OULUN YLIOPISTO
UNIVERSITY of OULU

A Novel Hybrid Recommendation System for Library Book Selection

University of Oulu
Department of Information Processing Science
Master's Thesis
Aatu Nykänen
17.12.2020

Abstract

Increasing number of books published in a year and decreasing budgets have made collection development increasingly difficult in libraries. Despite the data to help decision making being available in the library systems, the librarians have little means to utilize the data. In addition, modern key technologies, such as machine learning, that generate more value out data have not yet been utilized in the field of libraries to their full extent. This study was set to discover a way to build a recommendation system that could help librarians who are struggling with book selection process.

This thesis proposed a novel hybrid recommendation system for library book selection. The data used to build the system consisted of book metadata and book circulation data of books located in Joensuu City Library's adult fiction collection. The proposed system was based on both rule-based components and a machine learning model. The user interface for the system was build using web technologies so that the system could be used via using web browser.

The proposed recommendation system was evaluated using two different methods: automated tests and focus group methodology. The system achieved an accuracy of 79.79% and F1 score of 0.86 in automated tests. Uncertainty rate of the system was 27.87%. With these results in automated tests, the proposed system outperformed baseline machine learning models. The main suggestions that were gathered from focus group evaluation were that while the proposed system was found interesting, librarians thought it would need more features and configurability in order to be usable in real world scenarios.

Results indicate that making good quality recommendations using book metadata is challenging because the data is high dimensional categorical data by its nature. Main implications of the results are that recommendation systems in domain of library collection development should focus on data pre-processing and feature engineering. Further investigation is suggested to be carried out regarding knowledge representation.

Keywords
recommendation system, collection development, book selection, machine learning, deep learning, artificial intelligence

Supervisor
Ph.D., Postdoctoral Researcher, Umar Farooq
Ph.D., Professor, Minna Isomursu

Foreword

Some experiences in life fly past like a shooting star and other tend to take their time. The writing process of this thesis certainly belonged to the latter group. While I am happy that thesis is now finished, I am also happy that I got to experience this long journey. I wish to express my deepest gratitude to both my supervisors, Dr. Minna Isomursu and Dr. Umar Farooq, for their guidance and support that helped me to reach the finish line. I would also like to give special thanks to the participants of the focus group session for their willingness to support my thesis process by giving their precious time.

I remain thankful to many people who I have met during my life – too many to list here. Some of you I have known for years and some only a brief amount of time. Some I have had the chance to work with and with some I have had the joy to share the interest towards music and gaming. Some of you have taught me about technology, but each of you have taught me something about life. No matter when and where we have met the last time, I look forward to having a pint with you.

I am most grateful for having the best parents a child can hope for and a brother, who is not only wise but also always willing to help. Last, but certainly not the least, I want to thank the two most optimistic and energetic teachers I have the joy to have in my life, my wife and my daughter, for believing in me even during hard times.

Aatu Nykänen

Oulu, December 17, 2020

Abbreviations

API	Application Programming Interface
DSR	Design Science Research
DSRM	Design Science Research Methodology
HTTP	Hypertext Transfer Protocol
SGD	Stochastic Gradient Descent
SQL	Structured Query Language

Contents

1. Introduction

Libraries have provided both education and entertainment to their patrons for a long time. To fulfil their mission each library has had to take good care of their collection by evaluating the current state of the collection and making decisions on how to develop the collection further. This process of handling decisions related to how to develop library's collection is called collection development in the literature (Johnson, 2018, pp. 1-2). Through the years, the question of how to develop the library's collection to serve the community best has been answered in multiple different ways. Despite the different answers, one thing has remained constant: the tension between demand and value (Johnson, 2018, p. 21). Not everything can be acquired and kept in library's collection.

As time has passed the task of selecting and acquiring books has grown to be a separate position in all types of libraries (Johnson, 2018, p. 25). At the same time, the task of selecting books to be acquired has also grown to be more difficult. The number of books that are published each year has increased and libraries have had to learn how to stretch their decreasing budgets (Fieldhouse, 2012, p. 31). For example, in Finland the amount of fiction books published in a year almost doubled between the years 1985 and 2005 (Tilastokeskus, 2007). In the year 2017 the number of published fiction books in Finland surpassed the milestone of three thousand published books for the first time (Kansalliskirjasto, n.d.), which meant the number of books published in a year had tripled when compared to the year 1985. On the other hand, public libraries in Finland used around three and half million euros less to acquire library materials in 2019 than they did in 2010 which resulted in acquiring 200 000 books less (Kirjastot.fi, n.d.).

Different methods, such as developing formal collection development policy (Shaw, 2011, p. 165) and utilizing statistics (e.g. Adams & Noel, 2008), have been introduced to help librarians with collection development. Tools that provide visualizations of the data have also entered the market (e.g. Springshare, 2020). At the same time as these new ways of assisting collection development work have been discovered, usefulness of data visualizations in collection development has been acknowledged (Borrego & Levellen, 2014, p. 556). However, not all librarians have had access to these tools. Developing tools to help in collection development has not been prioritized by companies offering integrated library system solutions and librarians wishing to use collection analytics tools have had to acquire additional software to do so. As budgets in many libraries have been tight, chances are many librarians have continued to trust their own knowledge and experience in collection development instead of getting help from modern technology. In addition, some rising technologies, such as artificial intelligence and machine learning, have not yet been utilized to their full potential in the domain of libraries (Ex Libris, n.d., p. 10).

Requirements for utilizing machine learning to its full effect have been met during the past decade and lately it has become the choice of technology for many businesses (Kaggle, n.d.; MIT Technology Review, 2017). Today anyone with enough data can make use of machine learning to learn approximations of algorithms from the data they have (Alpaydin, 2014, pp. 1-4). These approximations, often referred as models, can then be used to make predictions. With this predictive power it has become possible to develop applications that are impossible to program manually (Mitchell, 2006, p. 3).

Furthermore, machine learning is not the only giant leap that has been taken in technology industry during the past decade. The advances in technology have made it possible that the act of giving recommendation is no longer limited only to a social process between persons. Instead, nowadays most people using internet services have received recommendations from a recommendation system (Jannach, Zanker, Felfernig & Friedrich, 2011, Ch. 1). Recommendation systems are computer systems that can detect dependencies between users and items and make recommendations based on these discoveries (Aggarwal, 2016, p. 2). In addition of providing help with decision-making, recommendation systems have also proven to be helpful in addressing the problem where we as humans feel overwhelmed by the amount of information. This problem is referred as the information overload problem in the literature (Good et al., 1999, p. 1).

Although recommendation systems usually recommend items to individual users, individual users are not the only one to benefit from recommendation systems: businesses also gain benefit from recommendation systems. The benefit for businesses comes from better personalization that improves both service's user satisfaction and sales (Ricci, Rokach, & Shapira, 2015, pp. 5-6). Most of today's big technology companies, such as Amazon, Google, and Netflix, utilize recommendation systems as part of their services (Aggarwal, 2016, pp. 5-7).

While machine learning and recommendation systems at their core are two different discoveries in the vast field of artificial intelligence, their relationship has grown to be a close one over the years. Use of many different types of machine learning algorithms have been investigated in domain of recommendation systems in attempt to enhance the personalization aspect of the recommendations (Portugal, Alencar & Cowan, 2018, Ch. 4). In today's reality, it is often a machine learning model that is making the recommendation behind the curtain (e.g. Covington, Adams & Sargin, 2016).

In domain of library and information science machine learning and recommendation system solutions have been applied to some extent. Some of these applications focus on making library processes more efficient (see e.g. Lyngsoe Systems, 2019; Suominen, 2019; Wagstaff & Liu, 2018) and other focus more on improving user satisfaction (e.g. Yelton, 2018). Yet, considering the generally acknowledged potential of using these methods relatively few academic libraries have been actively engaging on artificial intelligence projects as reported by Wheatley and Hervieux (2019). Based on the lack of reports and whitepapers, the situation is similar with public libraries.

The reason behind challenges in adopting the artificial intelligence technology might not be lack of interest as pointed out by Padilla (2019, p. 9). This hypothesis is further supported by the observation that solutions based on data analytics seem to be on high demand in libraries (Library Journal, 2018, p. 39). According to Ex Libris (n.d.) the challenges in adopting artificial intelligence solutions in libraries include cost of the solutions, library staff's resistance to adopt new technologies and librarians fear of artificial intelligence replacing human workers. In addition, not all desired solutions yet exist.

1.1 Research problem

While some of collection development processes have been explored from machine learning perspective (see e.g. Baba, Minami, & Nakatoh, 2016; Iqbal, Jamil, Ahmad, & Kim, 2020; Wagstaff & Liu, 2018), no solution has been introduced to assist librarians in the book selection process. Furthermore, the machine learning assisted collection development solutions have mainly been investigated in domain of academic libraries (see e.g. Baba et al., 2016; Iqbal et al., 2020) and no research have been carried out in context of public libraries. Book recommendation systems on the other hand have been studied extensively from perspective of recommending books to individual users (see e.g. Ali, Khusro, & Ullah, 2016; Hammais, Ketamo & Koivisto, 2019; Okon, Eke, & Asagba, 2018; Rana & Jain, 2012), but research regarding the perspective of making book selection recommendations to organizations, such as libraries, has not had the same popularity.

Identification of this clear research gap regarding recommendation systems that assist librarians in collection development, particularly in the book selection process, was the starting point for this thesis. The main task in this thesis was set to be to answer one research question:

How to implement a recommendation system that can assist in book selection in context of public libraries print collections?

To answer the research question a novel software solution that can assist librarians in the book selection process was implemented. Design science research guidelines were followed during the implementation. After the implementation was finished, the produced artifact was evaluated. The evaluation consisted of automated tests and user evaluation that was carried out using focus group methodology.

1.2 Contributions of the research

Contributions of this thesis followed closely the goal defined by Hevner, March, Park and Ram (2004, p. 80): the aim for this thesis was to produce utility. The main contribution of this thesis was the proposed hybrid recommendation system. For purpose of constructing the system, six different machine learning models were compared, and a set of heuristics were designed. These approaches were then combined to form a novel hybrid recommendation system for book selection. A set of features was determined with help of feature engineering to determine what features could be used by the overall system. After the implementation was done, the system was published as an open source tool[1] so that it could be used by libraries or extended further by software developers and researchers.

In addition of producing a recommendation system as the output artifact, knowledge contributions were gained from evaluating the artifact and reflecting the results of this study against results from prior research. These knowledge contributions aimed to contribute to both future research and business applications as defined by Hevner et al. (2004, pp. 79-81). As artificial intelligence and machine learning had not yet been extensively studied from perspective of public libraries during the time when this thesis was written, it was considered that both organizations and researchers could gather important insights from this thesis.

1.3 Thesis structure

This thesis consists of seven chapters of which each serve a unique purpose. The thesis began with this chapter, Chapter 1, which introduced the research problem. In Chapter 2

[1] https://github.com/aanykanen/selection-recommender

the prior research related to recommendation systems, machine learning and library collection development is portrayed. The foundations for understanding the proposed novel solution are laid down here. Chapter 3 then continues to present the design science research methodology that was used for finding the solution to the research question. The implementation of the novel software solution is discussed in detail in Chapter 4. Chapter 5 presents the results of this research and in Chapter 6 the results are discussed together with the limitations of the research and suggestions for future research.

2. Prior research

A recommendation system uses different criteria to suggest an entity, object, person or product. In this thesis both rule-based approach and machine learning based approach were investigated to develop a hybrid recommendation system. The relevant literature regarding the topic of this thesis is divided into three categories: machine learning, recommendation systems and collection development. Since recommendation systems often utilize machine learning models, Section 2.1 first introduces prior research regarding machine learning. After this the prior research regarding recommendation systems is presented in Section 2.2. Finally, Section 2.3 summarizes the parts relevant to this thesis regarding preceding research about collection development in libraries.

2.1 Machine learning

Machine learning is a subfield of artificial intelligence that is concerned with how computer systems can learn and adapt to changes automatically (Alpaydin, 2014, pp. 1-4; Mitchell, 2006, p. 1). In addition to being a subfield of artificial intelligence, machine learning also shares a relationship with fields such as statistics (Mitchell, 2006, p. 1; Mohammed, Khan & Bashier, 2017, Ch. 1.2.1). No universally accepted formal definition of machine learning exists, but following definition by Mitchell (1997, p. 2) is often quoted in the community:

> *"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"*

Series of ideas and discoveries that led to the creation of machine learning discipline started in 1950s. One of the most famous papers from the early times of the discipline was written by Alan Turing (1950) who not only introduced the concept of intelligent machine that is capable of learning, but also coined a wish that *"machines will eventually compete with men in all purely intellectual fields"* (Turing, 1950, p. 460). The advances in the field of machine learning were ground-breaking in 1950s and some innovations heavily utilized nowadays, like Perceptron (Rosenblatt, 1958), were discovered during this time.

There exist three main learning paradigms in machine learning under which learning algorithms are commonly categorized: supervised learning, unsupervised learning, and reinforcement learning (Alpaydin, 2016, pp. 38-39, 111, 125-128). While categorization of a learning algorithm in most cases is simple, Goodfellow, Bengio and Courcille (2016, pp. 103-104) remind that the line between supervised learning and unsupervised learning may not always be clear. In addition to these three main learning paradigms the subfield of deep learning, which focuses on utilizing neural networks, is often discussed as its own topic (see e.g. Alpaydin, 2016, pp. 104-109; Goodfellow et al., 2016). Next, some important terminology that is universal for all the different machine learning methods is presented.

