Dissertations

School of Computing

2015-05-09

An Evaluation of the Use of Diversity to Improve the Accuracy of Predicted Ratings in Recommender Systems

Gillian Browne
Technological University Dublin

Recommended Citation

An evaluation of the use of diversity to improve the accuracy of predicted ratings in recommender systems

Gillian Browne

A dissertation submitted in partial fulfilment of the requirements of

Dublin Institute of Technology for the degree of

M.Sc. in Computing (Data Analytics)

February 2015

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Analytics), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Dublin Institute of Technology and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: _____

Date: **28 February 2015**

1   ABSTRACT

The diversity; versus accuracy trade off, has become an important area of research within recommender systems as online retailers attempt to better serve their customers and gain a competitive advantage through an improved customer experience. This dissertation attempted to evaluate the use of diversity measures in predictive models as a means of improving predicted ratings. Research literature outlines a number of influencing factors such as personality, taste, mood and social networks in addition to approaches to the diversity challenge post recommendation.

A number of models were applied included DecisionStump, Linear Regression, J48 Decision Tree and Naive Bayes. Various evaluation metrics such as precision, recall, ROC area, mean squared error and correlation coefficient were used to evaluate the model types. The results were below a benchmark selected during the literature review. The experiment did not demonstrate that diversity measures as inputs improve the accuracy of predicted ratings. However, the evaluation results for the model without diversity measures were low also and comparable to those with diversity indicating that further research in this area may be worthwhile.

While the experiment conducted did not clearly demonstrate that the inclusion of diversity measures as inputs improve the accuracy of predicted ratings, approaches to data extraction, pre-processing, and model selection could inform further research. Areas of further research identified within this paper may also add value for those interested in this topic.

Key words: *Diversity, Recommender Systems, Classification, Knowledge Discovery, Data Mining*

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Luca Longo for his feedback and support which has been invaluable during this process.

I would also like to thank all of the lecturers that have contributed to my learning experience during the MSc in Computing.

Finally, I would like to thank all my friends and family, and in particular Alan, for their unfaltering support and encouragement.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 Background

There are a large volume of products available on many e-commerce sites. Amazon, for example, offers millions of products across 17 categories in conjunction with over 2 million third-party sellers (Amazon.com 2014). This presents a challenge for consumers with regard to reviewing and browsing products in a reasonable timeframe. Historically consumers could browse through a book or music store utilising facilities such as Virgin Megastore's in store headphones. This type of facility allowed a customer to sample or review a previously unknown product to see if they liked it prior to purchase. In addition, knowledgeable sales assistants were often available to make recommendations for products that the customer may like. Online retailers often choose to address this gap in their service offering through technology.

A recommender system is the online sales assistant that makes product suggestions that an e-commerce site user may like. It is called a recommender system because it presents recommendations of items that a user may find interesting. A sales assistant in a high street store can have a detailed conversation with the customer in order to make an informed product recommendation. The recommendation system does not have this advantage. Instead, the algorithms underpinning the recommender system may use previous purchase behaviour, browsing history, ratings or a series of questions that a user may or may not have chosen to answer as an alternative information gathering technique.

Users may provide feedback with regard to purchases made through ratings. Vozalis and Margaritis (2003) state that these ratings can be explicitly provided by the customer or gathered implicitly from their interactions. These ratings are often used as inputs to the recommendation system algorithms. Longo, Dondio and Barrett (2009) outline reading time, bookmarking, scrolling, form filling and editing as sources for determining preferences implicitly. Online retailers may also allow users to explicitly rate items. The following graphic shows the Amazon rating interface.
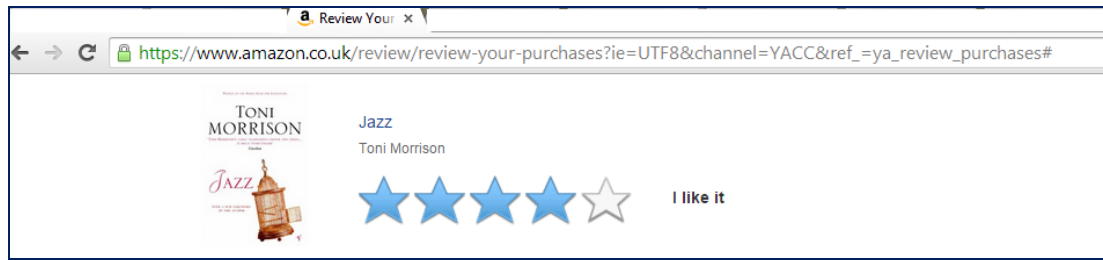
Figure 1 Amazon's customer rating interface.

Source: Amazon.com (2014)

In recent years there has been a big focus on accuracy in recommender systems but the challenge of dealing with accurate but poor value recommendations is becoming more prominent. Introducing diversity into recommendations systems is viewed as one approach to addressing this challenge.

Diversity is defined as "the state or quality of being different or varied" (Collins Dictionary 2014). It has a human aspect which influences its use within recommendations. Wu, Chen and Liang (2013) argue that personality influences choices. It is important to try to understand the human aspect and appetite for diversity. A one size fits all approach would not be appropriate as not all users are the same or have the same broad spectrum of tastes. Effort expended in addressing this challenge can have positive impacts for online retailers.

A level of diversity within the recommendations can add additional value for online retailers and customers alike. Adomavicius and Kwon (2009) state that an additional benefit to increased customer satisfaction is a reduction in cost to serve if diversity within recommendations can be applied effectively. *However, the desired diversity levels of customers can be difficult to identify.*

1.2 Research problem

Researchers have and continue to investigate ways to address the challenge of introducing an appropriate level of diversity into recommendations produced by recommender systems. Techniques used include search retrieval. Vargas, Castelis and Vallet (2011) suggest that using the user profile aspect allows the diversity approach used in search retrieval to be applied to recommender systems.

Alternative approaches are the use of customer profiles (Adomavicius and Tuzhilin 2005), social networks analysis (Pera and Ng 2011) and the use of personality attributes (Wu, Chen and Liang 2013) to introduce levels of diversity related to customer personality. Behavioural approaches include web browsing, opinion mining and sentiment analysis (Tao *et al.* 2013). Researchers are also investigating blunt approaches including segregated recommendation lists with higher or lower diversity (Linden, Smith and York 2003) excluding popular items (Adomavicius and Kwon 2009) and hybrid approaches (Bradley and Smith 2001). In addition, to approaches to include enhanced customer information, there is research measuring diversity post recommendation (Zeigler, McNee and Konstan 2005). While this research contributes to the body of knowledge, there are issues concerning the explicit nature of data capture, trust, accuracy and the fact that the measure of diversity is applied post recommendation. If users are requested to provide details of their connections with others or complete personality quizzes this may give rise to trust issues. Blunt approaches introduce the risk of an adverse affect on accuracy which in turn may reduce the perceived value of the recommender system. Some of the above studies include an approach to diversity after the recommendation has been selected. In this dissertation the goal is to investigate the application of diversity before recommendations are made with a view to improving the accuracy of predicted ratings.

As mentioned earlier, explicit data capture and the stage of implementation are drawbacks of some of the research explored during the preparation of this document. The research question proposed for evaluation in this paper is as follows:

Does diversity improve the accuracy of predicted ratings in recommender systems?

Many machine learning algorithms search for patterns in data to make accurate recommendations through training models. The research question is concerned with utilising measures of diversity as inputs to learning algorithms to predict future item ratings there by identifying them for potential inclusion as a recommendation.

1.3 Research objectives

The aim of this project is to assess if including measures of diversity using different classification approaches can assist with improving the accuracy of predicted ratings of previously unseen data.

The aim will be addressed through the preparation of a dataset including the calculation of a number of other measures of diversity. These additional metrics for each user will be calculated within the cleansed dataset. These data fields will be used as inputs to classification models that will be assessed for their suitability to the research problem. These models will be used to predict ratings for previously unseen items. Evaluation will be performed against a test dataset and a dataset where diversity measures were not used as inputs.

The objectives of the research are as follows:

a) Explore general issues, trends, diversity and algorithms used to predict ratings in recommender systems. This will involve the identification of gaps in current approaches

b) Obtain an understanding of the theory supporting the research question to assist with shaping an approach for quantitative analysis

c) Investigate an appropriate hypothesis with regard to the research question

d) Discuss and critically analyse the results of the investigation

e) Outline the contribution to the body of knowledge and identify areas for further research related to this project

1.4 Research methodology

In order to answer the research question a literature review of general issues, trends, diversity and algorithms used to predict ratings in recommender systems will be undertaken. This literature review will conclude with the identification of gaps in current approaches.

An experiment will be designed influenced by learnings from the literature review in support of the research question. This experiment will involve the use of a free dataset titled Book Crossing dataset [1] enhanced with an Amazon metadata[2] file.

An experiment aligned with the design that includes data analysis to facilitate data understanding and preparation will be undertaken. Data enrichment will be performed through the calculation of multiple measures of diversity. Quantitative analysis including the use of classification models will be used to predict ratings and to facilitate the empirical evaluation of the research question.

Analysis and discussion of the results including an overall evaluation of the experiment success or failure will be performed. The document will be concluded through the identification of the contribution to the body of knowledge and areas for further research related to this project.

1.5 Scope and limitations

A single dataset prepared using the Book Crossing and Amazon metadata datasets will be utilised. This is a limitation as further datasets of a similar nature are not available. A limitation of the dataset itself is the fact that there is no time dimension. The Book Crossings dataset was crawled in the summer of 2004 but there is no timestamp for each rating offered. The Amazon metadata dataset was obtained in summer 2006 and the date of customer ratings is available but this is not useful for this research paper. The creation of a recommendation system GUI is out of scope for this dissertation. Qualitative studies such as obtaining expert feedback and conducting participant tests and observations are also out of scope for this project.

---

[1] Book Crossing dataset sourced from http://www.informatik.uni-freiburg.de/~cziegler/BX/

[2] Amazon metadata dataset sourced from http://snap.stanford.edu/data/amazon-meta.html

1.6 Organisation of the dissertation

This dissertation is organised into a number of chapters. These chapters cover Literature Review, Experiment Design, Experiment Implementation, Experiment Evaluation, Conclusions and Future Work. The taxonomy below illustrates the structure of this dissertation.
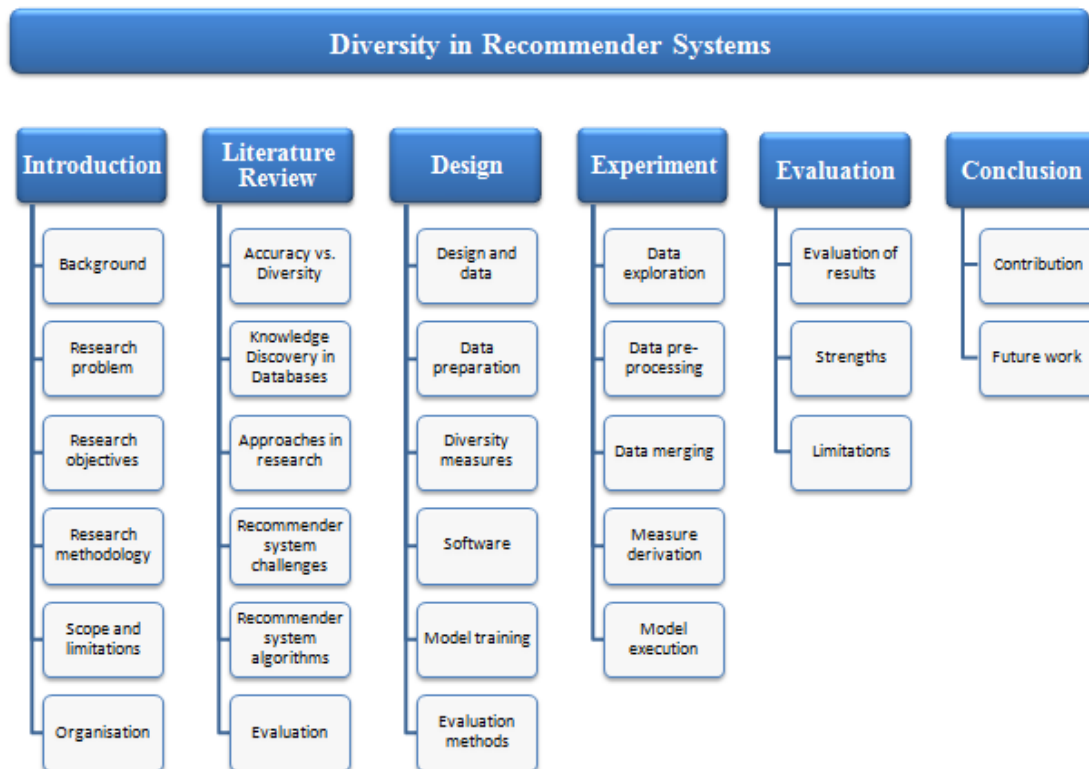
Figure 2 Dissertation structure

Chapter two will contain the literature review which will cover general challenges and trends related to algorithms underpinning recommender systems. Diversity and its application and impacts will also be reviewed. An examination of the algorithms used in predictive models, both those used in data mining in general and those used to predict ratings within recommender systems such as collaborative filtering and content filtering will be included.

Chapter three will cover the scope, design and implementation of the experiment. This chapter will also include the evaluation methods and details of the approach for comparative analysis of the results.

Chapter four will contain details of the data exploration and analysis conducted to facilitate data understanding in advance of building a model. This chapter will also include details of data transformation, cleansing and enrichment techniques applied to the dataset. Details of the experiment build including implementation of chosen models will be included.

Chapter five will include a detailed evaluation of the predictive model performance. Evaluation techniques include precision, recall, mean squared error and ROC. This detailed evaluation will refer back to the research examined within the Literature Review.

Chapter six will contain conclusions obtained from the research conducted and areas of future work identified throughout this analysis. This chapter will conclude the dissertation.

2 LITERATURE REVIEW

This Literature Review has been undertaken to explore research related to diversity and recommender systems. Figure 3 below provides an overview of the structure of this chapter.
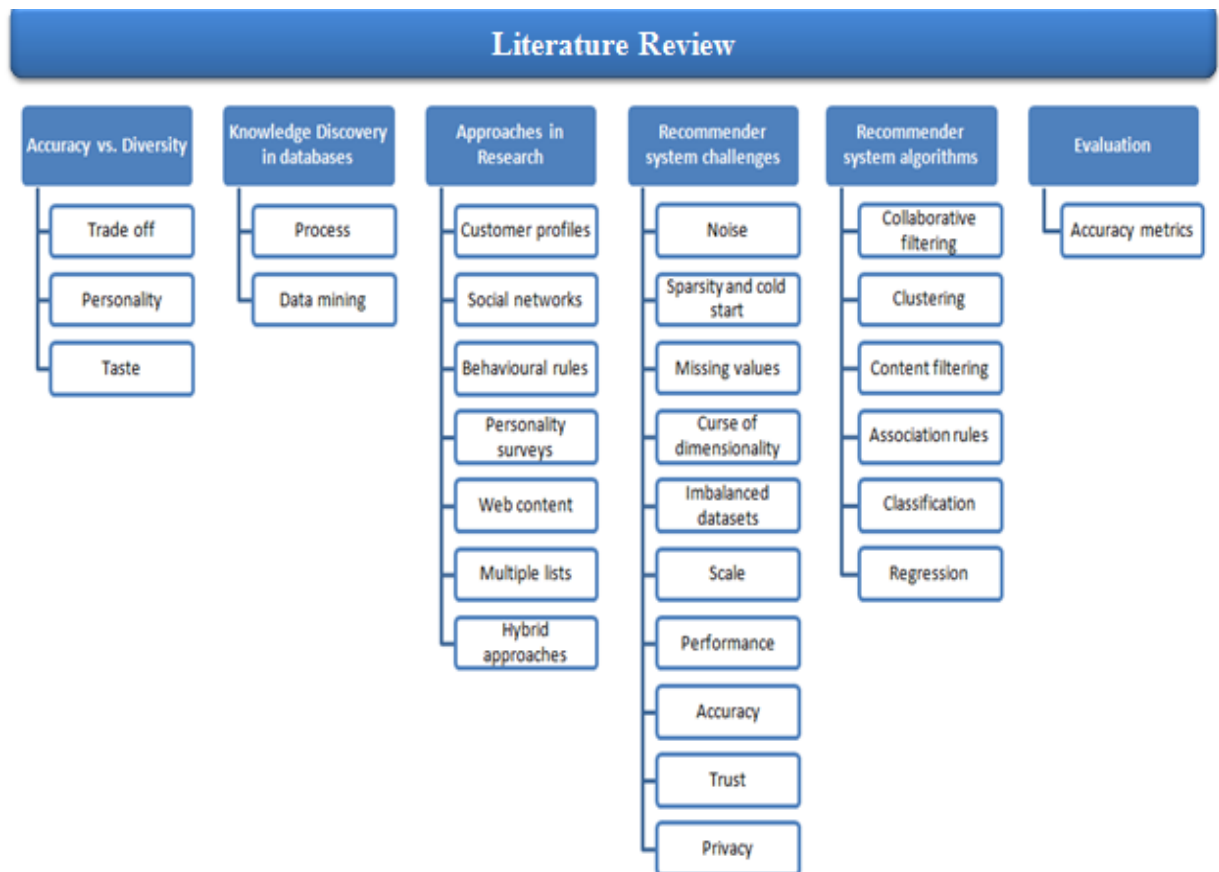
Figure 3 Structure of Literature Review

E-commerce websites have increasing amounts of content. Many businesses are using recommender systems to present suggestions to customers so that they do not have to search through lots of content to find items that may be of interest. The quality of the recommendation is a key challenge for recommender systems as recommendations that do not fit with the users preferences may negatively impact on the user experience. An inappropriate recommendation may discourage a customer from returning to the website.

Castells, Vargas, and Wang (2011) state that accuracy is just one metric that influences a successful recommendation. Diversity in the recommendation is also important but challenging to introduce.

Accurate recommendations that the user is very aware of will add little value, for example a book by a particular author recommended when the user has read other books by this author. As such a balance between accurate and diverse recommendations needs to be struck. Also, different users will have different appetites for diverse recommendations. The aim of this chapter is the provision of an overview of approaches to the application of diversity within recommender systems. In addition, the trends and challenges relating to data mining and recommender systems will be discussed. A discussion of data mining algorithms and those related to recommender systems and diversity will also be included in support of this research project.

2.1 The accuracy diversity challenge

Accuracy in recommender system algorithms has been a primary focus in recommender systems research. Adomavicius and Kwon (2009) state that accuracy has been a central theme in research promoted through competitions such as the Netflix prize. The focus on accuracy is underpinned by a need to foster user trust in the system. This encourages a better online experience and in turn increases sales. However this focus on accuracy is not without its disadvantages.

The issue with this focus on accuracy means that the user may be presented with the same type of product time and time again. If for example, they choose Harry Potter and the Philosophers' Stone, the first book in the series, it is likely that this user would be presented with further books in this series. While this may be very accurate it is likely that the user will already be aware of the subsequent books and not see much value in the recommendations. Sandoval (2012) illustrates this well with regard to music. He makes the point that a user will be presented with additional Beatles albums if they have an earlier purchase of a Beatles album such as Revolver or Abbey Road. This recommendation may be perceived as having a low level of usefulness.

Figure 2. A not so useful recommendation

Figure 4 Display of recommendations with reduced usefulness

Source: Sandoval (2012)

Rana and Jain (2012) state that there are a number of examples of book recommender systems that employ different methods to try to maintain the accuracy of their recommender systems. Whichbook.net allows a user to specify their mood and change this specification as their mood changes. WhatshouldIreadnext.com compares users reading lists where there is commonality. Lazylibrary.com recommends items to users by comparing the content of previously selected items with other available items. A recommendation will be made where there is similarity in the content. The problem with some of these techniques is that the same type of recommendations can be made time and again negatively impacting the user experience. This highlights the importance of introducing diversity within the recommendations made for particular users. The following shows the importance of diversity in Rana and Jain's survey.
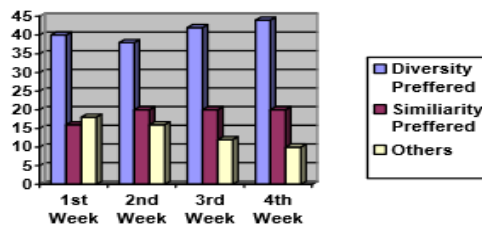
Figure 5 Survey results regarding preference for accuracy or diversity

Source: Rana and Jain (2012)

The level of diversity within the users tastes can be challenging to detect. Wu, Chen and Liang (2013) argue that a person's personality may influence their views on items within a recommenders catalogue. One user may have a limited palate when it comes to a particular websites' products. Another user may have an eclectic taste and appreciate a broader range of recommendations that are not so tightly linked to their previous purchasing behaviour. Introducing a certain amount of diversity into the recommendations may improve the user experience and the opportunity to up sell and cross sell. Accuracy is still important as users will not trust the system if they are receiving recommendations that they feel are not representative of their tastes. The challenge is to create a balance between accuracy and diversity. The Knowledge Discovery in Databases process and data mining are being utilised to address this challenge. The subsequent paragraphs provide an overview of this process and how it relates to recommender systems and the application of diversity.

2.2 Knowledge discovery and Data Mining

The accumulation of large volumes of data is necessitating the development of new techniques to store, manage and utilise this data for the benefit of both customers and corporations. The Economist (2010) provides examples of Walmart who process one million transactions per day and Facebook who retain billions of photos. Organisations across a range of disciplines are looking towards this data as a potential source of competitive advantage. Data is now inherent in key business processes such as decision making and planning. This is also true of online retailers who utilise customer data to make recommendations due to the large volume of products and services available. Schafer, Konstan and Riedl (1999) support this statement with regard to recommender systems when they describe these systems as core business tools for online retailers.

Traditional manual analysis often involved skilled resources with lots of domain knowledge, however this manual analysis is now often impractical. Fayyad, Piatetsky-Shapiro and Smyth (1996) argue that this reduction in relevance as an approach is due to increasing databases sizes, attributes and data volumes. Automation of this analysis to unlock value from data is required.

The increase in online purchases, the time constraint associated with serving customers and the disconnect from the traditional salesperson means that manual analysis is not appropriate for making recommendations. The application of automated analysis as suggested by Fayyad, Piatetsky-Shapiro and Smyth (1996) is appropriate to the business challenge of making accurate but diverse recommendations. The Knowledge Discovery in Databases (KDD) process assists with this need for automatic analysis.

