Country feature and observed the accuracy as 66.5% with PCA, 75.5% with SVM and 76.0% with ANNs. The movie features country also predict better results with the ANNs based approach than SVM and PCA. Language feature contributed in contextual recommendations of movies at the accuracy of 66.5% with PCA, 75.4% with SVM while 76.2% with ANNs. The *Language* feature shows better improvements with ANNs than SVM and PCA. The movie feature *Director* has shown a similar accuracy value of 65.5% with PCA, while SVM reports 75.3% accuracy. In comparison with PCA and SVM, ANN reports 75.5% accuracy. The notable value of director with the ANN-based approach makes it a better movie feature in comparison to other movie features that we have compared so far.

Similarly, PCA shows poor results for genre and actor features with accuracy 65.5% for both of the features. While, in comparison, SVM shows 75.5% for *Genre* feature and 75% for *actor* feature. Our proposed ANN based approach reports 76.6% accuracy for the genre and similar value of 76.6% for *actor* feature. In case of actor a close comparison is noted between the proposed approach and SVM as both show better accuracy for this feature. The observed results are presented in Table 5.3. The evaluation of each movie feature, individually, shows that the role of *Genre* and *Actors* is stronger with the higher accuracy rate in comparison to movie feature *year* which shows slightly worse results in comparison.

In order to try different sets of movie features, we also use a feature elimination approach. In this approach, we simply took all the movie features with the eight contextual features. Based on the results from Chapter 4 (Section 4.3.2), in which four contextual features {time, season, weather, social} provide less accuracy than other contextual features such as location etc. We simply eliminate those four features for the time being and perform our experiments on the remaining eight contextual features. In this section, we start trying all seven movie features and eliminate each movie feature after every experiment. So in this way, we start the experiment with all seven movie features and eight contextual features. After every entry, one movie feature is eliminated so we can observe the accuracy of the feature set. In all cases, the contextual features remain the same in number i.e. we use eight contextual features for every experiment in this section. Once the role of the ANN-based approach is confirmed, while predicting the recommendations, in our feature elimination technique we only focus on our proposed ANN based approach.

In this technique, we simply combine all of the movie features but in every instance, we eliminate (Remove) one movie feature from the combination. We use eight contextual features to run our experiments with the user-item matrix and rating data as target data. The experiment is performed using our proposed ANN based approach and results are given in Table 5.4. We observed the behavior of the recommendation process, first of all, by removing the *budget* feature. When we remove movie feature *Budget* from the combination we observed the accuracy of contextual recommendation process is 75.6%. When we remove the movie feature *Year* and added *Budget* back to the combination, we observed accuracy as 75.2%. The accuracy of the recommendation process further decreases to 74.9% when we remove movie feature *Country*.

However, the accuracy of the process increases to 76.5% when we remove movie feature *Language*. When we remove the movie feature *Director*, we observed accuracy as 75.3%. We noted more decrement in accuracy when we removed movie feature *Genre* from the process with the value of 72%. This confirms how important the role *Genre* can play from our test collections when generating contextual recommendations. Similarly, the role of the actor is also confirmed when we remove this movie feature, we also get less accurate results.

By looking at the results given in Table 5.3, we can say that ANN outperformed the well-established SVM and PCA algorithms in contextual recommendations using the Lods-Comoda dataset. Once the role of the ANN-based approach is confirmed with a continuous better performance, we used it on the next step to try different combinations so that we can look into the accuracy of different combinations of those movie features. This will further help to identify a combination of movie features that can be used to recommend movies more accurately. In the next stage, we tried every possible combination of movie features and reported the accuracy. By looking at the results, we decided to not use the SVM and

Table 5.4 The comparison of different movie features for contextual recommendations using the elimination technique.

Eliminated Movie Feature	ANN
Movie Feature	Accuracy
Budget	75.6
Year	75.2
Country	74.9
Language	76.5
Director	75.3
Genre	72
Actor	74.3

PCA for further experiments because of the continuous poor results with our test collections.

In order to try the combinations of different movie features, we again incorporate a selected eight contextual features and combinations of the user features. We tried the combinations starting from all user features with all movie features, then, eliminated one movie feature out of the combination to try different combinations of the movie features. One example of the ongoing work from the same domain is given in [68] from the literature. We used results from the work given in [68] as a baseline to our results while dealing with the combinations of movie features. The baseline method works with prism algorithm in the environment of Weka while we are dealing with ANNs using the platform of Matlab. We observed the results and reported in Table 5.5. In order to try different combinations, we calculated the combinations that could possibly be tried to analyse the recommendation accuracy. We find that we have 34 possible combinations of movie features in chosen test collections. We further cross compared each and every combination and reported our results in Table 5.5. The comparison of different combinations of movie features shows a better accuracy and resulted in two combinations that can result in a higher accuracy of recommendation while contextual attributes are considered. This further refers to our hypothesis that incorporating the contextual features into movie features and user features results in better accuracy in the recommendation process. A comparison with the results selected from the literature shows that an ANN based approach predicts better results than the baseline method as the prism algorithm. The following two combinations of movie features can better recommend the movies to the users.

1. {Actors, Genre, Director, Country}

2. {Actors, Genre Director, Country, Budget}

The results are also compared with the results reported in [68], and analysis shows that ANNs provide better results in terms of accuracy of recommendations than the reported results.

Discussion

In this experiment, we investigated the role of movie features in context-aware movie recommender systems. First, we formed our test collection as per requirements and evaluated the accuracy of Movie features as a single set of features. Later on, we used each feature as an individual subset which is used an input to generate recommendations. Initially, we used all movie features and cross-compared the accuracy of our input set by comparing the results from proposed methodology, and some well-known machine learning algorithms including PCA, SVM and Logistic Regression. Our results show that our proposed approach which is based on ANNs outperformed the machine learning algorithms by a fair margin. We then evaluate the accuracy of each movie feature individually and compare the results with SVM and PCA.

On our test collection, the ANN-based approach again outperformed SVM and PCA with a higher accuracy rate. At this stage, we can confirm the important role of our proposed approach in the domain of context-aware recommender systems. We further investigated the role of combinations of movie features in contextual recommendation assuming that we can find a combination that can provide more accurate value in our test collections. The results are observed in Table 5.5, and a comparison with a baseline from the literature are presented. The comparison shows that our proposed ANN-based methodology provides more accuracy as well as having two possible combinations that can recommend with accuracy more than 77.5%. Based on the experiments, in this section, we can conclude that if we remove features such as *Budget, Year, Language* from our movie features set, we can have better recommendation accuracy. However, the movie features such as $Genre$,$Actors$ are shown as high important user features. In our results, we have shown that if we remove the Genre feature the accuracy of recommendation drop down to 72% which is lower than average accuracy we are getting through our experiments. Similarly, if we remove the movie feature $actor$, the accuracy of the recommendation process drops down to 74.3%.

A comprehensive overview of how we can use movie features and contextual data into recommender system is described in this section and a comparison of movie features is given. In the next section, we describe the role of user features in the contextual recommendation by using our test collections. Since the user features also play a crucial role in context-aware

Table 5.5 The evaluation and comparison of the different Movie features when incorporating with the context to generate recommendations

Movie Features							ANNs	Prism Algorithm
Actor	Genre	Dir	Country	Lang.	Year	Budget	Accuracy	Accuracy
X	X	X	X	X	X	X	75.1	52.52
X	X	X	X	X	X	-	75.6	53.24
X	X	X	X	X	-	X	75.2	53.96
X	X	X	X	-	X	X	76.5	51.32
X	X	X	-	X	X	X	74.9	53.00
X	X	-	X	X	X	X	75.3	52.28
X	-	X	X	X	X	X	72	52.28
-	X	X	X	X	X	X	74.3	47.96
X	X	X	X	X	-	-	75.5	52.76
X	X	X	X	-	X	-	74.7	50.72
X	X	X	-	X	X	-	74.6	53.24
X	X	-	X	X	X	-	75.2	52.04
X	-	X	X	X	X	-	74.7	52.52
-	X	X	X	X	X	-	75.1	46.76
X	X	X	X	-	-	X	77.1	52.42
X	X	X	-	X	-	X	76	53.96
X	X	-	X	X	-	X	75.7	54.20
X	-	X	X	X	-	X	75.3	52.91
-	X	X	X	X	-	X	75.8	49.28
X	X	X	-	-	X	X	76.4	52.28
X	X	-	X	-	X	X	75.6	50.60
X	-	X	X	-	X	X	74.3	51.80
-	X	X	X	-	X	X	75.9	46.76
X	X	-	-	X	X	X	76.2	52.28
X	-	X	-	X	X	X	75.8	52.04
-	X	X	-	X	X	X	75.7	47.96
X	-	-	X	X	X	X	74.8	53.00
-	X	-	X	X	X	X	75.4	48.68
-	-	X	X	X	X	X	73.7	46.28
X	X	X	X	-	-	-	77.5	50.85
X	X	X	-	X	-	-	76.2	53.24
X	X	X	-	-	X	-	76.5	52.17
X	X	X	-	-	-	X	76.6	53.28
X	X	X	-	-	-	-	75.3	51.83

recommender systems, we aim to look at the accuracy of recommendations using our test collections with and without using the features of users.

5.4.2 Impact of User features in Contextual Movie Recommendations using the proposed approach

Once we have described the role of movie *features* for recommending movies from our test collections, we further refine our test collections based on the previous experiment. In this section, we described the role of user features which include *age, sex, city, country* in the contextual recommendations. For this purpose, we have conducted our experiments on two bases, contextual recommendations with movie features and user features as well as contextual recommendations with user features and without movie features. In order to generate recommendations, we, again, have divided movie ratings from our test collections into two categories zeros (0) and one (1) on the similar pattern of the previous experiment. We selected eight contextual features from the contextual features set of 12 and two combinations of the movie features that perform with better accuracy when we cross evaluate the movie features from our previous experiment. These combinations are as follows:

1. Combination:1 {Actors, Genre, Director, Country, Budget}
2. Combination:2 {Actors, Genre, Director, Country}

The selection of eight contextual features (Location, Mood, Physical, Interaction, Decision, DominantEmo, Daytype, EndEmo) comes from the results given in Table 4.6 of Chapter 4. We simply do not select the four features with the lower classification accuracy. On the similar pattern of the previous experiment, Category with zeros is representing the movies that are not recommended (TN) while rating category representing with one represent the movies that are recommended (TP) for the user. In this section, we describe our experiments with user feature. We by assuming eight contextual features and combinations of movie feature with higher accuracy. We further conduct our experiments on user features and presented our results in the following section.

Results

We consider that some or all of the user features can be used to generate recommendations. We tried all user features as part of the recommendation process with selected combinations

Table 5.6 The comparison of Impact of overall User Features for contextual Recommendations

SVM	ANNs		
-	Without Movie Features	With Movie Features both combinations	
Accuracy	Accuracy	C 1: Accuracy	C2: Accuracy
74.6%	76%	76.3 %	77 %

of movie features and without movie features. While, in both cases, we assumed eight contextual features for the recommendation process, we also tried each user feature individually as well as the combinations of the user feature so that we can define the role of each and every user feature that can give us better recommendation accuracy. Additionally, we tried combinations to see if we could find a better combination (or more than one combination) of user features that can improve the accuracy of the recommendation process. Initially, we chose three machine learning algorithms to compare the performance of our ANN based proposed approach. We concluded from our previous experiments that PCA and Logistic Regression models result in far lower accuracy than the SVM and ANNs. While SVM performs much better than PCA and logistic regression, in this experiment and onwards, we only compare the accuracy of ANNs for the contextual recommendation with SVM.

First of all, we ran our experiment on user features without assuming movie's feature using SVM and our proposed ANN-based system with the User-item matrix, eight contextual features, and user ratings. We observed the results of the recommendation process which are presented in Table 5.6. We observe that SVM performs at an accuracy of 74.6%. The ANN-based approach provides accuracy of 76.3%, when we do not use movie features as input to the recommendation process. We subsequently ran our experiment with an addition of movie features to observe the impact of user features in our context-aware recommendation approach. For this purpose, we used two combinations of movie features that performed better than other combinations from our previous experiment. The combination-1 consists of 4 movie features while combination-2 consists of 5 movie features. The ANN-based approach provides an accuracy of 76.3% of the combination-1. While the ANN-based approach shows more accuracy with the combination-2 with a rate of 77%. The combination-2 is the most prominent combination of movie features that show a better performance than all other combinations in our previous experiment When we take this combination into account to compare the accuracy of user features, the accuracy remains again better than combination-1. If we compare our results on overall user features, we can conclude that User features alongside all eight selected contextual features and a combination of movie features perform better in the recommendation process, in comparison, without movie features. This refers to our hypothesis H1 that the accuracy of the recommendation process can be further improved by incorporating the combinations of contextual and non-contextual features in the recommendation process using proposed ANNs methodology. The result of our experiment also confirms the important role of user features in the process of contextual recommendations. Once we confirmed the role of user features in the contextual recommendation process, we further analyze the impact of each individual user feature on contextual recommendation process from our test collections. Each user feature is cross-compared into contextual recommendation process by providing the selected eight contextual attributes, with movie features and without movie features. We use both combinations one by one from movie features that perform better when we analyzed them. We use our proposed ANNs based model to generate recommendations and evaluate results and reported in Table 5.7. When we use feature *Age* from user feature set, to generate contextual recommendations without using movie feature, Our results show that user feature *age* provide an accuracy 75.1% without using movie features. User feature *age* perform slightly better with combination-1 with an accuracy of 75.6% while even better accuracy with the combination-2 at 75.9%. The results are given in Table 5.7. This shows that age feature from the list of movie features can perform better when combined with the movie features, even better when it is combined with combination-2 from movie features, in the recommendation process.

Similarly, *sex* feature hit an accuracy of 75% without movie features and performed better when combined with movie features. When User feature *sex* is used with combination-1 from movie feature, the observed accuracy of the recommendation process is 75.7% while with combination-2, a slight improvement is observed with the rate of 75.8%. In the similar experimental pattern, *city* feature provide an accuracy rate of 75.9% without movie features and 76% with combination-1 of movie features as well as 76.4% accuracy is observed when *City* is used with combination-2. Later on, *country* user-feature receives an accuracy of 76% without using movie features and 76% with combination-1 of movie features while the accuracy is further improved of this feature when it is used with combination-2. In contextual movie recommendations, user features can be used alongside the movie features to increase the accuracy of the system. In most cases reported in the literature [20] [69], the impact of the user features is not discussed in detail and the justification of selecting user features as part of recommendation process is not provided. However, in proposed methodology, we tried each and each user feature and their combination as part of recommendation process to find a better match of user features along with the movie features.

Table 5.7 The comparison of Impact of individual User Features for contextual Recommendations

Movie Features	Without Movie Features	With Movie Features	
	Accuracy	Accuracy (C1)	Accuracy (C2)
Age	75.1%	75.6 %	75.9 %
Sex	75%	75.7 %	75.8 %
City	75.9%	76 %	76.4 %
Country	76%	76 %	76.6 %

Once we test the accuracy of different user features individually, we observed that user features can perform better when combined with movie features. We also noted that the user features $Country, City$ are more helpful in the recommendation process with higher accuracy than the remaining two user features. We further cross-compared the different combinations of all user features with each other to identify the combinations that can provide better accuracy when combined with the selected contextual features and 2 chosen combinations of movie features.

After observing the accuracy of each and every feature in user features set, we further ran our experiments to try different possible combinations. We observed the combinations and found a total of 11 possible combinations can be used with the contextual features and the two combinations of movie features. We run our experiments on each and every combination and observed the accuracy fo different combinations in the recommendation process. We used selected eight contextual features to include contexts in the recommendation process. We observed the accuracy of each and every possible combination and presented in Table 5.8. We observed that three combinations can predict better accuracy in the recommendation process when they are combined with the combination-2 of the movie features. The combination of all user features can be handy to use in the recommendation process when it is used with combination-2 of the movie features with an accuracy 77%. While a combination of two user features $City, Country$ can have even more accuracy in recommendation process when these are used with the features in combination-2 of the movie features. In next section, we summarize our findings from this experiment by discussing the outcomes of our experiment on user features.