Algorithm in context of machine learning commonly refers to the learning algorithm that is the method of learning (Perrier, 2017, p. 3). In general sense definition of an algorithm is often referred as *"a sequence of instructions that should be carried out to transform the input to output"* as it is defined by Alpaydin (2014, p. 2). In this thesis when the term algorithm is used in context of machine learning it refers to the learning algorithm that is used to train the machine learning model.

Dataset or *data set* commonly refers to the data that is made available to the learner. It consists of individual examples. Dataset can be considered to be a synonym to the term experience presented in Mitchell's (1997, p. 2) definition. As dataset serves as the source for the learning process, its quality and size has a great effect to how successfully the learner can make predictions after learning (Mohri, Rostamizadeh & Talwalkar, 2012, p. 1).

Features are attributes that individual examples have (Mohri et al., 2012, p. 3). They are often represented in vector form (Mohri et al., 2012, p. 3). Features can be either defined manually or learned automatically from the raw data using techniques such as deep learning (Goodfellow et al., 2016, pp. 3-6).

Model is the output of the learning process that is trained by the learning algorithm with the help of dataset (Perrier, 2017, p. 31). It can be either predictive, descriptive or both (Alpaydin, 2014, p. 3). If the model is predictive, it can make predictions of output when given an input, and in case the model is descriptive it knows how to obtain information from data (Alpaydin, 2014, p. 3). Synonyms for the term model in the machine learning literature include for example terms predictor, hypothesis, and classifier (Shalev-Shwartz & Ben-David, 2014, p. 34).

2.1.1 Supervised learning

Supervised learning is one of the three main learning paradigms in machine learning (Alpaydin, 2016, pp. 38-39). In supervised learning a model is learned from labeled data - i.e. data which consists of a pair including both a sample and label (Goodfellow, et al., 2016, p. 103). The label for each example is provided by a supervisor that may be either human or machine (Mohammed et al., 2017, Ch. 1.2.2). An example of labeled data is a photo with information associated to it that in the photo (sample) there is a dog (label).

Depending on the type of target value the learning task is categorized to be either classification or regression (Russell & Norvig, 2010, p. 696). An example of a classification problem would be to identify whether there is a dog or cat in a picture. An example of a regression problem on the other hand would be predicting price of an item, like car or house (see e.g. Alpaydin, 2016, pp. 29-31). The difference between these two problem types is that in the regression problem the label is not be restricted to be found from a finite set like in the classification problem. Since the number of plausible output values is multiple times greater in regression problems, regression problems are more complex to solve than classification problems (García, Luengo & Herrera, 2015, p. 7).

Results of a survey conducted by Kaggle (n.d.) show that using supervised learning methods, such as linear regression and decision trees, is popular in businesses. It is also generally acknowledged that supervised learning is the most common form of machine learning (see e,g, Kelleher, 2019, Ch. 1; LeCun, Bengio & Hinton, 2015, p. 436). However, the overall popularity of supervised learning over other machine learning paradigms is not supported by the search popularity data obtained from Google Trends as shown in Figure 1. The trend data suggests that especially reinforcement learning has generated a lot more interest in searches than supervised learning in the latter half of 2010s.

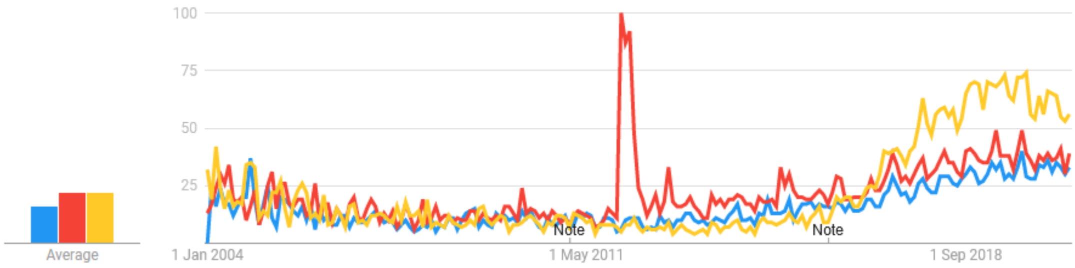

Figure 1. Interest towards different machine learning paradigms over time. Data source: Google Trends (https://www.google.com/trends). Retrieved August 23rd, 2020.

While supervised learning can offer a solution to multiple different types of problems, it comes with its own set of challenges. For example, some domains or problems may favor use of other machine learning paradigm: most of the state-of-art implementations in teaching computer to play video games have been achieved using reinforcement learning (e.g. Badia et al., 2020; Mnih et al., 2013). In addition, supervised learning suffers from challenges like the high cost of labeled data (Zhou, 2018, p. 44) and class-label noise (Kubat, 2015, Ch. 1.5). Techniques such as data augmentation (Krizhevsky, Sutskever & Hinton, 2012, pp. 5-6), semi-supervised learning (Kingma, Mohamed, Rezende & Welling, 2014) and generative adversarial networks (Antoniou, Storkey, & Edwards, 2018) have been discovered to solve the difficulty of gathering a large and good quality labeled dataset which is needed for supervised learning.

While there exists vast amount of different supervised learning techniques that can be used to train a model capable of solving a classification problem, only few of them have been picked to be tested in this thesis. The supervised learning techniques that are of interest in this thesis are following:

Decision trees are models that can be used to predict label by traveling down a tree-like structure that is constructed during the learning phase (Kubat, 2017, p. 113; Shalev-Shwartz & Ben-David, 2014, p. 250). The nodes in the structure define which path each example will take, and they are constructed based on either features of examples in training data or predefined rules (Shalev-Shwartz & Ben-David, 2014, p. 250). The main benefit of using decision tree models is that they can be displayed as a graph and because of this they are easy to understand (Shalev-Shwartz & Ben-David, 2014, p. 250). The major drawback of using decision trees is that decision trees tend to be unstable: a small change in training data may lead to a decision tree that makes very different predictions (Kuncheva, 2014, Ch. 2.2.7).

Support vector machine (SVM) is a well-established classification technique that performs particularly well in high-dimensional feature spaces (Mohri et al., 2012, p. 63; Shalev-Shwartz & Ben-David, 2014, p. 255). The mechanism SVMs uses for classification purposes of linearly separable data is by finding a hyperplane that separates the two classes (Mohri et al., 2012, pp. 63-65). The hyperplane is searched in such way that the margin, i.e. the distance from closest points of both classes to the hyperplane, is maximized (Kubat, 2017, pp. 85-86). This maximization of the margin helps the model to generalize. SVMs have been widely applied to many domains and some consider it to be one of the best available supervised learning techniques (see e.g.

Alpaydin, 2014, p. 381). Despite the praise, there are also challenges. For example, SVMs are known to have high algorithmic complexity and with large datasets training a model using SVM can take a long time (Cervantes, Garcia-Lamont, Rodríguez-Mazahua, & Lopez, 2020, pp. 195-197).

Random forest is an ensemble learning algorithm that consists of a collection of decision trees (Shalev-Shwartz & Ben-David, 2014, p. 255). The main idea behind the random forest algorithm is to introduce randomized data to the decision tree training and afterwards utilize the majority vote of the decision trees when making predictions (Shalev-Shwartz & Ben-David, 2014, p. 255). The benefit of this method is that it improves accuracy when compared to using one decision tree (Alpaydin, 2014, pp. 234-235).

Gradient boosting machine (GBM) is another ensemble learning method that seeks to convert weak learners into strong learners by constructing a sequential pipeline of weak learners (Natekin & Knoll, 2013, p. 1). The training process is iterative and consists of training new weak learners to find the best performing ensemble (Natekin & Knoll, 2013, p. 1). As the learners are placed in a sequential order the training result of previous learner in the sequence affects the training of the next one. Multiple different implementations of the GBM algorithm, such as Xgboost (Chen & Guestrin, 2016) and LightGBM (Ke et al., 2017), have been presented over the years.

2.1.2 Deep learning

Deep learning is a subfield of machine learning and representation learning which utilizes neural networks to learn how to build complex concepts by utilizing simpler concepts (Goodfellow et al., 2016, p. 5). The foundations for deep learning were discovered as early as 1940s but the modern era of deep learning can be seen starting in 2000s (Kelleher, 2019, Ch. 4). The major benefit of deep learning techniques is that they are able to find hidden structures even from high-dimensional raw data whereas conventional machine learning techniques can struggle when presented with similar data (LeCun et al., 2015, p. 436). A typical example of deep learning model is a feedforward neural network, which consists of input layer, output layer and selected number of hidden layers (Goodfellow et al., 2016, pp. 164-165). Each hidden layer consists of selected number of neurons (Kelleher, 2019, Ch. 1). Example of this type of architecture is pictured in Figure 2.

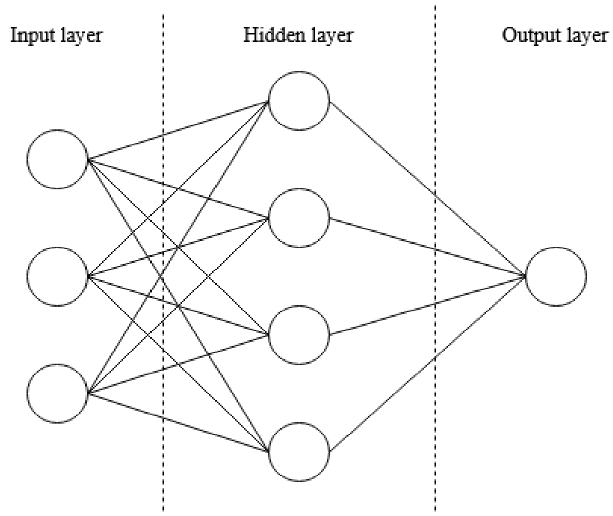

Figure 2. Feedforward neural network with a single hidden layer (adapted from Goodfellow et al., 2016, p. 170).

The working principle of feedforward neural networks is that each neuron located in hidden layer has a set of weights and an activation function associated with it (Kelleher, 2019, Ch. 3). The process of transferring the input to an output is a two-stage process where the input is first multiplied with weights and afterwards the sum of these calculations is given as an input to an activation function (Kelleher, 2019, Ch. 3). While there exist multiple different activation functions, the rectified linear activation, which limits the neurons output to a lower limit of zero, is considered to be the recommended activation function when constructing feedforward neural networks (Goodfellow et al., 2014, p. 170).

Deep learning can be used together with all the main learning paradigms of machine learning (Kelleher, 2019, Ch. 1). In addition to contributing to advancements in machine learning applications such as generative adversarial networks (Goodfellow et al., 2014), deep learning has also contributed to other advancements. For example, utilization of parallel computing and graphics processing units in machine learning has become a new standard (Jordan & Mitchell, 2015, p. 257) since training of deep neural networks can be effectively computationally parallelized.

2.1.3 Training a machine learning model

Training a machine learning model is a task of iterative nature that consists of multiple steps as pictured in Figure 3. Often the first step in training machine learning model is to define a problem and collect an appropriate dataset (Kotsiantis, Zaharakis, & Pintelas, 2007, p. 250). In some cases the data may also be obtained during the training through gaining feedback from an environment (Russell & Norvig, 2010, p. 830). No matter how the data is obtained, after obtaining the data it needs to be prepared and pre-processed before it is used for training the model (Kotsiantis et al., 2007, p. 250).

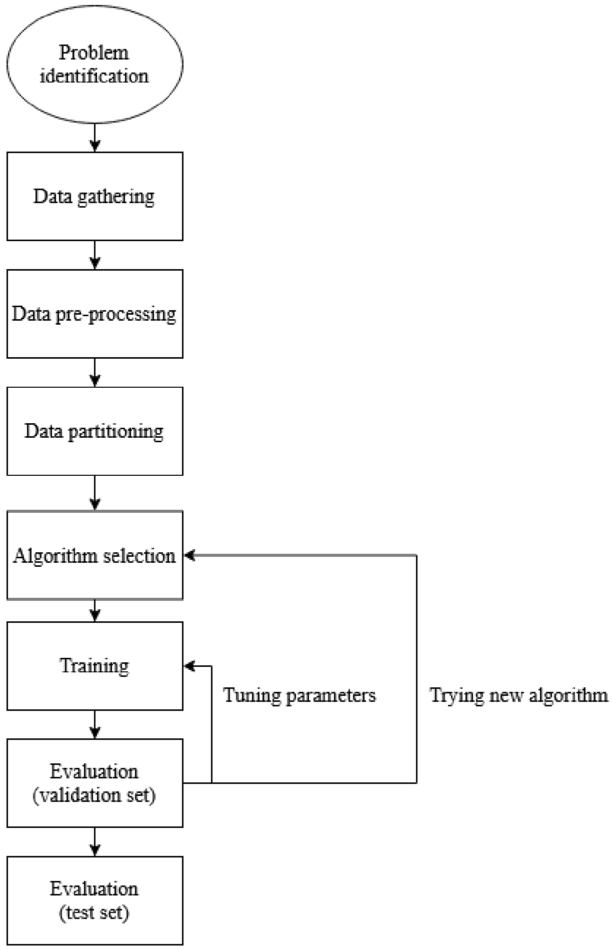

Figure 3. Process of training a machine learning model (adapted from Kotsiantis et al., 2007, p. 250).

After preprocessing the dataset needs to be partitioned. The general rule when training a machine learning model is to partition the dataset to three different parts: training set, validation set and test set (Alpaydin, 2014, p. 556). Having a separate test set allows producing metrics of how well the trained model can generalize after the model has been trained (Russell & Norvig, 2010, p. 695). Since the aim of the training is to learn to generalize, having a separate test set that is not used at all during training phase is important regarding model evaluation. Validation set on the other hand gives information about how well the model is generalizing during the training and allows choosing the best model out of all available options (Alpaydin, 2014, p. 40).