Fayyad, Piatetsky-Shapiro and Smyth (1996) provide an outline of the KDD process. This process has been influenced by disciplines including statistics, machine learning, databases, artificial intelligence and visualisation. It also has applications in a number of domains in addition to e-commerce such as marketing, astronomy, financial services and telecommunications. Applications within these domains include fraud detection, network fault management and data quality assessment. The authors state that the value add, originality and usefulness of the process for knowledge extraction must be clear and the complexity of the problem domain must be sufficient to warrant the use of the KDD process. The value add for making recommendations is an increase in sales overall, increased sales of diverse items, increased customer satisfaction and loyalty while also increasing knowledge of the customer base (Ricci, Rokach and Shapira 2011). Obtaining value from data requires a number of steps which are included in the KDD process and can be summarised as data preparation, pattern identification and evaluation. A graphical representation of the KDD process is available in Figure 6. The KDD process facilitates the extraction of value through the use of diverse recommendations.
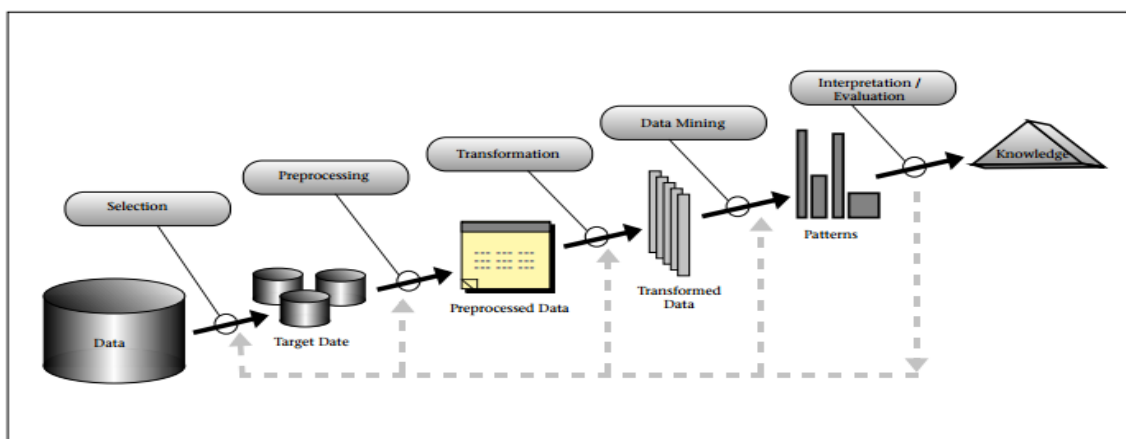
Figure 6 The Knowledge Discovery in Databases process
Source: Fayyad, Piatetsky-Shapiro and Smyth (1996). From Data Mining to Knowledge Discovery in Databases.

Data mining is related to the pattern identification phase of the KDD process. Similarly to mineral mining, data mining involves searching for value when a pinpoint location of this value is unknown. The knowledge value is encompassed in the entire KDD process and as such the data must be pre-processed so the information can be exposed to data mining algorithms. Evaluation follows in a post processing phase so the value can be assessed. The application of data mining to the recommender system requires a decision by the e-commerce retailer with regard to the level of knowledge sufficient for their recommender system. This will be linked to the appropriate level of value that they want to obtain. An e-commerce retailer may want to increase sales but may not be that concerned with increasing diverse sales for example. Another decision applicable to the KDD process with regard to recommender systems is the desired complexity of the data mining algorithm utilised. Ricci, Rokach and Shapira (2011) state that there are different options depending on the level of knowledge an organisation wants to include in the recommender system. Diversity is associated with increased knowledge with regard to understanding customers' likes and behaviours which potentially can increase the effort associated with the pre-processing and data mining steps of the KDD process.

The aim of the data mining step has an initial dichotomous split with regard to objective categorisation. This split is classification and regression. Fayyad, Piatetsky-Shapiro and Smyth (1996) describe classification as a method that assigns a data item to a predefined class. This classification can inform the action designed to address the problem outlined at the start of the KDD process. Regression identifies the relationship between variables for use in prediction. Further categorisation is provided by the authors including clustering which brings groups of items together, descriptive and summary data analysis, methods for identifying dependencies amongst attributes and analysis that assists with change and deviation detection. There are a number of options available for use within recommender systems which will be explored further in the next section.

The KDD process and data mining may have influencing factors that must be considered when undertaking this process. Influencing factors can include privacy and legal issues such as data protection and access.

Other considerations that often need to be addressed before the data mining step is embarked upon are data availability which may be too little or too much data. Data relevance means that the data available must be appropriate for the task at hand. Data quality and frequency and the availability of domain knowledge are also important. Model evaluation, statistical significance, interpretability and deployment are considerations downstream in the KDD process (Fayyad, Piatetsky-Shapiro and Smyth 1996). These influencing factors apply to recommender systems which can suffer from lack of data for new customers who have no previous purchasing behaviour or new or obscure items that have little or no purchase pattern. In addition, missing values may adversely affect data quality while the frequency of data capture is important as customers tastes change over time. Researchers are utilising the KDD process to address the diversity challenge. Section 2.3 provides an overview of the research approaches to diversity.

2.3 *Research approaches on diversity in recommendations*

The research community has taken a number of approaches to address the challenge of introducing diversity in recommendations. These include the use of customer profiles. Adomavicius and Tuzhilin (2005) outline their approach to building customer profiles. Customer profiles can be built using facts about the customer such as their gender and age. Also transactional information such as what they purchased, when and using what method can also be included. The authors also illustrate how these types of customer profiles can be expanded to include indicators of customer behaviour. They provide the example of rule identification using association or classification rules based on the customers previous purchasing behaviour. A rule that identifies that a customer always purchases milk and sugar at the same time on a Tuesday may be noted for example.

The rules defining the customer behaviour are formulated and validated iteratively as new data becomes available. The use of rules for different purchasing occasions facilitates the provision of different recommendation lists at different times for each customer. This in turn increases diversity within the recommendations based on each customers personal behaviour.

According to Rana and Jain (2012) Librarything.com makes an assumption that a user has read all books by an author and as such excludes that author from any recommendations. This attempts to create diversity in the recommendations but they may appear as random to the user and not helpful. Booklamp.com matches books on tone, action and dialogue style to try to introduce diversity. Goodreads.com uses social networking to enhance its recommendations. Recommendations are based on items rated by friends or similar users. Once again there can be too much overlap after a certain amount of time.

Pera and Ng (2011) experimented with a recommender system that uses the social network system Librarything to personalise recommendations. The premise for the experiment was that books rated favourably by a users' connections in their social network then influence the recommendations for the user in question and the books they are interested in. If the user connections have diverse ratings then it could be argued that the user will get more diversity in their recommendations. Though in this work the user has a personal catalogue where they express an interest in a particular book or books. If a member of the users' connections is rating more than one book in a genre then their rating has more weight. This approach may have some drawbacks if applied specifically to the diversity challenge.

Researchers are also investigating the importance of personality in recommender systems. The authors state that a person's personality may influence their views on items available within the recommender systems catalogue. Investigation into personality and its influence may help with the diversity trade off. Wu, Chen and Liang (2013) conducted a survey that showed that personality correlates with levels of diversity depending on the personality type.

Furthermore, the authors performed a comparative analysis between a system where personality influenced diversity and a system where it did not. This showed that the users preferred the recommendations that correlated diversity with their personality type. They also provide the example of the site Whattorent that uses a personality quiz to influence recommendations. In their study the user took a quiz which captured details of their preferences and personality.

Personality attributes are mapped to item attributes in the initially produced list of recommendations and then the level of diversity is adjusted.

Amazon.com uses a number of approaches to introduce diversity into the user experience. Amazon uses an item to item collaborative filtering method which organises a list of purchased, positively and negatively rated items and then each item is multiplied by the inverse frequency of the item to reduce the impact of best selling items (Linden, Smith and York, 2003). In this way some diversity is introduced though it is not influenced by user preferences. Amazon provides two recommendation lists in order to reduce the risk of mistrust of the system. One based on items in the users shopping cart and another through a separate "Your recommendations" menu presented to the user.
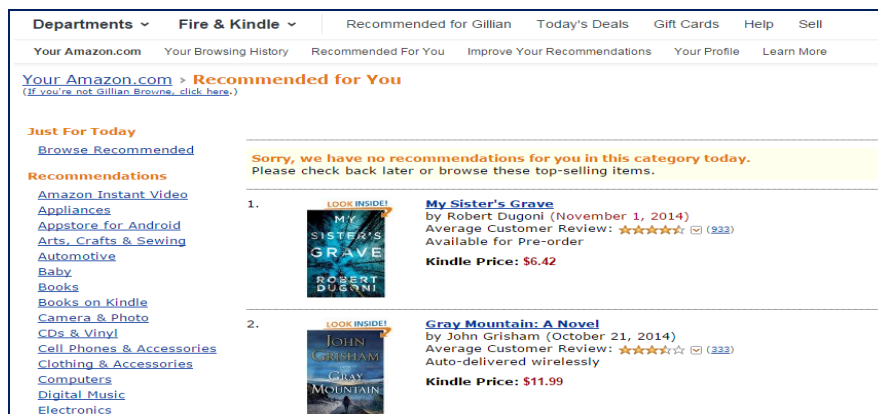
Figure 7 "Recommended for You" section on Amazon.com

Source: Amazon.com (2014)
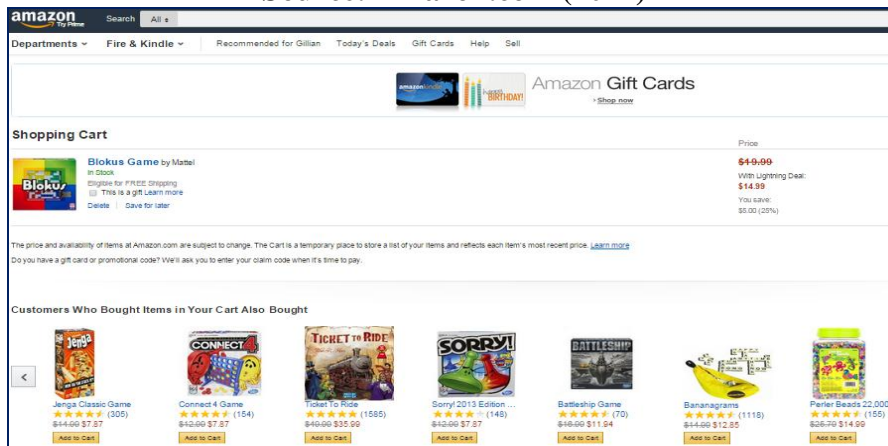
Figure 8 Shopping Cart recommendations on Amazon.com

Source: Amazon.com (2014)

Web information is also under research with a view to improving recommender systems. It is proposed that web information relating to a user may reflect their preferences. The web searches a user performs or user generated content such as blogs, comments, ratings, tagging and tweets can reflect their information needs and preferences. These may be formulated into a user profile which is then used to influence recommendation lists. The user profile may also be enhanced with browsing and click through behaviour. If a user has diverse browsing tagging and search history the potential to provide a more diverse list of recommendations can be provided. Tao *et al.* (2013) elaborate further by suggesting that opinion mining and sentiment analysis can also be used to adjust recommendation lists.

The issue of diversity and accuracy tradeoff also exists in information retrieval. Agrawal *et al.* (2009) mention maximal marginal relevance as a method to control this trade off. Alternatives outlined for diversity in information retrieval are comparison of item features and using explicit feedback. The authors also propose a greedy algorithm for calculating diversity and ranking results using probability. Diversity in web search results is in response to queries that could be interpreted in different ways. It has the potential to be used in recommender systems where user preferences are ambiguous. The experiments conducted by the authors had favourable outcomes when compared to commercial search engines which may make an application for recommender systems worthwhile.

Serendipity though different from diversity may result in a more varied list of recommendations. Ge, Delgado-Battenfeld and Jannach (2010) state that a serendipitous item is one that is previously unknown, surprising and interesting. The authors suggest providing an additional recommendation list that contains serendipitous items to mitigate the risk of user dissatisfaction. This could take a similar form to the Amazon dual recommendation list format previously discussed.

Blunt approaches to increasing diversity are also available (Adomavicius and Kwon, 2009). One approach to increasing diversity is to recommend less popular items though this can adversely affect accuracy significantly. The authors suggest applying the Pareto principle that 20% of the most often rated items within the catalogue are popular items.

The authors suggest a number of more sophisticated approaches such as parameterised ranking approaches. This approach ranks the items by the smallest number of ratings and then recommends them if they are above a rank threshold defined by the user. This allows for a configurable balance between accuracy and diversity. Alternatives also included using the predicted rating value as a measure for ranking, the average rating for an item, ranking how many users liked the item out of the population that rated it and ranking by the percentage of users who liked an item out of the population that rated it.

Bradley and Smyth (2001) also suggest a simple approach of choosing an algorithm that is less susceptible to the diversity problem or using a hybrid to reduce the problem. The authors provide a number of examples such as PTV which uses case based reasoning and collaborative filtering to introduce diversity. Another example provided is CASPER, a job recommender system that uses a combination of collaborative filtering and client side diversity.

In addition to hybrid approaches Bradley and Smith (2001) present three further options for dealing with the diversity problem. The first is the Bounded Random Selection method which randomly chooses items from a set of most similar items. The second is the application of quality metrics that balance similarity of items against diversity of items previously purchased or rated. Alternative versions of this approach are the use of weights or harmonic mean in the quality metric.

Zeigler, McNee and Konstan (2005) use a topic diversification method to reduce the similarity in item to item collaborative filtering. An intra-similarity calculation is used to measure the diversity within the recommended list of items in this study.

The preceding paragraphs outlined some of the research approaches to the introduction of diversity to recommendations. The selection of a research approach will be influenced by challenges associated with recommender systems and diversity and the availability of a range of algorithms which may have varying degrees of suitability to the problem at hand. The following two sections outline some of the factors that require consideration when approaching research in this area.

2.4 Recommender systems challenges

Researchers attempting to address diversity within recommender systems will often use various techniques utilised in the pre-processing and evaluation phases of the KDD process. Utilising the data without addressing complications inherent in the data is likely to result in a system of little use for many data mining applications. Pyle (1999) argues that an automated way to address the complications in data sets for use against a particular domain problem or mining tool is not currently available. It is necessary to make it as easy as possible for the data mining tool to utilise the data and also to eliminate or reduce any problems.

Problem items that often have to be addressed during the pre-processing phase can be many and complicated. These include data quality and transformation issues, sparsity and imbalanced datasets. Some of these problem items will be discussed further in the subsequent paragraphs.

2.4.1 Noise

It is likely that a dataset used in many data mining applications including recommender systems will not fully represent the real world concept to which it is concerned. Noise in the data will be present to varying degrees influenced by items such as data capture and storage and it can be hard to identify. The recommender system can often rely on implicit and explicit data capture. Bell and Koren (2007) state that this may be based on previous purchases or requested directly from the user.

The design of the implicit data capture can influence the usefulness of the data. Similarly the mechanism for explicit data capture from users may influence the completion rate and the quality of this data. This may influence the quality of recommendations and the ability to produce diverse recommendations. Pyle (1999) also states that training the data for too long can cause the algorithm to learn a noise pattern (overfitting). The separation of training and test datasets assists with assessing the level of noise learnt by the data mining algorithm. The training dataset is used to discover relationships in the dataset by the model. The test dataset is used to assess model performance and identify noise. Those implementing data mining algorithms for diversity in recommender systems need to cognisant of the level of noise inherent in the data and methods to mitigate the risk of model overfitting.

2.4.2 Sparsity and the cold start problem

Sparsity and the cold start problem is another data mining challenge related to recommender systems. The cold start problem relates to new users or items where there have been no or too few reviews or purchases to inform a recommendation. Adomavicius and Tuzhilin (2005) state that if you have a new user that has not yet purchased or viewed anything it will be very difficult to make a recommendation using a content based recommender system. This is less of an issue for collaborative filtering methods as long as a user profile is available.

Sparsity refers to the fact that there are usually many more items without a sufficient number of ratings than those that do. Sparsity may also refer to lack of user information. This also affects when a particular user has very unusual tastes and there are not many peers with similar tastes. A number of researchers have made proposals for addressing the cold start problem. Hybrid approaches can be used to address some of the challenges with content and collaborative techniques. In addition, Sarwar *et al.* (2000) state that dimensionality reduction techniques such as Principal Component Analysis and Latent Semantic Indexing can be used to address sparsity in the dataset. Lam *et al.* (2008) suggest the use of sample profiles. Schein *et al.* (2002) propose a two way aspect model to address the cold start problem. The aspect model hypothesises that there is likelihood that a user will like a particular item.
Zhang *et al.* (2010) suggest that tagging can be used to broaden the relations between users and items and can be used as a substitute where there is insufficient information available. The authors argue that social tags are strongly representative of user preferences and as such they can assist with creating balance between accuracy and diversity while addressing the cold start problem.

2.4.3 Missing values

Sparse datasets often have a high proportion of missing values. Missing values can be an issue depending on the problem domain and choice of data mining algorithm. Missing values cause a problem because they can create bias and reduce how representative the model is of the real world scenario it is trying to represent. Acuna and Rodriguez (2004) state that greater than 5% of missing values within a dataset constitutes a requirement for a method to handle these instances.

Acuna and Rodriguez (2004) categorise the approach to handling missing values in data mining as deletion, replacement and imputation. Pyle (1999) states that it is important that any technique utilised to address missing values does not damage the data set further. Collapsing the dataset through aggregation can be a method for addressing sparsity and missing values. The choice of algorithm may be influenced by the volume of missing values within the dataset. Decision trees can be effective for missing values but neural networks can be highly sensitive to this type of data for example. Recommender systems can suffer from sparsity and as such are susceptible to the issue of missing values.

2.4.4 Curse of dimensionality

The selection of pre-processing and technical approach to recommender system implementation with or without diversity can create further challenges that have to be addressed, one of which can be the curse of dimensionality. The curse of dimensionality is used to describe the scenario where there are many attributes available in the dataset which can cause data mining algorithms to fail to generalise well. A high number of dimensions can also mandate a requirement for large volumes of data which may be unobtainable (Pyle 1999). The number of features within a dataset can be increased if collaborate filtering is utilised.

Cayzer and Aickelin (2002) argue that this can make implementing successful recommendations harder and more laborious. Investigation of the relationships between variables is valuable initial analysis during the pre-processing phase. Principal component analysis and factor analysis are two methods for reducing the number of dimensions within a dataset.

2.4.5 Imbalanced datasets

Imbalanced datasets can be an issue when the objective is to predict a class that is naturally under represented within the dataset. Imbalanced datasets cause problems because data mining algorithms expect reasonably equal distributions. He and Garcia (2009) state that imbalance can be intrinsic or extrinsic. This means that the imbalance may be part of the domain, for example fraud or due to some anomaly in an associated data process, for example data capture. The authors further state that the complexity of the dataset coupled with imbalance can make model accuracy degrade further.

Further complications arise if the dataset is broad but has little depth. There can be class imbalance with regard to recommender systems underpinned by collaborative filtering as many users will only be interested in particular items (Zhang and Iyengar 2002).

He and Garcia (2009) provide a number of techniques to approach the imbalance. These include random under sampling which involves removing some of the dominant class. Random oversampling which replicates some of the minority class to balance the distribution. An informed version of under sampling is also outlined which may use ensemble methods and k-nearest neighbour to select which data points to remove. Synthetic sampling methods such as SMOTE and Adaptive Synthetic sampling may also be used to create new examples for the minority class rather than making copies. Less complex solutions involve cost sensitive learning where an assessment of misclassification is performed though the appropriate domain knowledge or cost matrix may not be available. Imbalanced datasets can benefit from additional evaluation metrics such as F-measure and G-mean for improved accuracy evaluation. The accuracy versus diversity challenge may complicated further if the imbalance in the dataset is not addressed.

2.4.6 Scale

Differences in scale amongst attributes can cause issues depending on the type of algorithm utilised. Range and distribution normalisation is often required. Pyle (1999) states most algorithms benefit from normalisation and some such as neural networks require it. The author states that benefits include enhancing linear prediction and reducing the influence of outliers. Normalisation may be a pre-processing requirement depending on the algorithms underpinning the recommender system.

2.4.7 Performance

Performance is also important for recommender systems. These systems have to make a recommendation within a tight timeframe or the user will move on and the opportunity for a sale will be missed. Adomavicius and Tuzhilin (2005) state that this can be addressed by calculating the similarity of all users in advance so that when a user interacts with the website a recommendation can be made quickly.

2.4.8 Accuracy

Sarwar *et al.* (2000) state that accuracy is still an important factor for recommender systems. The authors argue that it is important to avoid false positives as these represent products that have been recommended but the customer has no interest in them. Higher accuracy is likely if the algorithm has more time to make a recommendation, however if it takes too long the customer will have moved on. As such, a balance needs to be maintained between accuracy, performance and diversity.

2.4.9 Trust

The above challenges are mostly of a technical nature however there are others to which recommenders systems are susceptible. Trust is a key factor when embedding recommendations within the sales process. Resnick and Varian (1997) state that it is important for the recommendations to be unbiased and protect against users rating their own items highly and often. In addition the organisation must not let the cost model influence the recommendations at the detriment of levels of user trust within the system. O'Donovan and Smyth (2005) further elaborate that user ratings may not be reliable even though that user is similar to the target user. They mention that a user must have trust in the system overall and trust in the ratings.

They recommend the introduction of a trust measure weighted with similarity using the harmonic mean to address this issue.

2.4.10 Privacy

Another factor to be considered is privacy. Ramakrisknan, Keller and Mirza (2001) argue that this is more of an issue for users with diverse tastes as they may be identifiable from a recommendation. The privacy of the individual has to be protected as the user information could be combined with other data sources and abused or leaked. The authors state that this can be performed by setting a minimum number of users before a recommendation can be produced.

Jeckmans *et al.* (2013) state that legislation is a driver for increasing security within recommender systems. Data Protection and Article 29 Workers Party are influencing recommender system implementation.

Furthermore, initiatives such as the Platform for Privacy Preferences infer a move to standardise formats and make privacy policies more transparent. Cryptography can be used to enhance security. Randomising, aggregation and addition of noise to the data can help maintain user anonymity. Techniques to protect privacy can influence the accuracy of recommendations so once again there is a balance to be maintained.

2.5 Recommender system algorithms

There are various techniques employed in recommender systems. This section provides an overview of some of these algorithms including advantages and disadvantages. Rana and Jain (2012) state that Resnick and Varion are key authors regarding recommender systems and they are attributed with the idea of collaborative filtering. However, there are a number of methods available. These include collaborative filtering, content filtering, demographic knowledge based filtering, classification and regression. Adomavicius and Tuzhilin (2005) elaborate further by outlining additional options. Probability can be used also to identify the likelihood that a user will like a particular product. According to the authors research on ratings based recommender systems began in the 1990's. Recommender systems can also attempt to predict a user rating for a particular item and as such recommend the items or items with the highest predicted rating.