Table 5.8 The evaluation of the different combinations of user features

Combinations of User Features				Movie Features	
Age	Sex	City	Country	Accuracy (C1)	Accuracy (C2)
X	X	X	X	76.3	77
X	X	X	-	75.9	76
X	X	-	X	75.4	76.3
X	-	X	X	75.7	76.1
-	X	X	X	75.8	76.7
X	X	-	-	75	73.9
X	-	X	-	76	76.2
-	X	X	-	76	76.2
X	-	-	X	74	76.1
-	X	-	X	76.4	76.9
-	-	X	X	75.8	77.2

Discussion

In this experiment, we investigated the role of user features in context-aware movie recommender systems using our proposed ANN based approach. Our previous analysis showed that an ANN based approach provides better results than other machine learning techniques such as SVM which is a prominent method used for recommender systems. We compared our results for user features using contextual data and two combinations that perform better in movie features. When we used all user features without using movie features, we observed that our proposed methodology resulted in better accuracy than SVM. That is the reason why we have tried our proposed approach for later experiments when we assumed the two combinations of movie features. We formed our test collection as per requirements and evaluated the accuracy of user features. Initially, we used all movie features and cross-compared the accuracy of our input set by comparing the results from proposed methodology and SVM, a well-established machine learning technique. Later on, we tried each feature in user-feature as a separate input for contextual recommendations process to observe the accuracy of each feature. Our observations show that using the *country* feature from user feature set can result in better accuracy when it is used with the movie features, specifically, with combination-2. However, *City* feature can also result in better accuracy when it is used with movie features within contextual recommendation process. We also tried different combinations of the user features and observed that two combinations can provide the better level of accuracy within the process of contextual recommendations. These two combinations include all user features and user features having *City* as well as *Country* features.

Once we have analyzed the role of user features and movie features from our test collection, we further perform our experiments to analyze the role of different contextual features. Most of the context-aware recommender systems consider the time and location context without justifying a reason how and why the time feature is selected into contextual recommendations. Most of the datasets in context-aware recommender systems provide limited or few contextual features, where incorporating these contextual features into context-aware recommender system is another major challenge in this domain. The beauty of the LDOS-CoMoDa dataset is that is rich in terms of contexts in which the dataset provides 12 contextual features. These contextual features can be incorporated as individual as standalone features and, in the form of combinations to generate contextual recommendations. For our previous experiments, we have used only selected contexts that we chose as per the experiments that are given in Chapter 4. However, in the next section, we discuss the role of each contextual feature individually. In the next section, we investigate the role of the contextual feature *Location* as it is assumed to be very important in most cases of the context-aware recommender systems [81].

5.4.3 The role of Location in contextual Recommendations

In our test collections, the location feature is described with three contextual conditions Home, Public Place, Friend's house. The location attribute defines the context in which a user is watching the movie and providing the ratings. The numeric values in our location matrix are represented as the following subset of context features.

{Home = 1, Public Place = 2, Friend's house = 3}

In our test collections, we investigate the role of contextual feature *location* by providing it as an individual attribute to the recommendation process. The recommendation process follows the proposed ANN-based methodology to generate recommendations. We investigated the role of this feature in two ways: using all user features that includes {age, sex, gender, city} and two combinations from the movie features that results in better accuracy in our previous experiment given in Section 5.4.1. We give the term conditional movie features for both combinations that we select to compare the results of this experiment from the previous experiment. We called the condition conditional movie features which consist of two selected combinations. These combinations are as follows:

1. Combination:1 {Actors, Genre, Director, Country, Budget}

2. Combination:2 {Actors, Genre, Director, Country}

Table 5.9 The evaluation of the different combinations of user features

SVM		ANNs	
Accuracy(Comb-1)	Accuracy(Comb-2)	Accuracy(Comb-1)	Accuracy(Comb-2)
66.8	66.9	67.6	67.7

We test our proposed ANN-based approach and compare our results with the machine learning approach SVM. Later on, we evaluated the combinations (Combination:1, Combination: 2) on all users features and conditional movie features that perform better in the previous experiment. The detailed results are presented in next section.

Results

In order to run our experiments, we use the proposed system architecture described in chapter 3 (Section 3.1). Since *location* feature is a contextual attribute which is dynamic in nature, we normalize it before we can use it as an input to the proposed methodology. First of all, we use the user-item matrix with the location as a contextual feature to recommend a user with the movies based on features of the users i.e. the set of user features {age, sex, city, country}. we, also, have divided movie ratings from our test collections into two categories zeros (0) and one (1) on the similar pattern of the previous experiment. The category with one (TP) is representing the movies that are recommended while rating category representing with zero (TN) represent the movies that are not recommended for the user. We run our experiment using both SVM as baseline machine learning method and ANNs as the proposed methodology to compare our results. We observed the results and present them in Table 5.9. We ran our experiments with all user features, first, and conditional movie features. Contextual Feature *Location* hit an accuracy of 66.8%, when we tried SVM with all users features and first combination of movie features. We also observed SVM with an accuracy of 66.9% when it is tried with a second conditional combination. Later on, we tried our proposed ANNs based approach and observe that ANNs outperform SVM by 67.6% with combination-1 and 67.7% with combination-2 of movie features. We confirm that ANNs perform better than SVM on our chosen test collection for a given scenario. We also tried the different combinations of the user features with our given scenario and reported the results.

We further combined the *location* feature with users features and the movie feature to recommend based on the context provided. We evaluate the results of test data. Our evaluation shows that the contextual feature *location* predicts recommendations with better accuracy of 68.2%, when it is combined with the three user features (age, sex, country) with combination-2 of the movie features. We tried 13 different combinations to generate recommendations using contextual feature *location*, both conditional combinations of movie

Table 5.10 The evaluation of the different combinations of user features with location

All User Features				ANNs on Conditional Movie Features	
Age	Sex	City	Country	Accuracy(comb1)	Accuracy (comb2)
X	X	X	X	66.7	67.7
X	X	X	-	66.5	67.1
X	X	-	X	67.3	67.9
X	-	X	X	67.3	68.2
-	X	X	X	65.5	67
X	-	X	-	66.6	67.1
-	X	X	-	66.2	66.7
X	-	-	X	66.5	67.4
-	X	-	X	66.4	67.4
X	-	-	-	66.8	67.5
-	X	-	-	66.6	67.5
-	-	X	-	67	67.2
-	-	-	X	66.5	67.3

features for different user features and reported our results in Table 5.10. We identified that *location* performs better with combination-2 for almost all combinations which proved that the contextual recommendation can be better predicted with the combination-2. We identified the three most prominent combinations from all tested combinations with higher accuracy of contextual recommendations. We noticed a better improvement in the accuracy of 67.9%, when we combined the *age, sex, country* attributes from user features with the location using combination-2. We also observed that combining all user features {age, sex, city, country} can hit better recommendation accuracy using ANNs. Other than combinations of user features with *location* and combination-2. we also assumed individual user feature and observed that age and sex in given scenario can also predict the better accuracy of the recommendation process. However, the individual user features show less accuracy than the combinations, so, using an individual user feature with the location is not recommended. In the next section, we discuss our results briefly. Although the role of contextual feature *location* have been discussed several times in literature [13] [82] [116] [145], however in most of the cases contextual feature is used as standalone contextual feature for the recommendation process. In proposed methodology, we not only used the contextual feature *location* along with the user features and movie features to compare and achieve the better level of accuracy of the recommendation process but also combine *location* with the other contextual features as combinations to discuss its role within the contextual recommendation process in detail.

Discussion

The role of the *location* feature has been discussed in many scenarios of context-aware recommender systems. In our test collections, this role is different. We used our proposed methodology which is an ANN-based approach to describe the role of this contextual feature. Although the *location* feature performs well when it is combined with the individual and combinations of and conditional combinations of the movie feature, it is overall less than the average accuracy that we are getting for different recommendation scenarios. The accuracy of the contextual feature *location* is further decreased when it is combined with the combination-1 of the movie features as input to the recommendation process. So, combination-1 from movie features is not ideal input from the feature set. We also concluded that *location* gives more accurate results when it is combined with the user features and combination-2. So, in our experiment, we also tried different combinations of user features with our contextual attribute and refined the combinations that show better accuracy when incorporated into the contextual recommendation process.

We have described the role of *location* in contextual recommendations using our test collections, We further aim to look at the role of *emotions*, which is a hot topic in context-aware recommender systems [35]. By taking the results from this experiment into account, the pre-assumptions were used to drop the idea to use combination-1 for the next experiment, combination-1 is hitting low accuracy in the recommendation process on a consistent basis. So in the next experiment, we will have only one combination that will be used for the movie feature. We will call this condition of selecting one combination as conditional features from movie features. The details of experiments on the contextual feature *emotion* is given in the next section.

5.4.4 The role of Emotions in Contextual Recommendations

The role of emotions in contextual recommendations has a wide scope and is being investigated in a number of context-aware recommender systems [47]. The inspirations from the emotion-based recommender system and richness of LDOS-CoMoDa in terms of emotions enable the researchers to look at the contextual recommendations in movie domain. In our test collections, We have two type of emotions captured from the users. The dominant emotion defines the overall condition of emotions which come from the influence of movie and end-emotions are the end feelings of the users.

The dominant emotions and end emotions about the movie enable us to look at the contextual recommendation problem from a new angle of emotions. We used both features of emotions as separate entities and also combined them as one entity to look at the role of both types of emotions in the contextual recommendation process. We design our experiment by following our proposed ANN-based methodology and use the data transformer phase (described in section 3.1.1) to normalize the contextual feature $emotions$. In this experiment, we tried *emotions* in the contextual recommendation process with all user features first and a selected conditional movie feature. The selected feature from movie features consists of a combination of features that perform well in all previous experiments (Section 5.4.3) when combined as an input. We also tried to compare different combinations of user features to predict recommendations on user preferences and identify a combination of users features which perform better with higher accuracy in the contextual recommendation process. We called selected combination of movie features as conditional movie feature, preprocessed the data as per the approach defined in the methodology part and normalized it to run an ANN on it. We also split both *dominant emotions, end emotion*, and combined with user features and selected movie features to observe a variety of accuracy level of recommendation process. In the end, we report our results from our test data and present them in the following results section.

Hypothesis

H0 : There will be no improvement in accuracy between the traditional contextual recommendation approaches and contextual recommendations through proposed ANNs based methodology.

Results

In order to investigate the role of emotions in our test collections, we selected both features {end-emotions, dominant emotions} along-with all users features and a selected combination of movie features (Combination: 2 from section 5.4.3), to incorporate into the recommendation process as contextual features. We then used both features from emotions individually and in the form of combination to evaluate the comparison of our results with SVM. We tested both SVM and the proposed ANN-based methodology and report our results in Table 5.11. When we use $DominantEmotion$ as a stand-alone contextual feature in the recommendation process, we observe an accuracy of 72.4%. While in comparison, when we used ANNs, the accuracy of standalone $DominantEmotions$ is improved by up to 73.9%. Similarly,

Table 5.11 Results of contextual feature Emotions

SVM			ANN		
Dominant Emo	End Emo	Both	Dominant Emo	End Emo	Both
Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
72.4	74.4	75.1	73.9	75.7	76.2

EndEmotions provides an accuracy of 74.4%, when it is used as a standalone contextual feature in the recommendation process. While ANNs show accuracy as 75.7% which is much improved over the SVM. Later on, we combined both *DominantEmotions* and *EndEmotions* to compare and observe any change in the accuracy of the recommendation process. The accuracy of both features as one unit increases in both approaches- SVM and ANNs. SVM shows an improved accuracy of 75.1% when both features are combined. While ANNs in the same case shows a better accuracy level of 76.2% which is highest and dominant in this scenario. We also observed that the ANN methodology outperforms the SVM method in this experiment on our test collections. Once the role of ANNs is confirmed in the recommendation process with emotions as contextual features, we further tried different combinations of user features with the combination of both emotion features. Since the contextual feature emotion performs better in the combination of *dominantemotion* and *endemotions*, we use the same combination for further experiments.

The results from the proposed ANNs based methodology reports the improvements in accuracy in comparison with the traditional SVM method. The further statistical significant (paired test) test applied on two samples the initial set of ratings and the predicted ratings at Hypothesized Mean Difference as 0. The statistical significance of the results from proposed ANNs based methodology shows improvement at the value of p = 0.0158772 which is less than 0.05. This rejects the null hypothesis H0 for the proposed ANNs based methodology in the favor of an alternative hypothesis. The work reported in the literature [21] [54] [86] [133] focused on the contextual recommendations where only few features are used in the recommendation process as well as contexts selected in advance are assumed to perform better for the recommendation process. In proposed ANNs methodology, we compare the multiple contextual and non-contextual features and reported the accuracy of the recommendation process.

Once we confirmed the role of ANNs in the contextual recommendation process using emotions as context features, we further tried the combinations of emotion with the combinations and individual features from the set of user features. We assumed both emotion features along with the User and item matrix and one combination of movie features as input to the

Table 5.12 The evaluation of the different combinations of user features with Emotions

All User Features				ANNs with Conditional Movie Features
Age	Sex	City	Country	Accuracy
X	X	X	X	76.2
X	X	X	-	75.4
X	X	-	X	76.6
X	-	X	X	76.1
-	X	X	X	75.9
X	-	X	-	75.4
-	X	X	-	74.8
X	-	-	X	75.3
-	X	-	X	75.5
X	-	-	-	75.1
-	X	-	-	75.9
-	-	X	-	76
-	-	-	X	76.1

contextual recommendation process. We observed the results from different combinations of user features and present them in Table 5.12. When we use different combinations of users features such as {age, sex, city, country} with combined emotions features and conditional movie features, we observe a combination {age, sex, country} which performs better in recommendation process with 76.6% accuracy, in our proposed ANN methodology. We further noted that the user feature *Country* performs better with an accuracy of 76.1%, however, three combinations of user features with contextual feature *emotions* that results in higher accuracy. The detailed results are reported in Table 5.12. In the next section, we discuss our outcomes from this experiment.

Discussion

The role of emotions has been marked as a key role in many context-aware recommender system applications. It is an interesting contextual feature which can be used to improve the accuracy of contextual recommendation process. In this experiment, we investigate the role of end emotions and dominant emotions as individual contextual features and as combined one feature along-with the user features and a combination of movie features. We observed that a combination of both emotion features performs better than acting as a standalone contextual feature. We also observed that proposed ANNs outperform SVM. We further observed that different combinations of user feature can further improve the accuracy of context-aware recommender system with emotion as contextual features.

After this experiment, in the next section, we investigate the role of contextual feature *Mood* in context-aware recommendations using our test collections.

5.4.5 The Role of Mood in Contextual Recommendations

The role of *mood* in contextual recommendation process has been investigated in some recent studies [9, 55] on context-aware recommender systems. The mood of the user can be used as a key feature to predict what a user will like in the future. However, only a few applications and datasets provide mood information of a user in context-aware recommender systems. Our test collection LDOS-CoMoDa is rich in terms of mood context in which the mood of the user is used for contextual recommendations in the movie domain. The contextual features *mood* in LDOS-CoMoDa represents three different types of conditions {Positive, Neutral, Negative} which can be used to predict the recommendations of the movie. So, in this experiment, we aim to investigate the role of the contextual features *mood* when used in the contextual recommendation process to predict recommendations. We again used all user-features and one selected combination of movie-features from the dataset to compare our results with our proposed ANN-based methodology and SVM as baseline method. We perform our experiment and report the results in the following section results, then, discussed in the discussion.

Results

First, we pre-process the contextual feature *mood* as per the approach defined in Chapter 3 (Section 3.1), as part of the proposed methodology. Once the data is processed and normalized as per the requirements of the approach, we test SVM to predict recommendations using *mood* as contextual features with user-item matrix, user ratings in normalized form, all user features including age, gender, city, country and a selected combination of movie-features as per the conditions we defined in the previous section about the selected combination of movie-features. We observed the results and present in the Table 5.13. Our experiment reports that SVM recommends movies using *mood* as contextual features at the accuracy of 67.5%, while, in comparison, our proposed approach generate recommendations with a slight improvement of 67.8% accuracy. The comparison of our proposed ANN-based methodology and SVM shows that ANNs provide better accuracy when we use *mood* as a contextual feature in the recommendation process.