Before the data can be used as input for machine learning algorithm it needs to be properly formatted. One common technique that is used for encoding categorical variables to numerical format is one-hot encoding (Guo & Berkhahn, 2016, p. 1). The one-hot encoding however suffers from inability to represent relationships between values and thus techniques like entity embeddings, which maps categorical values to a pre-defined number of dimensions, have also been introduced (Guo & Berkhahn, 2016). If the data is numerical by nature, the data formatting stage may include steps like normalization to scale down the maximum and minimum values (Kotsiantis, Kanellopoulos & Pintelas, 2006, pp. 113-114).

Once the training dataset is transformed so that it can be used as an input for machine learning algorithm, the process of training a machine learning model continues to algorithm selection phase (Kotsiantis et al., 2007, pp. 250-251). During this phase different algorithms are compared to each other to find the algorithm that outputs the best performing model (Raschka, 2018, p. 4). After finding the best algorithm for learning, the algorithm's hyperparameters are tuned. Hyperparameters are configurations used for the learning algorithm which may affect the result in major way (Claesen & De Moor, 2015). When the hyperparameter tuning is finished the model is ready for final evaluation using the test set.

2.1.4 Challenges in machine learning

While machine learning has provided a solution to many previously unsolved problems, there are also challenges that need to be addressed when using machine learning methods to solve a problem. Some of commonly acknowledged challenges are following.

High dimensional data is problematic for two main reasons: a lot of computational resources are needed for model training and the learnability is not guaranteed (Kubat, 2017, pp. 204-205).

Imbalanced training set is a training set where number of examples between classes differ drastically (Kubat, 2017, pp. 194-198). For example, a training set consisting of hundred fiction books and one non-fiction book would be an imbalanced training set that could lead to underperforming in classifying non-fiction books.

Overfitting is a problem, where the machine learning model learns to remember the training dataset and is incapable of making generalized predictions (Dietterich, 1995).

Poor interpretability is a common problem with machine learning models and means that humans are not able to understand the reason behind the predictions made by the machine learning model (Rudin, 2019). This problem is referred also as the *black box problem* in the machine learning and artificial intelligence literature (e.g. Zednik, 2019). While it has been suggested by Doshi-Velez and Kim (2017, p. 3) that interpretability is not required for all machine learning systems, in some domains lack of interpretability has already had catastrophic consequences (see Rudin, 2019, p. 1).

Underfitting is an opposite problem to overfitting (Jabbar & Khan, 2015, p. 165). In this situation the model is unable to make good predictions to either training data or to data not included in training set.

2.2 Recommendation systems

Recommendation systems are type of information filtering systems targeted specifically to tackle the information overload problem (Good et al., 1999, p. 1; Khusro, Ali & Ullah, 2016, p. 1179). They generate and provide meaningful suggestions for items to a user (Good et al., 1999, p. 1; Khusro et al., 2016, p. 1180). While the first recommendation system, Tapestry, was designed to work with electronic documents that arrive in a continuous stream (Goldberg, Nichols, Oki & Terry, 1992), nowadays recommendation systems are able suggest items of various types. Some common item types that recommendation system research has focused on are music (Schedl, Zamani, Chen, Deldjoo, & Elahi, 2018), movies (e.g., Azaria et al., 2013; Li & Yamada, 2004) and e-commerce (Schafer, Konstan & Riedl, 1999).

In addition to the research interest, services have also identified the potential of recommendation systems. Many service providers have implemented a recommendation system to accompany their product (Table 1). In pursue of gaining benefits by developing better recommendation systems some companies have even organized competitions. The top prize in these competitions has reached to as high as one million dollars (Netflix Inc., 2009). Benefits for service providers who choose to implement a recommendation system have been also identified in the research. According to Ricci et al. (2015, pp. 5-6) these benefits include increasing the number of items sold, selling more diverse items, increasing user satisfaction, increasing user fidelity, and increasing understanding of users' needs.

Table 1. Services that have implemented a recommendation system.

Service	Item type	Source
Netflix	Streaming video	Gomez-Uribe and Hunt (2016)
Google News	News article	Das, Datar, Garg and Rajaram (2007)
LinkedIn	Job candidate	Geyik et al. (2018)
Steam	Video game	The Steam Team (2019)

2.2.1 Types of recommendation systems

Recommendation systems come in many shapes and forms. One commonly used taxonomy is by Burke (2007) which categorizes recommendation system approaches to five different classes: collaborative, content-based, demographic, knowledge-based and hybrid recommendation systems. However, the line between these categories is not always clear. For example, according to Jannach et al. (2011, Ch. 3) the border between content-based and knowledge-based methods is not exact in literature. Figure 4 describes the types and requirements for recommendation systems as presented by Burke (2007). Following sections introduce the categories in more detail.

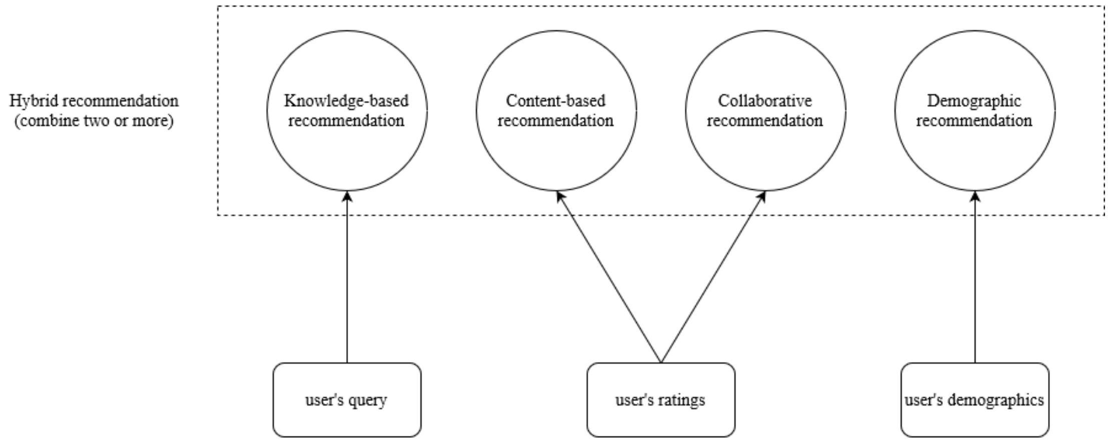

Figure 4. Types of recommendation systems (adapted from Burke, 2007, p. 379).

Collaborative filtering methods

Collaborative filtering was the first recommendation system approach introduced back in 1992 by Goldberg et al. (1992). The approach relies on core assumption that like-minded users have similar preferences (Burke, 2007, p. 378; Sarwar, Karypis, Konstan & Riedl, 2001, pp. 286-287). Recommendation systems utilizing collaborative filtering method in its purest form can make recommendations only based on rating histories and they do not need to know anything about items themselves (Jannach et al., 2011, Ch. 1.1.1). For example, if Alice and Bob shared similar rating history among food items, according to early collaborative filtering methods Alice would receive recommendations about items that she has not rated yet, but Bob has already rated highly.

Collaborative filtering methods offer good amount of customization options. For example, in addition of using an explicit statement of users' opinion (rating score), collaborative filtering methods can also utilize implicitly derived information from data like purchase records (Sarwar et al., 2001, p. 287). Also, in addition to user similarity information, similarity between two items can also be used as base of recommendation as introduced by Sarwar et al. (2001). Collaborative filtering approach also allows the target of recommendation to be a group instead of individual user (Schafer, Frankowski, Herlocker & Sen, 2007, p. 297). The collaborative filtering approach can be further divided into two categories: memory-based collaborative filtering methods and model-based collaborative filtering methods.

Memory-based collaborative filtering methods, also called neighborhood-based methods, memorize, and analyze all user and item data in continuous manner to produce recommendations (Drachsler, Hummel, & Koper, 2007, p. 21). They are often used in solving problems like predicting user ratings for an item (Breese, Heckerman & Kadie, 1998, p. 44). According to Yang, Wu, Zheng, Wang, and Lei (2016, p. 3277) memory-based collaborative filtering methods are used commonly because of their high-effectiveness and easy-implementation. Aggarwal (2016, p. 9) also mentions that the recommendations made using memory-based approach are easy to explain which in turn according to Zhang and Chen (2018, p. 5) has beneficial impact on user satisfaction and trustworthiness of the recommendation system. The two subclasses of memory-based collaborative filtering methods are user-based collaborative filtering and item-based collaborative filtering (Sharma, Gopalani & Meena, 2017, p. 2).

User-based collaborative filtering methods identify the nearest neighbors of the user and calculates prediction based on the neighbor values (Robillard, Maalej, Walker & Zimmermann, 2014, p. 17). According to Schafer et al. (2007, p. 303) these methods can be compared to word-of-mouth recommendation sharing. For example, considering data in Table 2 it can be observed that Alice and Bob share a similar type of preference regarding book genres. Since Bob also likes Sci-Fi books but Alice has not yet read them, Sci-Fi books could be recommended to Alice based on the similarity of their ratings of other genres. However, Bob might not be the only closest neighbor of Alice to be analyzed as the method allows k-value (i.e. the value which indicates how many neighbors to evaluate) to also be larger than one (Robillard et al., 2014, p. 17). In the example if k-value was 3, it would mean adding James's and John's (other two nearest neighbors of Alice) ratings of Sci-Fi to the equation and calculating mean of ratings given by Bob, James, and John. Since the mean in this scenario would be ~0.67, the system would still end up recommending Sci-Fi books to Alice.

Table 2. Example of user-item matrix with binary ratings (0=dislike, 1=like).

	Horror	Romance	Fantasy	Mystery	Sci-Fi
Alice	1	1	1	1	
Bob	1	1	1	1	1
James		0		1	1
Mary	0	1	1	0	
John	1	1		0	0
Jennifer		0	0		
Susan	0	0	0	0	

As opposed to user-based collaborative filtering method, item-based collaborative filtering method focuses on finding similar items rather than similar users (Dou, Yang & Deng, 2016). By using the item similarity together with user's history information, it is possible to solve problems like predicting users rating for unrated items (Schafer et al., 2007, p. 304). In item-based approach it could be analyzed from Table 2 that users that like romance books also often like fantasy books and users who do not like fantasy books also tend to not like romance books. Based on this information and John's own ratings, it would be appropriate to recommend fantasy books to John since he has liked romance books. An important difference between user-based and item-based approaches is that in user-based approach the recommendation is based on ratings given by similar users and in item-based method the recommendation in based on the user's own ratings (Aggarwal, 2016, pp. 29-30).

While both memory-based approaches of collaborative filtering methods can be powerful tools when building personalized recommendations, there are also disadvantages when using memory-based methods. Since all ratings must be included into calculations, memory-based methods are susceptible to scalability issues (Sharma, et al., 2017, p. 2; Dou et al., 2016, Ch. 3B). Sparsity of data can also prove to be an issue when using memory-based methods (Aggarwal, 2016, p. 9) and memory-based collaborative filtering methods also suffer from so called cold-start problem (Schafer et al., 2007, pp. 311-312). These problems are described further in Section 2.2.2 which discusses the challenges in recommendation systems.

Compared to the memory-based collaborative filtering approach that tries to calculate the optimal item to be recommended from the user-item matrix, *model-based* collaborative filtering methods try to generate a model from available data and use it for making predictions (Breese et al., 1998, p. 44). Typically, the model is generated with help of supervised or unsupervised machine learning methods (Aggarwal, 2016, p. 71). The commonly used algorithms for constructing the model include Bayesian methods, latent factor models and rule-based models among others (Aggarwal, 2016, p. 11; see also Jannach et al., 2011, Ch. 2.4). For example, given the data in Table 2 it could be modeled that Jennifer either does not like to read books at all or optionally maybe prefers to read only horror and mystery books. One of these models could be the correct one or both could be wrong: Jennifer might only like to read Sci-Fi books.

One of the greatest differences between memory-based and model-based methods is that model-based methods require model training phase before they can be used to make predictions (Aggarwal, 2016, p. 71). While the training phase requires extra time and computational resources compared to using memory-based techniques, model-based methods use less memory and make predictions faster than memory-based methods after the training has finished (Breese et al., 1998, p. 44). Model-based methods are also more resistant to overfitting (Aggarwal, 2016, p. 73).

While the model-based methods are easier to scale up than memory-based methods and do perform well, the flexibility when adding in new data is a big issue when comparing to memory-based methods. When data is handled in-memory it automatically is up to date after each change and recommendations can be made based on the latest information without any issue. However, with model-based methods a model training session is required after every change in data to make up-to-date recommendations. This requirement of training new model after each update is a problem in real-world situations as pointed out by Rendle and Schmidt-Thieme (2008, p. 251).

Content-based filtering methods

While collaborative filtering methods focus solely on the rating history, content-based filtering methods are also interested in analyzing the item descriptions (Pazzani, 1999, pp. 397-398). Opposite to collaborative filtering methods, which do not necessarily require any information about the items to work, content-based filtering methods rely on the detailed item descriptions (Jannach et al., 2011, Ch. 3). Lops, De Gemmis and Semeraro (2011, p. 75) define the process of content-based recommendation to be *"matching up the attributes of the user profile against the attributes of a content object"*.