Schafer, Konstan and Riedl (1999) categorise recommender systems as either automatic or manual. The automatic recommender collects data to support recommendations implicitly through the customers behaviour when they interact with the website. Manual recommender systems are those that ask customers to specify their preferences. Recommendations can be based on the most popular items which means that all customers get the same recommendations at a particular point in time which can heighten the diversity issue.

2.5.1 Collaborative filtering

Collaborative filtering compares one user to people who have a similar user profile and then recommends items that these similar peers have rated or purchased.

Measures of similarity can include Pearson coefficient, cosine similarity and Euclidian distance. Sarwar *et al.* (2000) provide further information regarding collaborative filtering by portioning the effort into three steps, getting the data into a suitable format, finding users that are similar to the target client and making recommendations. Once the dataset is in a suitable format, often because it has been reduced in size, neighbourhoods of similar users are created. These neighbourhoods can be aggregate or centre based. Recommendations can then be made based on the most frequently occurring item within the neighbourhood where the current user resides or using association rules for the products occurring in the chosen neighbourhood.
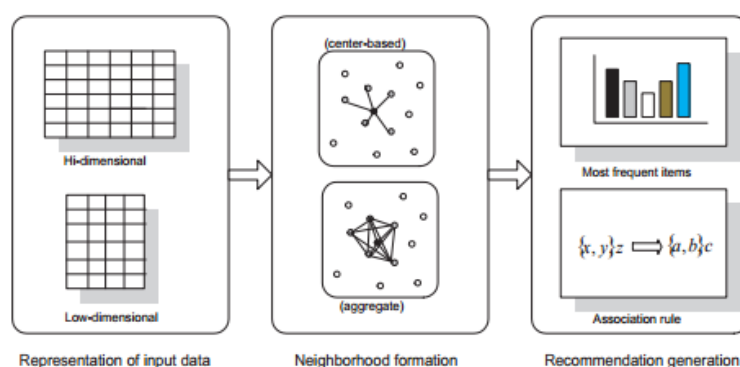
Figure 1: Three main parts of a Recommender System.

Figure 9 Graphical representation of steps within a recommender system
Source: Sarwar *et al.* (2000)

K nearest neighbours can be used to identify the neighbourhood to which a datapoint or user belongs and then assigns the class of this neighbourhood to the datapoint. The algorithm utilises a distance or similarity measure such as Euclidean distance or cosine similarity as aforementioned to identify the closest number of neighbours. K represents the number of neighbours to be utilised.

K nearest neighbour has some challenges. Wu *et al*. (2008) states that it can be difficult to select the appropriate number of neighbouring datapoints. The approach to combining class labels of the neighbours where they differ can influence the accuracy. Closer or more similar neighbours may be more accurate. Scaling of attributes is important for k-nearest neighbour to prevent a particular attribute dominating the selection of neighbours.

A method to address this is the weighting of the neighbour by its distance to a particular datapoint. It can be computationally expensive for large datasets. However this algorithm is easy to understand and implement despite these drawbacks.

Sarwar *et al.* (2000) conducted a number of experiments to evaluate recommender systems using the MovieLens dataset. Users with less than 20 ratings were excluded from the dataset. The dataset was then transformed into a binary user matrix that was split into training and test sets. Cosine similarity was used and recommendations limited to 10. Experiments were performed to identify the optimal size of the neighbourhood, the best number of dimensions to use in the model, to compare item based recommendation to association rule recommendation and measure the impact of different amounts of training data. The authors found that there was little difference in the results between item based and association rule analysis and that the algorithms made better recommendations when more training data was made available. The results also suggested that centre based neighbourhood formation was most appropriate for this dataset. The optimal dimensions is influenced by the dimensionality reduction technique performed.

2.5.2 Clustering

Clustering is useful for segmentation and understanding patterns within similar groups of customers. This can also be applied to recommender systems. Adomavicius and Tuzhilin (2005) outline that users may be grouped into a cluster with a defined class and recommendations associated with this class made to the user. K-means is a popular algorithm for clustering. Wu *et al.* (2008) attributes the discovery of the k-means to a number of people including Lloyd (1957,1982), Forgey (1965), Friedman and Rubin (1967) and McQueen (1967). K represents the set of clusters specified by the user. The algorithm works by selecting initial seed data points known as centroids through random sampling or exploration of a subset for example. Each data point is assigned to a cluster using the minimum sum of squared errors and the centroid reallocated to the mean of the cluster based on the shortest squared distance. This reassignment and centroid selection is performed iteratively until no further reassignment occurs. Euclidean distance is often used to assign a data point to its closet centroid. KL-divergence may also be used as an alternative.

K-means is popular for clustering because it is easy to understand and scalable accommodating both streaming and large datasets. It is efficient at processing large volumes of data in reasonable run times Chaturvedi *et al*. (1997). Huang (1998) challenges that k-means is either good at handling large datasets as long as the attributes are numeric or it can handle different types of data as long as the dataset is small. Non numeric data may need to be transformed to allow its use which can extend processing timelines and make the process more opaque. Huang (1998) states that alternatives such as k-modes and k-prototypes extend the algorithm for use with categorical data. However Chen, Ching and Lin (2004) state that while the k-means algorithm completes multiple runs over the data set it still outperforms other algorithms with regard to processing times.

Issues with k-means can be the initial selection of the centroids which can influence the quality of the cluster separation. It can also be sensitive to local minimum. Wu *et al.* (2008) offers methods for addressing these issues include running the algorithm a number of times utilising different centroids to identify the best outcome or using a hybrid algorithm of k-means and hierarchical clustering.

K-means is also sensitive to outliers as it uses the mean for centroid selection. Chaturvedi *et al.* (1997) state that outliers can mask valuable relationships and lead to misinterpretation. This can be addressed by using the median which is less sensitive to outliers, removing outliers before using the algorithm and merging or removing small clusters. This sensitivity to outliers may adversely impact diversity as outliers may represent the user with diverse tastes.

K-means has an inherent disadvantage as the project owner determines how many clusters should be produced rather than the system identifying the optimal count of clusters. Variation in the results can be caused by the initial selection of the centroids. The k-means algorithm tries to identify a local optimal centroid for the cluster that is appropriate or reflective of the overall dataset. Different results are produced if different initial data points are selected as the centroid. ISODATA algorithms can also allow the user to search for the appropriate number of clusters based on cost but selecting this is a challenge.

2.5.3 Content filtering

Adomavicius and Tuzhilin (2005) state that recommendations can be created in different ways including content based filtering. Content based recommendations are made by calculating the similarity between items and recommending those that are most similar to an item previously chosen by a user. The authors state that content based filtering has a close relationship with information retrieval and is often used for document recommendations. Attributes associated with an item are used to determine similarity. The accuracy of the recommendations produced can be limited by the features associated with the items. In addition, items may appear identical to the algorithm if they have the same attribute values. This algorithm is more susceptible to the cold start problem than collaborative filtering as mentioned previously. Rana and Jain (2012) argue that content filtering is not as popular as collaborative filtering however they conduct an experiment to include time in a content filtering recommender system to provide diverse recommendations that are updated on a regular basis demonstrating some success.

2.5.4 Association rules

Association rules analysis is one of the more popular techniques for recommender systems. Sarwar *et al.* (2000) state that they can encompass decision trees, apriori algorithms and tree projection algorithms for example. Association rules are often used in market basket analysis but can be used in science and medical fields. Association rule analysis provides an alternative to correlation analysis. Support and confidence are key metrics utilised by the association rule algorithm. The support count is the number of transactions that contain a particular itemset. The confidence states how often the items in the rule appear together. Support and confidence allow for the identification of significant relationships. Confidence is generally calculated on itemsets that meet a predefined support threshold to avoid unnecessary processing. The apriori algorithm is a commonly used association rule algorithm. It allows the system to discard many itemsets without having to calculate the support first. The Apriori algorithm creates buckets of candidate itemsets and stores them in a hash tree. This increases efficiency as a transaction is only compared to the candidate itemset in the same bucket Tan, Steinbach, and Kumar (2006).

Wu *et al*. (2008) offers an alternative to the Apriori algorithm. The FP growth (frequency pattern growth) algorithm maps each transaction to an FP tree. Initially it contains a single node. Next the support count is calculated for itemsets. Those that are infrequent are discarded and the remaining are sorted in descending order by support count. The tree is created and duplicate paths are merged until no further merging is possible. Frequent itemsets are then identified. The compression of the tree aids efficiency. The identification of related itemsets through the use of these algorithms allows for recommendations where items in an itemset that are not yet purchased can be recommended providing support and confidence thresholds are met. Thresholds can be adjusted to increase levels of diversity in recommendations for particular users. Davidson *et al.* (2010) performed this type of personalisation using user behaviour metrics for a YouTube video recommender system.

Association rule analysis as with many algorithms has some drawbacks. While discarding the subset rule based on the infrequency of the parent rule has significant benefits, the apriori algorithm can take a long time to run for large datasets as it performs multiple database scans. However, Wu *et al.* (2008) states that the apriori algorithm has been enhanced through new techniques for candidate itemset selection such as partitioning, subsampling hash functions and vertical data formats. A trade off between accuracy and efficiency is required as sampling may not be representative but a lower support value can be used. Other enhancements include the use of taxonomies, information gain, clustering and incremental mining. Tan, Steinbach and Kumar. (2006) states that it can be difficult to identify the appropriate support threshold though multiple support thresholds can be used across itemsets. Association rules may need to be validated by domain experts or using other means such as correlation analysis.

2.5.5 Classification

There are a number of algorithms available if classification has been chosen for implementing a recommender system. Classifiers define data items as being a member of a particular class based on their descriptive variables Adomavicius and Tuzhilin (2005) state that predicting the rating for unrated items is used to address the fact that there tends to be so many items that the dataset is sparse.

The users profile can be asked when the user registers on the website or can be formed through their browsing activity. Classification techniques that may be used are decision trees and clustering to build a model that will predict a users rating rather than using measures of similarity. The authors argue that Naive Bayesian classifiers have a high predictive accuracy. Further detail relating to these algorithms is provided below.

2.5.5.1 Decision trees

Decision trees offer a type of classifier. There are a number of decision tree algorithms available. Pazzani and Billsus (2007) state that decision trees can be beneficial for content based recommender systems because are easy to understand and perform well providing the dataset is not unstructured. C4.5 is an example of a decision tree algorithm that uses a divide and conquer approach. The decision tree starts with a root node and partitions the dataset into two or more subsets using a single attribute at a time. C4.5 uses information gain and gain ratio to decide on the partitioning.

The decision tree continues to partition the subsets until it reaches some stopping criteria or no further leaf nodes can be generated. A second method that prunes the tree is performed to avoid overfitting and improve comprehensibility. The pessimistic error estimate is used to prune the tree.

An alternative to the C4.5 tree is C4.5 rule sets. These rule sets are developed from the unpruned tree after which rules are dropped using the lowest pessimistic error rate identified. A set of rules is selected for each class, classes are ordered and a default class chosen.
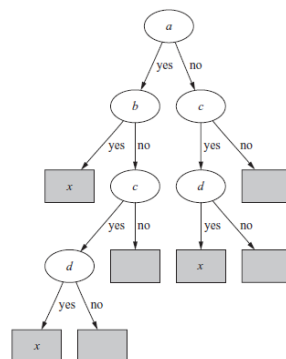
Figure 10 Simple decision tree example

Source: Witten,Frank and Hall. (2011). Data Mining: Practical machine learning tools and techniques 3rd edition.

An alternative decision tree algorithm is CART (Classification and Regression Trees) which uses gini index for partitioning and a cost complexity model for pruning. It only allows dichotomous partitioning. It can be used to create a number of trees with the optimal tree selected following completion of the pruning phase. An advantage of the CART algorithm is its ability to handle missing values with are likely to be a feature of sparse datasets associated with recommendation systems.

There are a number of issues with decision trees. Wu *et al.* (2008) elaborates that decision trees can be heavily influenced by the training set which can mean the error rate is higher on new cases. In addition, a different rule set outcome may be produced when a different training set is used.

Ensemble methods may be used to boost accuracy. AdaBoost is a common form of ensemble method that uses multiple learners to obtain better accuracy. Wu *et al.* (2008) states that the algorithm first assigns equal weights to all training examples and creates an initial simple learner. The results of the initial learner are tested and misclassified examples are weighted at a higher level resulting in distribution of weights. This creation of learners is performed iteratively. AdaBoost has also been adapted for regression also. The benefit of this algorithm is its reduced susceptibility to overfitting. Ensemble methods can be harder to interpret which negates a key benefit of decision trees. The C4.5 algorithm specifically can be computationally intensive for rule set generation. However, the next generation of C4.5 (C5.0) which became available in 1997 improved scalability, accuracy and interpretability.

2.5.5.2 Naive Bayes

Naive Bayes is often used as a classification technique as it is simple, quick and does not require multiple iterations while providing robust results. Wu *et al.* (2008) detail how Naive Bayes uses probability to assign previously unseen data points to a particular class or classes. The class with the highest probability is assigned to the instance. Adomavicius and Tuzhilin (2005) outline how Naive Bayes can be used in recommender systems through determining the probability that an item will be viewed positively.

This algorithm can easily be applied to large datasets and is easy to interpret. Naive Bayes assumes that variables are independent when it calculates the probability that a data point belongs to a particular class. Witten, Frank and Hall (2011) state that Naive Bayes has the additional advantages of not requiring large volumes of training data while not being sensitive to the curse of dimensionality or missing values. Another advantage of Naive Bayes is that it can be easily understood. It can perform better than more complex algorithms for reduced effort. However, this algorithm does not perform well if there are attributes that are related or contain a lot of the same information.

Naive Bayes often performs better with data that has a normal distribution. The application of standard estimation procedures for non normal distributions can be utilised to enhance performance. A Multimodal Bayes classifier may be more appropriate if there is skewness in the variable distribution. However binary data is required for this type of classifier. Witten, Frank and Hall (2011) continue that discretization of the data can be an appropriate pre-processing step in Naive Bayes however it can discard much of the data. Numeric input variables are usually assumed to be normally distributed. If there are missing values they are ignored. Naive Bayes is often used for document classification.

2.5.6 Regression

Regression can also be used for recommender systems if the target variable is numeric or binary such as a rating. A sample of regression algorithms and their use in recommendation systems is provided in this section.

2.5.6.1 Linear Regression

Linear regression is a statistical method appropriate for use when the target variable and input variables are numeric or binary. This method attempts to model the relationship between the dependent target variable and input variables that may be predictors. Vozalis and Margaritis (2003) state that linear regression can be used in recommender systems whereby the users previous ratings and unknown ratings are the dependent and independent variables. Witten, Frank and Hall (2011) state that linear regression is an example of a simple method that can often work well and it has been used as the basis for more complex methods such as neural networks.

Linear regression has a number of advantages in that it is easy to understand and explain and it is less likely to be computationally expensive. The disadvantage is that it may not be appropriate for use with non linear data and assumes that the data is normal. It is appropriate to look for a fanning affect in the variables. Logistic regression can be used where the target variable is not numeric.

2.5.6.2 Neural Networks

Neural networks were often heralded as the technique for classification of continuous data or large complex datasets. Adomavicius and Tuzhilin (2005) argue that this type of model can provide more accurate recommendations than memory based approaches. The objective of neural network is to predict the rating for a particular item for a user. Neural networks are similar to many other algorithms in that they have advantages and disadvantages. Zahedi (1991) provides an outline of neural networks. Neural networks were designed to copy human intelligence through the application of deduction. Their benefits are that they can handle incomplete patterns or patterns that are highly complex. They also do not need to know a target variable. The structure of a neural network consists of layers of nodes with connections and associated weights. The weight of a node is determined by its connection to other nodes. Feed forward neural networks, back propagation neural networks, kohonen self organising maps are all types of neural networks. The disadvantages are that they often need lots of training data and are sensitive to outliers and the curse of dimensionality. Categorical data must be transformed to numeric data in order to be utilised. This makes them less suitable for recommender systems unless dimensionality reduction techniques are used.

2.5.6.3 Support Vector Machines

Support Vector Machine (SVM) algorithms are regression based and offer an alternative when dealing with broad datasets. Pazzani and Billsus (2007) state that this approach is useful when recommendations are made under tight time constraints or need to utilise fast changing data. Wu *et al.* (1998) provide an overview of workings of SVM. SVM works by creating separation in the dataset using a separating hyperplane. These subsets are assigned classes. The best separation is selected by maximising the margin between the subsets represented as space in the hyperplane.

This use of margin means that SVM generalises well for previously unseen data reducing the reliance on the training dataset. Support Vector Regression (SVR) is used to perform numerical predictions. The accuracy is assessed when the difference between the actual and predicted value is within a very small positive amount. SVR is not sensitive to outliers but can be computationally intensive. SVM have a number of benefits. SVM are beneficial because they do not require lots of data for training and are not sensitive to the curse of dimensionality. SVM are applicable to continuous output variable and are not as complex as neural networks.

Research continues to identify new approaches or enhance existing approaches to recommendation formation within recommender systems. Pera and Ng (2011) expand on the use of correlation in LibraryThing book recommender systems illustrating how books are compared for similarity based on tag clouds. The authors used users' friends lists to make recommendations based on the theory that a user shares common interests with their friends. If a friend has rated more than one book in a genre, then that rating carries more weight. Similarly to the use of ensemble methods to improve accuracy, hybrid approaches are also under investigation to improve recommendation quality while addressing the need for diversity amongst other influencing factors (Adomavicius and Tuzhilin 2005).

2.6 Recommender system model evaluation

As aforementioned, accuracy is important for recommender systems. Adomavicius and Tuzhilin (2005) suggest that accuracy can be measured using the mean square error, mean absolute error, root mean square error and the correlation between prediction and ratings. Alternatives included precision and recall. Recommender systems performance measures can be put into two categories, coverage and accuracy. Coverage means how many users can they actually calculate recommendations for and accuracy compares the estimated versus the actual ratings.

Confidence and support are the measures of accuracy if association rule analysis is used in a recommender system according to Sarwar *et al.* (2000). Support measures how often items are purchased together and confidence measures the strength of the relationship between two items.

Herlocker *et al.* (2004) suggests a number of metrics to evaluate the accuracy of predicted ratings for recommender systems. These include precision, recall, mean squared error and ROC. An assessment of probability can also be used to measure the likelihood that a user will view an item favourably.

2.7 Discussion

This chapter outlined the accuracy versus diversity challenge while providing an overview of research used to address this challenge. The applicability of the KDD process, the factors that need to be considered and the range of solutions available was also presented. The literature review highlighted the complexity of creating balance between accuracy and diversity and the broad approaches utilised to address this challenge. Personality and taste are very hard to quantify and this adds magnitude to the challenge. However, the benefits of attempting to address this challenge are also understood from the literature review. Human behaviour can often be unpredictable and as such this makes for an interesting area of research. The number of different approaches and research available indicates the focus on this challenge.

The literature review has helped shape the research question as most of the research was measuring diversity post recommendation. This prompted the idea of using diversity as an input measure to explore if rating accuracy could be improved. Also the literature review provided measures of diversity that could be utilised in the design and experiment sections of this paper. This literature review has provided insight into the process and approaches to addressing the research question: Does diversity improve the accuracy of predicted ratings in recommender systems?

Insight into the limitations of the various techniques and the pre-processing that can be undertaken in an attempt to improve the results was also provided through the completion of the literature review. Exclusion of implicit ratings within the selection dataset has been informed by the literature review as the data capture or storage has reduced the usefulness of the field. This decision mitigates the ratings imbalance in the chosen dataset.

The literature review has also informed the usefulness of utilising of training and test datasets and influenced the choice of model selection based on the format of the target variable, the resources available and the advantages of each of the models. Models considered computationally intensive will be avoided. As such a selection of both regression and classification models including linear regression, decision trees and naive bayes have been selected for the design and experiment sections of this paper.

3   DESIGN AND EXPERIMENTS

This chapter presents the design of the experiment that will be used to predict ratings using models that include measures of diversity in their input metrics. Design details relating to data exploration, pre-processing and preparation will be included. An overview of the in scope attributes and software selections will be provided in this chapter. Lastly model choices and evaluation criteria will be outlined. The subsequent chapter will provide details of the implementation of this experiment design. Quantitative research methods will be used during the execution of this work supported by learnings gathering during the literature review. The graphic below provides an outline of this chapter.
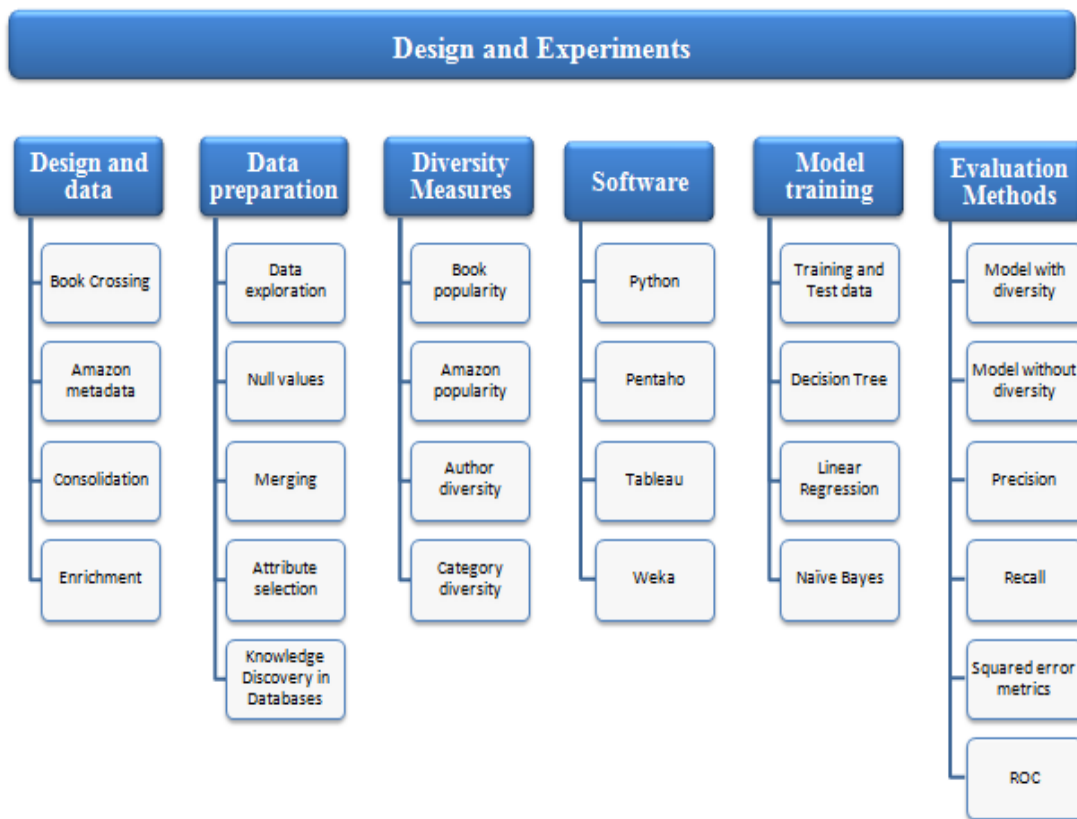
Figure 11 Outline of the Design and Experiments chapter

3.1 Design and data

The research question is concerned with evaluating if using measures of diversity as inputs to predictive models improves the accuracy of the predicted ratings for a recommender system. The experiment aims to assess if diversity has a favourable, adverse or neutral affect on the accuracy of predicted ratings. The hypothesis (H_1) associated with this research question is that the inclusion of diversity measures improves accuracy of predicted ratings when compared to models without diversity measures included. This chapter presents design details of an experiment undertaken to test this hypothesis.