Table 5.13 Accuracy of contextual feature mood in contextual recommendation process using SVM and ANNs

SVM	ANN
Accuracy	Accuracy
67.5	67.8

Once the role of ANNs has been defined for *mood* as a contextual feature, we further cross compare it with the different combinations of user-features. Since we have one movie-combination selected from our previous experiments that perform better, we tried the different combinations of user-features using our proposed methodology. We observed the results and report in Table 5.14. The experiment on combinations of user-features with *mood* as contextual feature shows that the accuracy of the recommendation process can be further improved by up to 68.1%, if we consider *age* and *city* from user-features as part of the recommendation process than all user features. While in comparison, if we use the combination of *sex* and country, the accuracy will be improved by up to 69.1% with the contextual feature *mood*. While the combination of *city* and *country* from user-features can predict recommendations up to 68.5% in the contextual recommendation process. We further discuss our findings in the discussion section. Mood aware recommendation is a hot topic in the domain of multimedia recommendations where the mood of the user can be used to predict/recommend what a user will like in the future. Few studies in the literature [33] [88] [93] reported the utilization of the *mood* as contextual feature to recommend music to the end user, however, the role of mood is not much discussed for the recommendations of the movies where it can play a crucial role to increase the accuracy of overall recommender system.

Discussion

The role of the contextual feature *mood* in LDOS-CoMoDa is of key importance which can be helpful to recommend movies to users based on the mood of the user. We investigated the role of *mood* as an individual contextual feature along with the user features and a combination of movie features in the recommendation process based on the user-item matrix and rating data. We observed the results from SVM as baseline method which is a well-established technique of machine learning in the domain of recommender systems and our proposed methodology as ANNs. We confirm the role of ANNs in contextual recommendation process with higher accuracy in the recommendation process than SVMs. Later on, we use ANNs to try further experiments on contextual feature *mood*, conditional movie features and different

Table 5.14 The accuracy of different combinations of user features with contextual feature mood and a selected conditional combination of movie features

All User Features				ANNs with Conditional Movie Features
Age	Sex	City	Country	Accuracy
X	X	X	X	67.8
X	X	X	-	67.4
X	X	-	X	67.5
X	-	X	X	68.5
-	X	X	X	67.1
X	-	X	-	68.1
-	X	X	-	66.8
X	-	-	X	65.6
-	X	-	X	69.1
X	-	-	-	66.6
-	X	-	-	67.1
-	-	X	-	67.5
-	-	-	X	67.9

combinations of user-features. We observed that *mood* as contextual feature can predict recommendations with better accuracy when we select a combination of *age* and *city* from user-features in recommendation process.

In the next experiment, we investigate the role of the contextual feature *physical* in the recommendation process.

5.4.6 The role of Contextual Feature "Physical" in Contextual Recommendations

The role of the physical condition of a user as a contextual feature can play a vital role in recommending movies, items and places etc., to the user based on the physical condition of the user. Physical condition as context can be further used to predict recommendations to recommend different types of users with different physical conditions. In our test collections, we have two conditions of the user {Healthy, Ill} that are described in the domain of movies. So in this experiment, we investigate the role of the contextual feature *physical* in the recommendation process. We investigated the role of *physical* as an individual contextual feature along with the user features and a combination of movie-features in the recommendation process based on the user-item matrix and rating data. We perform our

Table 5.15 Accuracy of contextual feature *Physical* in contextual recommendation process using SVM and ANNs

SVM	ANN
Accuracy	Accuracy
66.6	67.7

experiments and report the results in the following section of results, then, discussed the results in section discussion.

Results

In order to investigate the role of *physical* as a contextual feature from our test collections LDOS-CoMoDa, we use SVM as a baseline method and ANNs as a proposed approach on the same pattern of the previous experiment. We use all user-features with the contextual feature *physical* and a combination of movie-features along with ratings to analyze the accuracy of recommendation process. We perform our experiment and present our results in following Table 5.15. Our experiment reports that SVMs recommend movies using *physical* as contextual features at the accuracy of 66.6%, while, in comparison, our proposed approach generates recommendations with a slight improvement at 67.7% accuracy with a slight improvement. The comparison of our proposed ANNs based methodology and SVM shows that ANNs provide better accuracy when we use *physical* as a contextual feature in the recommendation process.

Once the role of ANNs has been defined for *physical* as a contextual feature, we further cross compare it with the different combinations of user-features. Since we have one movie-combination selected from our previous experiments that perform better, we tried the different combinations of user-features using our proposed methodology. We observed the results and report in Table 5.16. The experiment on combinations of user-features with the *physical* as contextual features show that the accuracy of recommendation process can be further improved up to 68.1%, if we consider *age* and *city* from user-features as part of recommendation process rather than all user features. While in comparison, if we use the combination of *sex* and country, the accuracy will be improved by up to 67.71% with the contextual feature *physical*. While the combination of all user-features can predict recommendations up to 67.7% in the contextual recommendation process. In most cases, the scope of contextual recommender system is limited to the contextual features such as location, mood, emotions etc. that can be used to predict/recommend, however, some of the other features such as physical condition of the user in the case of movie recommender system

Table 5.16 The accuracy of different combinations of user features with contextual feature *physical* and a selected conditional combination of movie features

All User Features				ANNs with Conditional Movie Features
Age	Sex	City	Country	Accuracy
X	X	X	X	67.7
X	X	X	-	67.8
X	X	-	X	67.1
X	-	X	X	67.6
-	X	X	X	67.1
X	-	X	-	68.1
-	X	X	-	67.0
X	-	-	X	67.2
-	X	-	X	67.7
X	-	-	-	66.7
-	X	-	-	66.9
-	-	X	-	67.2
-	-	-	X	67.1

can also play vital role. The physical contextual feature has not been exploited in the past for the recommendation purpose although it is available as part of LDOS-CoMoDa dataset as discussed in [100]. In this experiment, we observed the role of the *Physical* contextual feature in detail as part of contextual movie recommendations and discussed in detail. We further discuss our findings in the discussion section.

Discussion

The role of the contextual feature *physical* in LDOS-CoMoDa is observed as vital which can be helpful to recommend movies to users based on the physical condition of the user. We investigated the role of *physical* as an individual contextual feature along with the user features and a combination of movie-features into recommendation process based on the user-item matrix and rating data. We observed the results from SVM as baseline method which is a well-established technique of machine learning in the domain of recommender systems and our proposed approach based on ANNs. We confirm the role of ANNs in the contextual recommendation process with a slightly higher in the accuracy of recommendation process than SVM. Later on, we use ANNs to try further experiments on contextual feature *physical*, conditional movie features and different combinations of user-features. We observed that *physical* as a contextual feature can predict recommendations with better accuracy when we select a combination of *age* and *city* from user-features in the recommendation process.

In the next experiment, we investigate the role of the contextual feature *decision* in the recommendation process.

5.4.7 The role of Contextual Feature "Decision" in Contextual Recommendations

The role of Contextual Feature "Decision" can be investigated based on the choices that user is given. The contextual feature *decision* from our test collections LDOS-CoMoDa can also play an important role to recommend user with the movies, based on the choices: whether a user has made the choice or user is being asked to watch the movie. In our test collections, we have two options for *decision* of the user {user decided which movie to watch, User was given a movie} that described in the domain of movies. So, in this experiment, we investigate the role of contextual feature *decision* in the recommendation process. We investigated the role of *decision* as an individual contextual feature along-with the user features and a combination of movie-features into recommendation process based on the user-item matrix and rating data. We perform our experiments and report the results in following the section of results, then, discussed the result in the discussion.

Results

In order to investigate the role of *decision* as the contextual feature from our test collections LDOS-CoMoDa, we use SVM as baseline method and ANNs as the proposed methodology on the same pattern of the previous experiment. We use all user-features with the contextual feature *decision* and a combination of movie-features along-with ratings to analyze the accuracy of recommendation process. We perform our experiment and present our results in following Table 5.17. Our experiment reports that SVM recommends movies using *decision* as contextual features at the accuracy of 72.4%, while, in comparison, our proposed approach generate recommendations with improvements at 73.4% accuracy level with a slight improvement. The comparison of our proposed ANNs based methodology and SVM shows that ANNs provide better accuracy when we use *decision* as a contextual feature in the recommendation process.

Table 5.17 Accuracy of contextual feature *Decision* in contextual recommendation process using SVM and ANNs

SVM	ANN
Accuracy	Accuracy
72.4	73.4

Once the role of ANNs has been defined for *decision* as a contextual feature, we further cross compare it with the different combinations of user-features. Since we have one movie-combination selected from our previous experiments that perform better, we tried the different combinations of user-features using our proposed methodology. We observed the results and report in Table 5.18. The experiment on combinations of user-features with *decision* as a contextual feature shows that the accuracy of recommendation process can be further improved by up to 73.5%, if we consider *age* and *city* or *sex* and *country* as two separate combinations from user-features as part of the recommendation process rather than all user features. The contextual feature decision is another example of an available feature for the recommendation purpose which has not been exploited in the past in detail. It is only reported in the literature [100] as part of LDOS-CoMoDa dataset where it can be used to recommend a movie based on the two conditions; the decision of the user to watch a movie or user was given a choice. In this experiment, we also used this feature to find its role for the recommendations of the movies. We further discuss our findings in the discussion section.

Discussion

The role of the contextual feature *decision* in LDOS-CoMoDa is observed as vital which can be helpful to recommend movies to users based on the choices that a user has made to watch a movie. We investigated the role of *decision* as an individual contextual feature along-with the user features and a combination of movie-features in the recommendation process based on the user-item matrix and rating data. We observed the results from SVM as baseline method which is a well-established technique of machine learning in the domain of recommender systems and our proposed approach. We confirm the role of ANNs in contextual recommendation process with better results in terms of accuracy of recommendation process than SVM. Later on, we use ANNs to try further experiments on contextual feature *decision*, conditional movie features and different combinations of user-features. We observed that *decision* as contextual feature can predict recommendations with better accuracy when we select a combination of *age* and *city* or *sex* or *country* from user-features as two separate

Table 5.18 The accuracy of different combinations of user features with contextual feature *Decision* and a selected conditional combination of movie features

All User Features				ANNs with Conditional Movie Features
Age	Sex	City	Country	Accuracy
X	X	X	X	73.4
X	X	X	-	73.3
X	X	-	X	71.9
X	-	X	X	72.3
-	X	X	X	72.9
X	-	X	-	73.5
-	X	X	-	72.5
X	-	-	X	72.2
-	X	-	X	73.5
X	-	-	-	73.4
-	X	-	-	72.8
-	-	X	-	73.1
-	-	-	X	72.8

combinations in recommendation process.

In the next experiment, we investigate the role of the contextual feature *interaction* in the recommendation process.

5.4.8 The Role of Interaction in Contextual Recommendations

The role of the contextual feature *interaction* can be investigated based on the interaction of the user with the movie in the domain of movie recommendation. The contextual feature *interaction*, from our test collections LDOS-CoMoDa can also play an important role to recommend user with the movies, based on their interaction with the movies. In our test collections, we have two options for *interaction* of the user {first interaction with a movie, nth interaction with a movie} that can be described in the domain of movies. In this experiment, we investigate the role of contextual feature *interaction* in the recommendation process. We investigated the role of *interaction* as an individual contextual feature along with the user features and a combination of movie-features in the recommendation process based on the user-item matrix and rating data. We perform our experiments and report the results in the following section of results, then, discuss the result in the discussion section.

Table 5.19 Accuracy of contextual feature *Interaction* in contextual recommendation process using SVM and ANNs

SVM Accuracy	ANN Accuracy
66.9	68.8

Results

In order to investigate the role of *interaction* as a contextual feature from our test collections LDOS-CoMoDa, we use SVM as baseline method and ANNs as a proposed methodology on the same pattern of the previous experiment. We use all user-features with the contextual feature *interaction* and a combination of movie features along with ratings to analyze the accuracy of the recommendation process. We perform our experiment and present our results in the following Table 5.19. Our experiments report that SVMs recommend movies using *interaction* as contextual features at an accuracy of 66.9%, while, in comparison, our proposed approach generates recommendations with improvements at 68.8% accuracy level with better and improved level. The comparison of our proposed ANN-based methodology and SVM shows that ANNs outperform SVMs in this experiment when we use *interaction* as a contextual feature in the recommendation process.

Once the role of ANNs has been defined for *interaction* as a contextual feature, we further cross compare it with the different combinations of user-features. Since we have one movie-combination selected from our previous experiments that perform better, we tried the different combinations of user-features using our proposed approach. We observed the results and report in Table 5.20. The experiment on combinations of user-features with *interaction* as a contextual feature shows that the accuracy of the recommendation process remains on top when we used all user-features So in this case combinations of user-feature do not seem ideal to work with *interaction*. In contextual movie recommendation, social interaction feature is used in [43] for group recommendations where the recommendation is made to a group of users rather than individuals. However, the interaction of the user with a particular movie is not considered which can also be used as a vital feature for the movie recommendations. The feature is available as part of dataset LDOS-CoMoDa [100] and can be used to predict/recommend the movies. We further discuss our findings in the discussion section.

Table 5.20 The accuracy of different combinations of user features with contextual feature *Interaction* and a selected conditional combination of movie features

All User Features				ANNs with Conditional Movie Features
Age	Sex	City	Country	Accuracy
X	X	X	X	68.8
X	X	X	-	66.5
X	X	-	X	67.7
X	-	X	X	68.4
-	X	X	X	66.5
X	-	X	-	67.1
-	X	X	-	66.6
X	-	-	X	66.7
-	X	-	X	66.8
X	-	-	-	67.2
-	X	-	-	67.1
-	-	X	-	67.6
-	-	-	X	67.6

Discussion

The role of contextual feature *interaction* in LDOS-CoMoDa is observed as crucial and can be used to recommend movies to users based on the interaction of the user with the movies. We investigated the role of *interaction* as an individual contextual feature along-with the user features and a combination of movie-features into the recommendation process based on the user-item matrix and rating data. We observed the results from SVM as baseline method which is a well-established technique of machine learning in the domain of recommender systems and our proposed methodology using ANNs. We confirm the role of ANNs in contextual recommendation process with better results in the accuracy of recommendation process than SVM. Later on, we use ANNs to try further experiments on contextual feature *interaction*, conditional movie features and different combinations of user-features. We observed that *interaction* as a contextual feature can predict recommendations with better accuracy when we use all user-features combined.

In the next experiment, we investigate the role of the contextual feature *day-type* in the recommendation process.

Table 5.21 Accuracy of contextual feature $day - type$ in contextual recommendation process using SVM and ANNs

SVM	ANN
Accuracy	Accuracy
66.9	67.4

5.4.9 The role of day-type in Contextual Recommendations

The role of Contextual Feature *day-type* can be investigated based on the type of day a user watched a movie which can further help to predict on what day which type of movie a user will watch in the future. The contextual feature *day-type*, from our test collections LDOS-CoMoDa can also play an important role to recommend movies to the user, based on their patterns of movies watched in the past. In our test collections, we have three attributes for *day-type* such as {Working day, Weekend, Holiday} that can be used as contextual features to predict over the ratings for movies. So, in this experiment, we investigate the role of contextual feature *day-type* in the recommendation process. We investigated the role of *day-type* as an individual contextual feature along-with the user features and a combination of movie-features into recommendation process based on the user-item matrix and rating data. We perform our experiments and report the results in the following section of results, then discuss the result in the discussion section.

Results

In order to investigate the role of *day-type* as a contextual feature from our test collections LDOS-CoMoDa, we use SVM as the baseline method and ANNs as the proposed approach on the same pattern of the previous experiment. We use all user-features with the contextual feature *day-type* and a combination of movie-features along with ratings to analyze the accuracy of recommendation process. We perform our experiment and present our results in the following Table 5.21. Our experiments report that SVMs recommend movies using *day-type* as contextual features at the accuracy of 66.9%, while, in comparison, our proposed approach generate recommendations with improvements at 67.4% accuracy level with more better and improved level. The comparison of our proposed ANN-based methodology and SVM shows that ANNs outperform SVMs in this experiment when we use *day-type* as a contextual feature in the recommendation process.

Once the role of ANNs has been defined for *day-type* as a contextual feature, we further cross-compare it with the different combinations of user-features. Since we have one movie-combination selected from our previous experiments that perform better, we tried the different combinations of user-features using our proposed approach. We observed the results and report in Table 5.22. The experiment on combinations of user-features with the *day-type* as contextual features show that the accuracy of the recommendation process can be slightly improved by up to 67.5%, if we consider *age* and *country* as a combination of user-features as part of the recommendation process rather than all user features. As in Section 5.4.4, we rejected the null hypothesis H0 in the favor of an alternative hypothesis H1, we presented H1 in this section.