Adding on to the example given previously, item information could be for example genres as presented in Table 3. If this information would be combined with the information given previously, James could be recommended to try out horror books since he has liked mystery books and these two genres share two features (suspenseful and thrilling). Also, since Jennifer does not like romance or fantasy, it could be derived that only books that are not emotional or slow-paced might suit her taste and be worth recommending.

Table 3. Example of descriptive information with unary ratings (1=has feature).

	Horror	Romance	Fantasy	Mystery	Sci-Fi
Suspenseful	1			1	
Emotional		1	1		1
Thrilling	1			1	
Mysterious				1	
Futuristic					1
Slow-paced		1	1		

Benefits of using content-based filtering methods over collaborative filtering methods include for example user independence, transparency and being able to recommend new items (Lops et al., 2011, p. 78). However, there are also situations where content-based filtering methods do not work. For content-based filtering methods to work properly enough information to distinguish items from each other is needed (Lops et al., 2011, p. 78). Also, while broad databases containing ratings of thousands of users are not required when using content-based filtering methods, difficulties may arise when recommendations are to be made for new users who do not have rating history (Aggarwal, 2016, p.14). Recommending new types of items is also difficult when using content-based recommenders as they do not support exploration outside the observed preferences of the user (Lops et al., 2011, p. 79).

Demographic methods

Demographic profile of user is the key point of interest in recommendation systems using demographic methods (Aggarwal, 2016, p. 19). The key assumption when using the demographic approach is that users sharing demographic attributes share also views regarding items (Safoury & Salah, 2013, p. 303). While the core assumption of introducing a correlation between demographic attributes and other variables can be criticized for giving opportunity for producing dangerous heuristics, similar to what was demonstrated for example by Amazons recruiting tool that learned to favor men candidates (Dastin, 2018), demographic recommendation systems are aimed to be used for legitimate purposes. Examples of these purposes include personalizing website content based on user country or language (Ricci et al., 2015, p. 13) and predicting ratings of tourist attractions (Wang, Chan, & Ngai, 2012) among others.

According to Krulwich (1997, p. 38) the information from where user profile is constructed can include variables like owning a pet in addition to the commonly used demographic information. Example of demographic information that could be used in a recommendation system for recommendation generation can be seen in Table 4. After a user profile has been constructed, it can then be linked to the explicit or implicit ratings which leads on learning how the demographic profile links to different actions (Pazzani, 1999, p. 400). As reported by Safoury and Salah (2013) benefits gained from using demographic approach include improved recommendation quality and the help towards tackling the cold-start problem. This is because the similarity can be observed through investigating the similarity of demographic information saved to user profile. The downside of using demographic approach is that the information required for constructing a user profile is sensitive and therefore problematic from both security and privacy perspectives (Jain, Grover, Thakur & Choudhary, 2015, p. 957).

Table 4. Example of demographic information.

Income	Age	Gender	Marital status	Occupation
14 150	21	Female	Single	Student

Knowledge-based methods

Knowledge-based recommendation systems can help in situations where there are not a lot user ratings available or users wish to explicitly define requirements for the recommendations they wish to receive (Aggarwal, 2016, pp. 167-168; Jannach et al., 2011, Ch. 4). While the requirement of needing to have descriptive item information available when using knowledge-based methods can be compared to content-based filtering methods (Jannach et al., 2011, Ch. 4.2), Aggarwal (2016, pp. 168-169) suggests that the difference between content-based and knowledge-based recommendation systems is that knowledge-based recommendation systems do not require historical data of user ratings like content-based filtering methods. Additionally, interactivity between the recommendation system and the user receiving recommendations plays a key part in knowledge-based methods (Jannach et al., 2011, Ch. 4) and can be used to distinguish the content-based filtering method and knowledge-based method. According to Aggarwal (2016, p. 170), the three different forms of how knowledge-based recommendation systems may implement interaction between the recommendation system and a user are following:

- *Conversational systems*: user preferences are found by using feedback loop.
- *Search-based systems*: user preferences are found by asking questions from user.
- *Navigation-based systems*: user preferences are found by receiving change-requests to current recommendation iteratively.

In addition to the options in implementing the interaction between system and user, knowledge-based systems can be divided to two sub-groups: constraint-based systems and case-based systems (Aggarwal, 2016, pp. 172-188). While both of these approaches require user to specify requirements, they differ in the way they produce the recommendations: case-based approach produces recommendations based on item similarity and constraint-based focuses on applying requirements as strict rules (Janic et al., 2011, Ch. 4.1). Example of knowledge-based recommendation system that would fall in the sub-group of constraint-based recommendation systems derived from domain of Table 2 could be following: Mia, a new user of the system, would like to have recommendations regarding books that have been written by some popular author, are part of some ongoing series, and are made in United States of America. Based on information saved to the knowledge base of the system, the recommendation system could recommend a book that satisfies these pre-defined requirements. After receiving the initial recommendation, Mia could modify the requirements to receive new set of recommendations. For example, she might decide to drop the requirement about the book having to be a part of some series and add a new requirement that recommended books cannot be more than 250 pages long. It is important to note, that for Mia to be able to use a specific filter, the knowledge base would need to have information regarding the feature the selected filter is targeting.

Although it would be possible to adapt knowledge-based recommendation systems to multiple domains, they are best suited for recommending highly customized items (Aggarwal, 2016, p. 168; Jannach et al., 2011, Ch. 4.1). As rating information is not a strict requirement for the system to work, knowledge-based recommendation systems may also be used in situations where user information is not available. Because of this benefit, knowledge-based recommendation systems also avoid the cold-start problem (Burke, 2000, p. 2). The difficulty of not having information about the user however may severely limit the level of personalization the system offers (Aggarwal, 2016, p. 195).

Hybrid recommendation systems

Each of the different recommendation system approaches have their strengths and weaknesses and there does not exist one solution that works best for all recommendation problems (Jannach et al., 2011, Ch. 5). To overcome limitations that are present when using any one of the approaches while building a recommendation system, multiple approaches can be combined (Aggarwal, 2016, pp. 222-223; Hussein, Linder, Gaulke, & Ziegler, 2014, p. 127). These types of systems that combine multiple recommendation approaches into one system are called hybrid recommendation systems (Burke, 2007, p. 380). Total of seven different hybrid recommendation system categories have been proposed by Burke (2007, pp. 380-392):

- *Weighted*: Each system calculates its own score and weighted sum calculated from these scores is used.
- *Mixed*: Both recommenders generate recommendations, and all generated recommendations are used.

- *Switching*: Recommender that is used for generating recommendation may change based on calculated confidence level.
- *Feature Combination*: One system's recommendations are used as additional features and one system calculates the recommendation based on this and all other features.
- *Feature Augmentation*: One system uses domain knowledge the generate a new feature for the item that is afterwards processed by the other (actual) recommendation system.
- *Cascade*: Primary recommender chooses recommendation candidates, but secondary recommender influences the end score the candidate receives.
- *Meta-level*: Model learned by contributing recommender is used as input for actual recommender.

Another, a more general, way to categorize hybrid recommendation systems is to sort them by their system design architecture as suggested by Jannach et al. (2011, Ch. 5):

- *Monolithic Hybridization Design:* While multiple recommenders may influence data pre-processing, single recommender is responsible for making recommendation. Feature Combination and Feature Augmentation class hybrids from Burke (2007) taxonomy belong into this category.
- *Parallelized Hybridization Design:* Several recommender work side by side and may contribute to the recommendation. Includes hybrids from Weighted, Mixing and Switching classes from Burke (2007) taxonomy.
- *Pipelined Hybridization Design:* Recommenders form a pipeline where output of one is input of another. Output may be either recommendation list or a model. Includes both Cascade and Meta-level class hybrids from Burke (2007) taxonomy.

Aggarwal (2016, p. 201) offers a third view to classifying hybrid recommendation systems and classifies hybrids into three main categories: Ensemble design, Monolithic design, and Mixed systems. Comparing to the categorization provided by Jannach et al. (2011, Ch. 5), Aggarwal suggests that Mixed class would be its own main category and Feature Augmentation class could also be part of Monolithic design approach. In addition, Parallelized and Pipelined designs (named Sequential in this model) would be subclasses of the Ensemble design class instead of being main classes. This way of categorizing hybrid recommendation systems is pictured in Figure 5.

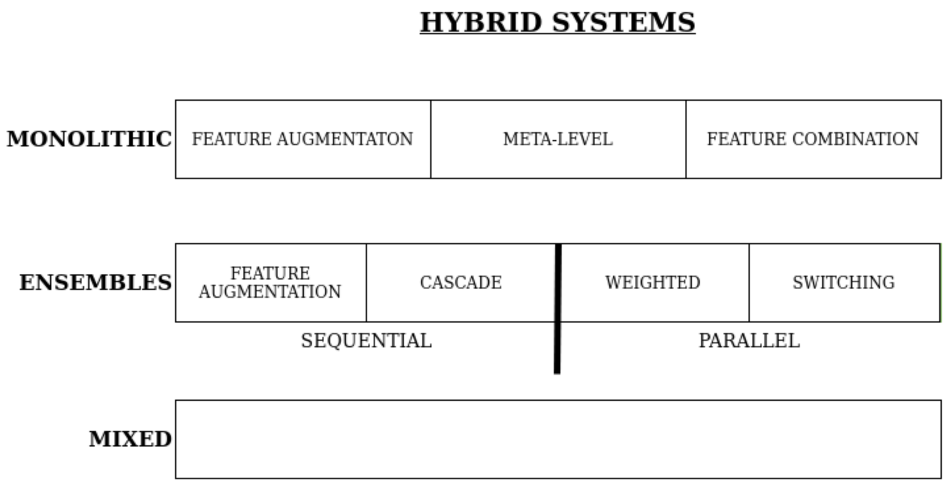

Figure 5. Categorization of hybrid recommendation systems (adapted from Aggarwal, 2016, p. 201).

According to the evaluation of hybrid recommendation systems done by Burke (2007, pp. 398-405) hybrid recommenders perform better than basic types of recommenders. Jannach et al. (2011, Ch. 5.5) however reminds that only limited amount of research effort has been put into comparing main recommendation strategies to their hybrid counterparts as finding a suitable dataset for performing comparisons is difficult. Despite lacking conclusive empirical proof of the superiority of hybrid recommendation systems, it is difficult to argue against their potential as they do allow reducing the number of drawbacks compared to using only one type of system (Burke, 2002, p. 339).

2.2.2 Challenges in recommendation systems

To make recommendation system as effective as possible some challenges must be overcome. The identification work of such challenges began already during the early years of recommendation systems. Some challenges identified during this time were for example concerns regarding data privacy and forged reviews as presented by Resnick and Varian (1997). Since then many more challenges have been identified. In this section commonly discussed challenges in recommendation systems are briefly described.

Data sparsity can be a problem in recommendation system handling large but sparse datasets (Jain et al., 2015, pp. 957). For example, a user might have rated hundred items, but that is only a small portion out of million items. In this type of situation finding similar users in trustworthy manner is challenging if using collaborative filtering methods (Guo, Zhang, & Thalmann, 2014, p. 57). Using hybrid recommendation systems have been suggested as solution to the data sparsity problem (Jain et al., 2015, p. 958; Su & Khoshgoftaar, 2009, p. 3).

Cold start is a specific type of data sparsity problem (Su & Khoshgoftaar, 2009, p. 2). The cold start problem happens when there is not enough rating history data available to make proper recommendations (Jain et al., 2015, pp. 957-958). The lack of data may consider either new user's lack of rating history or new item's lack of rating history (Jain et al., 2015, pp. 957-958). Utilizing demographic and content-based filtering methods have been suggested as solution to cold start problem (Khusro et al., 2016, pp. 1182-1183).

Scalability might become a problem in recommendation system when the amount of data grows (Jain et al., 2015, pp. 957; Khusro et al., 2016, p. 1185). As presented by Su and Khoshgoftaar (2009, p. 4) this problem is present especially when using traditional collaborative filtering algorithms. Dimensionality reduction techniques and clustering have been proposed as solution to the scalability problem as they reduce the amount of computation needed to calculate the recommendation (Khusro et al., 2016, p. 1185; Su & Khoshgoftaar, 2009, p.4).

Synonymy refers to situation where same or two very similar items are handled as two separate entries by the recommendation system (Khusro et al., 2016, p. 1183; Su & Khoshgoftaar, 2009, p. 4). For instance, this type of problem could occur with categories football and soccer when recommending what sport to watch next. Ontologies and Single Value Decomposition techniques have been proposed as solution to the synonymy problem (Khusro et al., 2016, p. 1183; Su & Khoshgoftaar, 2009, p. 4).

Overspecialization in context of recommendation systems means that the recommendation system is not capable of recommending novel items. Overspecialization may become a problem especially in content-based recommendation systems (Jain et al., 2015, p. 957). An example situation where the problem might occur is listening to music: if feedback given by the user indicates that the user likes melodic death metal music, the system might only recommend melodic death metal music to the user. Yet, the user might wish to explore other similar types of music as well, and recommendation system not suffering from overspecialization might recommend for example some other metal sub-genres like trance metal, folk metal or melodic metalcore. Introducing neighborhood-based collaborative filtering methods have been suggested as solution to overspecialization problem (Jain et al., 2015, p. 957).

2.3 Collection development

Collection development in domain of libraries can be summarized to be answering the question of which resources should be made available to library patrons (Johnson, 2018, pp. 1-2: Evans & Saponaro, 2012, p. 22). It is often used synonymously with the term collection management. Johnson (2018, p. 1) however suggests that collection development focuses on the phase when the collection is being built and collection management should be used to refer what happens after the collection has been developed. Evans and Saponaro (2012, p. 22) agree with Johnson together with Fieldhouse (2012, p. 28) on that the scope of collection management is broader than scope of collection development. Nonetheless, both terms describe a process which is a mean towards a single uniform goal: managing collections to serve the community best possible way (Fieldhouse, 2012, pp. 27-28).