The data that will be used in the experiment is a free dataset titled Book Crossing dataset[3]. This dataset was mined by Cai-Nicholas Ziegler (2005) in summer 2004. The dataset consists of three csv files. The first file BX-users contains details of 278,858 users of the recommendation system including their anonymised User Id, location and age.

The second file BX-books contains 271,379 records of books with the attributes Book-Title, Book-Author, Year-of-Publication, Publisher and URL details. The final Book Crossing file titled BX-Book-Ratings contains the ratings provided by the users. This file contains 1,149,780 ratings and contains the User ID, the ISBN of the book and the book rating.

An amazon metadata file[4] will also be used to add additional attributes to the dataset. This file was sourced from Stanford University's SNAP website and contains metadata for amazon books, music, CD's, DVD's and video tapes. The file also contains details of the Amazon salesrank, ids of similar items, categorisation, ratings and votes.

The datasets will be merged, explored, pre-processed and cleansed in preparation for use by classification and regression models. This involves merging the above data files to provide a single consolidated dataset with greater breadth.

---

[3] Sourced from http://www.informatik.uni-freiburg.de/~cziegler/BX/

[4] Sourced from http://snap.stanford.edu/

Figure 12 illustrates the consolidation of the datasets. Data anomalies specific to this dataset such as missing values will be handled during data pre-processing phase.
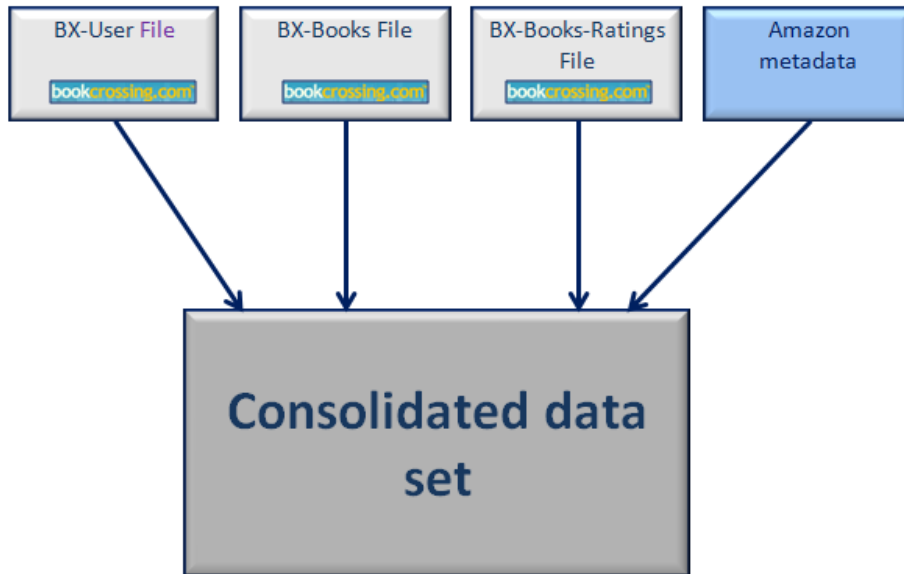
Figure 12 Consolidation of the datasets.

In addition, a complement of diversity measures will be added to the above configuration to further broaden the dataset. Figure 13 provides a graphical illustration of the enhanced dataset.
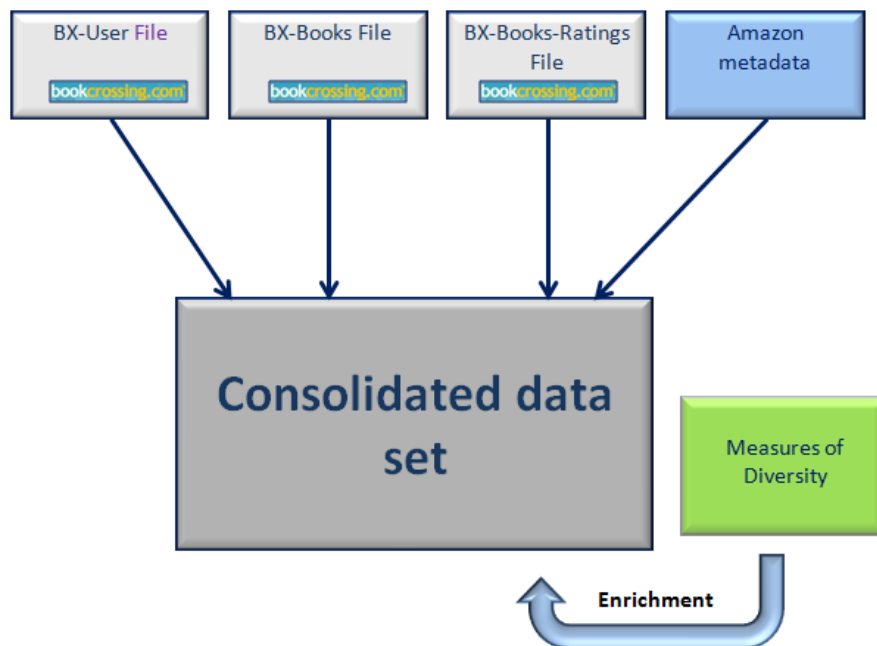
Figure 13 Enrichment of the consolidated dataset.

Classification and regression models will be utilised following preparation of the dataset. These models, using diversity metrics as inputs will be executed to predict user ratings on items within a test dataset. If diversity helps increase the accuracy of predicted ratings, it could be argued that this approach could be of value to organisations using recommender systems as increased accuracy builds trust in their systems and can increase sales and customer satisfaction.

3.2 Data Preparation

There are a number of considerations that have to be made due to variations in the datasets. The BX-users file has both null and zero values for age. The BX-books file does not appear to have nulls in the attributes identified for use in the experiment. However, there are ISBN records that have unusual formats. A new ISBN is provided for a new edition of a book. This may be viewed as the same item from a user and diversity perspective however there is no link between the previous and subsequent ISBN numbers.

The ratings file has a number of ratings of 0 that represent implicit ratings but they are not useful for this experiment and will be removed. There are no null values within the User-ID or ISBN on this file. The dataset will be joined using the User-ID and ISBN to form a consolidated dataset. The Amazon unique identifier (ASIN) for books is the ISBN which facilitates the joining of the amazon metadata file to the Book Crossing files.

There are a number of attributes that will be utilised within the predictive models. These will be extracted from the source files and utilised to prepare a final dataset for introduction to the data mining software. Table 1 shows the details of the in scope attributes sourced from the input datasets.

The data exploration phases will be conducted to provide an assessment of data quality and inform any data preparation decisions made during the data pre-processing phase. Data exploration will include variable metrics such as minimum, maximum, mean, standard deviation and null values. Further assessment will be performed to identify the presence of constants, outliers, duplicates and data inconsistencies.

Missing values are present in the dataset. These need to be addressed while minimising the risk of introducing bias to the dataset. Techniques to address missing values include removing these records or replacement. Age, for example has a number of missing values. It is not possible to provide a reasonably reliable estimator for the age of the user so removal may be considered. This decision will be supported by analysis performed in the data exploration phase. Different approaches may be required depending on the algorithm utilised.

Id	File ID	Column Name	Derived YN	Data Type	Description
1	1,3	User_ID	N	Integer	Unique id that represents each user. Used as a key to create consolidated file.
2	1	Location_Line_1	N	String	The first line of the Location field parsed.
3	1	Location_Line_2	N	String	The second line of the Location field parsed.
4	1	Country	N	String	The country parsed from the Location field.
5	1	Age	N	Integer	Age of the user
6	2,3,4	ISBN	N	String	Unique identifier for each book used to join on ratings file and Amazon metadata.
7	2	Book_Author	N	String	Book-Author field
8	2	Year_Of_Publication	N	Date	Year-Of-Publication
9	2	Publisher	N	String	Publisher
10	3	Book_Rating	N	Integer	Book-Rating
11	4	Group	N	String	Used for filtering and validation
12	4	Categories	N	Integer	Sourced from Category
13	4	Subcategory	N	String	Sourced from Category Detail
14	4	Salesrank	N	Integer	Salesrank field.
15	4	Average_Rating	N	Integer	Sourced from review Detail field

File ID Legend - 1 = BX-Users, 2 = BX-Book-Ratings, 3 BX-Books, 4 Amazon-meta.txt

Table 1 In scope fields from source files

3.3 Diversity Measures

The use of diversity as an input to the models is an integral portion of this experiment. This requires the creation of measures of diversity. A number have been selected for this experiment. Adomavicius and Kwon (2009) provide ranking calculations that can be used as a measure of diversity. These are pareto popularity, average popularity and relative average popularity. The authors use these metrics as ranking criteria but for this experiment it is assumed that popular items included in user's ratings represent reduced diversity. As such pareto popularity will be included through two fields Book-Popularity_Category and Amazon_Popularity_Category. A metric denoting author diversity will be derived. It is assumed that a list of recommendations for books all of the same author will also have reduced diversity.

The Amazon metadata file provides a source of derived attributes also. A number of categories are available through the Category_Detail field. The level of diversity across each category will be created for each user. The top 20% of books ranked by the Amazon salesrank will be deemed popular and therefore less diverse if strongly represented within user's ratings. In addition, the Amazon average rating will be calculated. Descriptive statistics at the user and book level will also be derived.

Trust is a factor that should be considered during this experiment. It is assumed that users differ in their rating behaviour and different levels of confidence in their ratings exists. The user ratings frequency will be calculated as a measure of this trust. In addition the frequency of the ratings by rating number will also be calculated. If a user is rating all their items with the same value then this could be an indicator of spurious rating behaviour.

In summary, the measures of diversity that will be added to the dataset are as follows:
- Book_Popularity_Category
- Amazon_Popularity_Category
- Author_diversity
- Category3 diversity
- Category4 diversity
- Category5 diversity

- Category6 diversity
- Category7 diversity
- Category8 diversity

Table 2 shows the full set of derived attributes and metrics.

Id	File Name	Column Name	Derived YN	Data Type	Description
1	1	Location_Line_1	Y	String	Parsed from Location
2	1	Location_Line_2	Y	String	Parsed from Location
3	1	Country	Y	String	Country description derived from Location
4	1	User_Ratings_Count	Y	Integer	Count of ratings per user
5	1	User_Average_Rating	Y	Integer	Average rating per user
6	1	User_Min_Rating	Y	Integer	Maximum rating value per user
7	1	User_Max_Rating	Y	Integer	Minimum rating value per user
8	1	User_Rating_Std_Dev	Y	Number	Standard deviation of user rating
9	1	User_Distinct_Ratings	Y	Integer	Count of distinct ratings per user
10	3	User_Author_Count	Y	Integer	Count of distinct authors per user
11	2	Book_Ratings_Count	Y	Integer	Count of ratings per book
12	2	Book_Max_Rating	Y	Integer	Maximum rating per book
13	2	Book_Min_Rating	Y	Integer	Minimum rating value per book
14	2	Book_Average_Rating	Y	Integer	Average rating value per book
15	2	Book_Std_Deviation	Y	Number	Standard deviation of book rating
16	2	Book_Distinct_Ratings	Y	Integer	Count of distinct ratings per book
17	2	Book_Popularity_Category	Y	String	Popular or Less Popular selected if the count of ratings per book is within top 20%

18	4	Maximum_Amazon_Rating	Y	Integer	Maximum amazon rating received for each ISBN
19	4	Minimum_Amazon_Rating	Y	Integer	Minimum amazon rating received for each ISBN
20	4	Distinct_Amazon_Ratings	Y	Integer	Count of distinct amazon rating received for each ISBN
21	4	Amazon_Popularity_Category	Y	String	Popular or Less Popular selected if the salesrank per book is within top 20%
22	1	Author diversity	Y	Integer	Count of distinct authors per user
23	1,2,3,4	Category3 diversity	Y	Integer	Count of distinct category 3 instances per user
24	1,2,3,4	Category4 diversity	Y	Integer	Count of distinct category 4 instances per user
25	1,2,3,4	Category5 diversity	Y	Integer	Count of distinct category 5 instances per user
26	1,2,3,4	Category6 diversity	Y	Integer	Count of distinct category 6 instances per user
27	1,2,3,4	Category7 diversity	Y	Integer	Count of distinct category 7 instances per user
28	1,2,3,4	Category8 diversity	Y	Integer	Count of distinct category 8 instances per user

File ID Legend - 1 = BX-Users, 2 = BX-Book-Ratings, 3 BX-Books, 4 Amazon-meta.txt

Table 2 Derived fields

The Book-ID (ISBN) will be used as the identifier of each book as it is unique and facilitates the joining of the files. Data will be aggregated to user level where appropriate. Categorical variables will be numerated for use in appropriate models. In addition, range and distribution normalisation will be performed as required.

3.4 Software

A number of software selections have been made for this project. Python will be used to parse the input files and create text file inputs for use in Pentaho. Python has been chosen due to its flexibility and open source nature. Python has the advantage that it is extremely fast at processing large files. There are online forums such as stack overflow and python tutorials to assist with learning and trouble shooting.

Disadvantages include the fact that the Python syntax has a steep learning curve. In addition it is not very verbose when an error is encountered which can elongate time spent on trouble shooting.

Pentaho will be used to upload, cleanse and consolidate the datasets. The source data will be introduced in text file format following parsing in Python as required. Each dataset will be introduced to Pentaho as an individual file. Calculations and consolidation into a single dataset will be performed through various Pentaho graphs to allow for iteration, testing and ease of refinement should a calculation need adjustment. Separate graphs will be used as required for performance reasons. Pentaho has been selected as it's node based interface increases usability and the in-built data profiler allows for rapid data exploration. In addition, this software was selected following initial assessment of MySQL which was discounted due to sensitivity to special characters. Pentaho's graphical user interface which condenses implementation time and change control. There is an online forum with technical information also. Tableau will be used to produce visualisations and perform additional data exploration. This software has the advantage of being easy and quick to use.

Weka has been chosen to execute the predictive models due to it's graphical user interface, open source nature and wide range of tutorials and training material available online. A disadvantage of Weka is that it is memory bound.

3.5 Model training

A set of classification and regression models will be trained using the training dataset. The Weka algorithms identified for use are linear regression, DecisionStump, J48 and Naive Bayes. DecisionStump is a form of decision tree suitable for continuous outcomes. J48 is a decision tree algorithm based on C4.5 that is suitable for nominal output variables. A portion of the overall dataset will be used for training and the remainder used for testing. Each model will be evaluated against a test dataset. A number of runs will be conducted through the use of 10 fold cross validation. The list below shows the split of data between training and test datasets. The models will individually decide which data items are most important as part of the training phase.

Training and Test Data set split %	Count of records
90/10	196,878/21,876
80/20	175,003/43,751
70/30	153,127/65,627
60/40	131,252/87,502

A number of models have been selected for this experiment. Table 3 below provides details of these models and their associated inductive bias.

Model	Underlying approach	Inductive bias
Decision Tree	Information based	Shorter trees are preferred over longer trees.
Linear Regression	Least squares	The relationship between the attributes x and the output y is linear.
Naive Bayes	Probability	Assumes variable independence

Table 3 Selected algorithms

3.6 Evaluation Methods

The chosen models will be evaluated against a baseline model that will not include the diversity metrics.
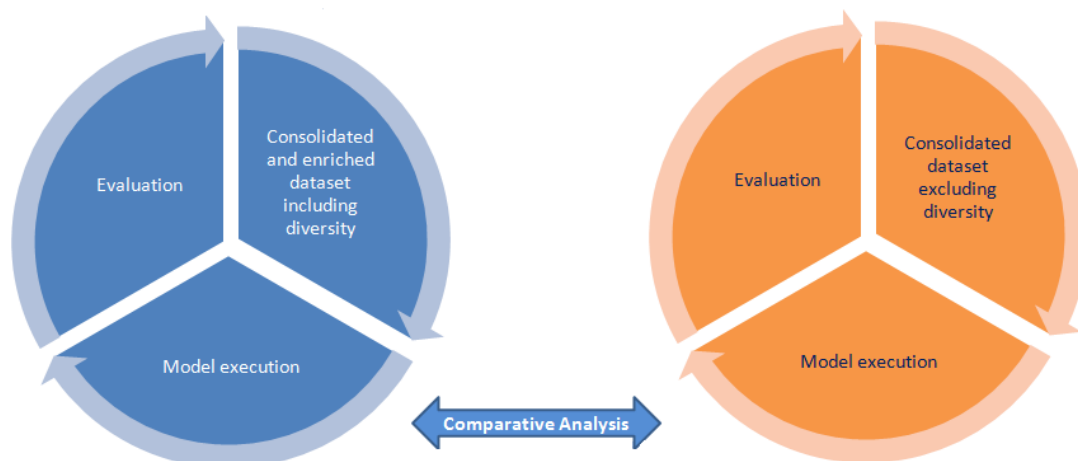
Figure 14 Model evaluation

Herlocker *et al.* (2004) suggests a number of metrics to evaluate the accuracy of predicted ratings for recommender systems. These include precision, recall, mean squared error and ROC. Herlocker *et al.* (2004) state that many newer and existing algorithms have a mean absolute error of 0.73 when utilised on movie ratings datasets with a five point rating scale. This absolute error rate will be used as a benchmark comparison even though the rating scale (10 point) and dataset domain differ. These techniques will be used in the evaluation phase of this project. Where a regression model is used correlation coefficient, mean absolute error and, root mean squared error will be provided. Conclusions will be formulated based on the results of the experiment conducted as part of this project.

In summary, the solution outlined in this design chapter encompasses a number of design layers including data extraction from different file formats, data pre-processing including enrichment and transformation to expose as much information as possible and model building and evaluation. This solution has been selected as it appropriate for the data utilised for this research question and suitable for the infrastructure and technical resources available.
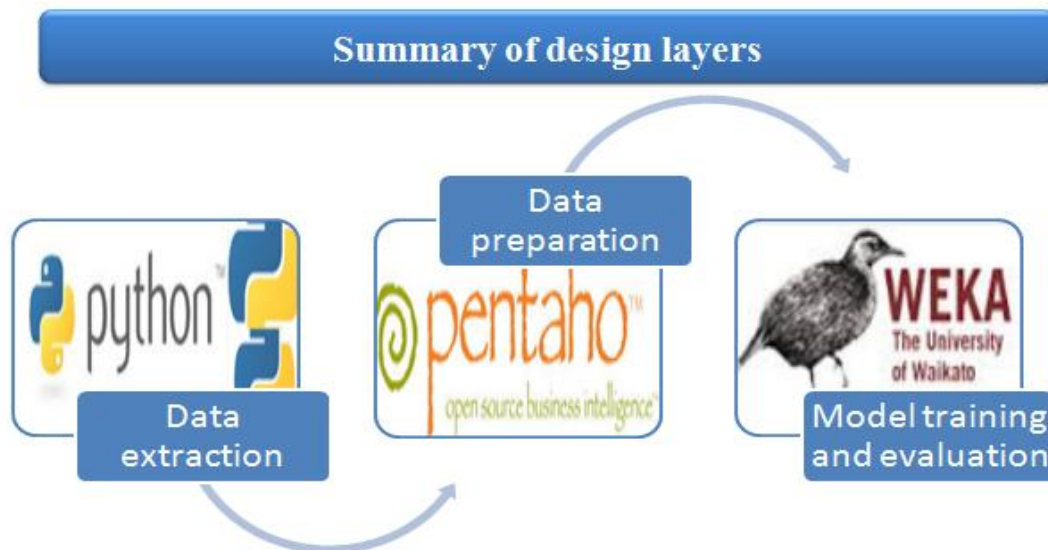
Figure 15 Summary of design layers

This solution is intended to allow for the testing of the hypothesis that diversity measures improve accuracy of predicted ratings when compared to models without diversity measures included.

4   EXPERIMENT

The experiment seeks to execute a number of predictive models that utilise measures of diversity to test the hypothesis outlined in the previous chapter. The consolidated and enriched datasets outlined in chapter 3 have been used in this endeavour. Training datasets have been utilised for model training. There were a number of pre-processing steps involved in this experiment including data exploration, merging and transformation. Details of these steps initially outlined in the design chapter are provided in the sections below. There were also specific pre-processing steps utilised for each model which are detailed in the section dedicated to each model run.

Test datasets have been used for model evaluation. In addition, comparison against a model of the consolidated dataset without measures of diversity has been used for evaluation. The results obtained from model execution will be outlined within this chapter but discussed in more detail in chapter five.

4.1 Data Exploration

An initial step before commencement of pre-processing was exploration of the input datasets. Four disparate datasets of varying complexity were utilised in this experiment as outlined in the design chapter. The Book Crossing datasets consists of three csv files titled BX-Ratings, BX-Users and BX-Books. The BX-Books file contains 271,379 records of books. The BX-User file contains details of 278,858 users of the Book Crossing website. The BX-Ratings file contains 1,149,780 ratings. The amazon metadata dataset used appeared to be in XML format though tags were missing. The amazon metadata dataset represents 548,552 products of which 393,561 are books. This provided additional attributes for use in model execution. Graphical output relating to the input and consolidated file can be found in Appendix A.

The BX-Users file showed that the User_ID field is fully populated. The User_ID ranged between 1 and 6 characters in length with no evidence of letters or special characters or unnecessary spaces. The user id's range from 1 to 258,858. However, on this file the age field is very poorly populated as 110,761 records have a null value for age (42%). In addition NULL is a value populated within the Age field.

The Literature review suggested that any field with greater than 5% missing values would need to be addressed. Replacement with the median or mode values is unlikely to be representative of the user's true age and may introduce noise into the dataset. Exploration of this attribute also shows a minimum value of zero and a maximum value of 244, both of which are likely to be spurious values. The combination of missing and spurious data reduces the usability of this field. The Location field has a maximum character length of 105 and a minimum value of 3. On review this appeared to be a default value used to represent missing values. The Location field contains 11,317 special characters.