Hypothesis

H1 The accuracy of a recommendation process can be further improved by incorporating the combinations of contextual and non-contextual features in recommendation process using proposed ANNs methodology.

The results from the proposed ANNs based methodology reports the improvements in accuracy in with the different combinations of the contextual and non-contextual features. The further statistical significant (paired test) test applied on two samples the initial set of ratings and the predicted ratings. The statistical significance of the results from proposed ANNs based methodology shows improvement at the value of p = 0.003913769 which is less than 0.05. This accepts the hypothesis H1 for the proposed ANNs based methodology, the accuracy of the recommendation is significantly improved by incorporating the combinations of contextual and non-contextual features in recommendation process using the proposed methodology. We further discuss our findings in the discussion section.

In the domain of the movies recommendations, type of the day can play an important role where a user can be recommended based on the based on the nature of the day on which a movie is watched. Most of the movie recommender systems reported in the literature [56] [88] assume the common features such as location, mood, and emotions. However, the contextual features that define the interaction of the user on a particular day and time can also play an important part in the particular domain of movie recommender system. In this experiment, we also consider the type of the day as an important feature for contextual recommendation. A summary of all individual contextual features that are incorporated in

Table 5.22 The accuracy of different combinations of user features with contextual feature $day-type$ and a selected conditional combination of movie features

All User Features				ANNs with Conditional Movie Features
Age	Sex	City	Country	Accuracy
X	X	X	X	67.4
X	X	X	-	66.4
X	X	-	X	67.2
X	-	X	X	66.5
-	X	X	X	67.3
X	-	X	-	66.5
-	X	X	-	66.1
X	-	-	X	67.5
-	X	-	X	66.6
X	-	-	-	66.9
-	X	-	-	67.1
-	-	X	-	67.2
-	-	-	X	66.8

the contextual recommendation process, along with all users features and combination-2 of the movie features, is given in Table 5.23.

Discussion

The role of contextual feature *day-type* in LDOS-CoMoDa is observed as of key importance and can be used to recommend movies to users based on the type of day on which users have watched movies in the past. We investigated the role of *day-type* as an individual contextual feature along with the user features and a combination of movie-features in the recommendation process based on the user-item matrix and rating data. We observed the

Table 5.23 Summary of individual contextual features with all user features and selected combination (Comb-2) of the movie fearures

Contextual Feature	SVM (Accuracy)	ANNs (Accuracy)
Emotions (end emo and dominant emo)	75.1	76.2
Decision	72.4	73.4
Interaction	66.9	68.8
Mood	67.5	67.8
Location	66.9	67.7
Physical	66.6	67.7
Day-type	66.9	67.4

results from SVM as baseline method which is a well-established technique of machine learning in the domain of recommender systems and our proposed approach. We confirmed the role of ANNs in the contextual recommendation process with better results in the accuracy of recommendation process than SVMs. Later on, we use ANNs to try further experiments on contextual feature *day-type*, conditional movie features and different combinations of user-features. We observed that *day-type* as contextual feature can predict recommendations with better accuracy when we use a combinations of *age* and *country* from user-features.

Once we discussed the role of all eight selected contextual features from our test collections LDOS-CoMoDa, we further evaluated the combinations of these contextual features. The evaluations of combinations of contextual features further help to identify if the combination of two or more contextual feature can provide better accuracy in the recommendation process than the feature as standalone context. Since the results from our proposed ANN-based methodology and baseline method SVM have the close difference, we decided to observe both methods on combinations of contextual features. In order to try different combinations of contextual features, we used all user-features and one combination of movie-features with prominent accuracy in all previous experiments. We further observe the results and reported in Table 5.24. Our comparison with the baseline method shows that ANNs outperform SVM in every combination with a higher accuracy rate of recommendation process. In our results, we can see that a combination of all contextual features that we selected (described in Chapter 4) results in higher accuracy of recommendation process with the proposed ANNs based methodology. This refers to our hypothesis that incorporation of context into the recommendation process can improve the performance of recommender systems.

5.5 Summary and Conclusion

The majority of the literature on recommender systems focuses on a few contextual features with pre-assumptions that the feature will contribute to better accuracy [9] [23] [29]. That is because only a few contextual features are available to incorporate into the recommendation process. LDOS-CoMoDa is a context-rich dataset that provides almost 12 contextual features to incorporate into context-aware movie recommender systems with user-features and movies features. We cross-compared the different contextual features with our proposed ANNs based methodology and find that eight contextual features are highly relevant. In this chapter, we provide a detailed experiment based comparison of different combinations of user-features, movie-features, and contextual features. We observed our results based on our proposed

Table 5.24 The evaluation and comparison of the different contextual features when incorporated into the recommendation process

Contextual Features							ANNs	SVM
Loc	Mood	Physical	Decision	Interaction	Day	Emotion	Accuracy	Accuracy
X	X	X	X	X	X	X	77.0	76.0
X	X	X	X	X	X	-	67.7	67.1
X	X	X	X	X	-	X	76.3	75.1
X	X	X	X	-	X	X	75.6	74.6
X	X	X	-	X	X	X	76.6	74.8
X	X	-	X	X	X	X	76.1	75.1
X	-	X	X	X	X	X	76.3	75.1
-	X	X	X	X	X	X	76.4	75.0
X	X	X	X	X	-	-	68.0	67.4
X	X	X	X	-	X	-	67.6	67.1
X	X	X	-	X	X	-	67.7	66.9
X	X	-	X	X	X	-	68.1	66.8
X	-	X	X	X	X	-	67.8	67.4
-	X	X	X	X	X	-	69.0	67.7
X	X	X	X	-	-	X	76.4	75.0
X	X	X	-	X	-	X	76.1	75.0
X	X	-	X	X	-	X	76.3	74.1
X	-	X	X	X	-	X	76.1	75.1
-	X	X	X	X	-	X	76.4	75.6
X	X	X	-	-	X	X	76.2	74.8
X	X	-	X	-	X	X	76.2	75.3
X	-	X	X	-	X	X	76.3	74.8
-	X	X	X	-	X	X	75.8	74.7
X	X	-	-	X	X	X	76.6	75.5
X	-	X	-	X	X	X	76.2	74.8
-	X	X	-	X	X	X	76.6	75.2
X	-	-	X	X	X	X	76.4	75.1
-	X	-	X	X	X	X	75.7	75.0
-	-	X	X	X	X	X	76.4	75.1
X	X	X	X	-	-	-	67.5	66.8
X	X	X	-	X	-	-	68.5	66.6
X	X	X	-	-	X	-	67.6	66.9
X	X	X	-	-	-	X	76.5	74.9
X	X	X	-	-	-	-	66.9	66.5

approach and compare them with some well-established machine learning algorithms. We also cross-compared our results with the different results reported in the literature [68] for the same test collections. We discussed our findings based on the experiments in detail as well as recommended the combinations of different features with higher recommendation accuracy in the contextual recommendation process.

Chapter 6

Conclusion and Future Work

Although the domain of contextual recommendation has made significant progress over the last few years, the problem of selection of appropriate context, when plenty of contextual features are available, remained fairly unexplored. In context-aware recommender systems, different machine learning techniques have been developed to predict the ratings that come in the form of initial ratings from the users. Different machine learning techniques have been adopted for recommender systems such as Principle Component Analysis (PCA), Support Vector Machine (SVM) and Matrix Factorization (MF), however, Artificial Neural Networks (ANNs) have not been widely explored in this particular domain of context-aware recommender systems. The incorporation of contextual data into the recommendation process is challenging and requires the data to be divided into proper forms of features. Whereas, in open datasets that are available for research purposes most of them have few contextual features that are presumed to perform better in selected contexts. A comprehensive study in this context can help in how to compare the contextual features when you have a context-rich dataset with plenty of contextual features.

LDOS-CoMoDa is an open source dataset collected by a group of researchers with 12 contextual features along with a variety of user-features and movie-features in the domain of movie recommendations, where contextual features include the personal preferences of the users for the real-time movies in different scenarios. We find it an interesting dataset that can be used for a detailed analysis of contextual features, user-features, and movies features. We observe from LDOS-CoMoDa that contextual features are more dynamic by nature, in comparison to the user-features and movie-features, which are observed as static features. In this work, we compare both static features and dynamic features as well as finding the relevant features from all categories.

We introduced an Artificial Neural Network (ANN) based methodology that can be used to compare the different contextual, user and movie features that can be further used to predict recommendations. The proposed approach is divided into three phases which are described in the proposed methodology. Once the data is transformed into an appropriate form, ANNs can be trained for feature selection and contextual recommendation purposes. In order to compare the results from our proposed methodology, we selected three well-established methods of machine learning in the domain of recommender systems which are SVM, PCA, and Linear Regression. We also selected the results given in the literature on the same dataset as baseline results to compare the output of the proposed methodology. In chapter 1, we introduced the problem of recommendation process and describe our motivation. In chapter 2, we investigate the different recommendation techniques that refer to over objective O1 which focus on investigation and analysis of existing recommendation techniques for different recommender systems. The main contribution of this thesis is in two areas: contextual feature selection and contextual recommendations with ANNs. This dissertation also contributes to define the role of ANNs in the domain of context-aware recommender systems, especially, in the scenario where plenty of techniques from machine learning algorithms are being used, ANNs can be a better alternative option.

In chapter 3 we define a proposed methodology based on ANNs which refers to our contribution C3. We also define how ANNs can be used to cross compare the role of different contextual, user and movie features that further help to ease the selection process of features or their combinations. The results from the feature selection approach using ANNs can further be used in the process of contextual recommendations. We use a test collection from an open source dataset from the domain of context-aware recommender systems. In our contribution to the feature selection domain, while using LDOS-CoMoDa as our test collections, we train ANNs to define the role of the different type of available features (user features, movie features, contextual features). We also define a way to handle features by dividing them into different representations/categories based on the nature of features (static, dynamic). We further use our ANN-based methodology and confirm the important role of contextual features in comparison with the subordinate role of user features and movie features. Dynamic features have better classification accuracy that can be used as a base to select a feature in the contextual recommendation process. We also compare the results from our ANN based approach with a well-established approach Matrix Factorization that is used to find the relevance of the contextual features in literature. We reported our results as our proposed methodology outperforms the baseline method in detection of relevant contexts.

6.1 Contribution Revisited

C1 In Chapter 4, we describe the categorisation of contextual and non-contextual features and evaluation. We defined how ANNs can be used to cross compare the role of different contextual, user and movie features that further help to ease the selection process of features or their combinations. The results from the feature selection approach using ANNs can further be used in the process of contextual recommendations. We used our test collection from an open source dataset from the domain of context-aware recommender systems. In this chapter, our contribution to the feature selection approach refers to our objective O2 which focus on analysis and evaluation of different feature sets available for contextual recommendations., We used LDOS-CoMoDa dataset as test collections for the experiments. We further trained ANNs to define the role of the different type of available features (user features, movie features, contextual features). We also defined a way to handle a large number of features by dividing them into different representations/categories based on the nature of features (static, dynamic). We further used our ANN-based methodology and confirmed the role of contextual features in comparison with the subordinate role of user features and movie features.

C2 In chapter 5, we selected the eight top-performing contextual features based on the results of our proposed methodology and take them on a further step to use in the contextual recommendation process. We apply and evaluate MLP model for rating prediction in both chapter 4 and chapter 5 which focused on our contribution C2. Our main aim is to incorporate the relevant contextual features into the recommendation process along with user features and movie features so that we can find a balanced combination of three types of features that can predict better accuracy in the contextual recommendation process. We also compared the results of the ANN-based methodology with some other well-established machine learning algorithms to compare the accuracy of the contextual recommendation process. We investigated the role of movie features in the contextual recommendation process by using context and user-features. We explore the link between the features of movies and the context when the movie is selected or watched to better understand how contextual information from the chosen dataset LDOS-CoMoDa, such as location, weather and time of the day can influenced movie selection. We showed with experiments that by using preferences in particular contexts, we are able to make recommendations to a user in a given context based on the preferences of the users. We also demonstrate that there is an improvement in accuracy in the contextual recommendation process using the proposed ANN-based approach which rejects our null hypothesis H0 and leaves a space for an alternative hypothesis. We cross compared our results from this experiment with the results reported in the literature on the same dataset. We observe that our proposed approach predicts better in terms of the accuracy of the recommendation process in comparison with the results from baseline algorithms. The contribution C2 which focus on applying and evaluating multilayer perceptron model for rating prediction also extends our objectives O2 (Analyze and evaluate different feature sets available for contextual recommendations) and O3 (Analyze and evaluate performance by defining a minimum contextual attribute subset which can generate more accurate contextual recommendations).

We further cross-compared the different combinations of movie features and reported a combination as finding that performed consistently better in the recommendation process. We used economization of movie features to evaluate the performance of different user features in contexts. We found that user-features have a crucial role in the context-aware recommendation process and they are better in combination with movie features and contextual features. We investigated the role of user features into context-aware movie recommender system using our proposed ANN-based approach. Our previous analysis showed that an ANN-based approach gives better results than other machine learning techniques such as support vector machines SVM which is a prominent method used for recommender systems [40]. We compared our results for user features using contextual data and two combinations that perform better in movie features. When we used all user features putting them with/ without using movie features, we observed that our proposed methodology results in better accuracy than SVM.

Once we have analyzed the role of user features and movie features from our test collection, we further perform our experiments to analyze the role of different contextual features. In a context-aware recommender system, contextual features have a very crucial role. Most context-aware recommender systems consider the time and location context without justifying a reason how and why the time feature is selected for contextual recommendations. Most of the datasets in context-aware recommender systems provide limited or few contextual features, where incorporating these contextual features into context-aware recommender system is another major challenge in this domain. The beauty of the LDOS-CoMoDa dataset is that is rich in terms of contexts in which dataset provides 12 contextual feature. These contextual features can be incorporated as individual as a standalone feature and, in form of combinations to generate contextual recommendations. We also discussed the role of each individual contextual feature as well as the role of their combinations. We observed that combinations of contextual features perform better than the standalone contextual features. We also observed that the accuracy of the recommendation process remains higher with the user features on a consistent basis. We also observed that emotion as standalone and with the combinations is the only contextual feature that results in better accuracy with the process of contextual recommendations. Since SVM is a close competitor in all the experiments we performed, we also noticed that the ANN-based approach outperform SVM in every experiment with a fair margin that confirms the role of ANNs in the domain of context-aware recommender systems. Based on the facts and results of our experiment, we can recommend our ANN based approach in the selected domain of context-based movie recommender systems.

C3 In chapter 5, we introduced an ANN-based approach which integrated both contextual and non-contextual features into the contextual recommendation process. We confirmed the role of our proposed methodology with the experiments 5 by generating contextual recommendations and comparing our results with some established machine learning algorithms. Our results confirmed the role of our proposed methodology which also confirms our contribution C3 which also refers to our objective O4. Our proposed ANN based approach outperforms SVM and PCA by improving the accuracy of contextual recommendation processes which also confirmed/accepted our hypothesis H1 that states the accuracy of the recommendation process can be further improved by incorporating the combinations of contextual and non-contextual features in the recommendation process using proposed ANNs methodology.

6.2 Limitations of proposed ANN based approach

The proposed ANNs based approach has some limitations the contextualisation of the data remains challenging for the large datasets in the domain of contextual recommendations. In this section, we summarise the limitations.

The proposed ANN based methodology can only be applied to available contextual information

In this thesis, we focused on the contextual information that is available as part of test collections. The proposed ANNs based approach can recommend the movies for different contextual, non-contextual features and the combinations. The approach can be useful for the contextual recommendations in music recommendation domain where songs can be recommended to the user based on the mood of the user and/or the genres of the music. The approach is also relevant for the location and interaction based recommendations which are common in food recommenders or trip recommendation where similar contextual information is utilized for contextual recommendation process. However, the proposed approach may not be relevant to the other recommendation such as job search recommendation. The approach does not introduce a way to infer the contexts from the user reviews like some of the recommendation methods do. The issue is common in context-aware recommendation techniques as in context generalisation, different generalisation rules are required for different contextual information.