2.3.1 Selection process

Stock selection, i.e. choosing what materials should be acquired to collection, is a fundamental task in collection development (Fieldhouse, 2012, p. 28). Edelman (1979, p. 34) defines it to be the second level in the overall hierarchy of collection development terminology that is followed by acquisition. According to Johnson (2018, p. 121) selection can be thought to be partly art and partly science as both librarian's experience as well as intuition play important role in the process. In addition to the librarian's experience and intuition, features such as book reviews, author reputation, cost, and usage statistics can affect the selection decision as pointed out by Fieldhouse (2012, p. 28).

Reasoning behind an inclusion or exclusion decision of a book can be hard to explain even to an experienced librarian (Johnson, 2018, p. 121). However, since librarians are making purchases with community money, according to Evans and Saponaro (2012, pp. 23-24) it can be argued that librarians should be able to justify their decisions. Shaw (2011, pp. 165-166) suggest that each library should have a formal collection development policy to help with both public relations and decision-making process. However, in a survey conducted by Horava and Levine-Clark (2016, p. 1) five out of sixteen academic libraries responded that they do not have any type of collection development policy.

Other challenges that may be encountered during selection process include for example the challenge of deciding the format in which the book should be acquired as well as challenges related to time constraints (Library Journal, 2018, p. 36). Furthermore, shrinking budgets and increasing number of publications have also increased the difficulty of selection process (Fieldhouse, 2012, p. 28, 31). In some libraries the collection budget has also expanded to include new items such as discovery services and metadata costs (Horava & Levine-Clark, 2016, p. 4) which in turn leaves less money toward acquiring new books. The increasing amount of self-publishing has also affected the selection process as formerly it was easier to justify not acquiring a self-published book to library collection (Gregory, 2019, pp. 3-4).

Use of data, such as circulation statistics, has been successfully applied to evaluate and enhance collection development policies (Adams & Noel, 2008; Knievel, Wicht & Connaway, 2006). In research conducted by Library Journal (2018, p. 37) usage statistics and circulation data were the most requested data sources to help with collection development. Although visualizing library data has also be explored and some prototypes of visualization tools have been developed (see e.g. Borrego & Lewellen, 2014; Eaton, 2016; Finch & Flenner, 2016), only a few tools in domain of data-driven collection management exist (OCLC, 2020a; OCLC, 2020b).

2.3.2 Library collections and machine learning

While collection data visualization has been explored and some machine learning applications have been developed to help in various library processes, it can be argued that the usage of modern machine learning techniques in collection management have not been researched very broadly. Decreasing the workload in cataloguing by utilizing machine learning is one of the topics that has had some active research interest during the past five years. For example, Brygfjeld, Wetjen & Walsøe (2018) researched automated classification of articles to reduce workload of cataloguers. Automated subject indexing tool, Annif, developed by Suominen (2019) is also a working example of a recently developed tool that helps in reducing the workload of cataloguers by utilizing modern machine learning techniques. Other novel machine learning applications in context of collection management include for example tool for logistics and space management (Lyngsoe Systems, 2019).

The topic of machine learning assisted weeding has only been studied on one occasion previously. Wagstaff and Liu (2018) researched this topic and proposed a machine learning solution that can automatically classify weeding candidates. The proposed solution showed encouraging results as a statistically significant agreement between human decisions and machine learning model predictions was found (Wagstaff & Liu, 2018, p. 245). While machine learning assisted recommendation systems targeting selection process in context of libraries has not been researched before, previous research linked to selection process does exist in collection management context. For example, the interest on predicting book use started as early as 1970s (e.g. McGrath,

1971) and is still going strong (e.g. Baba et al., 2016; Iqbal et al., 2020). One of the latest research efforts towards predicting book use was done by Iqbal et al. (2020). The research proposed a novel method to predict rental book data utilizing modern machine learning techniques, such as deep neural networks (Iqbal et al., 2020).

3. Research methodology

The goal of this study was to explore, analyze and evaluate how selection process in library collection development context could be assisted by using a recommendation system. To reach the goal of this study design science research (DSR) was selected to be the research method. Section 3.1 introduces the DSR methodology and its history in concise form and Section 3.2 continues by discussing how the DSR methodology was applied in the thesis. In Section 3.3 the research activities that were carried out are presented.

3.1 Design science research

DSR is a fairly new research approach that has had increasing interest in the field of Information Systems during the past decade (Iivari & Kuutti, 2017, p.2). It can be summarized to target the gap between relevance and rigor in the research conducted in field of information systems (Baskerville, Baiyere, Gregor, Hevner & Rossi, 2018). While there does not exist one widely accepted definition of DSR, one common nominator between many of the definitions can be considered that it aims to be human-centered improvement achieved through problem-solving process (Iivari & Venable, 2009; March & Smith, 1995; Hevner et al., 2004; Peffers, Tuunanen, Rothenberger & Chatterjee, 2008).

Creation of an artificial artifact is a key element in DSR (March & Smith, 1995; Hevner et al. 2004; Peffers et al., 2008). Artifacts are broadly defined to be *"constructs, models, methods, and instantiations"* (March & Smith, 1995, p.256; Hevner et al., 2004, p.81), but more accurate type chart has also been identified through a literature review conducted by Offermann, Blom, Schönherr and Bub (2010). The artifact type classifications identified by Offermann et al. (2010, p.83) include such categories as system design, method, algorithm, or guideline among other.

To perform DSR in a legitimate and scientifically valid way a methodology is needed in addition to descriptions of essential elements in DSR (Peffers et al. 2008). One identifiable starting point for defining DSR methodology was when March and Smith (1995) identified two main activities in DSR process: building and evaluating. These activities are described to be counterpart to discovery and justification from natural science (Marc & Smith, 1995, p. 254). The difference between natural science and DSR was further described to be that natural science research is interested in how and why things are, and DSR is more interested in asking how a pre-defined goal can be achieved and why the artifact build to achieve to goal actually works (Marc & Smith, 1995).

The work towards definition of a commonly accepted methodology was continued by Hevner et al. (2004) by describing seven guidelines (Table 5). The goal of defining these guidelines was to provide tools to understand, execute and evaluate DSR better. It was further proposed by Hevner et al. (2004) that DSR should be both proactive and reactive in its nature so that it can benefit both academia and industry. This attempt of flexibility shows also in guidelines that were written in such way that each researcher could make decision on how to implement each guideline in their study (Hevner et al., 2004).

Table 5. Suggested guidelines by Hevner et al. (2004).

#	Guideline	Description
1	Design as an Artifact	Design-science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
2	Problem Relevance	The objective of design-science research is to develop technology-based solutions to important and relevant business problems.
3	Design Evaluation	The utility, quality and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
4	Research Contributions	Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
5	Research Rigor	Design-science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.
6	Design as a Search Process	The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.
7	Communication of Research	Design-science research must be presented effectively both to technology-oriented as well as management-oriented audiences.

Based on guidelines proposed by Hevner et al. (2004) and other prior research, Peffers et al. (2008) defined a design science research methodology (DSRM) with the aim of creating a commonly accepted framework for DSR. The framework includes total of six different research activities which are in nominally sequential order (Figure 6). Peffers et al. (2008) have suggested that researchers should choose the starting point based on the initiation of the research process.

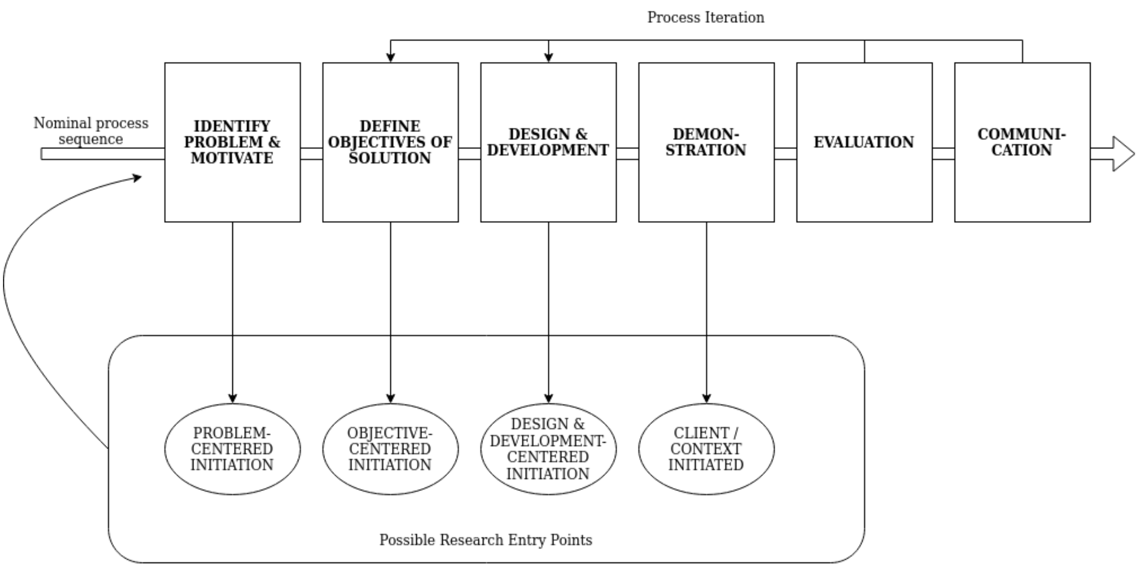

Figure 6. Cyclical nature of design science research (adapted from Peffers et al., 2007).

3.2 Application of DSR in this thesis

Guidelines suggested by Hevner et al. (2004) were followed in this study. In addition, the importance of theorizing as part of conducting a DSR, as proposed by Venable

(2006), was respected. However, the contributions to science were made as appropriate to goals as suggested by Baskerville et al. (2018).

The type of artifact that was created maps to system design using the categorization defined by Offermann et al. (2010). As applying the approach of building a new type of recommendation system was a novel solution but library collection development was an old problem, the contribution of this thesis matched the Gregor's and Hevner's (2013) definition of improvement-contribution (Figure 7).

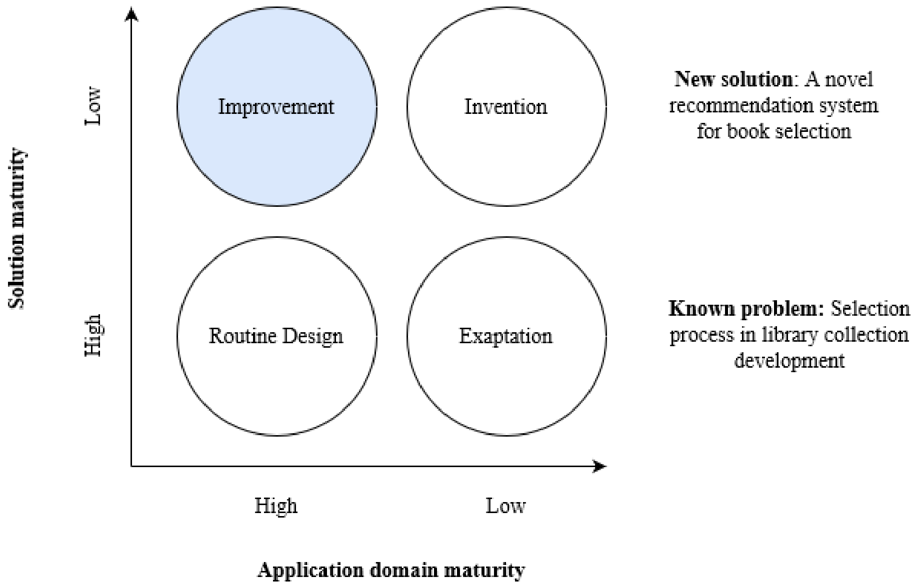

Figure 7. Knowledge contribution of this thesis within DSR knowledge contribution framework (adapted from Gregor and Hevner, 2013, p. 345).

3.3 Research activities

This thesis followed design and development-centered approach defined by Peffers et al. (2008). The aim was to design and deploy a system that can solve the research problem. Each research activity that was taken during the research will be discussed next in detail.

Problem identification and motivation

The main motivation of this work was to help librarians with the book selection process that has become increasingly difficult because of the increased number of books published in a year and budget cuts in libraries. The suggested solution seeks to solve the identified problem by allowing librarians to utilize the data stored in library systems better with help of a novel hybrid recommendation system. Utilizing data better allows librarians to make better decisions faster and thus benefits both the librarians and library's community. In addition to the main motivation, the secondary motivation was to give a tool for librarians to justify purchases better as this problem was identified from the literature. Ability to provide better justifications regarding selected books was seen as path leading to increased transparency and better public relations.

The research problem was originally observed by the author in middle of 2010s. During this time, the author was managing a library collection as part of his job as a manager of a public library. The main observation was that since no additional tools were available, decision-making was highly based on librarian's personal experience and insight about the collection. For new librarians entering the field, the lack of tools was problematic as they did not have prior expertise to rely on when making decisions. Furthermore, even experienced librarians sometimes had to spend a lot of time thinking about collection development decisions. The data to help decision making was already available in the library systems when these initial observations were made, but it was not utilized as software development in domain of library systems had other priorities at the time.

Although the idea of having an assisting tool for the selection process was lost for many years, it was rediscovered by the author in 2020. While during these five years some data visualization tools had been developed in domain of libraries, they had not solved the problem which was originally observed. An accessible tool that could assist in collection development, particularly in book selection process, was still needed.