The BX-Books file shows an ISBN field that is alphanumeric and containing different formats with 21,924 that are entirely uppercase and 411 entirely lowercase. While most instances have a record length of 10 characters there is evidence of whitespace and three records with 13 characters. These were cross validated against the book ratings file. This field is particular important as it will be used as a join key for the files. There were no special characters. The Book-Author field is alphanumeric showing varying formats of both uppercase (5,914) and lowercase characters (54). 32 records containing digits in the authors name, for example. '3rd Duke of' or the number of a government agency. 2086 records have special characters. The Book Publisher field has 596 records containing numeric characters such as Channel 4. There are varying formats displayed (1,146 entirely uppercase and 158 entirely lowercase). 1,796 have ASCII characters of which there are 144 distinct values. A review of distinct values highlights some slight variation in spelling which creates duplicates Frommer's, Frommer. Database translation issues such as &amp are also visible. The Year of publication shows data quality issue as the maximum year is 2050. 4619 records show a year of publication of 0.

The BX-Ratings file represents the cleanest file though it has the fewest attributes. THE ISBN field is alphanumeric with a maximum length of 13 and contains unusual entries such as NONFICTION and SELFPUBLISHED. 95,036 records are entirely uppercase and 605 are entirely lowercase. 10 records have ACSII characters are visible. Similarly to the BX-User file the User_Id on the Bx-Ratings file is fully populated and in integer format.

The Book Rating attribute has a maximum value of 888,809,228 and a minimum value of 0. 0 represents implicit ratings but ratings should only be between 0 and 10. There are no nulls and the attribute has an integer format.

4.2 Data Pre-processing

The design and data exploration conducted informed the data pre-processing undertaken to arrive at a consolidated data set enriched with measure of diversity. The subsections below outline the pre-processing steps taken on each file. As aforementioned additional pre-processing suitable for each model was undertaken details of which are outlined in the section in this chapter dedicated to each model.

4.2.1 Book Crossing pre-processing

Data preparation commenced with the Book Crossing dataset. The individual data files were initially profiled to understand potential issues that would need to be addressed before data merging could occur.

The ACSII or special characters identified as part of the data exploration phase were replaced or removed as appropriate in the BX-Users, BX-Books and BX-Ratings files. The details of this pre-processing step are available in Appendix C. These characters were addressed to ensure that values were consistent across the consolidated file. An instance may not be recognised as having the same value as another if one has a special character included and another does not. These special characters were ACSII characters often utilised in non-English languages. The Location field was parsed on ',' into three new fields titled Location_Line_1, Location_Line_2 and Country to make it more usable. A check was performed on duplicates based on User_ID, Location_Line_1, Location_Line_2, Country and Age. 8 duplicate records were identified within the Bx-Users file. The field Book Title within the BX-Books file has too much variation reducing its usefulness. This field was removed in addition to the image URL fields. Duplicate records were removed where the ISBN, Book_Author, Year of Publication and Publisher were the same. This accounted for 319 records.

Data exploration identified ratings between zero and ten. However a rating of zero described as indicating an implicit rating does not provide any associated metrics with regard to the user behaviour underpinning this rating. As such, ratings of zero were removed for this reason. Duplicates records where the same user had rated the same ISBN were removed. This equated to 33 records.

4.2.2 Amazon metadata pre-processing

A number of approaches were taken to parse the amazon metadata file. While it is suspected that this file was in XML format originally the missing tags meant that parsing of this file proved challenging. This coupled with the fact that the file was too large to review with a text editor meant that different options had to be explored. Python was selected to parse the file due to its speed and open source nature. Initially SQL injection to a MYSQL database was chosen. However, mySQL had issues with accepting the file due to special characters which could not be easily identified due to the file size. An alternative was utilised whereby an output csv was produced by Python following parsing of the file. The level of parsing selected was aligned to the required fields used for testing or in downstream models. Python parsed the amazon metadata file by iterating through each row in the input file to produce the condensed csv file. ASIN was also renamed to ISBN for ease of use in Pentaho. The output csv file contained ID, ISBN, Title, Group, Salesrank, Similar, Categories, CategoryDetail, Reviews, ReviewDetail. The parsing of the file condensed the file into 548,552 rows facilitating further data pre-processing.

The amazon file was introduced into Pentaho once parsed. The fields ID, Title, Similar and ReviewDetail were removed as they were not required for downstream analysis. The CategoryDetail field was parsed on '|' into eight subcategories which were used for diversity calculations. Additional subcategories could be used but it was felt that there would be little consensus at that level of granularity. The Reviews field was also split on ':' to obtain the amazon average rating field. Subcategory 1 and 2 were constants representing no information and were therefore removed for this reason. The amazon file contained ACSII characters which were removed or replaced as appropriate.

In addition, the file was filtered to remove any records that did not have a Group equal to book as the file contained information relating to other products such as music and dvd's which are not of interest to this experiment.

4.3 Data merging

The BX-Users, Bx-Books and Bx-Ratings file were joined. The Bx-Ratings file had one rating per user and so was joined to the BX-Books file using ISBN. This output was then joined to the Bx-User file using User_ID. The Amazon dataset was joined on ISBN once a consolidated Book Crossing dataset was obtained. A reasonably low hit rate was obtained. Only 218,754 records could be found in the Amazon dataset with most fields populated out of the 433,639 available following merging of the data sets.

4.4 Measure derivation

The design chapter outlined a number of new metrics for use in the model execution. These calculations were performed using the consolidated dataset at either a ISBN or user level as appropriate. Some of the measures were produced for use in further calculations. Table 4 provides details of the calculations.

Id	Aggregation level	Column Name	Calculation
1	User_ID	User Ratings Count	Count of Book_Ratings
2	User_ID	User Average Rating	Average (Mean) Book_Rating
3	User_ID	User Min Rating	Minimum Book_Rating
4	User_ID	User Max Rating	Maximum Book_Rating
5	User_ID	User Rating Std Dev	Standard deviation of the Book_Rating field
6	User_ID	User Distinct Rating	Count of distinct Book_Ratings
7	User_ID	User Author Count	Count of distinct Book_Authors
8	User_ID	User Category3 Count	Count of distinct SubCategory3
9	User_ID	User Category4 Count	Count of distinct SubCategory4
10	User_ID	User Category5 Count	Count of distinct SubCategory5
11	User_ID	User Category6 Count	Count of distinct SubCategory6
12	User_ID	User Category7 Count	Count of distinct SubCategory7
13	User_ID	User Category8 Count	Count of distinct SubCategory8
14	User_ID	Maximum Amazon Rating	Maximum Average_Rating
15	User_ID	Minimum Amazon Rating	Minimum Average_Rating

16	User_ID	Distinct Amazon Rating	Count of distinct Average_Ratings
17	ISBN	Book Ratings Count	Count of Book_Ratings
18	ISBN	Book Average Rating	Average (Mean) Book_Rating
19	ISBN	Book Min Rating	Minimum Book_Rating
20	ISBN	Book Max Rating	Maximum Book_Rating
21	ISBN	Book Rating Std Dev	Standard deviation of the Book_Rating field
22	ISBN	Book Distinct Rating	Count of distinct Book_Ratings

Table 4 Calculation of derived measures

The Literature review informed the use of diversity measures through the concept of popularity. This was used in the experiment measure derivation through the calculation of Amazon_Popularity_Category and Book_Popularity_Category. The number range node in Pentaho was used to determine the top 20% of books based on the Book_Ratings_Count field. Records were stamped with Popular or Less Popular. This node was also used in to determine the top 20% of books based on the Amazon salesrank to populate the Amazon_Popularity_Category field. Data Exploration of the final dataset was undertaken. Visualisations relating to this exploration are available in Appendix A.

4.5 Model execution

The output data set comprising of 218,754 records was used to formulate training and test datasets as outlined in the design chapter. Four training and test sets were produced using the Weka Resample filter before any model execution commenced. The noreplacement parameter was set to true to ensure that the training and test sets contained different instances. A version of each training and test dataset combination was augmented to remove diversity measures facilitating model evaluation. Ten fold cross validation was utilised in all iterations to avoid the possibility of a random favourable outcome due to the selection of the training instances.

4.5.1 DecisionStump

The Book_Rating field was numeric in the input file and initially the DecisionStump algorithm was utilised as it provided a decision tree for a numeric output variable.

The Literature Review informed that decision trees are adept at handling missing values so this model was executed on the training and test datasets with no additional pre-processing as a baseline comparison. Table 5 below shows the output of DecisionStump execution for input files with and without diversity measures. The most notable aspect is that there is no difference between the model run with diversity measures and the model without these measures. The correlation coefficient shows a low level of positive correlation between the input variables and target variable. Inspection of the predicted values shows that numeric precision is included in the predicted value which influences the accuracy of the predicted model. Relative errors over 87% show that this is a poor model and indicates that this technique is not suitable for the research problem.

Model with diversity						
Dataset Category	% Split	Correlation Coefficient	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	0.4632	1.253	1.5938	87.035 %	88.6271 %
Test	40	0.4574	1.2522	1.5917	-	-
Training	70	0.4625	1.2537	1.5944	87.0922 %	88.6637 %
Test	30	0.4571	1.251	1.5896	-	-
Training	80	0.462	1.255	1.596	87.117 %	88.6878 %
Test	20	0.4562	1.2442	1.5809	-	-
Training	90	0.4614	1.2544	1.5947	87.1413 %	88.7191 %
Test	10	0.4557	1.2392	1.5773	-	-

Table 5 - Decision Stump model utilising measures of diversity

Model without diversity						
Dataset Category	% Split	Correlation Coefficient	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	0.4632	1.253	1.5938	87.035 %	88.6271 %
Test	40	0.4574	1.2522	1.5917	-	-
Training	70	0.4625	1.2537	1.5944	87.0922 %	88.6637 %
Test	30	0.4571	1.251	1.5896	-	-
Training	80	0.462	1.255	1.596	87.117 %	88.6878 %
Test	20	0.4562	1.2442	1.5809	-	-
Training	90	0.4614	1.2544	1.5947	87.1413 %	88.7191 %
Test	10	0.4557	1.2392	1.5773	-	-

Table 6 - DecisionStump model without diversity measures

4.5.2 Linear Regression

A linear regression model was selected based on the Literature Review due to the numeric input and target variables. This model was chosen to validate if the numeric precision issue identified during the DecisionStump model iteration was due to the simplicity of the model selection. As Pyle (1999) mentioned model selection can be an art in itself. Pre-processing was performed using a number of Weka filters in advance of any model iteration. Nominal variables were removed (Age, Location_Line_1, Location_Line2). These fields had a high level of variability and missing values reducing their usefulness. The Linear Regression model requires numeric input attributes only. The training and test data files were normalised to avoid issues with scale.

The results differed between the model with diversity measures and the model without these measures. The model with diversity measures performed marginally better than the model without these measures. The correlation coefficient is close to one which suggests that there is a relationship between the input variable and target variables. The 80% and 20% split of training to test data resulted in the best results in both the model with diversity and the model without diversity metrics. The relative errors are high between 65 to 71% which indicates that this is a poor model. Inspection of the predicted outcomes shows that the rating precision issue identified with the DecisionStump model persists and as such alternative models were utilised.

Model with diversity						
Dataset Category	% Split	Correlation Coefficient	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	0.7139	0.9443	1.2592	65.593 %	70.0211 %
Test	40	0.713	0.9415	1.2554	-	-
Training	70	0.7143	0.9434	1.2584	65.5409 %	69.9798 %
Test	30	0.7118	0.9433	1.2556	-	-
Training	80	0.7124	0.9387	1.246	65.9222 %	70.1969 %
Test	20	0.7124	0.9387	1.2469	-	-
Training	90	0.7143	0.9434	1.2579	65.5396 %	69.9836 %
Test	10	0.7071	0.9465	1.2536	-	-

Table 7 - Linear Regression model utilising measures of diversity

Model without diversity						
Dataset Category	% Split	Correlation Coefficient	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	0.7137	0.945	1.2597	65.6416 %	70.0487 %
Test	40	0.7126	0.9423	1.256	-	-
Training	70	0.714	0.9442	1.259	65.5936 %	70.0131 %
Test	30	0.7115	0.944	1.2561	-	-
Training	80	0.7136	0.9454	1.2608	65.6304 %	70.0598 %
Test	20	0.7122	0.9394	1.2472	-	-
Training	90	0.714	0.9442	1.2586	65.5904 %	70.0172 %
Test	10	0.707	0.9466	1.2538	-	-

Table 8 - Linear Regression model without measures of diversity

4.5.3 J48 Decision Tree

An alternative approach was taken to model execution in an attempt to improve results. The Weka filter NominalToBinary was used to convert the Book Popularity Category and Amazon_Popularity_Category field to binary fields. The RemoveType filter was used to remove nominal variables (Age, Location_Line_1, Location_Line_2, Country). The NumerictoNominal filter was applied to the class variable Book_Rating to allow for use of the J48 decision tree model. The Literature Review suggested that decision trees are not susceptible to outliers so the data was not normalised.

The best performing iteration was the 90% training and 10% test data set split with 55.1042% and 54.9232% correctly classified instances respectively. This combination provided the most data for training purposes. The test dataset displayed a reduction in correctly classified instances but it was not a large amount. The Kappa statistic is greater than zero which indicates that the correctly classified instances are unlikely to be due to chance. Overall the error rate is high for this model.

Model with Diversity								
Dataset Category	% Split	Correctly Classified Instances	Incorrectly Classified Instances	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	54.7184 %	45.2816 %	0.4524	0.0948	0.2678	57.3283 %	93.1655 %
Test	40	54.8913 %	45.1087 %	0.4544	0.0947	0.2668	-	-
Training	70	54.7905 %	45.2095 %	0.4533	0.0944	0.2672	57.1355 %	92.9327 %
Test	30	54.7305 %	45.2695 %	0.4522	0.0944	0.2668	-	-
Training	80	54.7928 %	45.2072 %	0.4534	0.0945	0.2667	57.191 %	92.7605 %
Test	20	54.7919 %	45.2081 %	0.453	0.0942	0.2658	-	-
Training	90	55.1042 %	44.8958 %	0.4571	0.0939	0.2656	56.8206 %	92.3797 %
Test	10	54.9232 %	45.0768 %	0.454	0.0947	0.2664	-	-

Table 9 - J48 Decision tree model utilising diversity metrics summary evaluation

The precision and recall supports the summary evaluation metrics with quite low results for this model.

Model with Diversity							
Dataset Category	% Split	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Training	60	0.547	0.096	0.546	0.547	0.546	0.782
Test	40	0.549	0.095	0.547	0.549	0.548	0.785
Training	70	0.548	0.095	0.547	0.548	0.547	0.785
Test	30	0.547	0.096	0.546	0.547	0.546	0.785
Training	80	0.548	0.096	0.547	0.548	0.547	0.785
Test	20	0.548	0.096	0.546	0.548	0.547	0.787
Training	90	0.551	0.095	0.549	0.551	0.55	0.788
Test	10	0.549	0.096	0.547	0.549	0.547	0.785

Table 10 - J48 Decision Tree model utilising diversity metric detailed accuracy (weighted averages)

The confusion matrix below shows the correctly classified records (in green) for the best performing iteration (90/10% training to test dataset split). The correctly classified instances range from 42% to 69% with instances with a rating of 1 or 10 showing the highest number of correctly classified instances.

Confusion Matrix - 90/10% split										
A	B	C	D	E	F	G	H	I	J	
45	2	0	2	2	1	4	4	2	3	A = 1
0	79	3	1	10	7	3	6	6	9	B = 2
3	5	102	8	18	10	28	25	7	17	C = 3
1	6	5	189	30	19	44	38	23	17	D = 4
5	9	23	28	1338	121	188	215	124	127	E = 5
2	8	12	47	158	735	249	301	148	94	F = 6
6	12	36	30	255	250	1765	778	346	271	G = 7
4	11	31	55	232	252	691	3027	634	505	H = 8
6	11	19	29	149	132	392	682	1846	532	I = 9
3	7	18	12	129	82	226	432	373	2889	J = 10

Table 11 - J48 Decision tree confusion matrix (90/10% training and test dataset with diversity measures)

Similar results are visible for the model without diversity measures. This model appears to perform marginally better with a higher percentage of correctly classified instances on the best performing iteration (90/10% training to test split).

Model without Diversity								
Dataset Category	% Split	Correctly Classified Instances	Incorrectly Classified Instances	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	54.8761 %	45.1239 %	0.4543	0.0947	0.2662	57.3061 %	92.6038 %
Test	40	55.0925 %	44.9075 %	0.4567	0.0945	0.2648	-	-
Training	70	54.878 %	45.122 %	0.4544	0.0944	0.2654	57.1399 %	92.3323 %
Test	30	54.8128 %	45.1872 %	0.4532	0.0944	0.2653	-	-
Training	80	54.8476 %	45.1524 %	0.4538	0.0945	0.2652	57.1717 %	92.2627 %
Test	20	55.1233 %	44.8767 %	0.4566	0.0942	0.2642	-	-
Training	90	55.1844 %	44.8156 %	0.458	0.094	0.2638	56.8957 %	91.7793 %
Test	10	54.9781 %	45.0219 %	0.4548	0.0947	0.265	-	-

Table 12 - J48 Decision tree model without diversity metrics summary evaluation

Model without Diversity							
Dataset Category	% Split	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Training	60	0.549	0.095	0.547	0.549	0.548	0.785
Test	40	0.551	0.095	0.549	0.551	0.55	0.788
Training	70	0.549	0.095	0.547	0.549	0.548	0.787
Test	30	0.548	0.096	0.547	0.548	0.547	0.788
Training	80	0.548	0.096	0.547	0.548	0.547	0.788
Test	20	0.551	0.096	0.55	0.551	0.55	0.79
Training	90	0.552	0.095	0.55	0.552	0.551	0.791
Test	10	0.55	0.096	0.548	0.55	0.548	0.788

Table 13 - J48 Decision Tree model without diversity metrics detailed accuracy (weighted averages)

The confusion matrix for the model without diversity metrics shows the correctly classified records (in green) for the best performing iteration (90/10% training to test dataset split). The correctly classified instances range from 42% to 69% with instances with a rating of 10 showing the highest number of correctly classified instances. This is very similar to the model with diversity metrics.

Confusion Matrix										
A	B	C	D	E	F	G	H	I	J	
43	1	0	0	2	3	5	6	2	3	A = 1
1	83	1	2	4	4	5	9	7	8	B = 2
4	2	110	6	21	11	23	25	6	15	C = 3
1	3	2	188	35	23	43	41	21	15	D = 4
4	8	24	38	1321	121	206	199	125	132	E = 5
3	10	10	48	158	734	275	295	131	90	F = 6
6	17	27	39	260	258	1800	758	317	267	G = 7
4	10	38	49	238	240	683	3018	673	489	H = 8
6	9	16	27	163	125	411	676	1852	513	I = 9
1	6	18	22	120	80	229	433	384	2878	J = 10

Table 14 - J48 Decision Tree confusion matrix (90/10% training and test dataset without diversity measures)

4.5.4 Naive Bayes

The Naive Bayes model utilised the same pre-processing performed for the J48 Decision Tree model. In addition the FilteredClassifier specified for use with Naive Bayes was selected. This allowed for the data to be discretized as the data attributes are not normally distributed. The selection of this filter mitigates the risk of incompatibility between the training and test data sets. The model utilising 80% of the training data and 20% for testing has the best results based on the correctly classified instances. The Kappa statistics is greater zero which indicates that the correctly classified items did not occur purely by chance. This is also supported in Table 16 as the ROC area is greater than .50. Recall as a measure of overall accuracy at 0.546 is low and the mean absolute error at 0.0992 is much lower than the benchmark rate of 0.73.

Model with Diversity								
Dataset Category	% Split	Correctly Classified Instances	Incorrectly Classified Instances	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
Training	60	54.3131 %	45.6869 %	0.4508	0.0996	0.2337	60.2651 %	81.281 %
Test	40	54.3016 %	45.6984 %	0.4508	0.0996	0.2342	-	-
Training	70	54.3431 %	45.6569 %	0.4509	0.0996	0.2336	60.2489 %	81.2441 %
Test	30	54.4989 %	45.5011 %	0.4529	0.0993	0.2337	-	-
Training	80	54.3682 %	45.6318 %	0.4514	0.0993	0.2336	60.0844 %	81.2528 %
Test	20	54.6319 %	45.3681 %	0.4546	0.0992	0.2338	-	-
Training	90	54.4916 %	45.5084 %	0.453	0.0991	0.2336	59.9738 %	81.2437 %
Test	10	54.4067 %	45.5933 %	0.4518	0.0994	0.2342	-	-

Table 15 - Naive Bayes model utilising diversity metrics summary evaluation

Model with Diversity							
Dataset Category	% Split	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Training	60	0.543	0.092	0.554	0.543	0.54	0.871
Test	40	0.543	0.092	0.553	0.543	0.54	0.871
Training	70	0.543	0.093	0.553	0.543	0.54	0.871
Test	30	0.545	0.092	0.554	0.545	0.542	0.872
Training	80	0.544	0.093	0.553	0.544	0.541	0.872
Test	20	0.546	0.092	0.556	0.546	0.544	0.872
Training	90	0.545	0.092	0.553	0.545	0.542	0.872
Test	10	0.544	0.092	0.552	0.544	0.541	0.869

Table 16 - Naive Bayes model utilising diversity metric detailed accuracy (weighted averages)

The confusion matrix for the best performing iteration which is the 80% training and 20% test dataset split is shown below. There is a larger range of correctly classified instances than those displayed for the J48 model. This range is between 42% and 78%. Rating 2 had the highest number of correctly classified items and rating 6 had the lowest.

Confusion Matrix - 80/20										
A	B	C	D	E	F	G	H	I	J	
86	2	12	1	3	1	2	1	1	6	A = 1
2	186	8	9	11	1	5	1	3	13	B = 2
10	15	314	1	27	7	21	7	15	27	C = 3
9	20	43	475	45	35	68	41	29	49	D = 4
27	36	125	117	2719	130	509	272	182	348	E = 5
13	47	95	94	316	1451	492	389	197	331	F = 6
28	64	155	106	414	416	4096	792	607	821	G = 7
32	78	145	137	418	391	1732	5069	1179	1656	H = 8
20	41	105	83	214	159	894	1028	3413	1548	I = 9
24	39	81	42	144	80	427	841	637	6093	J = 10

Table 17 - Naive Bayes confusion matrix (80/20% training and test dataset with diversity measures)

The model without diversity had improved evaluation results. The 70/30% training to test dataset split shows the highest percentage of correctly classified instances.