The proposed ANN based approach is used with the explicit feedback (ratings)

In our proposed methodology, we only use the users' feedback that is collected explicitly. Recommender systems should be able to make use of implicit data when it is available to incorporate into the recommendation process.

The proposed ANN based approach cannot handle the change of preferences or ratings

The proposed approach can only handle the pre-recorded preferences while in recommender systems, the preferences of the user can change. In order to obtain the recommendations from the updated preferences, the process has to rerun with the updated feedback. The contextual information in test collections is driven by the dataset which is available for research purposes in the domain of contextual recommendation. The LDOS-CoMoDa dataset is an interesting data set when it comes to providing a rich set of dynamic contextual features. The dominance of such features for the given rating prediction task is remarkable, but may not be observable in different datasets.

6.3 Future Work

The LDOS-CoMoDa dataset is a rich set of contextual and non-contextual features which provide multiple contexts to incorporate into the recommendation process along with user features and movie features. In the future, we will look into similar data sets and investigate the role of dynamic contextual features compared to static, non-contextual ones. In this respect, we will also check if there is still a way to combine non-contextual features with dynamic, contextual ones, given that other data sets do not possess a dominant feature set like we find with LDOS-CoMoDa. One potential idea is borrowed from the principle of polyrepresentation in information retrieval [58], which is also a reason why we called feature sets *representations* in this work. If documents are recommended by different classifiers using different representations (feature sets), we would expect that the set of documents recommended by all classifiers to exhibit a high precision. This would also give rise to a more interactive and personalized approach to recommendation using machine learning and neural networks, for instance by presenting to the user those recommendations first that are confirmed by different representations and let the user decide which set of recommendations to visit next (for instance those that match the current mood vs. those that match other features like age, location or genre). Polyrepresentation, as well as user interaction, would inform the feature selection process in this case. Whether we can actually observe something 'polyrepresentation-like' in machine learning based recommendation and artificial neural networks are subject to further investigation.

In the light of limitations of the proposed approach, we, specifically, would like to work on the following points in the future.

- In the proposed ANN-based methodology, we can bring more algorithms other than MLP such as GMDH that can be tested on the input and give an opportunity to decide on the one with the better performance based on the nature of the data.

- We would also like to test our methodology in an application where both implicit data and explicit data are available. We aim to test/modify our approach to use it for job recommendation task by observing the click/search behavior of the users on job searching website. The implicit data can be useful for the job recommendation process which can be exploited in several forms, in future, for the recommendation tasks.

- In future, we would also like to introduce a way to infer contexts from the user reviews so that the scope of the proposed methodology can be extended to text classification and recommendations.

In future, we also plan to extend our ANN-based methodology to other datasets so that the effectiveness of the ANNs can be further evaluated in the domain of recommender systems.

We also plan to develop a recommender system kit based on the ANN approach and test more ANN techniques on different datasets.

References

[1] Abbas, A., Zhang, L., and Khan, S. U. (2015). A survey on context-aware recommender systems based on computational intelligence techniques. *Computing*, 97(7):667–690.

[2] Abdollahi, B. and Nasraoui, O. (2016). Explainable Restricted Boltzmann Machines for Collaborative Filtering. *ArXiv preprint*, (Whi).

[3] Adomavicius, G. and Tuzhilin, A. (2001). Extending recommender systems: A multidimensional approach. *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI01 Workshop on Intelligent Techniques for Web Personalization ITWP2001 Seattle Washington August*, pages 4–6.

[4] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

[5] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer.

[6] Adomavicius, G., Tuzhilin, A., and Luca, E. W. D. (2010). Context-Awareness in Recommender Systems : Research Workshop and Movie Recommendation Challenge Categories and Subject Descriptors. page 60558.

[7] Aïmeur, E., Brassard, G., Fernandez, J. M., and Mani Onana, F. S. (2008). Alambic: a privacy-preserving recommender system for electronic commerce. *International Journal of Information Security*, 7(5):307–334.

[8] Amatriain, X. and Pujol, J. M. (2015). Data mining methods for recommender systems. In *Recommender Systems Handbook*, pages 227–262. Springer.

[9] Andjelkovic, I., Parra, D., and O'Donovan, J. (2016). Moodplay: Interactive & Affective-aware Music Discovery and Recommendation. *Umap 2016*, pages 275–279.

[10] Asabere, N. (2013). Towards a Viewpoint of Context-Aware Recommender Systems (CARS) and Services. *Ijcst.Org*, 4(1).

[11] Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., and Weintraub, O. (2013). Movie Recommender System for Profit Maximization.

[12] Baltrunas, L. and Amatriain, X. (2009). Towards time-dependant recommendation based on implicit feedback. *Workshop on context-aware recommender systems (CARS'09)*, pages 25–30.

[13] Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2011). Context-aware places of interest recommendations and explanations. *CEUR Workshop Proceedings*, 740:19–26.

[14] Baltrunas, L., Ludwig, B., Peer, S., and Ricci, F. (2012). Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5):507–526.

[15] Bari, P., Consulting, A., Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A. (2009). Experimental Comparison of Pre- vs . Post-Filtering Approaches in Context-Aware Recommender Systems. pages 3–6.

[16] Beel, J., Gipp, B., Langer, S., and Breitinger, C. (2016). Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338.

[17] Bell, R., Koren, Y., and Volinsky, C. (2007). Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 95–104. ACM.

[18] Bellogín, A., Castells, P., and Cantador, I. (2011). Precision-Oriented Evaluation of Recommender Systems : An Algorithmic Comparison.

[19] Bennett, J. and Lanning, S. (2007). The Netflix Prize. *KDD Cup and Workshop*, pages 3–6.

[20] Biancalana, C., Gasparetti, F., Micarelli, A., Miola, A., and Sansonetti, G. (2011). Context-aware movie recommendation based on signal processing and machine learning. *Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation - CAMRa '11*, pages 5–10.

[21] Bianchi, M., Cesaro, F., Ciceri, F., Dagrada, M., Gasparin, A., Grattarola, D., Inajjar, I., Metelli, A. M., and Cella, L. (2017). Content-based approaches for cold-start job recommendations. In *Proceedings of the Recommender Systems Challenge 2017*, RecSys Challenge '17, pages 6:1–6:5, New York, NY, USA. ACM.

[22] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

[23] Bogers, T. and Koolen, M. (2016). Report on RecSys 2015 Workshop on New Trends in Content-Based Recommender Systems. *ACM SIGIR Forum*, 49(2):141–146.

[24] Bogers, T. and Van Den Bosch, A. (2009). *Collaborative and content-based filtering for item recommendation on social bookmarking websites*, volume 532.

[25] Bostandjiev, S., O'Donovan, J., and Höllerer, T. (2012). Tasteweights: a visual interactive hybrid recommender system. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 35–42. ACM.

[26] Brons, M., Nijkamp, P., Pels, E., and Rietveld, P. (2008). A meta-analysis of the price elasticity of gasoline demand. a sur approach. *Energy Economics*, 30(5):2105–2122.

[27] Buder, J. and Schwind, C. (2012). Learning with personalized recommender systems: A psychological view. *Computers in Human Behavior*, 28(1):207–216.

[28] Burke, R. (2007). Hybrid web recommender systems. *The adaptive web*, pages 377–408.

[29] Campos, P. G., Díez, F., and Cantador, I. (2014). Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols. *User Modeling and User-Adapted Interaction*, 24(1-2):67–119.

[30] Carolis, B. D., Novielli, N., Plantamura, V. L., Gentile, E., Informatica, D., and Orabona, V. (2009). Generating Comparative Descriptions of Places of Interest in the Tourism Domain. pages 277–280.

[31] Chang, C. C., Chen, P.-L., Chiu, F.-R., and Chen, Y.-K. (2009). Application of neural networks and kano's method to content recommendation in web personalization. *Expert Syst. Appl.*, 36(3):5310–5316.

[32] Chou, P. H., Li, P. H., Chen, K. K., and Wu, M. J. (2010). Integrating web mining and neural network for personalized e-commerce automatic service. *Expert Systems with Applications*, 37(4):2898–2910.

[33] Colombo-Mendoza, L. O., Valencia-García, R., Rodríguez-González, A., Alor-Hernández, G., and Samper-Zapater, J. J. (2015). Recommetz: A context-aware knowledge-based mobile recommender system for movie showtimes. *Expert Systems with Applications*, 42(3):1202–1222.

[34] Colomo-Palacios, R., GarcÃa-PeÃ±alvo, F. J., Stantchev, V., and Misra, S. (2017). Towards a social and context-aware mobile recommendation system for tourism. *Pervasive and Mobile Computing*, 38:505 – 515. Special Issue IEEE International Conference on Pervasive Computing and Communications (PerCom) 2016.

[35] Costa, H. and Macedo, L. (2013). Emotion-based recommender system for overcoming the problem of information overload. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 178–189. Springer.

[36] De Clercq, O., Schuhmacher, M., Ponzetto, S. P., and Hoste, V. (2014). Exploiting framenet for content-based book recommendation. *CEUR Workshop Proceedings*, 1245:14–20.

[37] Deng, Z.-j., Zhang, F. E. I., and Wang, S. P. S. (2016). SHILLING ATTACK DETECTION IN COLLABORATIVE FILTERING. pages 10–13.

[38] Desarkar, M. S. and Sarkar, S. (2012). Rating prediction using preference relations based matrix factorization. In *UMAP Workshops*.

[39] Devi, M. K., Samy, R. T., Kumar, S. V., and Venkatesh, P. (2010). Probabilistic neural network approach to alleviate sparsity and cold start problems in collaborative recommender systems. In *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*, pages 1–4. IEEE.

[40] Diaby, M., Viennet, E., and Launay, T. (2013). Toward the next generation of recruitment tools: an online social network-based job recommender system. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 821–828. ACM.

[41] Fagá-Jr, R., Furtado, B. C., Maximino, F., Cattelan, R. G., Pimentel, M. D. G. C., and Fagá, R. (2009). Context information exchange and sharing in a peer-to-peer community: a video annotation scenario. *Proceedings of the 27th ACM international conference on Design of communication - SIGDOC '09*, pages 265–272.

[42] Ganesan, K. (2010). Opinrank review dataset.

[43] Gartrell, M., Xing, X., Lv, Q., Beach, A., Han, R., Mishra, S., and Seada, K. (2010). Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work*, pages 97–106. ACM.

[44] Gaudioso, E. (2008). Evaluation of recommender systems : A new approach. 35:790–804.

[45] Georgiev, K. and Nakov, P. (2013). A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines. *Icml*, 28:1–9.

[46] Guy, I., Jaimes, A., Agulló, P., Moore, P., Nandy, P., Nastar, C., and Schinzel, H. (2010). The RecSys 2010 Industry Panel Will Recommenders Kill Search ? Recommender Systems – An Industry Perspective. pages 7–12.

[47] Han, B.-j., Rho, S., Jun, S., and Hwang, E. (2010). Music emotion classification and context-based music recommendation. *Multimedia Tools and Applications*, 47(3):433–460.

[48] Hariri, N. and Burke, R. (2012). Context-aware music recommendation based on latent topic sequential patterns. In *Proceedings RecSys 2012*, pages 131–138.

[49] Herlocker, J. L., Konstan, J. a., Terveen, L. G., and Riedl, J. T. (2004a). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53.

[50] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004b). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

[51] Hidasi, B. and Tikk, D. (2012). Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 67–82.

[52] Hidasi, B. and Tikk, D. (2016). Speeding up als learning via approximate methods for context-aware recommendations. *Knowledge and Information Systems*, 47(1):131–155.

[53] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554.

[54] Honrado, J. I., Huarte, O., Jimenez, C., Ortega, S., Perez-Aguera, J. R., Perez-Iglesias, J., Polo, A., and Rodriguez, G. (2016). Jobandtalent at recsys challenge 2016. In *Proceedings of the Recommender Systems Challenge*, RecSys Challenge '16, pages 3:1–3:5, New York, NY, USA. ACM.

[55] Hu, Y., Park, F., Koren, Y., and Volinsky, C. (2008). Collaborative Filtering for Implicit Feedback Datasets. pages 263–272.

[56] Huang, S.-L. (2011). Designing utility-based recommender systems for e-commerce: Evaluation of preference-elicitation methods. *Electronic Commerce Research and Applications*, 10(4):398–407.

[57] Hussein, T., Linder, T., Gaulke, W., and Ziegler, J. (2014). Hybreed: A software framework for developing context-aware hybrid recommender systems. *User Modeling and User-Adapted Interaction*, 24:121–174.

[58] Ingwersen, P. and Järvelin, K. (2005). *The turn: integration of information seeking and retrieval in context*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[59] Inzunza, S., Juárez-Ramírez, R., and Ramírez-Noriega, A. (2016). *User and Context Information in Context-Aware Recommender Systems: A Systematic Literature Review*, pages 649–658. Springer International Publishing, Cham.

[60] jae Kim, K. and Ahn, H. (2008). A recommender system using GA K-means clustering in an online shopping market. *Expert Systems with Applications*, 34(2):1200–1209.

[61] Jaitly, N., Nguyen, P., Senior, A., and Vanhoucke, V. (2012). Application of Pretrained Deep Neural Networks to Large Vocabulary Speech Recognition. *Interspeech*, (Cd):2–5.

[62] Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, pages 135–142.

[63] Jiang, M., Cui, P., Liu, R., Yang, Q., Wang, F., Zhu, W., and Yang, S. (2012). Social contextual recommendation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 45–54. ACM.

[64] Jiang, M., Cui, P., Wang, F., Zhu, W., and Yang, S. (2014). Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2789–2802.

[65] Kanetkar, S., Nayak, A., Swamy, S., and Bhatia, G. (2014). Web-based personalized hybrid book recommendation system. In *International Conference on Advances in Engineering and Technology Research*, pages 1–5.

[66] Karatzoglou, A. (2013). Recommender Systems Telefonica Research in Barcelona. Technical report.

[67] Karatzoglou, A., Amatriain, X., and Oliver, N. (2010). Multiverse Recommendation : N-dimensional Tensor Factorization for Context-aware Collaborative Filtering. pages 79–86.

[68] Karpus, A., di Noia, T., Tomeo, P., and Goczyła, K. (2016). Using contextual conditional preferences for recommendation tasks: a case study in the movie domain. *Studia Informatica*, 37(1):7–18.

[69] Katarya, R. and Verma, O. P. (2016). Effectivecollaborative movie recommender system using asymmetric user similarity and matrix factorization. In *Computing, Communication and Automation (ICCCA), 2016 International Conference on*, pages 71–75. IEEE.

[70] Kefalas, P., Symeonidis, P., and Manolopoulos, Y. (2013). New perspectives for recommendations in location-based social networks. *Proceedings MEDES 2013*, pages 1–8.

[71] Konstas, I., Stathopoulos, V., and Jose, J. M. (2009). On Social Networks and Collaborative Recommendation. *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 195–202.

[72] Koren, Y. (2008). Factorization meets the neighborhood: a multifaceted collaborative filtering model. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434.

[73] Koren, Y. and Bell, R. (2015). Advances in collaborative filtering. In *Recommender systems handbook*, pages 77–118. Springer.

[74] Koren, Y., Bell, R., and Volinsky, C. (2009a). Matrix factorization techniques for recommender systems. *Computer*, 8:30–37.

[75] Koren, Y., Bell, R., and Volinsky, C. (2009b). Matrix factorization techniques for recommender systems. *Computer*, 42(8).

[76] Kothari, A. A. and Patel, W. D. (2015). A Novel Approach Towards Context Based Recommendations Using Support Vector Machine Methodology. *Procedia Computer Science*, 57:1171–1178.

[77] Krstic, M. and Bjelica, M. (2012). Context-aware personalized program guide based on neural network. *IEEE Transactions on Consumer Electronics*, 58(4):1301–1306.

[78] Ledesma, S., Cerda, G., and Avi, G. (2008). Networks. pages 351–359.

[79] Lee, J., Sun, M., and Lebanon, G. (2012). PREA : Personalized Recommendation Algorithms Toolkit. *The Journal of Machine Learning Research*, 13:2699–2703.

[80] Lemos, F. D. a., Carmo, R. a. F., Viana, W., and Andrade, R. M. C. (2012). Towards a context-aware photo recommender system. *Proceedings of the 4th Workshop on Context-Aware Recommender Systems.*, 889.

[81] Levandoski, J. J., Sarwat, M., Eldawy, A., and Mokbel, M. F. (2012). Lars: A location-aware recommender system. In *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*, pages 450–461. IEEE.