Objectives

Objectives for the solution were defined according to the DSR methodology in this thesis: objectives were deducted from the problem definition in a manner that they were accomplishable. The main objective of the solution was to implement a novel recommendation system that could assist librarians in collection development, particularly in book selection process. The solution was to be considered to satisfy the main objective in case it reached to appropriate score with selected metrics and satisfied librarians working in public libraries.

Secondary objective was to make recommendations as transparent as possible. This objective did aim to bring means of justifying decisions to librarians. Since data visualization had been established to be a desirable feature within the realm of collection development in prior studies, one objective was for the solution to have support to data visualization feature at least to some level in addition to providing transparency via displaying statistics.

Lastly, the solution was to aim to serve as a base for future research regarding the topic as its final objective. This objective was determined because the book selection process is only one of the processes of collection development and there are even more processes in the broader area of collection management where similar types of solutions could be provided in future. It was determined that this final objective would be fulfilled if the system design would emphasize extensibility.

Design and development

Creation of the artifact, the third activity of DSR research process, was done between July and October 2020. This activity consisted of developing a software solution with three different parts: hybrid recommendation system, backend service and a web-based user interface. The activity followed an iterative process of exploring, building, and evaluating all the components of the artifact. As the artifact consisted of multiple components, each component was revised multiple times during the development phase. Component-level iterations were conducted until all the objectives defined for the solution were achieved. The full development process is described in detail in Chapter 4.

Demonstration

According to the DSR process the artifact needs to demonstrate its effectiveness in solving the problem defined in first activity. As the main objective was to provide tools for librarians to evaluate if a book should be selected for acquisition or not, demonstration for this objective could be done by retrieving recommendations for books in validation dataset and comparing if the recommendations matched the pre-defined label (i.e. if the classification resulted into true positive or true negative outcome). Because of the nature of the demonstration, accuracy metric along with precision, recall and F1-score metrics were chosen to be used for these tests. The demonstration regarding the visualization and transparency capabilities could be done by demonstrating that the backend service could serve statistical data and frontend service could display it.

The final objective of the artefact being capable to be extended as part of future research can be argued to be difficult to demonstrate. However, as the code of the artifact was designed and implemented in a way that extending it should be easy and the source code for the application was made publicly available, it was determined that the demonstration requirement for this objective was fulfilled to the extend it was possible.

Evaluation

According to the DSR process, the performance of the artifact needs to be observed and measured after the artifact has demonstrated its capability to solve the research problem. To achieve this, evaluation of the artefact produced as output of this thesis was split in two sections: first, a new dataset that had not been used in any of the previous phases was used to determine how well the proposed recommendation system performed against a set of baseline options. This part of evaluation was named automated tests as the testing did not include any human interaction.

In addition to automated tests, evaluation from user perspective was carried out. Evaluating the artefact from user perspective allowed to gain insight regarding how well the solution could solve the research problem in real-world context. For this evaluation purpose, focus group methodology was chosen to be used. Reasoning for choosing to use focus group methodology was that it was already established in prior research to be a good fit for evaluating potential solutions (Kontio, Bragge & Lehtola, 2008, p. 99). In addition, the value of the focus group methodology had been acknowledged in both academic research as well as in other domains (Morgan, 1997, p.1) which perfectly reflected the goals of DSR research. Focus groups had also already been successfully applied in settings where people are unable to gather to one location (e.g. Greenbaum, 1998, pp. 89-97; Steward & Shamdasani, 2017, p. 49, 52). This was important as the evaluation was conducted during the time when global coronavirus pandemic was ongoing, and the evaluation was needed to be arranged in a manner that allowed participants to safely attend.

Section 5.1 gives further details regarding how automated tests and focus group session were organized. Results of the evaluation are displayed in Section 5.2 and Section 5.3.

Communication

Communicating the knowledge gathered during the research process along with the produced artifact to researchers and other audiences is the last activity defined in DSR process. Reporting the research process, result, and discussion regarding the implications of the research was done through this thesis that follows the guidelines present in University of Oulu as of 2020. The communication of the built artefact was handled by making source code public. Releasing both the source code and the research publicly in Internet may lead into other scientific contributions. Furthermore, releasing the source code allows libraries who have access to software development services to start utilizing the tool in real-world if they so wish.

During the research process a handful of librarians were consulted regarding the heuristic rules for the system in addition of some librarians participating to the focus group evaluation. Both of these events can be seen as communication which results into spreading information about the existence of the solution to the target user base.

4. Implementation

This part of the work discusses the implementation of the proposed solution. In Section 4.1 the tools that were used in building the solution are presented. Section 4.2 then proceeds to describe the dataset that was used to construct and test the proposed solution. Steps that were taken during data preprocessing are explained in Section 4.3. Development of the three different parts that formed the overall recommendation system are reviewed in Section 4.4.

4.1 Programming languages and libraries

The proposed solution, that is the output artifact of this thesis, was developed using Python[2] and JavaScript[3] programming languages. These programming languages were chosen mainly because of their popularity. Libraries such as TensorFlow[4], scikit-learn[5], xgboost[6] and LightGBM[7] were used for research regarding machine learning implementation options as they allowed access to a broad variety of options for training machine learning models using supervised learning paradigm. By including all these libraries to the project, it was possible to both construct a deep feedforward neural network model and compare it to models trained using other machine learning algorithms.

Data processing was done using NumPy[8] and pandas[9] libraries for Python. These libraries were chosen as they were observed to be commonly used in scientific research and statistical workloads at the time. Matplotlib[10] library was included so that different types of graphs could be created from the data. Generating graphs allowed tuning the heuristics of the proposed recommendation system effectively in the development phase and provided a way to produce graphs for analysis purposes after the development of the artifact was finished.

---

[2] https://www.python.org/

[3] https://developer.mozilla.org/en-US/docs/Web/JavaScript

[4] https://www.tensorflow.org/

[5] https://scikit-learn.org/stable/

[6] https://xgboost.ai/

[7] https://lightgbm.readthedocs.io/en/latest/

[8] https://numpy.org/

[9] https://pandas.pydata.org/

[10] https://matplotlib.org/

The backend service, that serves resources regarding both statistics and recommendations over HTTP protocol, was constructed using Flask[11] library. To extend the capabilities of the Flask library further, a set of additional libraries regarding web development were also added to the project. This set of libraries included Flask-RESTful[12], Flask-Marshmallow[13] and Flask-SQLAlchemy[14].

The frontend, that serves as user interface for the solution, was built with React[15] library that at the time was one of the most popular JavaScript libraries for building websites. Since the main goal of this research was not to focus on user-interface design, a card-based approach was selected to be used for its efficiency. Material-UI[16] library was added to the project so that this card-based approach could be effectively used. SQLite[17] was applied as the database solution because it was determined to be the most simple and lightweight solution for creating a relational database. SQLAlchemy[18] library for Python was used to interact with the database because it had extensive capabilities in object relational mapping.

4.2 Dataset

The dataset used for constructing the artifact was provided by Joensuu City Library. The data provider approved the use of data for the purposes of this research (Appendix B). The dataset consisted of both circulation logs and item level information regarding items located in the adult fiction collection of the Joensuu City Library. The data was first acquired by applying SQL queries to the library system's database. This step was done by the administrator of Koha library software[19] used in Joensuu City Library. The said administrator also contributed to creating working versions of the said queries. The output received after applying the queries to the database was four comma separated value files that were then concatenated to produce two separate files: one file consisting of item information and one file consisting of circulation logs.

The circulation log data consisted of data saved to the system from July 2014 to August 2020. The book item information included information from an undetermined date to August 2020 as book item information had been transferred from previous library systems used in Joensuu City Library opposed to the circulation log data. The book item information consisted of total 53171 entries. Of these entries 41389 considered books

---

[11] https://flask.palletsprojects.com/

[12] https://flask-restful.readthedocs.io/

[13] https://flask-marshmallow.readthedocs.io/

[14] https://flask-sqlalchemy.palletsprojects.com/

[15] https://reactjs.org/

[16] https://material-ui.com/

[17] https://www.sqlite.org/index.html

[18] https://www.sqlalchemy.org/

[19] https://koha-community.org/

that were available in the collection and 11782 considered books that had been deleted from the collection.

During the development of the solution it was discovered that some datapoints that would be beneficial for the system under construction were not included to original SQL queries. As these datapoints were available on the database but were not included into the original SQL queries a new decision regarding data acquisition was made. Rather than iterating over developing the SQL queries it was decided to reacquire the data through Finna API[20], a service used by Joensuu City Library among many other libraries, archives, and museums in Finland. The decision was heavily affected by the observation that it would be faster to iterate over data collection using the API compared to re-developing SQL queries and having a system administrator run them against database snapshot. Support for using data in comma separated value format was left in place to the final version of the proposed solution so that it would be possible to both fix and extend the support for different types of data acquisition in future.

4.3 Data preprocessing

Data preprocessing was conducted to produce pre-defined type of objects out of raw data. Producing these objects was important for both machine learning model and for the overall recommendation system. To produce these objects, the data was first cleaned. The data cleaning steps included normalizing all textual features by removing unwanted characters, such as starting and ending whitespaces, from the strings and making all strings lowercase. Cleaning the data ensured that string data referencing to same entity would connect properly to the entity in question. For example, without data cleaning strings " Adams, Douglas " and "adams, Douglas" would be treated as separate entities, but after data cleaning both connect to the same entity of "adams, douglas". The output objects from data preprocessing phase contained following information:

- *acquisition date:* date when book was acquired to library's collection
- *author:* name of the book's author
- *biblio id:* unique identifier for the piece-of-work the book links to
- *circulation sequence:* an array consisting of integers that represent the circulation of the book for each month
- *deletion date:* date when book was deleted from library's collection in case it had been deleted from the collection
- *genres:* names of genres the book is associated with
- *item id:* unique identifier for the book item
- *last borrowed:* date when the book was last borrowed
- *publication year:* year when the book was published
- *publisher:* name of the publisher of the book
- *series:* names of series the book is part of
- *subject words:* subject words the book is associated with
- *title:* title of the book

[20] https://www.kiwi.fi/pages/viewpage.action?pageId=53839221

4.3.1 Training, validation, and testing datasets

During the data preprocessing stage, the data was also split into different partitions (Table 6). One partition was made to include all the data up until March 2019. This partition served as the development data for the proposed hybrid recommendation system. Data of books that were acquired between July 2014 and July 2018 were used as training data for the machine learning model. Partition made of books acquired between August 2018 and March 2019 served as the validation data for the machine learning models. The data of books acquired between April 2019 and September 2019 was used as test data for both the proposed hybrid recommendation system and other models.

Table 6. Data partitions for machine learning models.

Dataset name	Items acquisition date	Samples total	Excluded samples	Accepted samples	% of accepted samples
Hybrid recommendation system development	-03/2019	46614	n/a	n/a	n/a
Training	07/2014-07/2018	9490	5889	3601	69.96
Testing	08/2018-03/2019	2421	1650	771	14.98
Validation	04/2019-09/2019	2339	1564	775	15.06

From machine learning perspective it proved to be problematic that a lot of the data consisted of duplicate items. One problem was that duplicate entries of same item could have different labels attached to them with same feature properties. In other words, one copy of same book could have been borrowed ten times during it's first year after acquisition while other copy of the book could have not circulated at all. In addition, even if the duplicate items had the same label it was problematic to include both items to the dataset as in this case the item would be over presented in the dataset when compared to other items. To solve this problem, it was decided that only the best label for each unique title would be used and each title could have only one entry in dataset.

Another problem was that some items had only few common features included in their information. This was problematic as with items having only few common features, it would be problematic to make distinction between different items. To provide machine learning algorithms enough information to make clear distinction between items, samples having less than five features available were excluded from the partitions.

All data partitions, with the exception of the development partition for hybrid recommendation system, were saved as serialized objects to disk using Pickle[21] object serialization module. This was done to provide easy and fast access to the data in all development phases. Data for hybrid recommendation system development was imported to a newly constructed SQLite database so that it could be queried by the rule-based recommendation system.

[21] https://docs.python.org/3/library/pickle.html

4.3.2 Feature engineering

Features for the system to use were selected already during the data gathering phase. Total of five features were chosen to be used for the implementation of the proposed hybrid recommendation system: author, publisher, series, genres, and subjects. These features were chosen as they were determined to be information that could distinguish item from another while still being easy for collection librarian to use as an input for the system.

As all the chosen features were categorical in their nature, there were three main options on how to encode the features to machine learning model: one-hot encoding, word embeddings and entity embeddings. While one-hot encoding was determined to present the data well, entity embeddings and word embeddings were chosen to be investigated as well as they could reduce memory usage and speed up the training process of the neural network. A major drawback regarding use of word embeddings was discovered during the investigation as a lot of words found in data were missing from the pre-trained embeddings that were available for Finnish language. In the end it was decided that entity embeddings would be used to train the deep feedforward neural network model and one-hot encoding would be used to train models using off-the-shelf algorithms as they did not have support for use of entity embeddings.

4.4 Recommendation system

A hybrid recommendation system was proposed based on heuristics and machine learning to achieve the objective of providing reliable recommendations while similarly introducing transparency to the system. The final system consisted total of five different parts: machine learning model, rule-based system, database, backend, and frontend (Figure 8). The development of all parts of the final system was done in iterative blocks: one part was developed at a time and the part that was next revised depended on what was done or observed in the previous iteration block. The development started by gathering initial information about available machine learning models for deciding which model would be chosen to be part of the hybrid system.