Model with Diversity									
Dataset Category	**% Split**	**Correctly Classified Instances**	**Incorrectly Classified Instances**	**Kappa statistic**	**Mean absolute error**	**Root mean squared error**	**Relative absolute error**	**Root relative squared error**
Training	60	54.9706 %	45.0294 %	0.4572	0.0999	0.2312	60.4552 %	80.441 %
Test	40	54.9587 %	45.0413 %	0.4571	0.0999	0.2315	-	-
Training	70	54.9923 %	45.0077 %	0.4573	0.0999	0.2312	60.4399 %	80.4362 %
Test	30	55.1998 %	44.8002 %	0.4595	0.0997	0.2313	-	-
Training	80	54.9608 %	45.0392 %	0.4571	0.0997	0.2313	60.3024 %	80.4402 %
Test	20	55.1347 %	44.8653 %	0.4589	0.0995	0.2314	-	-
Training	90	55.0356 %	44.9644 %	0.4579	0.0995	0.2312	60.2086 %	80.4295 %
Test	10	55.1243 %	44.8757 %	0.4586	0.0998	0.2319	-	-

Table 18 - Naive Bayes model without diversity metrics summary evaluation

The Precision and Recall figures also support the summary evaluation showing more favourable results than the model where diversity metrics were utilised.

Model without Diversity							
Dataset Category	**% Split**	**TP Rate**	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**
Training	60	0.55	0.094	0.56	0.55	0.548	0.874
Test	40	0.55	0.094	0.56	0.55	0.548	0.874
Training	70	0.55	0.094	0.56	0.55	0.549	0.874
Test	30	0.552	0.094	0.562	0.552	0.551	0.874
Training	80	0.55	0.094	0.56	0.55	0.548	0.875
Test	20	0.551	0.094	0.561	0.551	0.55	0.874
Training	90	0.55	0.094	0.56	0.55	0.549	0.875
Test	10	0.55	0.094	0.561	0.551	0.55	0.872

Table 19 - Naive Bayes model utilising diversity metric detailed accuracy (weighted averages)

The confusion matrix for the model iteration utilising 70% of the dataset for training and 30% for testing shows a range of correctly classified records between 41% and 86%. Instances with a rating of 1 had the highest number of correctly classified records and instances with a rating 6 of had the lowest number of correctly classified records.

Confusion Matrix										
A	B	C	D	E	F	G	H	I	J	
161	2	3	3	6	2	5	1	0	5	A = 1
7	286	3	15	13	1	9	1	1	22	B = 2
14	23	522	4	40	11	32	22	11	35	C = 3
15	36	69	692	62	39	114	85	22	54	D = 4
33	63	191	185	4108	159	813	588	215	446	E = 5
27	69	147	135	405	2079	821	813	214	346	F = 6
49	99	244	159	537	567	6228	1680	715	919	G = 7
51	132	249	200	545	502	2765	8529	1350	1875	H = 8
32	83	170	122	273	217	1419	2194	4875	1830	I = 9
50	78	120	77	199	98	691	1767	876	8746	J = 10

Table 20 - Naive Bayes confusion matrix (70/30% training and test dataset without diversity measures)

The results outlined in this chapter will be further evaluated in the subsequent chapter where an overall evaluation of the research project as a whole will be undertaken.

5 EVALUATION

The focus of this chapter is the evaluation of the results obtained from the experiment. Each model will be evaluated independently and against other models executed as part of the experiment. This evaluation involves assessment with regard to learnings gained from the literature review. Strengths and limitations of the overall approach to this research will be discussed

5.1 Evaluation of results

The results outlined in the previous chapter will be discussed in more detail in this section. The models have different evaluation measures depending on their regression versus classification objective. The model results will be compared where they are comparable. The regression models where the target variable was in numeric format will be compared based on correlation coefficient, mean absolute error and root mean squared error. Precision, recall and ROC area will be used for models that utilised the target variable in nominal format. The recall metric shows how many instances were correctly classified. Commentary will be provided with regard to the best training and test data set pair.

5.1.1 Regression models

Table 21 below summarises the performance of the DecisionStump and Linear Regression models. The DecisionStump model performs particularly badly mean absolute error of 1.2392 and relative absolution error of 1.5773. The DecisionStump model also did not display any difference between the data set containing diversity metrics and the data set that did not contain diversity metrics.

Model	Test dataset %	Diversity included	Mean absolute error	Relative absolute error
DecisionStump	10%	Y	1.2392	1.5773
DecisionStump	10%	N	1.2392	1.5773
Linear Regression	10%	Y	0.9465	1.2535
Linear Regression	20%	N	0.9394	1.2472

Table 21 Regression model comparison

The Linear Regression model also shows high mean absolute error and relative absolute errors. This poor performance by both regression models is likely due to the numeric but not continuous format of the target variable which reduces the suitability of these model. Missing values and scale are unlikely to be a contributor to the poor performance as nominal values had to be removed for use in these models and the linear regression model was normalised. The nominal values within the dataset such as Age were the main source of missing values. In addition, the literature review indicated that decision trees are less susceptible to missing values. The poor performance of the linear regression model could also be attributed to a non linear relationship between the input variables though the correlation coefficient indicates a positive relationship between actual and predicted target values.

5.1.2 Classification models

Table 22 shows the results of the best performing J48 Decision Tree and Naive Bayes model iteration. The Naive Bayes classifier outperforms the decision tree for the model without diversity when compared using the cost sensitive measure ROC area. The Naive Bayes classifier displays higher performance with regard to precision and ROC area for the model with diversity. However, recall is lower than the J48 decision tree model.

Model	Test dataset %	Diversity included	Precision	Recall	ROC area
J48 Decision Tree	10%	Y	0.547	0.549	0.785
J48 Decision Tree	10%	N	0.548	0.551	0.788
Naive Bayes	20%	Y	0.556	0.546	0.872
Naive Bayes	30%	N	0.562	0.552	0.874

Table 22 Classification model comparison

The Literature Review highlighted that often models such as Decision Trees and Naive Bayes are viewed as simplistic but can be very robust and performant. In this instance the models do not display a high level of accuracy though the ROC area results are high. Overall the Naive Bayes classifier trained on 70% of the dataset without using measures of diversity is the best performing model. The ROC curves available for each rating type (1-10) show variation with instances where the book rating equals to 1 showing ROC area of 0.9984 and those with a rating of 8 showing ROC area of 0.8239. These are also the categories with the least and most number of instances. This indicates that skewness in the dataset may be affecting the results though discretization was performed. It was noted in the Literature Review that discretization can discard a lot of information which may mean that alternative methods of normalisation may incur better results. Naive Bayes can also perform poorly if much of the same information is held by different input variables. This could be a source of performance degradation.
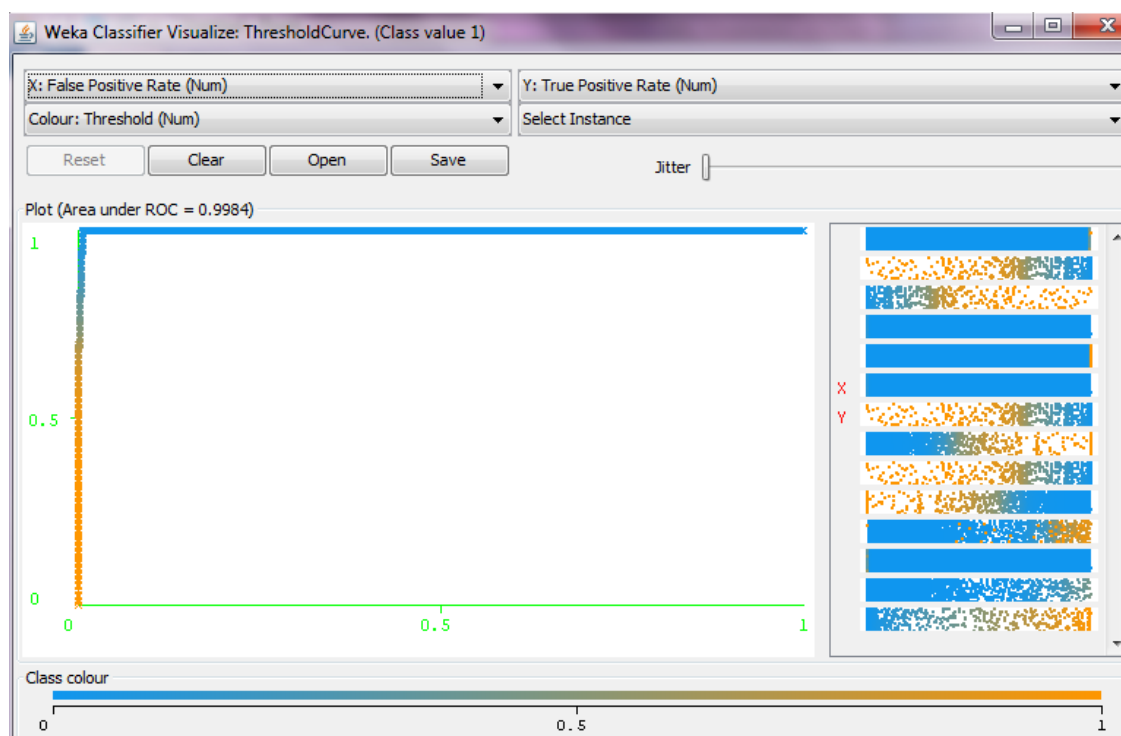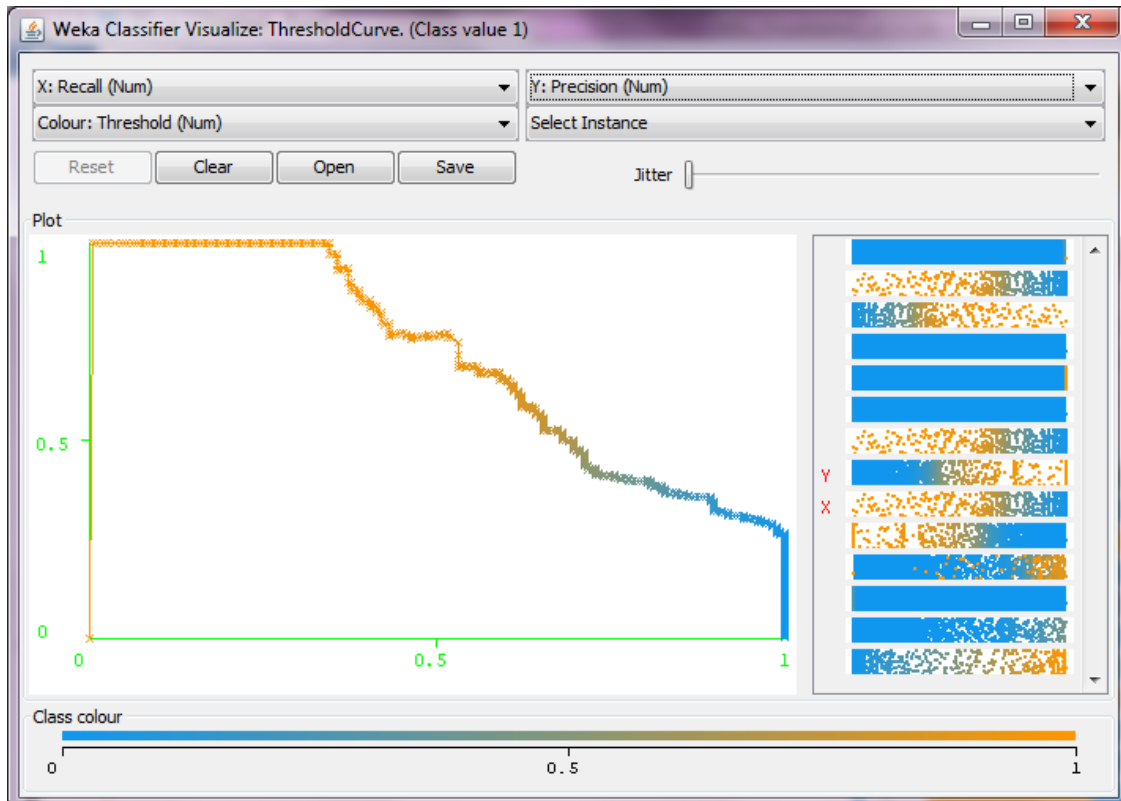

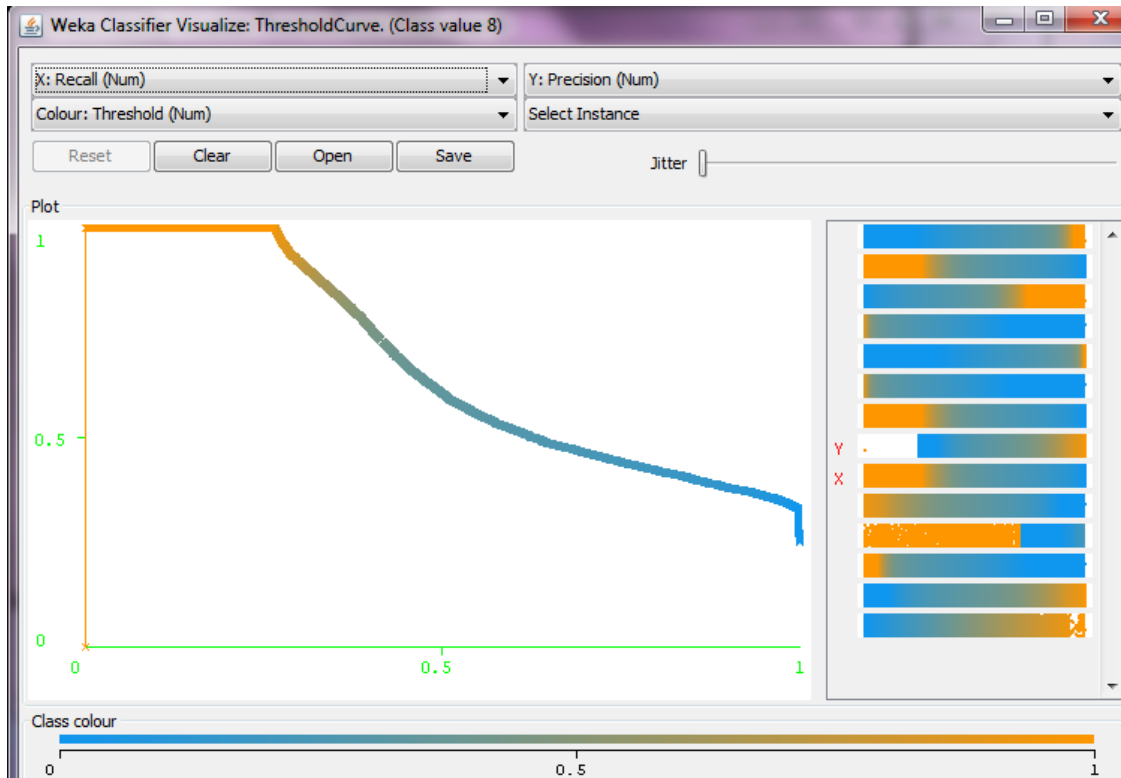**Figure 16 ROC curve for instances with a Book_Rating = 1**
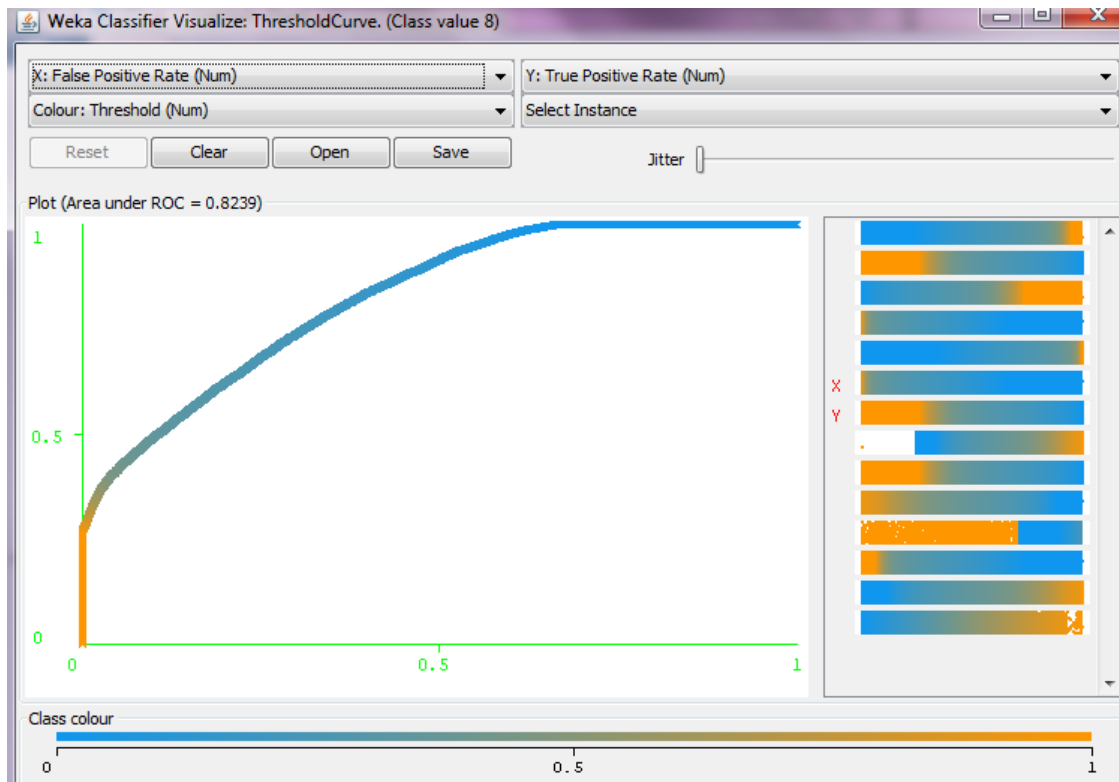
Figure 17 ROC curve for instances with a Book_Rating = 8

In addition, the Precision and Recall curves are consistent with instances with ratings of 1 or 8 showing the best curve. However, most of the curves are consistent in displaying a decline before 50% Recall.

Figure 18 Precision and Recall curve for instances with a Book_Rating = 1

Figure 19 Precision and Recall curve for instances with a Book_Rating = 8

The results would suggest that the models are influenced by the number of instances per rating type. The experiment did not demonstrate that diversity measures as inputs improve the accuracy of predicted ratings so we fail to reject the null hypothesis. However, the evaluations results for the model without diversity measures were low also and comparable to those with diversity indicating that further research in this area may be worthwhile.

5.2 Strengths of including diversity measures

The introduction of this paper outlined the challenge associated with defining diversity for users as it is closely linked to personality and taste which can vary from project to project. While the experiment conducted did not clearly demonstrate that the inclusion of diversity measures as inputs improves the accuracy of predicted ratings, some learnings from the literature review were supported. The Decision Tree and Naive Bayes models had the best time performance as was identified as part of the literature review. Additional strengths associated with this experiment include the use of cross validation in model iterations and the use of multiple model iterations based on different training and test dataset splits. The dichotomous approach to pre-processing at the overall data and model level also added value to this experiment.

5.3 Limitations of including diversity measures

The project had some limitations most noticeably the poor experiment outcomes. This may be due to the unavailability of alternative data sources including a time dimension, the approach to sampling for training and test datasets, the diversity measures selected and the choice of pre-processing and models. Aggregation or consolidation of the dataset may have improved results. Much of the effort utilised was on trying to get the data in a suitable format for use within models. An alternative approach to data normalisation may have improved accuracy. It could be argued that the datasets selected were not appropriate for the challenge as the models without diversity measures performed poorly also. Additional processing power could have facilitated the use of other models that are more computationally intensive.

6 CONCLUSIONS

This chapter concludes the dissertation outlining the contribution to the body of knowledge and areas of future work. The dissertation was concerned with the evaluation of the use of diversity to improve the accuracy of predicted ratings in recommender systems. *Does diversity improve the accuracy of predicted ratings in recommender systems?* was the specific research question being explored. This research question was selected as diversity can be hard to identify as it can be influenced by a users personality. Strengths and limitations of the approach to each objective is outlined in the subsequent paragraphs.

6.1 Summary of dissertation

The first objective of this dissertation was the completion of a literature review of general issues, trends, diversity and algorithms used to predict ratings in recommender systems including identification of gaps in current approaches. This objective was completed providing an overview of the diversity challenge and general challenges applicable to research regarding recommender systems. Coverage was broadened through the discussion of algorithms used in recommender systems and the implications for the introduction of diversity. The advantages and disadvantages of each algorithm were outlined and the associated pre-processing discussed. The Literature Review also influenced the subsequent design and experiment chapters. A definitive step by step guide to appropriate pre-processing is difficult to ascertain as it is often determined by the data itself. Finding enough detailed information regarding appropriate pre-processing was challenging.

The design of an experiment in support of the research question was the second objective of this dissertation. The design chapter provided an overview of the approach to the experiment and the rationale based on initial data exploration. The analysis conducted during the design informed the selection of suitable data for model derivation. The initial consideration of MySQL meant that much exploration work was undertaken that expended time and this software ultimately had to be abandoned.

It could be argued that the time cost could have been used to explore more complex diversity measures that may have had a more favourable impact on the experiment results.

The experiment chapter provided details of the data exploration, pre-processing and enrichment undertaken. A strength of this chapter is the breadth of model iterations utilised. The use of training and test datasets and cross validation is also a favourable aspect. Models were evaluated with regard to the best prediction results. An alternative approach could have been the selection of a single model with deeper focus on parameters and exploration of approaches to pre-processing.

Analysis and discussion of the results including an overall evaluation of the experiment success or failure was completed in the evaluation chapter. Models were built though low levels of success regarding accuracy of predicting ratings was demonstrated resulting in failure to reject the null hypothesis. However the evaluation chapter demonstrated critical analysis through identification of strengths and limitations and potential alternative applications.

6.2 Contribution to the body of knowledge

Contributions to the body of knowledge include the literature review and the approach to parsing the Amazon metadata file. Further research could utilise the code produced as a starting point for analysis. The outline of the limitations within this project could assist with further research allowing the avoidance of pain points. The testing of more complex models highlights that simpler models such as naive bayes from a technology infrastructure, time and accuracy point of view as beneficial. The review of this paper could prompt a further research idea in the area of diversity within recommender systems.

6.3 Future work

There are a number of areas of future work that have been identified during the completion of this project. Further and more complex metrics of diversity could be produced such as comparison of text similarity increasing the use of Amazon categorisation and book titles.

Alternative pre-processing steps could be utilised including bootstrapping for sample selection and statistical approaches to data normalisation with alternative tools. An alternative approach to model implementation using deeper modelling could result in different evaluation results. A data gathering exercise could be conducted to avail of enhanced data sources and a time dimension. This could be conducted through the creation of online tests to explore the preference for diversity and the creation of a GUI for experiment purposes. A further project could also consist of a survey of experts to enhance implementation approaches, provide domain knowledge and potentially the addition of a cost model if this could be ascertained. The process of completing this dissertation highlighted the importance of appropriate data preparation and model selection. While there are a number of areas identified for future work an interesting endeavour would be the enhancement of an existing successful system from a diversity perspective combined with user feedback from a test user group. This may likely involve collaboration with industry which would further enhance the learning experience.