[82] Levi, A., Mokryn, O. O., Diot, C., and Taft, N. (2012). Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings RecSys 2012*, pages 115–122.

[83] Li, J. and Zhang, W. (2014). Conditional Restricted Boltzmann Machines for cold start recommendations. *arXiv preprint arXiv:1408.0096*.

[84] Lian, J., Zhang, F., Hou, M., Wang, H., Xie, X., and Sun, G. (2017). Practical lessons for job recommendations in the cold-start scenario. In *Proceedings of the Recommender Systems Challenge 2017*, RecSys Challenge '17, pages 4:1–4:6, New York, NY, USA. ACM.

[85] Liu, F. and Lee, H. J. (2010). Use of social network information to enhance collaborative filtering performance. *Expert Systems with Applications*, 37(7):4772–4778.

[86] Liu, K., Shi, X., Kumar, A., Zhu, L., and Natarajan, P. (2016). Temporal learning and sequence modeling for a job recommender system. In *Proceedings of the Recommender Systems Challenge*, RecSys Challenge '16, pages 7:1–7:4, New York, NY, USA. ACM.

[87] Liu, T., Wang, F., and Agrawal, G. (2010). Stratified Sampling for Data Mining on the Deep Web. *2010 IEEE International Conference on Data Mining*, pages 324–333.

[88] Liu, X. and Aberer, K. (2013). Soco: a social network aided context-aware recommender system. In *Proceedings of the 22nd international conference on World Wide Web*, pages 781–802. ACM.

[89] Liu, Z., Wang, Y.-X., and Smola, A. (2015). Fast differentially private matrix factorization. In *Proceedings RecSys 2015*, pages 171–178.

[90] Mahmood, T. and Ricci, F. (2009). Improving recommender systems with adaptive conversational strategies. *Proceedings of the 20th ACM conference on Hypertext and hypermedia - HT '09*, pages 73–82.

[91] Marafi, S. (2014). Collaborative Filtering with R.

[92] Massa, P. and Avesani, P. (2007). Trust-aware recommender systems. *Proceedings of the 2007 ACM conference on Recommender systems RecSys 07*, 20:17–24.

[93] Mezei, Z. and Eickhoff, C. (2017). Evaluating music recommender systems for groups. *arXiv preprint arXiv:1707.09790*.

[94] Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2005). Effective Attack Models for Shilling Item-Based Collaborative Filtering System. *Proceedings of the seventh International Workshop on Knowledge Discovery from the Web (WEBKDD 2005)*, pages 13–23.

[95] Mustafa, G. and Frommholz, I. (2015). Performance comparison of top N recommendation algorithms. *2015 4th International Conference on Future Generation Communication Technology, FGCT 2015*, (Fgct):100–105.

[96] Mustafa, G. and Frommholz, I. (2016). Comparing contextual and non-contextual features in ANNs for movie rating prediction. In Krestel, R., Mottin, D., and Müller, E., editors, *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", Workshop Information Retrieval*, Potsdam, Germany. CEUR-WS.org.

[97] Nilashi, M., Bin Ibrahim, O., Ithnin, N., and Sarmin, N. H. (2015). A multi-criteria collaborative filtering recommender system for the tourism domain using Expectation Maximization (EM) and PCA-ANFIS. *Electronic Commerce Research and Applications*, 14(6):542–562.

[98] Noia, T. D. I. and Tomeo, P. (2016). Using Contextual Conditional Preferences for Recommendation Tasks : a Case Study in the Movie. 37(1).

[99] Odić, A., Tkalčič, M., Tasič, J. F., and Košir, A. (2012). Relevant context in a movie recommender system: Users' opinion vs. statistical detection. *CEUR Workshop Proceedings*, 889.

[100] Odić, A., Tkalčič, M., Tasič, J. F., and Košir, A. (2013). Predicting and detecting the relevant contextual information in a movie-recommender system. *Interacting with Computers*, 25(1):74–90.

[101] Panniello, U. and Gorgoglione, M. (2011). A contextual modeling approach to context-aware recommender systems. *CEUR Workshop Proceedings*, 791.

[102] Panniello, U., Tuzhilin, A., and Gorgoglione, M. (2014). Comparing context-aware recommender systems in terms of accuracy and diversity. *User Modeling and User-Adapted Interaction*, 24(1):35–65.

[103] Park, M.-h., Hong, J.-h., and Cho, S.-b. (2007). Location-Based Recommendation System Using Bayesian User ' s Preference Model in Mobile Devices. *Ubiquitous Intelligence and Computing*, pages 1130–1139.

[104] Pazzani, M. J. and Billsus, D. (2007). *Content-Based Recommendation Systems*, pages 325–341. Springer Berlin Heidelberg, Berlin, Heidelberg.

[105] Pilászy, I. (2009). Recommending New Movies : Even a Few Ratings Are More Valuable Than Metadata Categories and Subject Descriptors. pages 93–100.

[106] Qiao, X., Li, X., Su, Z., and Cao, D. (2011). A Context-Awareness Dynamic Friend Recommendation. *International Journal of Advanced Intelligence*, 3(2):155–172.

[107] Ramos, C., Augusto, J. C., and Shapiro, D. (2008). Ambient Intelligence—the Next Step for Artificial Intelligence. *IEEE Intelligent Systems*, 23(2):15–18.

[108] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134. ACM.

[109] Renaud-Deputter, S., Xiong, T., and Wang, S. (2013). Combining collaborative filtering and clustering for implicit recommender system. In *27th International Conference on Advanced Information Networking and Applications (AINA 2013)*, pages 748–755. IEEE.

[110] Rendle, S., Gantner, Z., Freudenthaler, C., and Schmidt-Thieme, L. (2011). Fast context-aware recommendations with factorization machines. *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pages 635–644.

[111] Ricci, F. (2010). Mobile Recommender Systems. *Information Technology & Tourism*, 12:205–231.

[112] Sae-Ueng, S., Pinyapong, S., Ogino, A., and Kato, T. (2008). Personalized shopping assistance service at ubiquitous shop space. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pages 838–843.

[113] Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th international conference on Machine learning - ICML '07*, pages 791–798.

[114] Sandvig, J. J., Mobasher, B., and Burke, R. (2007). Robustness of collaborative recommendation based on association rule mining. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 105–112. ACM.

[115] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th ...*, 1:285–295.

[116] Sarwat, M., Eldawy, A., Mokbel, M. F., and Riedl, J. (2013). Plutus: leveraging location-based social networks to recommend potential customers to venues. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on*, volume 1, pages 26–35. IEEE.

[117] Schafer, J. (2009). The application of data-mining to recommender systems. *Encyclopedia of data warehousing and mining*, 1:44–48.

[118] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg.

[119] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 02*, 46(Sigir):253–260.

[120] Serrano, A. J., Soria, Emilio, M. J. D., Magdalena, R., and Gómez, J. (2010). Feature selection using ROC curves on classification problems. In *International Joint Conference on Neural Networks (IJCNN 2010)*, pages 1–6.

[121] Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R. (2008). Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 259–266. ACM.

[122] Sivapalan, S., Sadeghian, A., Rahnama, H., and Madni, A. M. (2014). Recommender systems in e-commerce. *World Automation Congress Proceedings*, pages 179–184.

[123] Smucker, M. D., Allan, J., and Carterette, B. (2007). A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 623–632. ACM.

[124] Stern, D., Herbrich, R., and Graepel, T. (2009). Matchbox : Large Scale Online Bayesian Recommendations. *Proceedings of the 18th International Conference on World Wide Web (WWW'09)*, pages 111–120.

[125] Su, X. and Khoshgoftaar, T. M. (2009a). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009(Section 3):1–19.

[126] Su, X. and Khoshgoftaar, T. M. (2009b). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009(Section 3):1–19.

[127] Su, X. and Khoshgoftaar, T. M. (2009c). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4.

[128] Symeonidis, P., Tiakas, E., and Manolopoulos, Y. (2011). Product recommendation and rating prediction based on multi-modal social networks. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 61–68. ACM.

[129] Tuzhilin, G. A. and Alexander (2011). *Advances in Collaborative Filtering*.

[130] Unger, M., Bar, A., Shapira, B., and Rokach, L. (2016). Towards latent context-aware recommendation systems. *Knowledge-Based Systems*, 104:165–178.

[131] Van den Oord, A., Dieleman, S., and Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651.

[132] Vargas-Govea, B., González-Serna, G., and Ponce-Medellın, R. (2011). Effects of relevant contextual features in the performance of a restaurant recommender system. *ACM RecSys*, 11(592):56.

[133] Volkovs, M., Yu, G. W., and Poutanen, T. (2017). Content-based neighbor models for cold start in recommender systems. In *Proceedings of the Recommender Systems Challenge 2017*, RecSys Challenge '17, pages 7:1–7:6, New York, NY, USA. ACM.

[134] Wang, H., Wang, N., and Yeung, D.-Y. (2015a). Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM.

[135] Wang, J., de Vries, A., and Reinders, M. (2006). On Combining User-based and Item-based Collaborative Filtering Approaches. pages 501–508.

[136] Wang, S. L. and Wu, C. Y. (2011). Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. *Expert Systems with Applications*, 38(9):10831–10838.

[137] Wang, X., Wen, J., Luo, F., Zhou, W., and Ren, H. (2015b). Personalized Recommendation System Based on Support Vector Machine and Particle Swarm Optimization. *International Conference on Knowledge Science, Engineering and Management*, pages 489–495.

[138] Weinberger, M., Kirkman, M. S., Samsa, G. P., Cowper, P. A., Shortliffe, E. A., Simel, D. L., and Feussner, J. R. (1994). The relationship between glycemic control and health-related quality of life in patients with non-insulin-dependent diabetes mellitus. *Medical care*, pages 1173–1181.

[139] Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[140] Xia, M. J. and Zhang, J. G. (2010). Research on personalized recommendation system for e-commerce based on web log mining and user browsing behaviors. *ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings*, 12(Iccasm):408–411.

[141] Xin, L. and Aberer, K. (2013). Personalized Point-of-Interest Recommendation by Mining Users ' Preference Transition. *Proceedings CIKM 2013*, pages 733–738.

[142] Xu, G. (2008). Web Mining Techniques for Recommendation and Personalization. 2(5).

[143] Yakut, I. and Polat, H. (2012). Estimating nbc-based recommendations on arbitrarily partitioned data with privacy. *Knowledge-Based Systems*, 36:353–362.

[144] Yang, J. B., Shen, K. Q., Ong, C. J., and Li, X. P. (2009). Feature selection for MLP neural network: The use of random permutation of probabilistic outputs. *IEEE Transactions on Neural Networks*, 20(12):1911–1922.

[145] Yin, H., Cui, B., Chen, L., Hu, Z., and Zhang, C. (2015). Modeling location-based user rating profiles for personalized recommendation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(3):19.

[146] Yu, X. Y. X. and Sun, S. S. S. (2010). Research on Personalized Recommendation System Based on Web Mining. *E-Business and E-Government (ICEE), 2010 International Conference on*, 2:1–5.

[147] Zammali, S., Arour, K., and Bouzeghoub, A. (2015). A Context Features Selecting and Weighting Methods for Context-Aware Recommendation. *2015 IEEE 39th Annual Computer Software and Applications Conference*, 5:575–584.

[148] Zhang, Y., Lai, G., Zhang, M., Zhang, Y., Liu, Y., and Ma, S. (2014). Explicit Factor Models for Explainable Recommendation based on Phrase-level Sentiment Analysis Categories and Subject Descriptors. In *Proceedings SIGIR 2014*, pages 83–92.

[149] Zheng, Y., Burke, R., and Mobasher, B. (2013). The role of emotions in context-aware recommendation. *CEUR Workshop Proceedings*, 1050:21–28.

[150] Zheng, Y., Burke, R., and Mobasher, B. (2014). Splitting approaches for context-aware recommendation: An empirical study. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 274–279. ACM.

[151] Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R., and Zhang, Y.-C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences*, 107(10):4511–4515.

Appendix A

ANNs Confusion Matrices

Some of the confusion matrices are given for the results presented in the form of tables in Chapter 5. The confusion matrices for table 5.23 (All contextual combinations) from ANNs are given.

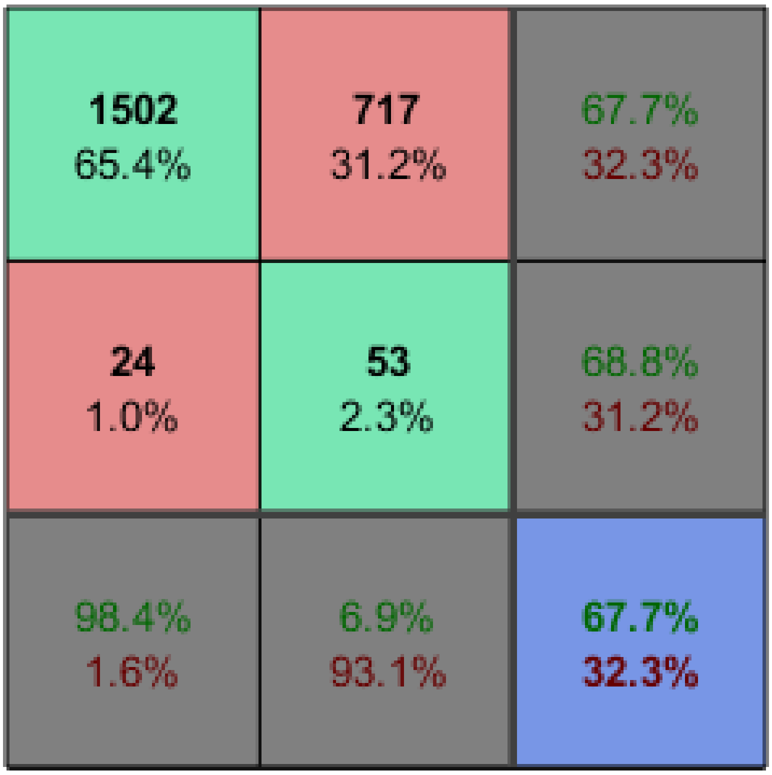

Fig. A.1 ANNs results for Location with combination C2 in table 5.9 with accuracy of 67.7% and precision 98.4%.