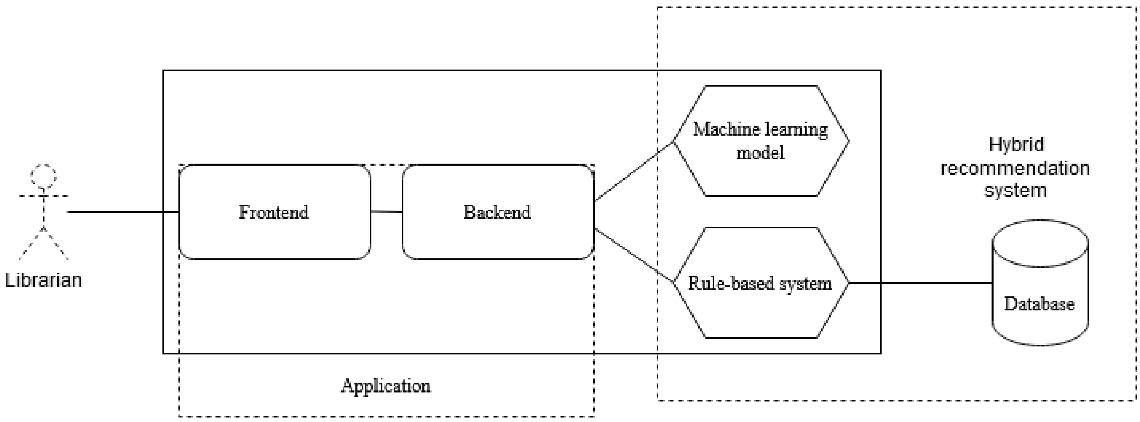

Figure 8. Overview of the proposed novel hybrid recommendation system.

4.4.1 Machine learning model

Multiple different approaches for training a machine learning model were studied and tested. Most emphasis was put in training a working version of deep feedforward neural network model. Classification problem approach was chosen for the development of the machine learning model instead of linear regression approach. The decision was supported by the fact that the dataset could be described to be high dimensional low sample size data. The decision of defining the machine learning problem to be a classification problem meant that the machine learning model would predict a label for each item. This target label was defined to be based on circulation statistics of the first year after the item was acquired. To simplify the problem further, a binary labeling schema was chosen: the target label of each example could either be 0 or 1.

The rule for defining target label was defined based on the author's own experience and information gathered from two phone calls with librarians who were actively participating to managing a library collection at the time. Figure 9 displays the rule in its entirety. Items with target label 0 are referenced as class 0 members and items with target label 1 are referenced as class 1 members from here on.

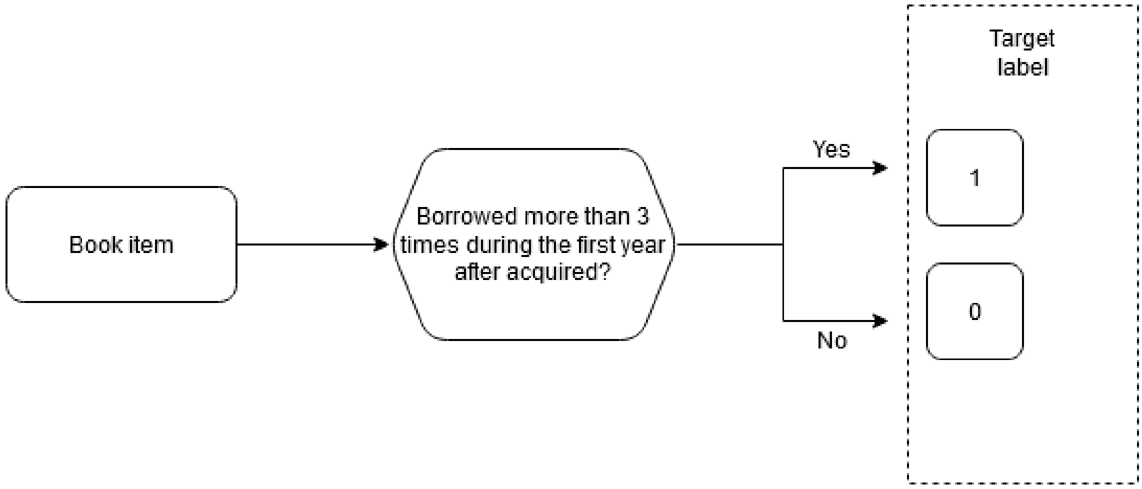

Figure 9. Rule that defines target label for each item.

After target labels were defined for all the items, it was observed that the training set was clearly imbalanced: class 0 had a lot less members than class 1. To reduce the imbalance between classes, a data augmentation function was developed. Augmentation function created a new member out of existing members of class 0 if the existing member had more than ten features that could be used in training the model. The creation of new member was achieved by deleting one feature randomly from a feature class that had over three features. Implementing the augmentation function solved the problem with class imbalance.

After solving the problems related to data, the development of deep feedforward neural network was started. From the start, the configuration of input and output layers was static. The input layer consisted of five different embedding layers. One input layer was reserved for one feature type: author, genre, series, subject, and publisher. The output layer was configured to be a dense layer outputting one neuron per class using softmax activation function. With the static configurations of input and output layers, a series of tests were conducted to obtain the best architecture and hyperparameters for the model. These tests included tests regarding number of embedding dimensions, number of hidden layers, number of neurons in each hidden layer, optimizers, learning rate, use of dropout layers and batch size (Table 7). All the tests were done by manually altering the parameters, training the model using training set and evaluating the model loss and accuracy using validation set.

Table 7. Tested hyperparameters and architectures with value ranges.

Parameter	Tested values	Best value	Description
Optimizer	SGD, RMSProp, Adam, Adadelta, Adagrad, Adamax, Nadam	Nadam	Optimizer used for updating the weights of neural network to minimize the loss calculated by loss function.
Learning rate	$1 * 10^{-1}$, $1 * 10^{-2}$, $1 * 10^{-3}$, $1 * 10^{-4}$, $1 * 10^{-5}$, $1 * 10^{-6}$	$1 * 10^{-6}$	Rate of which weights are adjusted by the optimizer.
Embedding dimensions	1, 25, 50, 100, 150, 200, 250	50	Number of dimensions used for entity embeddings in embedding layers of the network.
Hidden layers	1-5	2	Number of dense hidden layers in the neural network.
Neurons (1^{st})	50, 100, 500, 1000	100	Number of neurons in first hidden layer.
Neurons (2^{nd}-n^{th})	50, 250, 500	50	Number of neurons in the subsequent hidden layers.
Dropout layers	0, 1, 2	0	Number of dropout layers in network architecture.
Batch size	1, 4, 8, 16, 32, 64	8	Number of samples processed between weight updates.

Best performing deep feedforward neural network model was achieved using following configuration: 50 embedding dimensions for each embedding layer, 2 hidden layers with 100 and 50 neurons respectively, Adam optimizer with Nesterov momentum using learning rate of 1e-6, and batch size of 8. The parameters that affected the outcome most were the optimizer and learning rate. Use of dropout layers did not enhance the performance of the model and thus dropout layers were left out of the final architecture. Figure 10 displays the final architecture of deep feedforward neural network model that was selected to be compared against other models trained using various supervised learning techniques before selecting which model would be used for the hybridization purposes of the recommendation system.

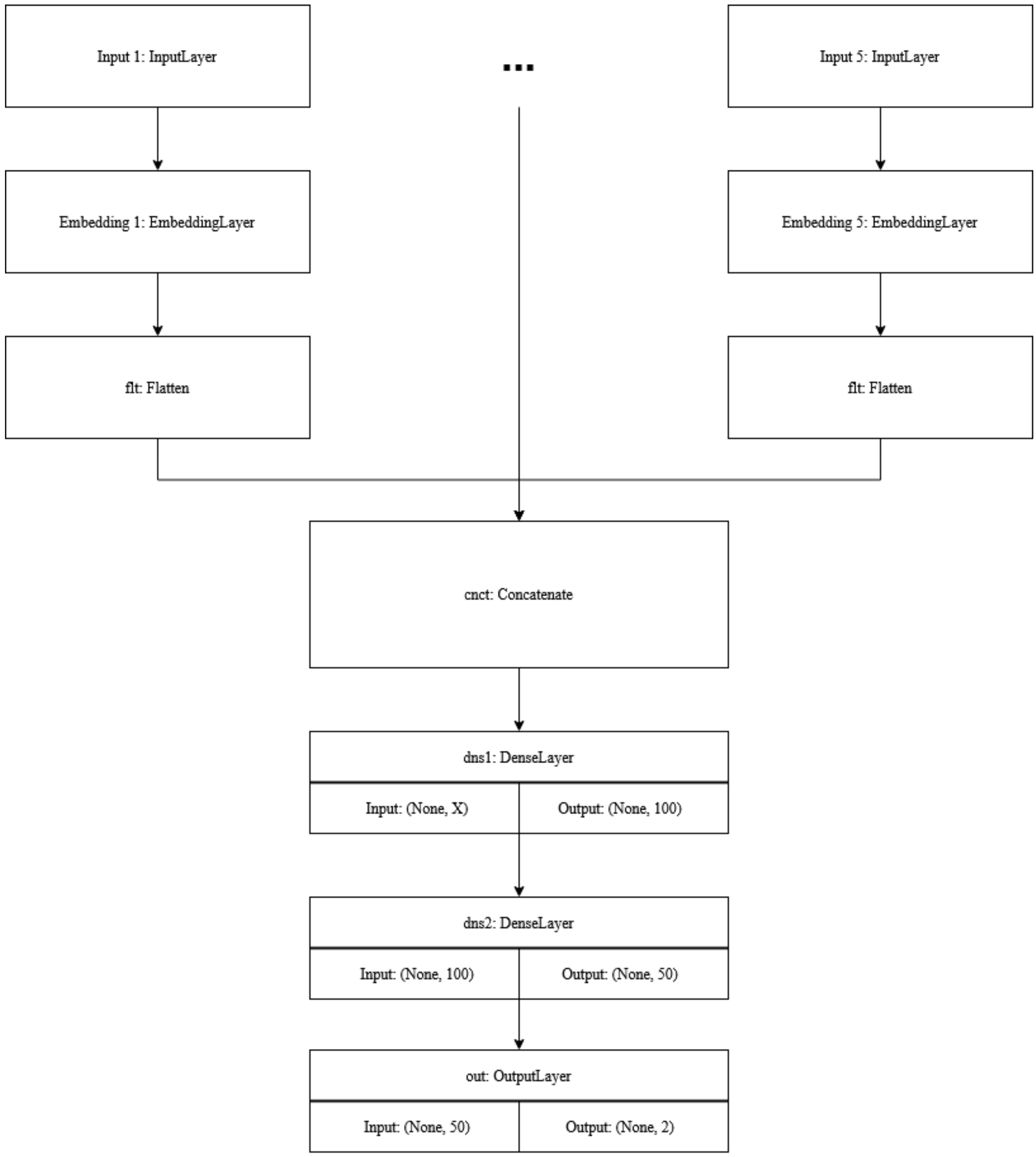

Figure 10. Final architecture for deep feedforward neural network model.

The deep feedforward neural network model was compared against other models that were trained using off-the-shelf supervised learning algorithms available in scikit-learn, xgboost and LightGBM libraries. Same datapoints from training set were used to train all the models and same datapoints were used for validating all models. The preprocessing included the data augmentation for class 0 members and encoding the features appropriately in all cases. The results of the development phase tests are described in Table 8.

Table 8. Comparison of different machine learning models in development phase.

Model name	Accuracy (%)	Precision	Recall	F1
Decision tree	79.66	0.87	0.88	0.87
Support Vector Classification	80.57	0.87	0.90	0.88
Random forest	82.77	0.87	0.94	0.90
Xgboost	80.96	0.89	0.88	0.88
LightGBM	80.31	0.89	0.86	0.87
Deep feedforward neural network	81.35	0.88	0.90	0.89

Model trained using random forest algorithm slightly outperformed other models in the test. To see if reducing features would help the accuracy of models using one-hot encoded data, an additional test was run where top 1000 features where chosen by using SelectKBest[22] function from scikit-learn library. The results of this test revealed that all models using one-hot encoded data performed worse when using the reduced feature set. While there were not big differences observed between the different models at this point, it was decided that the deep feedforward neural network model would be chosen to accompany the rule-based system in the final hybrid recommendation system.

4.4.2 Rule-based system

The rule-based system was developed with a score-based system in mind. The aim was to provide a rule-based score for each feature type and in the end hybridize the system by summing all the scores in order to obtain a total score. Machine learning model prediction was also determined to contribute towards the total score, although the contribution to the total score was made limited so that the system would be as transparent as possible. The purpose for the machine learning model was to reinforce or counter what rule-based system had determined in edge cases. Each of the features could contribute an integer value between -2 and 2 towards the total score and machine learning model could contribute either -1 or 1 to total score. The total score of the proposed recommendation system could therefore have any integer value between -11 and 11.

The calculation of the total score was divided so that a recommendation score for each feature would be calculated separately. Each feature was to have something unique in the score calculation. Author, publisher, and series features were designed a scoring rule that was affected by the item count rank and normalized circulation rank. In addition, it was defined that a positive boost would be given to new authors and negative boost would be given to new publishers. This was done to promote novelty but at same time to prevent boosting score of independent publishers too much. While it was debated if new series should receive a small negative boost towards total score as acquiring new series usually translates into a commitment, in the end it was decided that new series would not receive positive or negative boost. Pseudocode (Algo 1) explains in detail how the author, publisher and series feature score were calculated in the proposed rule-based system.