BIBLIOGRAPHY

1. Acuna, E and Rodriguez, C (2004), 'The treatment of missing values and its effect on classifier accuracy'. Classification, Clustering, and Data Mining Applications, 639-47. Springer Berlin Heidelberg.

2. Adomavicius, G and Kwon, Y (2009), 'Toward more diverse recommendations: Item re-ranking methods for recommender systems'. *Workshop on Information Technologies and Systems*.

3. Adomavicius, G and Tuzhilin, A (2005), 'Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions'. *IEE Transactions on Knowledge and Data Engineering*, Vol. 17(6), 734–49.

4. Agrawal, R, Gollapudi, S, Halverson, A, and Ieong, S (2009), 'Diversifying search results'. In Proceedings of the Second ACM International Conference on Web Search and Data Mining, 5-14. ACM.

5. Al-Taie, M and Kadry, S (2012) 'Applying Social Network Analysis to Analyze a Web-Based Community'. *International Journal of Advanced Computer Science and Applications,* Vol. 3(2), 29-41.

6. Amazon.com (2014) Company Facts. Available: http://phx.corporate-ir.net/phoenix.zhtml?c=176060&p=irol-factSheet. [Date Accessed: 15 October 2014]

7. Bell, R M and Koren, Y (2007), 'Improved neighborhood-based collaborative filtering'. In *KDD Cup and Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Jose, California, United States of America.

8. Bradley, K and Smyth, B (2001), 'Improving recommendation diversity'. *Proceedings of the Twelfth National Conference in Artificial Intelligence and Cognitive Science* (AICS-01), 85–94.

9. Castells, P, Vargas, S, and Wang, J (2011), 'Novelty and diversity metrics for recommender systems: choice, discovery and relevance'. International Workshop on Diversity in Document Retrieval (DDR 2011) at the *33rd European Conference on Information Retrieval (ECIR 2011)*, Dublin, Ireland, 29-36.

10. Cayzer, S, and Aickelin, U (2002), 'A recommender system based on the immune network', Evolutionary Computation, CEC'02,Vol. 1, 807-12, IEEE.

11. Chaturvedi, A, Carroll, J D, Green, P E and Rotondo, J A (1997), 'A Feature-Based Approach to Market Segmentation Via Overlapping K-Centroids Clustering', *Journal of Marketing Research,* Vol. 34, 370–77.

12. Chen, J S, Ching, R K H, Lin, Y S (2004), 'An Extended Study of the K-Means Algorithm for Data Clustering and Its Applications'. *The Journal of the Operational Research Society*, Vol.55, 976–87.

13. Collins Dictionary (2014), Available: http://www.collinsdictionary.com/, [Date Accessed: 25 October 2014]

14. Davidson, J, Liebald, B, Liu, J, Nandy, P, Van Vleet, T, Gargi, U, ...and Sampath, D (2010), 'The YouTube video recommendation system'. *In Proceedings of the fourth ACM conference on Recommender systems*, 293-6, ACM.

15. Economist (2010), Data, data everywhere Available: http://uk.emc.com/collateral/analyst-reports/ar-the-economist-data-data-everywhere.pdf, [Date Accessed: 14 December 2014]

16. Fayyad, U, Piatetsky-Shapiro, G and Smyth P (1996), 'From 'Data Mining to Knowledge Discovery in Databases', *AI Magazine*, 38-54

17. Ge, M, Delgado-Battenfeld, C and Jannach, D (2010), 'Beyond accuracy: evaluating recommender systems by coverage and serendipity.' *In Proceedings of the fourth ACM conference on Recommender systems*, 257-60.

18. Hahsler, M (2011), Recommenderlab: A Framework for Developing and Testing Recommendation Algorithms.

19. He, H, and Garcia, E A (2009), 'Learning from imbalanced data', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21(9), 1263-84.

20. Herlocker, J L, Konstan, J A, Terveen, L G, and Riedl, J T (2004), Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems (TOIS)*, Vol. 22(1), 5-53.

21. Huang, Z (1998) Extensions to the k-Means Algorithms for Clustering Large Data Sets with Categorical Values, *Data Mining and Knowledge Discovery*, Vol. 2 (3), 283-304.

22. Jeckmans, A J, Beye, M, Erkin, Z, Hartel, P, Lagendijk, R L, and Tang, Q (2013), Privacy in recommender systems, *Social Media Retrieval*, 263-81, Springer London.

23. Lam, X N, Vu, T, Le, T D, and Duong, A D (2008), 'Addressing cold-start problem in recommendation systems', In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, 208-11.

24. Longo, L, Barrett, S, and Dondio, P (2009), 'Toward Social Search-From Explicit to Implicit Collaboration to Predict Users' Interests', *WEBIST*, 693-6.

25. Linden, G, Smith, B and York J (2003) 'Amazon.com recommendations: Item-to-item collaborative filtering', *Internet Computing*, IEEE Vol. 7(1),76-80.

26. O'Donovan, J, and Smyth, B (2005), 'Trust in recommender systems', *Proceedings of the 10th international conference on Intelligent user interfaces*, 167-174, ACM.

27. Pazzani, M J, and Billsus, D (2007), 'Content-based recommendation systems'. *The adaptive web*, 325-41, Springer Berlin Heidelberg.

28. Pera, M S and Ng Y K (2011),' With a Little Help From My Friends : Generating Personalized Book Recommendations Using Data Extracted from a Social Website', *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology,* 1. Lyon, France.

29. Pyle, D (1999), Data preparation for data mining Vol. 1. Morgan Kaufmann.

30. Ramakrisknan, N, Keller , J and Mirza, B (2001), 'Privacy Risks in Recommender Systems', *EEE Internet Computing*, Vol. 5(6), 54-62.

31. Rana, C and Jain, S K (2012), 'Building a Book Recommender system using time based content filtering', *WSEAS Transactions on Computers*, Vol. 2 (11), 27-33.

32. Resnick, P and Varian, H R (1997), 'Recommender systems', *Communications of the ACM*, Vol. 40(3), 56-8.

33. Ricci, F, Rokach, L, and Shapira, B (2011), 'Introduction to recommender systems handbook'. *Recommender systems handbook*, 1-35, Springer US.

34. Sandoval, S V, (2012), Novelty and Diversity Enhancement and Evaluation in Recommender Systems MSc. in Computer Science and Telecommunications, unpublished dissertation, Autonomous University of Madrid, Spain. [Accessed: 30 December 2013]

35. Sarwar, B, Karypis, G, Konstan, J, and Riedl, J (2000), 'Analysis of recommendation algorithms for e-commerce', *Proceedings of the 2nd ACM conference on Electronic commerce,* 158-67, ACM.

36. Schafer, B, Konstan, J and Riedl, J (1999), 'Recommender Systems in E-Commerce', *Proceedings of the 1st ACM conference on Electronic commerce.* ACM.

37. Schafer, J B, Frankowski, D, Herlocker, J, and Sen, S (2007), 'Collaborative filtering recommender systems', *The adaptive web*, 291-324, Springer Berlin Heidelberg.

38. 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*, pp. 253-60, ACM.

39. Tan, P N, Steinbach, M and Kumar, V (2006), *Introduction to Data Mining*, 3rd ed, Addison-Wesley Longman Publishing, Boston.

40. Tao, X, Zhou, X, Lau, C H, and Li, Y (2013), 'Personalised information gathering and recommender systems: techniques and trends', *ICST Transactions on Scalable Information Systems*, Vol. 13, 1-3.

41. Vargas, S, Castells, P and Vallet, D (2011), 'Intent-oriented diversity in recommender systems', *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. Beijing, China.

42. Vozalis, E and Margaritis, K (2003), 'Analysis of recommender systems algorithms.' In *Proceedings of the 6th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA-2003), Athens, Greece, 1211-2*.

43. Witten, I H, Frank, E and Hall, M A (2011), *Data Mining: Practical machine learning tools and techniques,* 3rd ed. Morgan Kaufmann.

44. Wu, W, Li, C and Liang, H (2013), 'Using personality to adjust diversity in recommender systems', *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, 225-229.

45. Wu, X, Kumar, V, Quinlan, J R, Ghosh, J, Yang, Q, Motoda, H, ... and Steinberg, D (2008), 'Top 10 algorithms in data mining', *Knowledge and Information Systems*, Vol. 14(1), 1-37.

46. Zahedi, F (1991), 'An introduction to neural networks and a comparison with artificial intelligence and expert systems', *Interfaces*, Vol. 21(2), 25-38.

47. Zhang, T, and Iyengar, V S (2002), 'Recommender systems using linear classifiers', *The Journal of Machine Learning Research*, Vol. 2, 313-34.

48. Zhang, Z K, Liu, C, Zhang, Y C and Zhou, T (2010), 'Solving the cold-start problem in recommender systems with social tags', *EPL (Europhysics Letters)*, Vol. *92*(2).

49. Ziegler, C N, McNee S M and Konstan J A (2005), 'Improving recommendation lists through topic diversification', *Proceedings of the 14th international conference on World Wide Web, ACM, Japan*, 22-32.
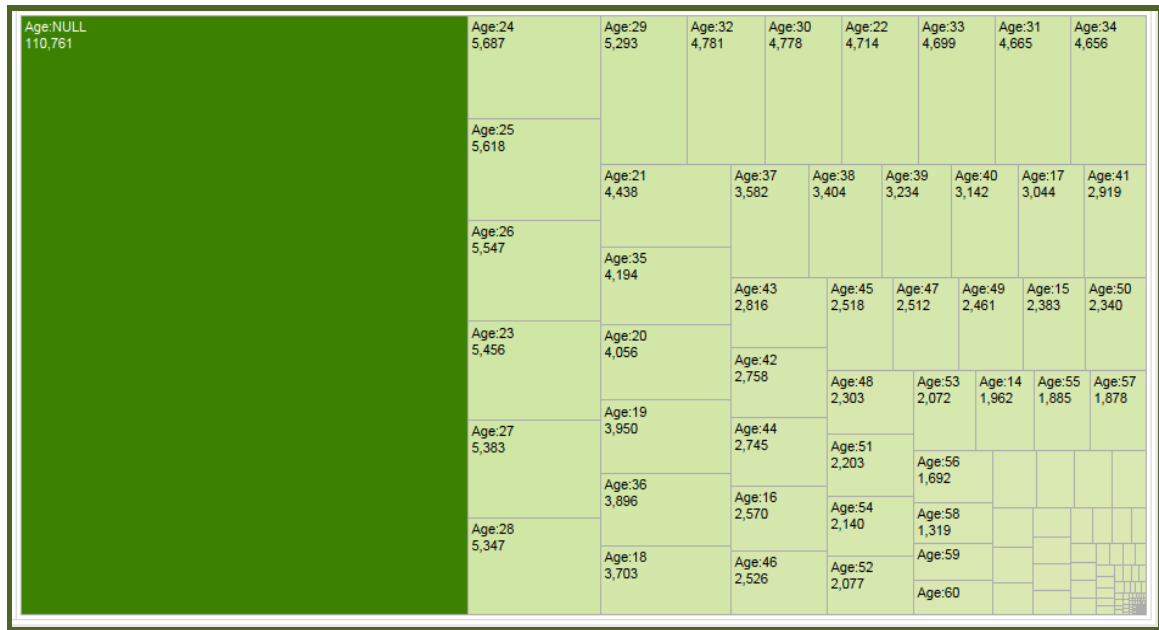
APPENDIX A

<u>BX-Book Ratings file</u>
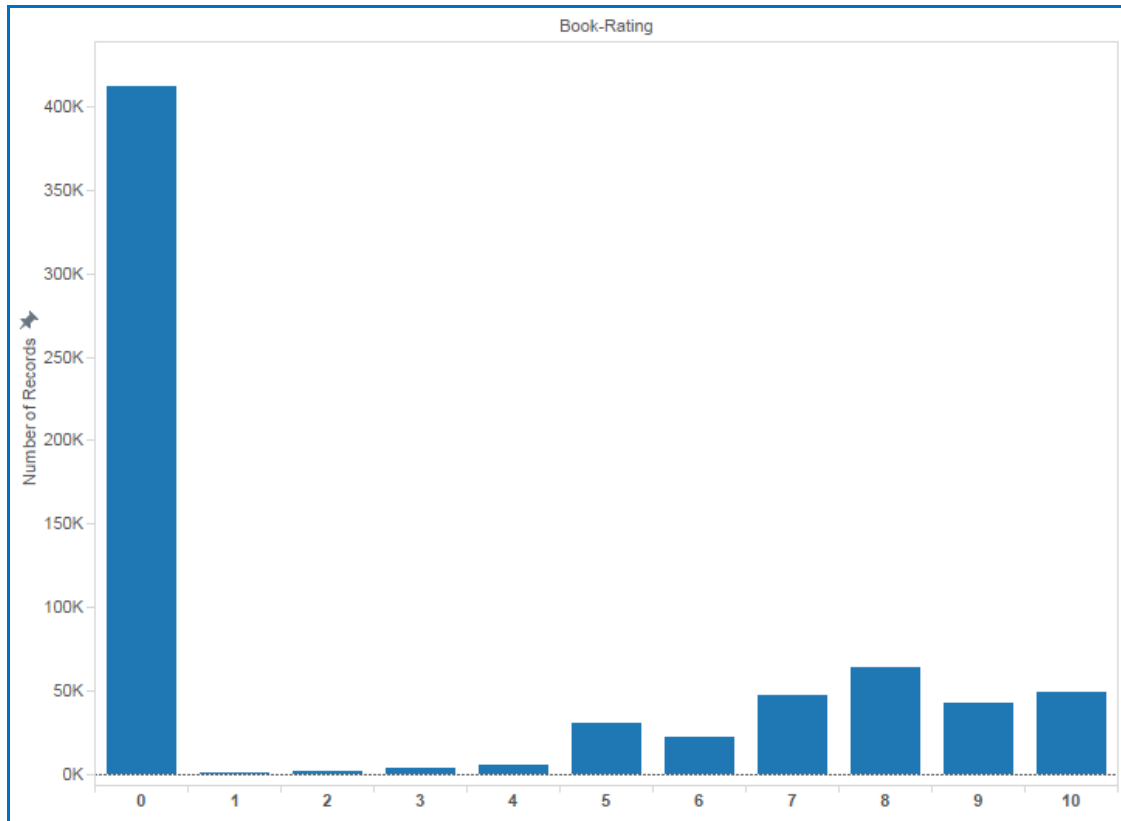
Histogram produced using Tableau for the variable Book-Rating showing negative skewness.
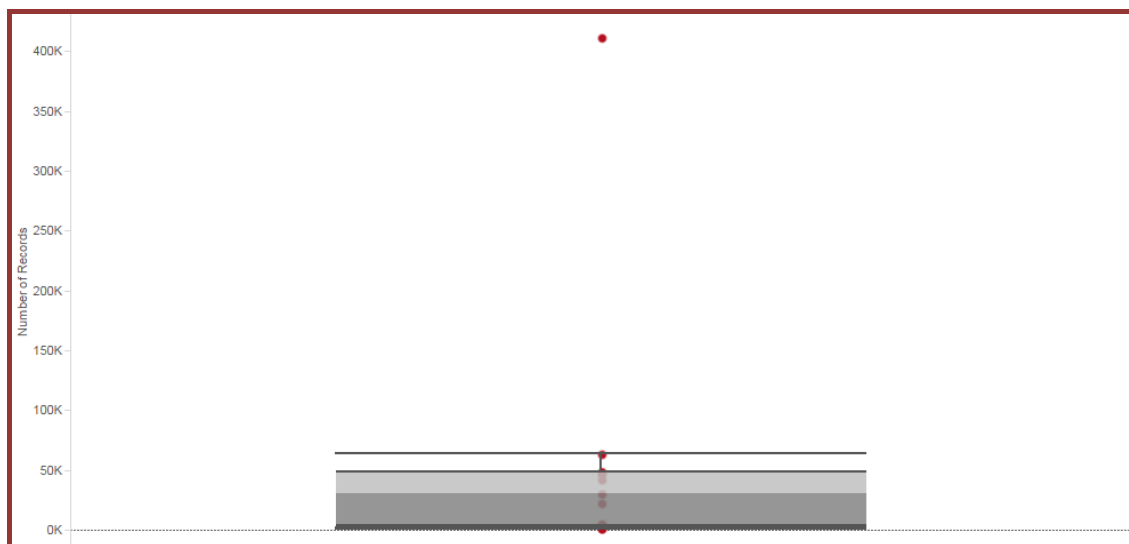
Box and whisker plot produced using Tableau shows a condensed range for the variable Book Ratings. This plot supports the histogram through the large number of zero ratings.
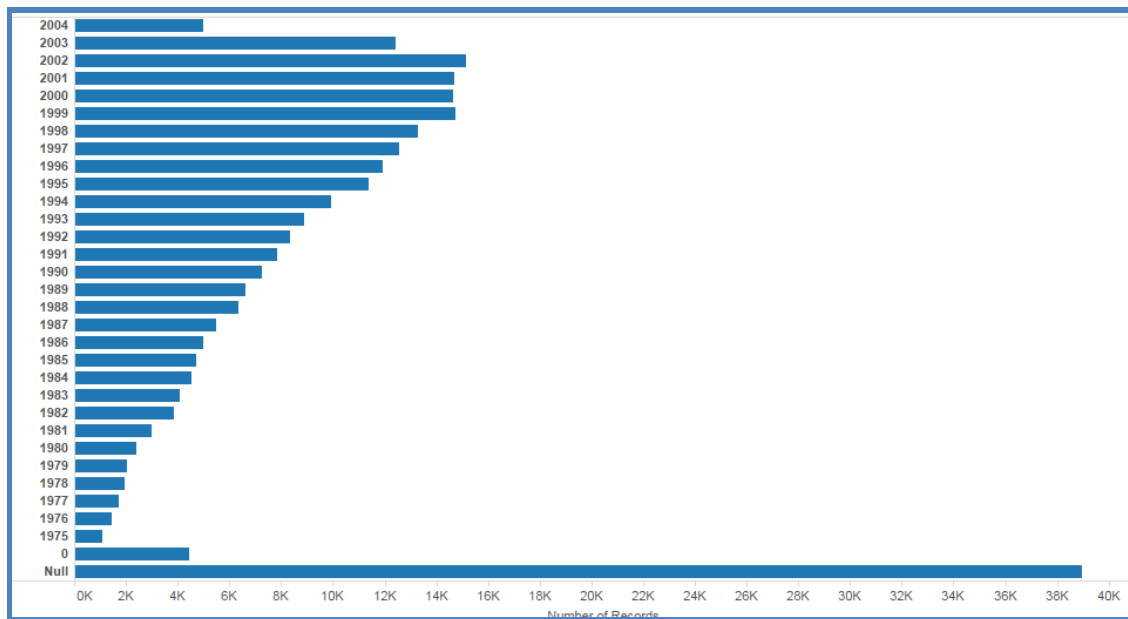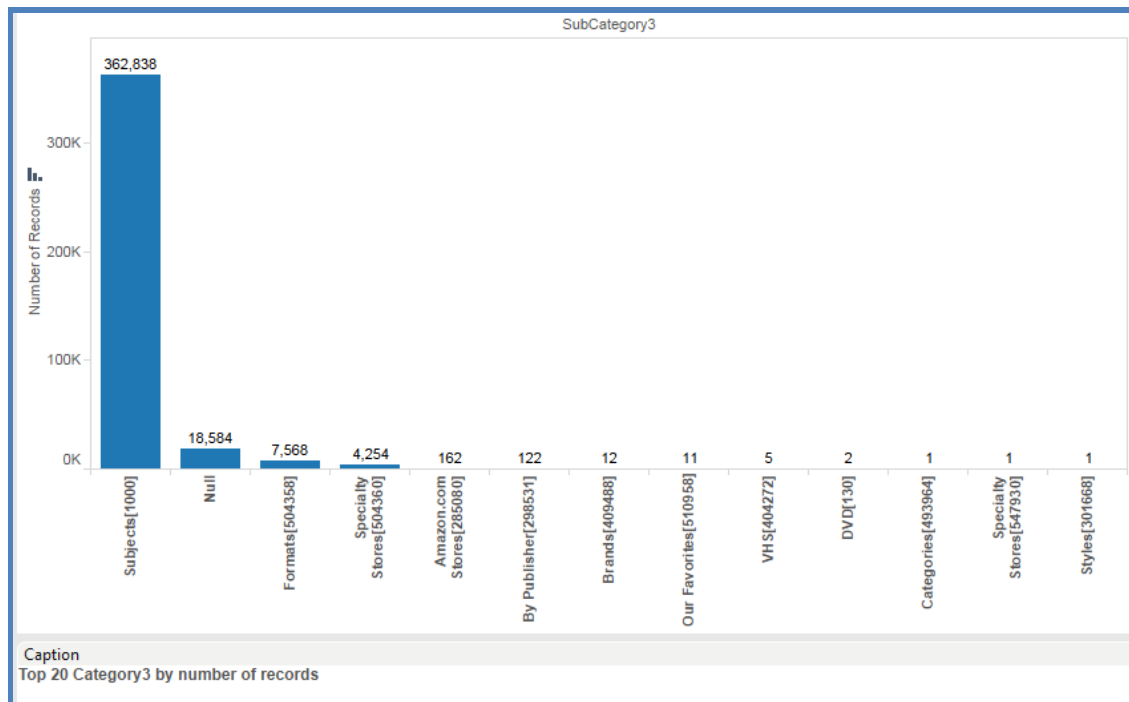
BX-Users file