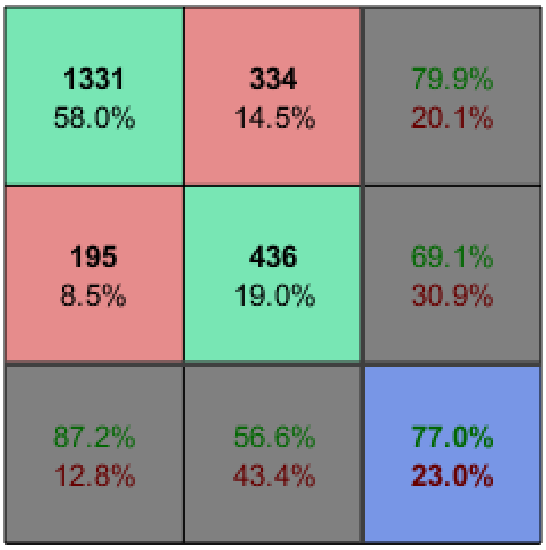

Fig. A.2 Confusion matrix for Combination number 1 of contextual features given in table 5.23

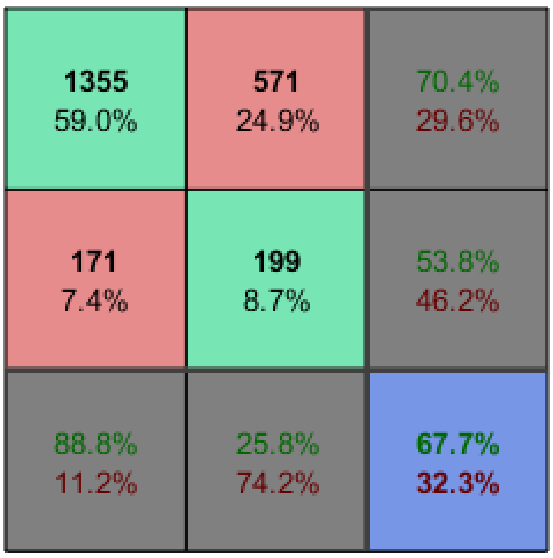

Fig. A.3 Confusion matrix for Combination number 2 of contextual features given in table 5.23

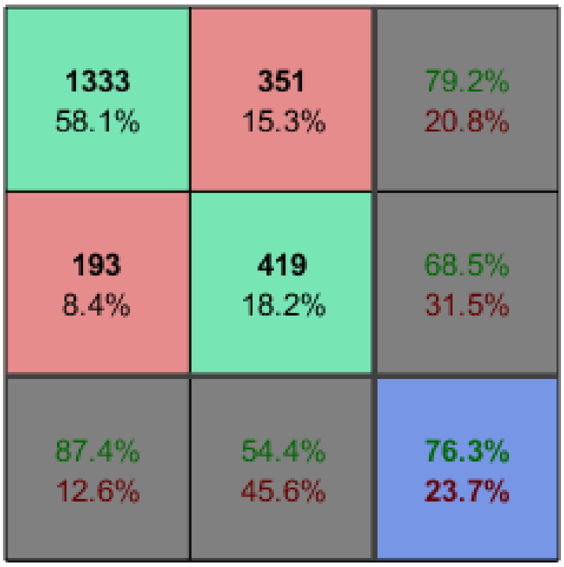

Fig. A.4 Confusion matrix for Combination number 3 of contextual features given in table 5.23

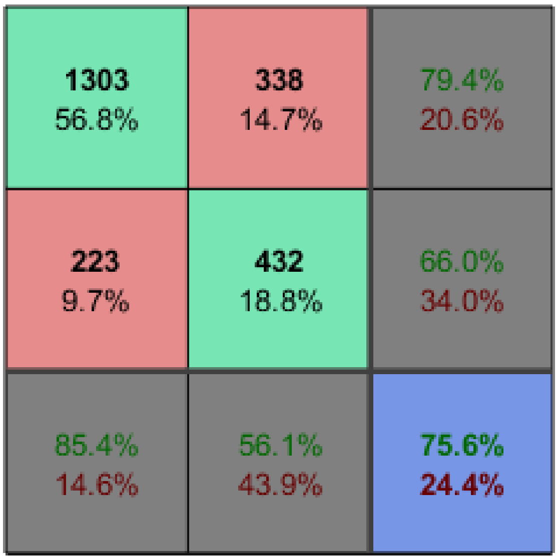

Fig. A.5 Confusion matrix for Combination number 4 of contextual features given in table 5.23

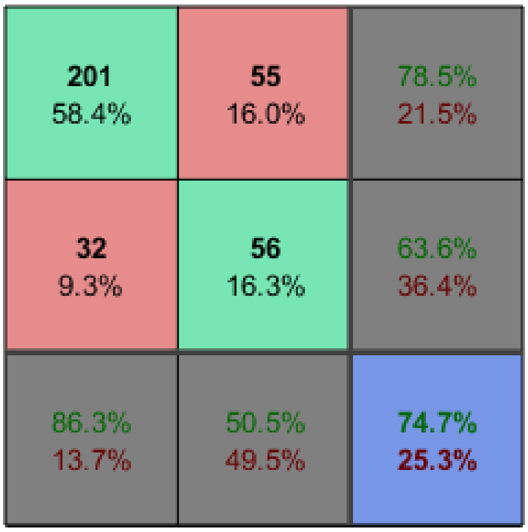

Fig. A.6 Confusion matrix for Combination number 5 of contextual features given in table 5.23

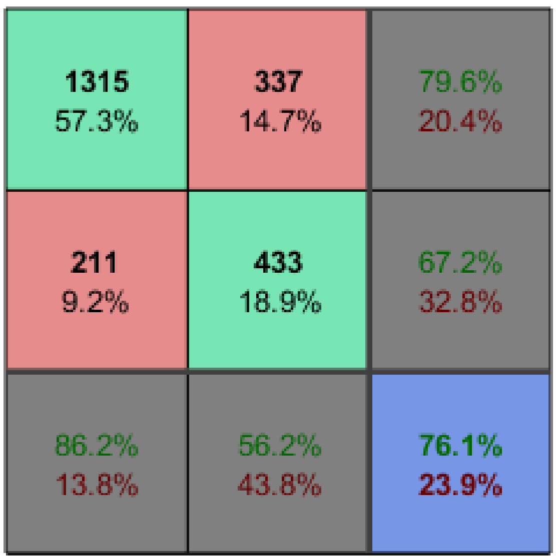

Fig. A.7 Confusion matrix for Combination number 6 of contextual features given in table 5.23

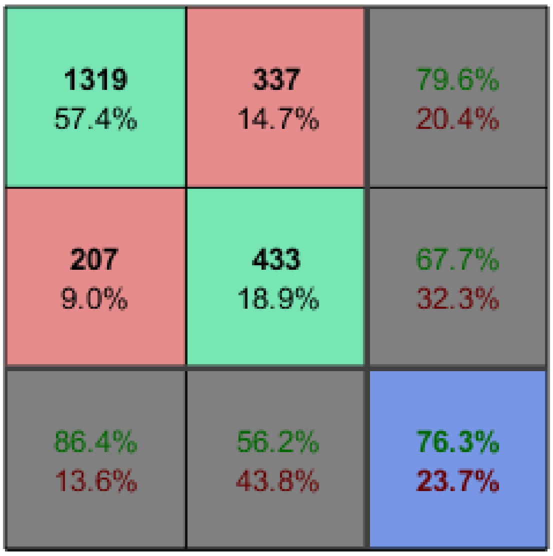

Fig. A.8 Confusion matrix for Combination number 7 of contextual features given in table 5.23

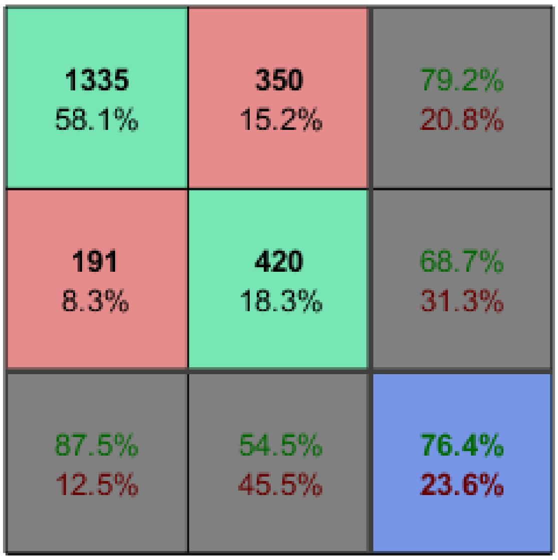

Fig. A.9 Confusion matrix for Combination number 8 of contextual features given in table 5.23

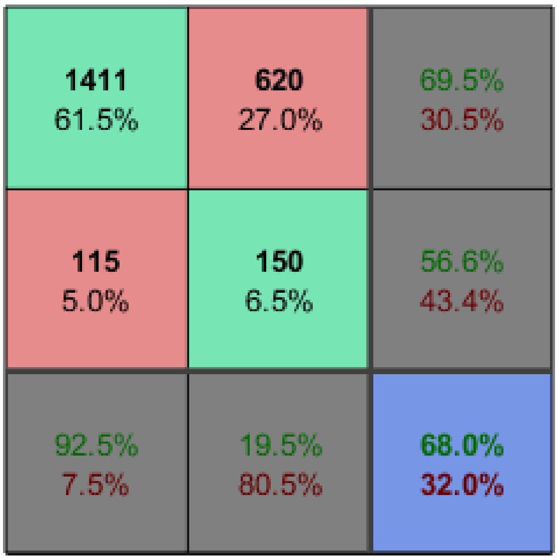

Fig. A.10 Confusion matrix for Combination number 9 of contextual features given in table 5.23

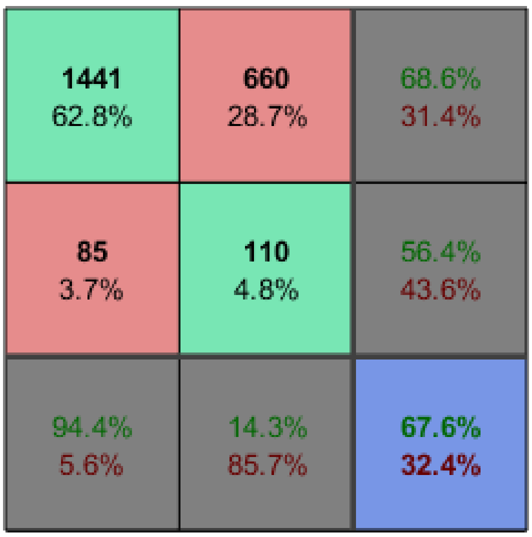

Fig. A.11 Confusion matrix for Combination number 10 of contextual features given in table 5.23

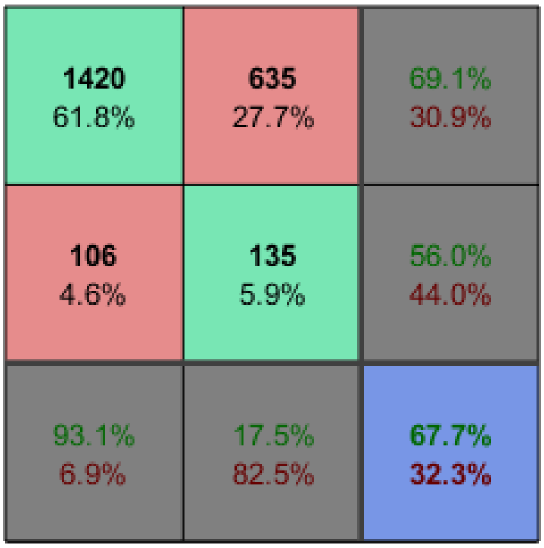

Fig. A.12 Confusion matrix for Combination number 11 of contextual features given in table 5.23

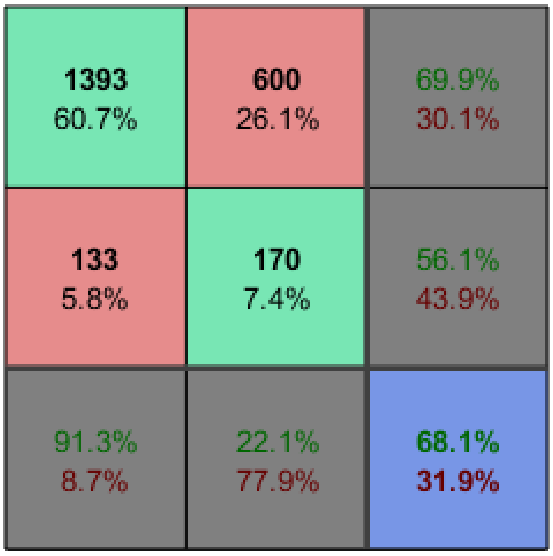

Fig. A.13 Confusion matrix for Combination number 12 of contextual features given in table 5.23

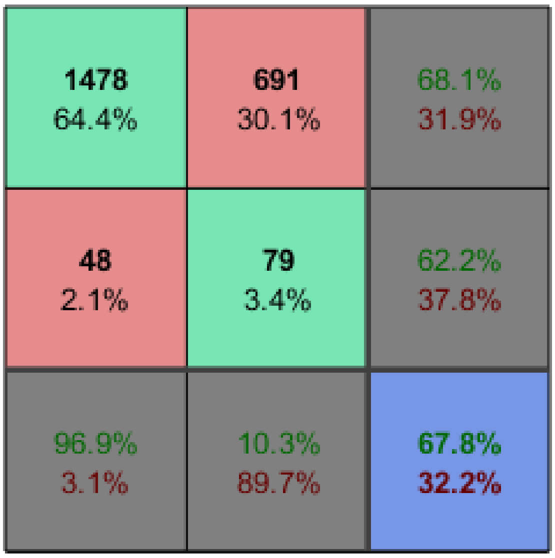

Fig. A.14 Confusion matrix for Combination number 13 of contextual features given in table 5.23

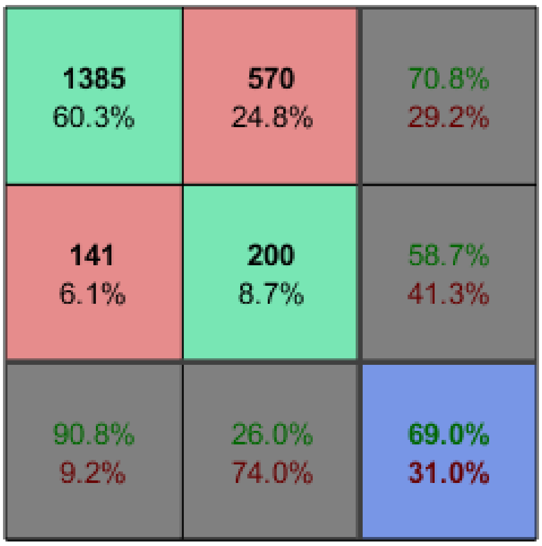

Fig. A.15 Confusion matrix for Combination number 14 of contextual features given in table 5.23

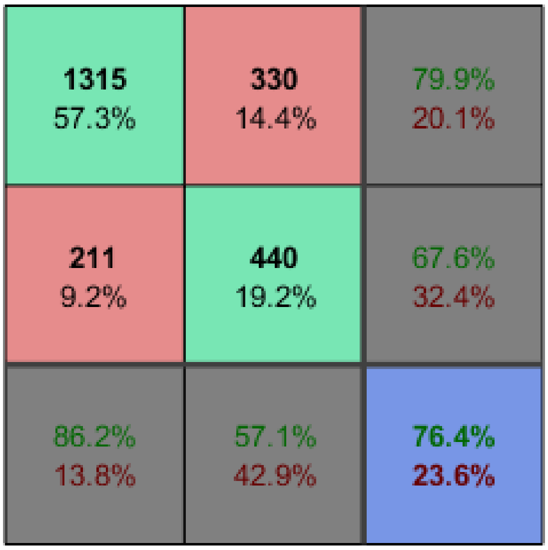

Fig. A.16 Confusion matrix for Combination number 15 of contextual features given in table 5.23

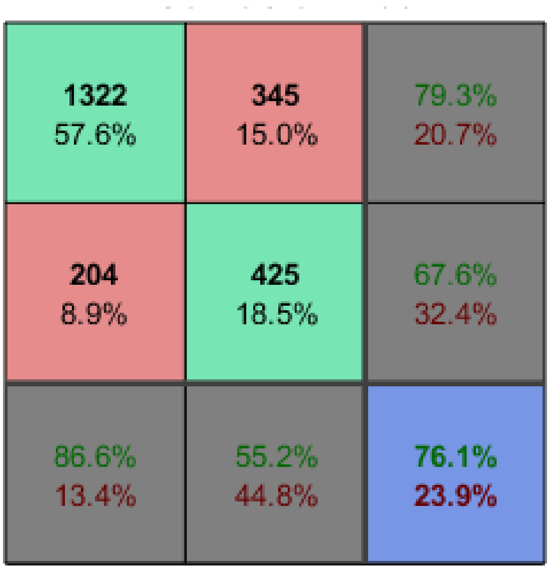

Fig. A.17 Confusion matrix for Combination number 16 of contextual features given in table 5.23

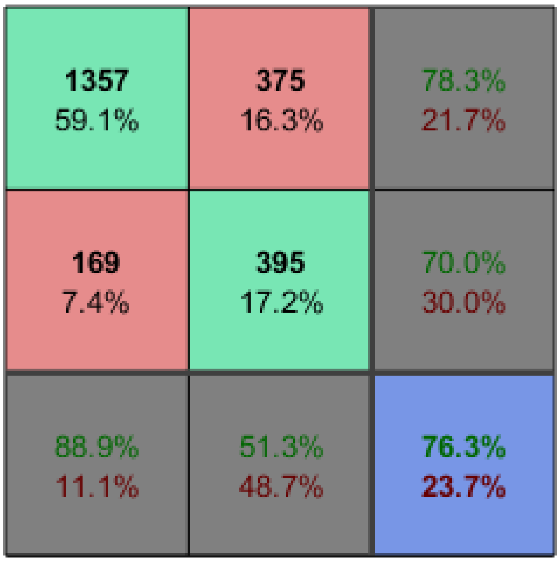

Fig. A.18 Confusion matrix for Combination number 17 of contextual features given in table 5.23