Tree map produced using Tableau shows a large number of null values for Age. The bottom right illustrates a number of ages with low record counts.
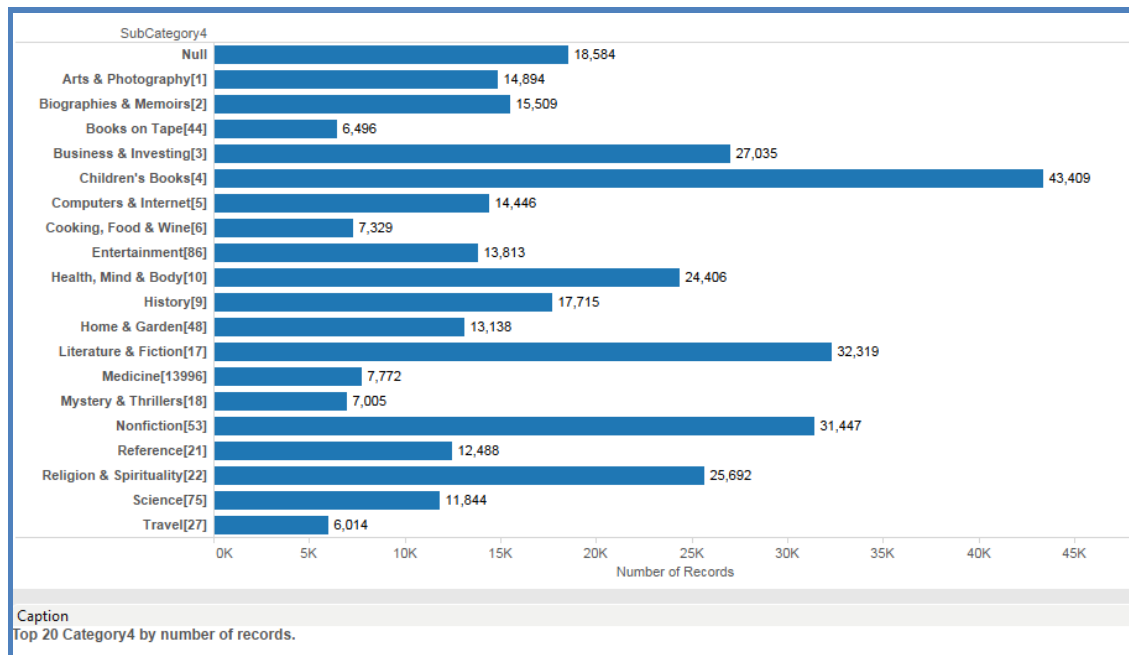
BX-Books file

Barchart produced using Tableau for the variable Year of Publication showing skewness and a large number of null values.
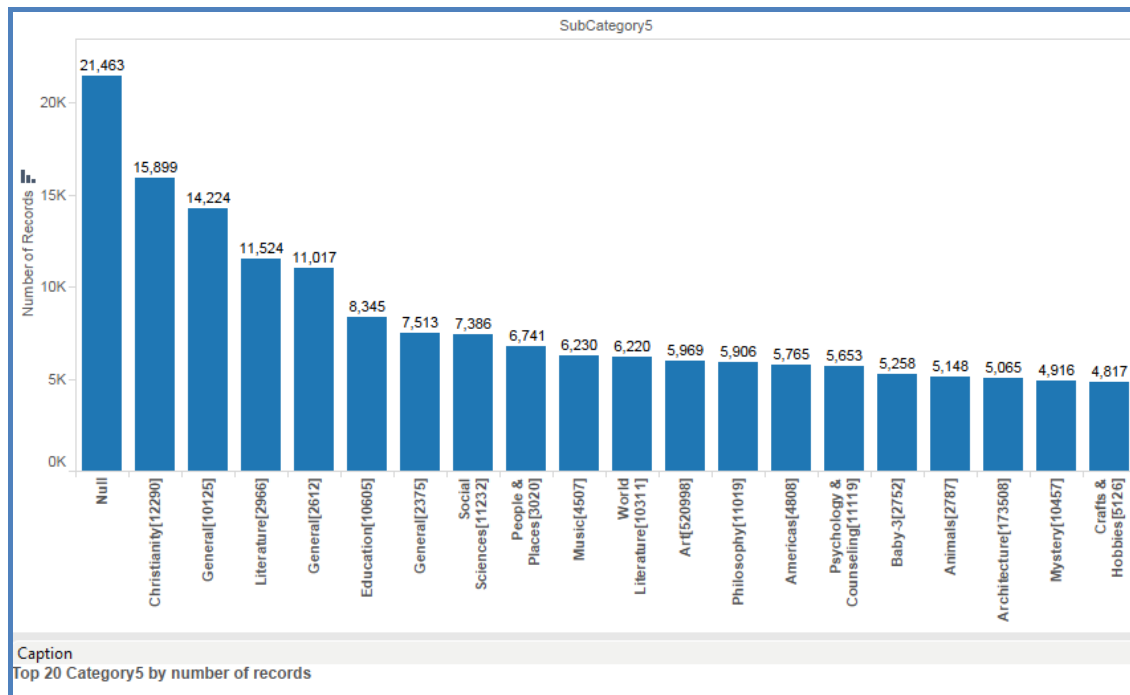
Amazon metadata file
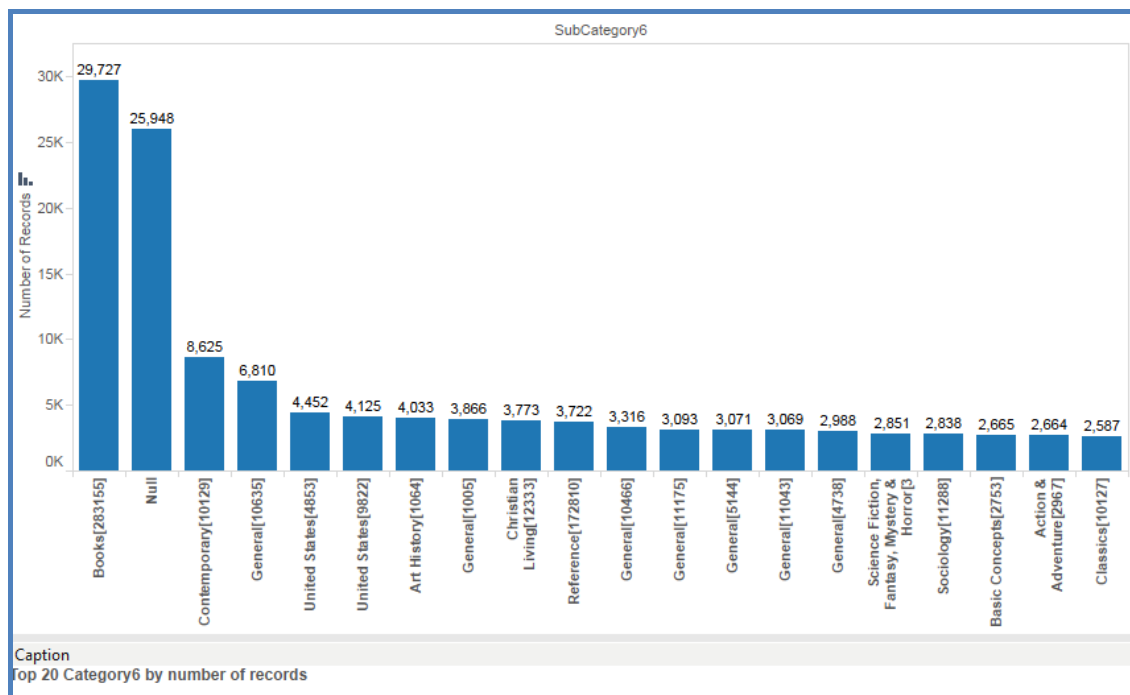

Barchart showing top 20 category3 values by number of records. This shows that subjects is the predominant category.
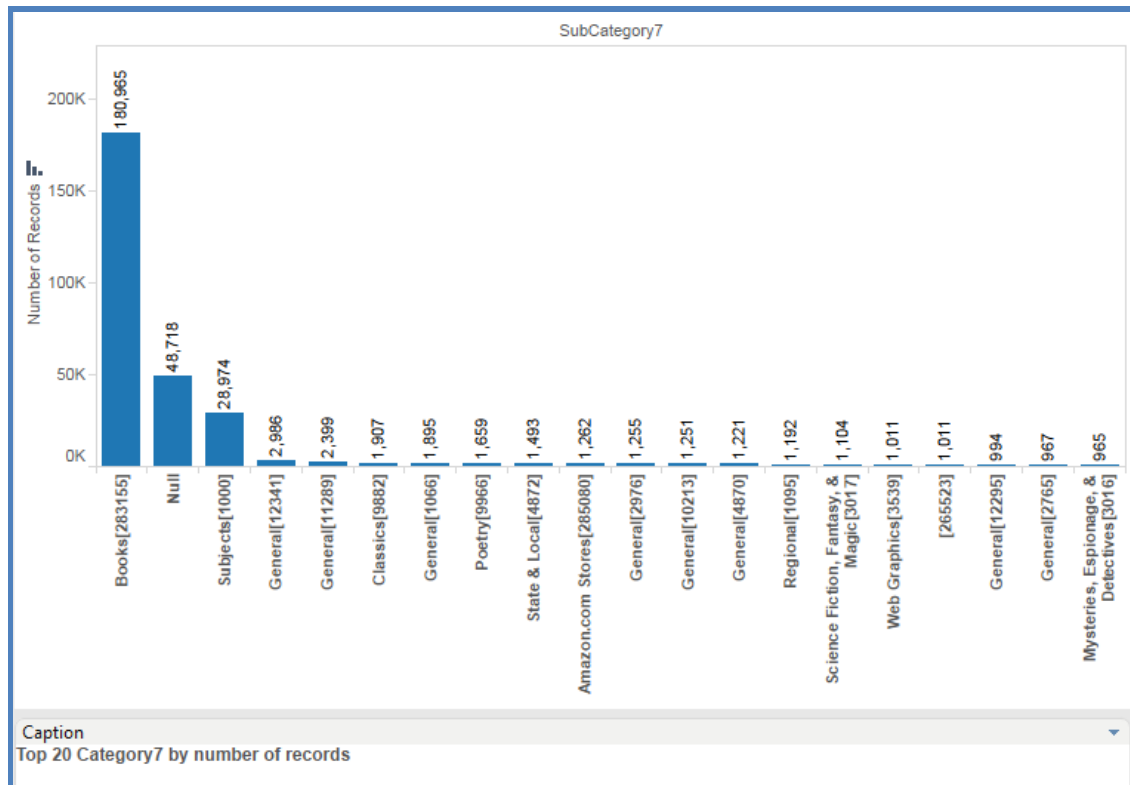

Barchart showing top 20 category4 values by number of records. This shows that Childrens Books is the predominant category. There is a high proportion of null values.
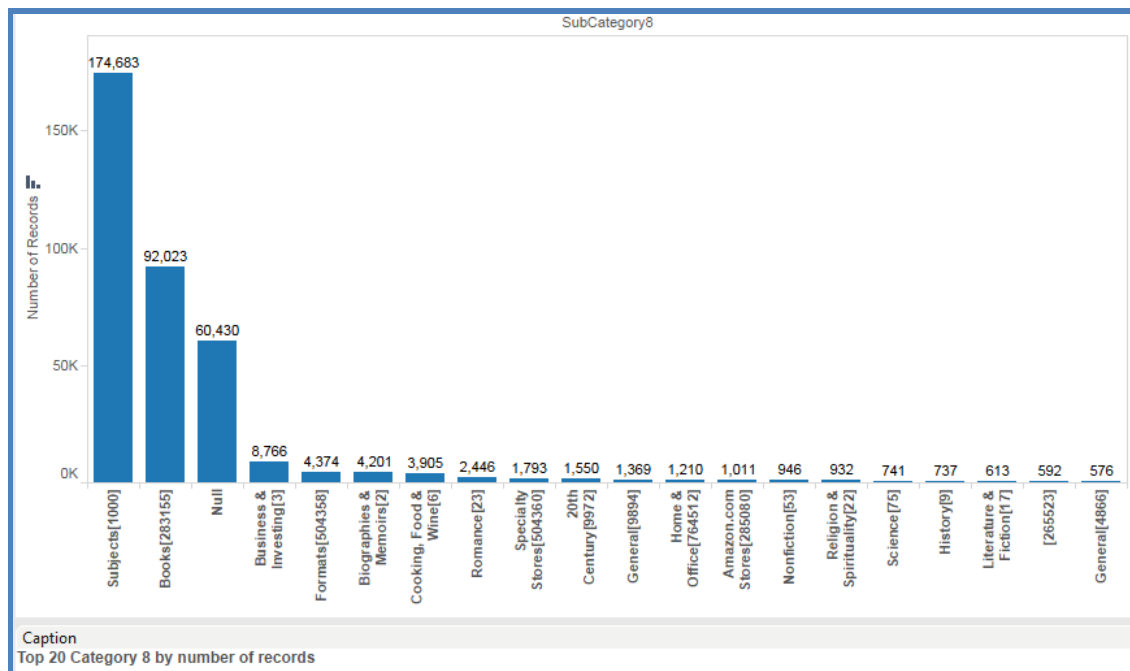
Histogram showing top 20 category5 values by number of records. Negative skewness is displayed. Null is the predominant category.

Histogram showing top 20 category6 values by number of records. This shows that Books is the predominant category closely followed by null. Negative skewness is displayed.

Histogram showing top 20 category7 values by number of records. This shows that Books is the predominant category. Negative skewness is displayed with a low spread of records across the other values.

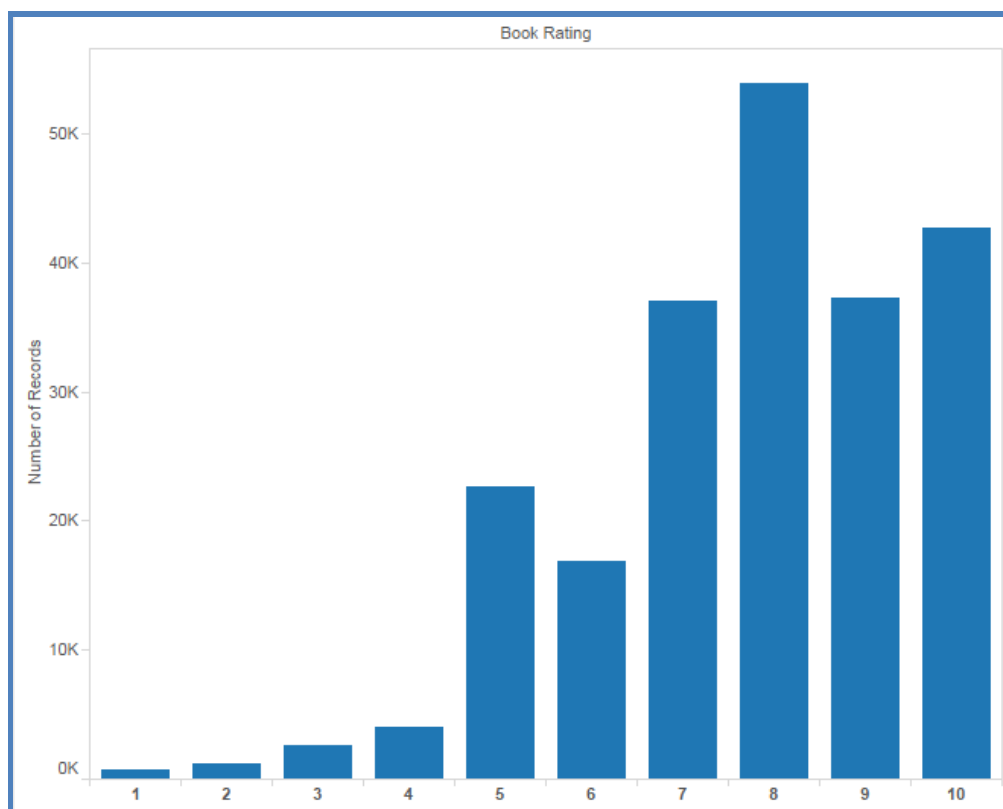Histogram showing top 20 category8 values by number of records. This shows that Subjects and Books are the predominant categories closely followed by null. Negative skewness is displayed.

Consolidated File with Diversity
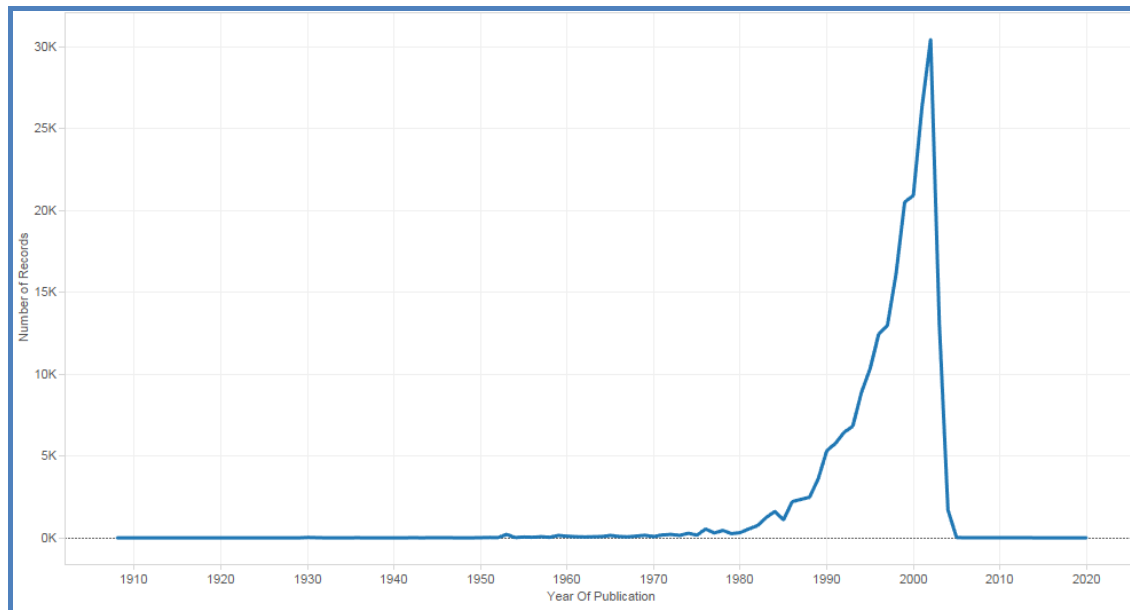
The above visualisation shows a majority of instances with a year of publication between 1990 and mid 2000's.

Histogram showing of Book Rating by number of records. Positive skewness is displayed.

APPENDIX B

Python Code utilised in the experiment.

```python
import csv
import os

sDirectory = 'C:\\pythonworkdirectory'

os.getcwd()
os.chdir(sDirectory)
sCurrent_Directory = os.getcwd()
print(sCurrent_Directory)

#Variable Declaration
sDirectory = 'C:\\pythonworkdirectory'
sFileName = '\\amazonmeta.txt'
iValidate = 0
sString = ""
sString1 = ""
sID = ""
sASIN = ""
sTitle = ""
sGroup = ""
sSalesrank = ""
sSimilar = ""
sCategories = ""
sCategoryDetail = ""
sReviews = ""
sReviewDetail = ""
sCSVstring = ""
```

```
#1 = categories
#2 = reviews


def file_import(sFileName,sDirectory):
    sObject = sDirectory + sFileName
    with open(sObject,'r',encoding="utf8") as source_file:
        imported_file = source_file.readlines()
    source_file.close()
    return imported_file


active_file=file_import(sFileName,sDirectory)


with open('amazonoutput.csv','w',newline='') as csvfile:
    container = csv.writer(csvfile, delimiter='~')


container.writerow(['^ID^~^ISBN^~^Title^~^Group^~^Salesrank^~^Similar^~^Categ
ories^~^CategoryDetail^~^Reviews^~^ReviewDetail^'])



##Putting data into csvfile
    for row in active_file:
        if("Id:" in row):
            sCSVstring                                              =
['^'+sID+'^'+'~'+'^'+sASIN+'^'+'~'+'^'+sTitle+'^'+'~'+'^'+sGroup+'^'+'~'+'^'+sSalesrank
+'^'+'~'+'^'+sSimilar+'^'+'~'+'^'+sCategories+'^'+'~'+'^'+sCategoryDetail+'^'+'~'+'^'+sR
eviews+'^'+'~'+'^'+sReviewDetail+'^']
            container.writerow([sCSVstring])

            sID = ""
            sASIN = ""
            sTitle = ""
            sGroup = ""
            sSalesrank = ""
            sSimilar = ""
```

```
        sCategories = ""
        sCategoryDetail = ""
        sReviews = ""
        sReviewDetail = ""


        sID = row.replace("Id:","")
        sID = sID.strip(" ")
        iValidate = 0
    if("ASIN:" in row):
        sASIN = row.replace("ASIN:","")
        sASIN = sASIN.strip(" ")
    if("title:" in row):
        sTitle = row.replace("title:","")
        sTitle = sTitle.strip(" ")
    if("group:" in row):
        sGroup = row.replace("group:","")
        sGroup = sGroup.strip(" ")
    if("salesrank:" in row):
        sSalesrank = row.replace("salesrank:","")
        sSalesrank = sSalesrank.strip(" ")
    if("similar:" in row):
        sSimilar = row.replace("similar:","")
        sSimilar = sSimilar.strip(" ")
    if("categories:" in row):
        sCategories = row.replace("categories:","")
        sCategories = sCategories.strip(" ")
        iValidate = 1
    if((iValidate == 1) and ("reviews:" not in row) and ("rating" not in row)):
        sString = ""
        sCategoryDetail = sCategoryDetail + "~" + row
        sCategoryDetail = sCategoryDetail.strip(" ")
    if("reviews:" in row):
        sReviews = (row)
        iValidate = 2
```

```python
        sReviewDetail = ""
    if((iValidate == 2) and ("id:" not in row)):
        sReviewDetail = sReviewDetail +  "~"  + row


  sCSVstring                                                              =
['^'+sID+'^'+'~'+'^'+sASIN+'^'+'~'+'^'+sTitle+'^'+'~'+'^'+sGroup+'^'+'~'+'^'+sSalesrank
+'^'+'~'+'^'+sSimilar+'^'+'~'+'^'+sCategories+'^'+'~'+'^'+sCategoryDetail+'^'+'~'+'^'+sR
eviews+'^'+'~'+'^'+sReviewDetail+'^']
  container.writerow([sCSVstring])


csvfile.close()
print("I'm FINISHED")
```

APPENDIX C

ACSII and special characters identified during data exploration.

Character(s)	Action taken
\n	Removed
~	Removed
;	Removed
ß	Removed
É	Replaced with E
º	Removed
Ü	Replaced with U
Ô½crosoft	Replaced with Microsoft
è	Replaced with e
>>	Removed
#	Removed
/	Removed
\	Removed
(	Removed
*	Removed
.	Removed
-	Removed
'	Removed
,	Removed
x	Replaced with X
+	Removed
!	Removed
?	Removed
.	Removed
n/a	Replaced with not applicable
£	Removed
¤	Removed
¨	Removed
©	Removed
ª	Removed
«	Removed
±	Removed
²	Removed

³	Removed
¶	Removed
¹	Removed
¼	Removed
½	Removed
¾	Removed
à á â ã ä å å	Replaced with a
æ	Replaced with ae
ç	Replaced with c
è é ê ë	Replaced with e
ì í î ï	Replaced with i
ð ò ó ô õ ö ø	Replaced with o
ñ	Replaced with n
ù ú û ü	Replaced with u
ý ÿ	Replaced with y
&	Replaced with and
>	Removed
þ	Replaced with p