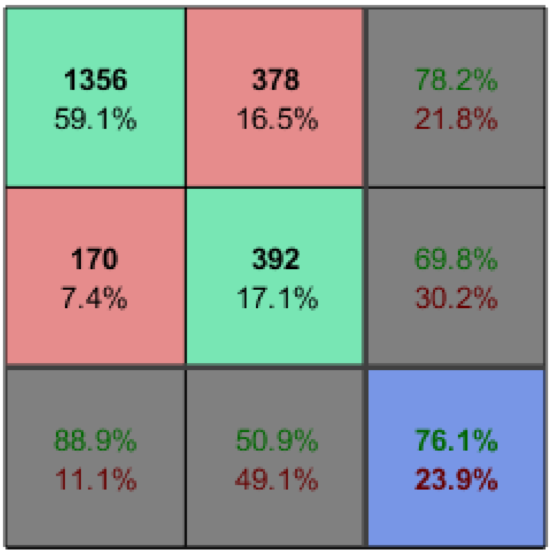

Fig. A.19 Confusion matrix for Combination number 18 of contextual features given in table 5.23

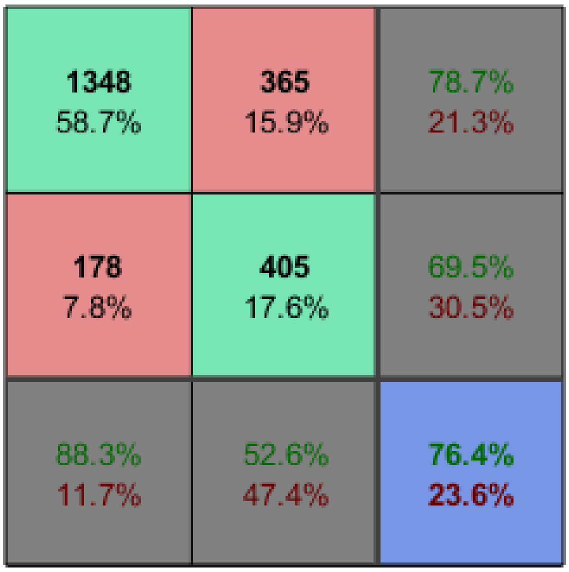

Fig. A.20 Confusion matrix for Combination number 19 of contextual features given in table 5.23

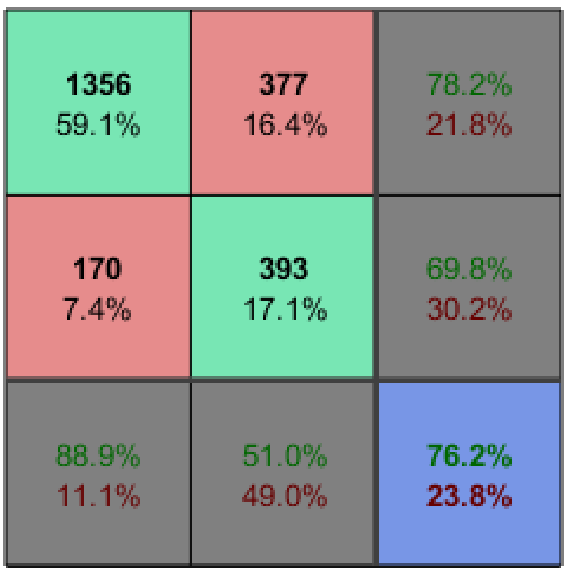

Fig. A.21 Confusion matrix for Combination number 20 of contextual features given in table 5.23

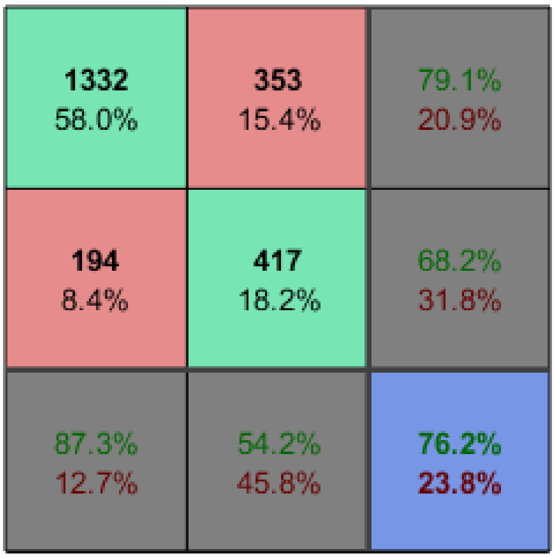

Fig. A.22 Confusion matrix for Combination number 21 of contextual features given in table 5.23

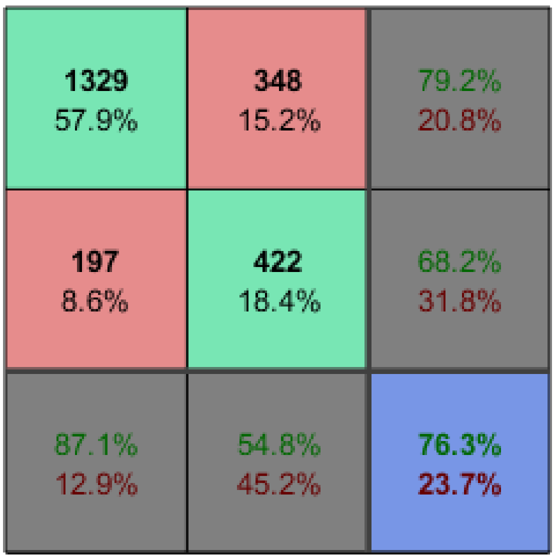

Fig. A.23 Confusion matrix for Combination number 22 of contextual features given in table 5.23

1343 58.5%	**372** 16.2%	78.3% 21.7%
183 8.0%	**398** 17.3%	68.5% 31.5%
88.0% 12.0%	51.7% 48.3%	**75.8%** **24.2%**

Fig. A.24 Confusion matrix for Combination number 23 of contextual features given in table 5.23

1328 57.8%	**340** 14.8%	79.6% 20.4%
198 8.6%	**430** 18.7%	68.5% 31.5%
87.0% 13.0%	55.8% 44.2%	**76.6%** **23.4%**

Fig. A.25 Confusion matrix for Combination number 24 of contextual features given in table 5.23

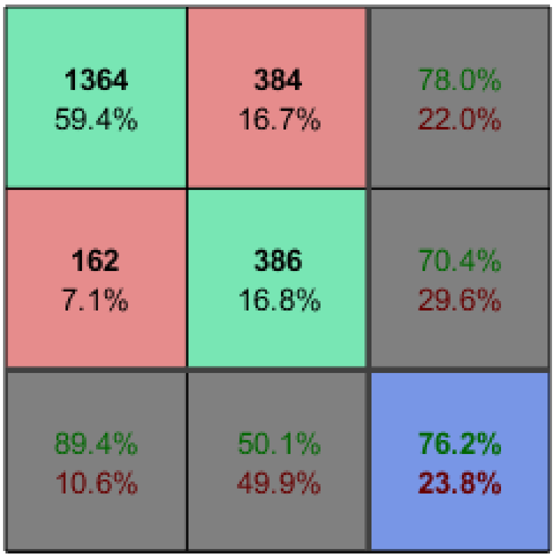

Fig. A.26 Confusion matrix for Combination number 25 of contextual features given in table 5.23

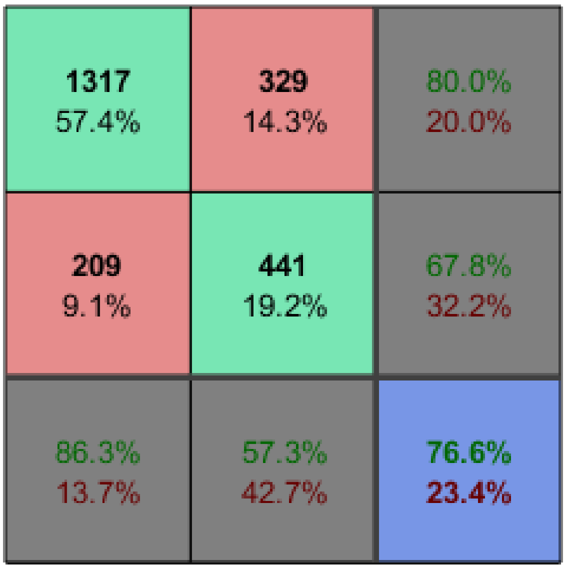

Fig. A.27 Confusion matrix for Combination number 26 of contextual features given in table 5.23

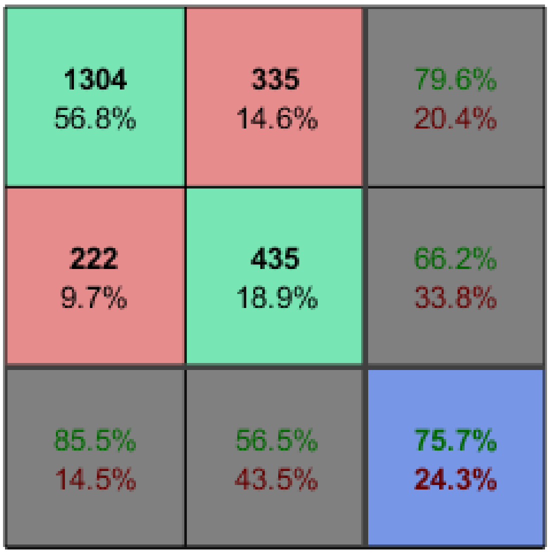

Fig. A.28 Confusion matrix for Combination number 28 of contextual features given in table 5.23

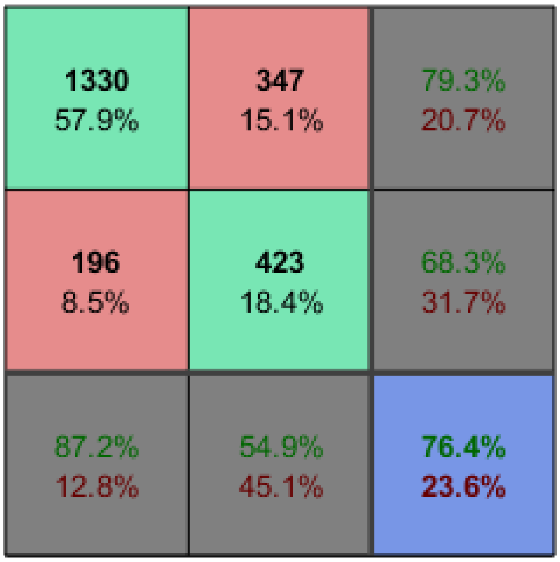

Fig. A.29 Confusion matrix for Combination number 29 of contextual features given in table 5.23

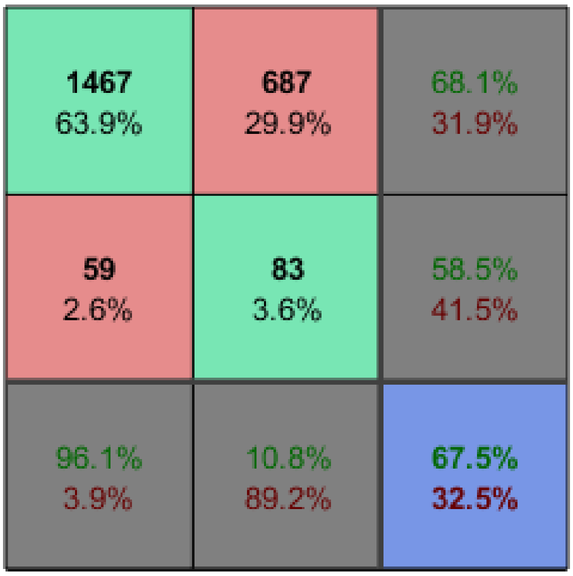

Fig. A.30 Confusion matrix for Combination number 30 of contextual features given in table 5.23

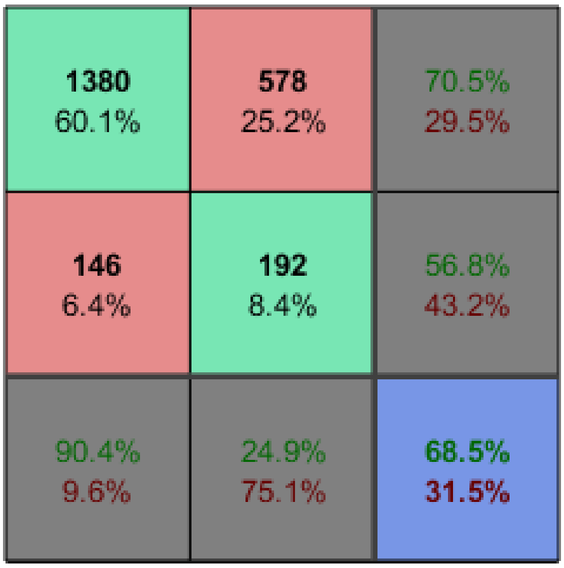

Fig. A.31 Confusion matrix for Combination number 31 of contextual features given in table 5.23

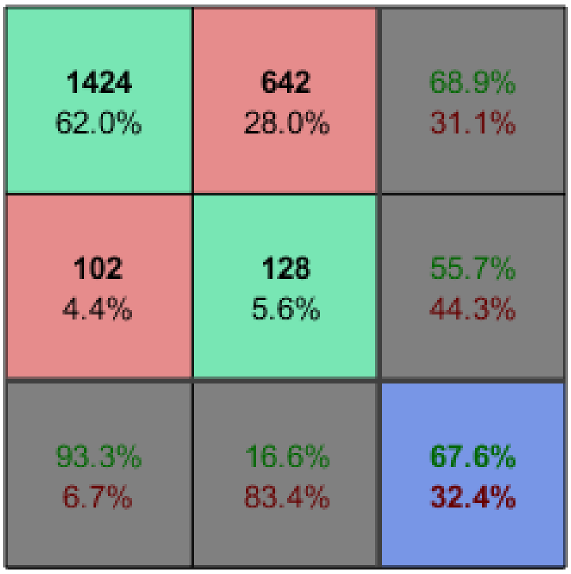

Fig. A.32 Confusion matrix for Combination number 32 of contextual features given in table 5.23

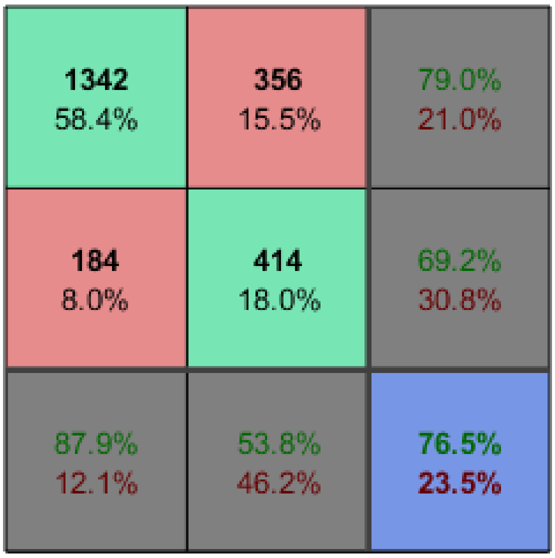

Fig. A.33 Confusion matrix for Combination number 33 of contextual features given in table 5.23

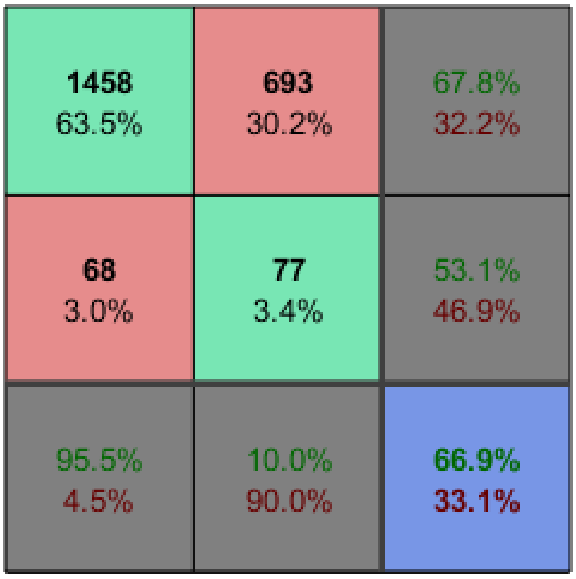

Fig. A.34 Confusion matrix for Combination number 34 of contextual features given in table 5.23