[22] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

<u>Feature Scoring (Algo.1):</u>

Input:

```
F: Feature
FT: Feature type
```

Output:

```
S: An integer containing the score for feature


S <- 0
FEATURES <- get_features_from_db(FT)

if F is in FEATURES
        IC_RANK <- calculate_ic_rank(F, FEATURES)
        CIRC_RANK <- calculate_circ_rank(F, FEATURES)

        if IC_RANK <= length(FEATURES) / 100 * 10
                S <- S+1
        elseif IC_RANK > length(FEATURES) / 100 * 50
                S <- S-1
        endif

        if CIRC_RANK <= length(FEATURES) / 100 * 10
                S <- S+1
        elseif CIRC_RANK > length(FEATURES) / 100 * 50
                S <- S-1
        endif
else
        S <- calculate_newitem_score(FT)
endif
```

Genre and subject features were designed to be handled as a set of independent features from where a sum of these independent scores would be gathered (see Algo 2). As the number of unique subjects was much greater than the number of unique genres, it was decided that penalties applied towards genres score would be halved. The score was capped for both subject and genre features to +2 and -2 values so that the scoring would be balanced with other features.

<u>List Feature Scoring (Algo.2):</u>

Input:

```
FL = {F1, F2,……, Fn}, Input list containing n-features
FT: Feature type
```

Output:

```
S: An integer containing the score for list of features


S <- 0
WP <- 0
BP <- 0
FEATURES <- get_features_from_db(FT)

for each Fi in FL do
        if Fi is in FEATURES
                CIRC_RANK <- calculate_circ_rank(Fi, FEATURES)
                if CIRC_RANK > 0
                        WP <- WP+1
                elseif CIRC_RANK < 0
                        BP <- BP+1
                endif
        else
                WP <- WP+1
        endif
end for

if FT is subject
        S <- WP-BP
elseif FT is genre
        S <- WP-(0.5*BP)
endif

if S > 2
        S <- 2
elseif S < -2
        S <- -2
endif
```

By designing the rule-based components as described, recommendations became more transparent for the user as the components used for scoring were same as the ones that would be used for visualization purposes. In addition, since each feature's score could be mapped to a number between 1 and 5, it would be easy to visualize the individual feature score for the user. The only exception regarding this was the score from the machine learning model, but since that score could have only two values, it could be presented with any binary type of visualization.

As finalizing touch for the proposed hybrid recommendation system, an uncertainty threshold was introduced to limit the system from being too positive. Total score of 0 or 1 was defined to result into a message that the system is too uncertain to make any recommendation. For these cases, it was also further configured that the system suggests the user to study the statistics and visualizations before deciding whether to select the book or not.

4.4.3 Application development

Development of backend functionalities was started after the results were observed regarding the machine learning models performance. The recommendation system logic and the backend service were developed simultaneously. The main backend functionalities included serving data in a manner that it could be visualized in the frontend and serving pathway to obtaining recommendations.

Developing a script to construct database from the serialized object file was first step in backend development. Having a database allowed both testing the heuristics and producing statistics. After the database was initialized, features to serve both statistical information and overall recommendations were built synchronously. Statistical features were made accessible by building an endpoint for a GET[23] request for all different feature types. The response served from these endpoints included the heading of the feature in addition of statistical information. A normalized trend showing the interest towards the feature was also included to the response.

The backend service for recommendations was created so that a recommendation could be retrieved using parameterized GET request. Parameter names were chosen based on the features chosen in feature engineering. Parameter values were determined to be integers referencing unique identifiers in case the feature could be found from the database. This design decision was made to speed up the processing in the backend service.

The frontend development started with creating a blueprint of what components would be needed. As the system would require user's input to work, it was first determined to use a web form as the starting page. This decision was reversed in later stages of the development to introduce easy extensibility to the system (Figure 11). To demonstrate how the system could be extended, an additional feature was developed. This additional feature allowed user to input the item id and observe the information of features similar to what was displayed by the recommendation system in the recommendation page.

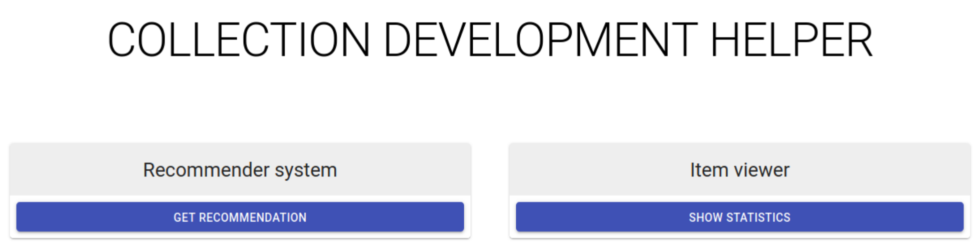

Figure 11. Starting page which allows user to select action.

The input form (Figure 12) was developed using Material-UI's Autocomplete-component. Use of the Autocomplete-component made it possible to use identifier values that were already available in the SQLite database in the requests that were send to backend service. This was considered important for the recommendation system to work properly since the input value needed to connect to the feature in the database for

---

[23] https://tools.ietf.org/html/rfc7231#section-4.3.1

the rule-based scoring to work properly. If feature would not be found in the database, the feature would be labeled as a new feature which would result into different scoring scheme. Each feature was given a separate field in the input form. While some fields, like author and publisher field were restricted to having only one value, fields like genres and subjects could have multiple values as input.

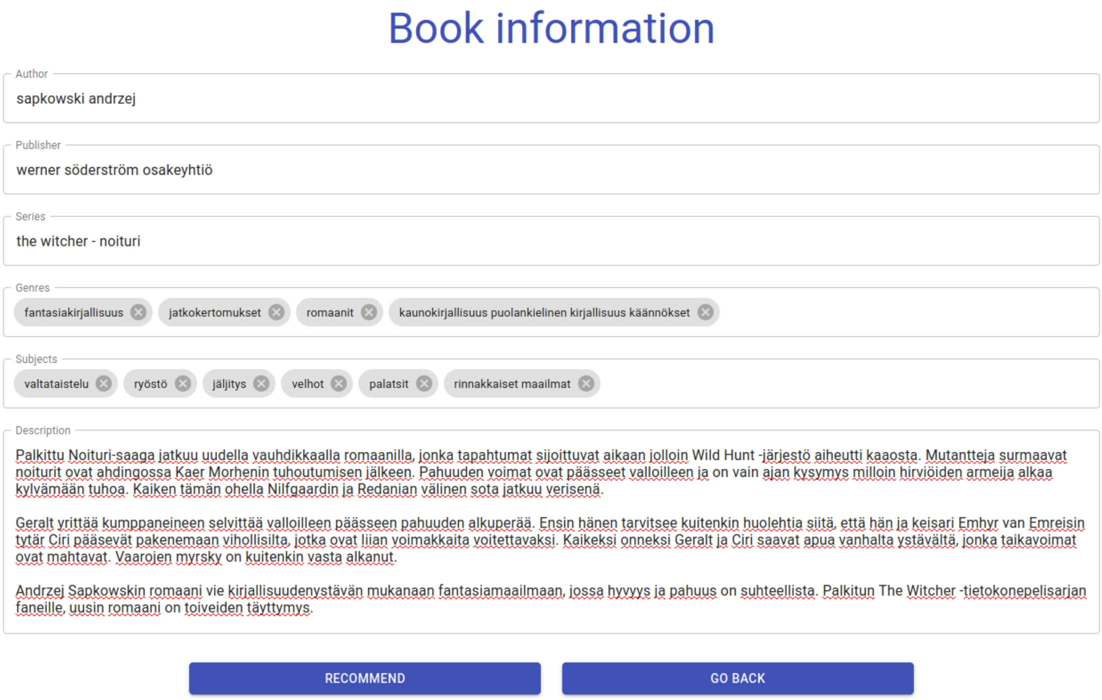

Figure 12. Form component that handles user input.

In addition of including an input field for every feature type, a field for the book description was added. Reasoning behind adding this field was that the recommendation system could utilize the automated subject words generation offered by API service of Annif[24]. With the added functionality librarians would be able to gain more insights based on the book's back cover text if the text was written in Finnish, Swedish or English.

After developing the input form, an investigation on how to represent both the statistical data and the recommendation simultaneously was conducted. Since the focus of this research was not on user-interface design, a card-based approach was selected also for these purposes. Each card was built an option to either display the statistical data (Figure 13) or the trend chart (Figure 14). In case the feature did not exist in the database, a plain card indicating that the feature is new was designed to be shown to user.

24 https://annif.org/

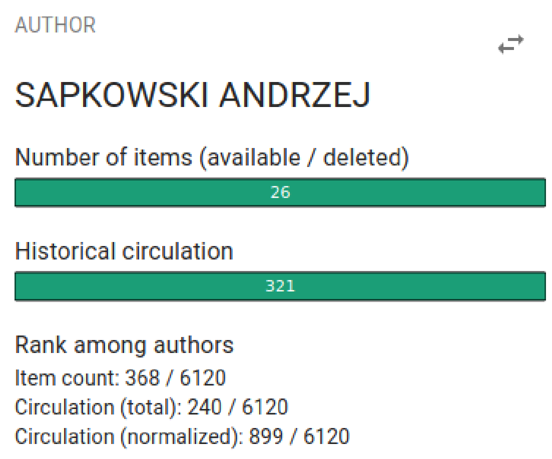

Figure 13. Example of a statistical data in feature card.

Figure 14. Example of trend data in feature card.

The overall recommendation given by the hybrid recommendation system was designed so that it would display in the top of the screen and consist of textual data, numerical data represented as stars, and a thumb icon. The textual data was designed to represent the final recommendation made by the system. The data presented with the stars was to represent the scores calculated for each feature. Finally, the thumb icon characterized the prediction made by machine learning model. In attempt to highlight the use of machine learning technique in the recommendation generation, a robot personality, HA-1-APU, was developed to display as the giver of the recommendation. All the design choices were based on what the author thought would be simple yet informative. Example of a recommendation given by the final version of the proposed hybrid recommendation system is shown in Figure 15.

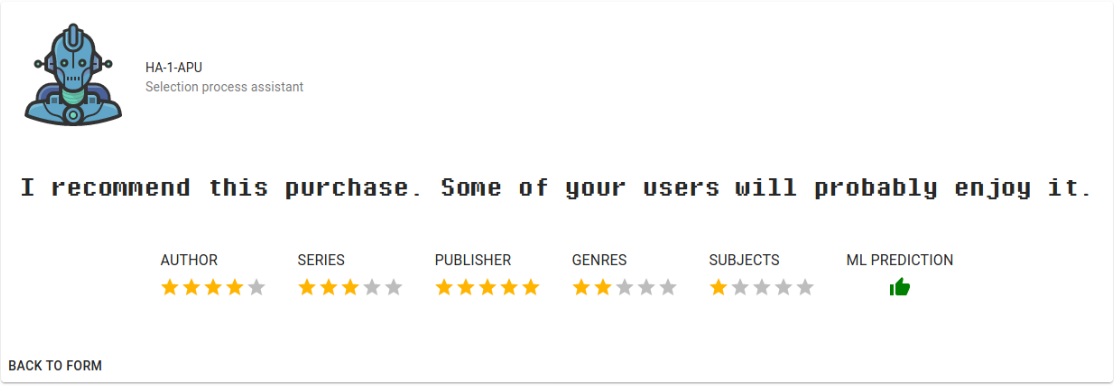

Figure 15. Example of recommendation given by the recommendation system.

5. Results

This chapter presents the results that were observed during the final evaluation of the finished artifact. Section 5.1 first presents how the tests were organized and conducted. Then, in Section 5.2 the results of automated tests are reviewed. Finally, results regarding the focus group evaluation are discussed in Section 5.3.

5.1 Test setup

The evaluation of the recommendation system was carried out in two parts. First, the system performance of overall recommendation system was evaluated using automated test. In these tests the proposed system was analyzed and compared to other baseline methods. Second, a user evaluation session was carried out using focus group methodology. In this part of the evaluation the focus was on how the proposed solution was observed from user perspective.

5.1.1 Automated tests

The test scenario for automated tests was following: Features chosen in the feature engineering stage were given as input for all the systems. The features were encoded separately for each tested system, for different systems required different type of encoding. Machine learning models trained using off-the-shelf algorithms were given one-hot encoded input, deep feedforward neural network was given input in entity embedding format, and the proposed hybrid recommendation system was given the input based on the identifiers found in the database of the developed system. Task for all systems was to solve a classification problem as based on the item classification a recommendation could be derived. For machine learning models a binary output was directly compared to the target label of each item. For the proposed hybrid recommendation system the positive score of more than 1 was mapped to value 1 and negative score lower than 0 was mapped to label 0 before comparing the output value to target label. If the proposed recommendation system output value was 0 or 1, it was recorded that the system did not give any recommendation as these values were within the defined range of uncertainty. From these cases a separate uncertainty metric was calculated for the proposed recommendation system.

Evaluation regarding system performance was carried out using the dataset specifically determined to be used only in evaluation. The automated tests were run using a separately developed script. The system used to build all compared systems and run the evaluation script had following components: Intel i5-6600K with 16GB of RAM and Nvidia GeForce GTX 970 with 4GB of dedicated memory.

5.1.2 User evaluation

The user evaluation session was carried out using Microsoft Teams[25] service on 28.10.2020. The aim of this part of the evaluation was to evaluate if the librarians thought that the system could help them in the collection development, particularly in the book selection process. Three participants participated the session which was guided by the author of this thesis who acted as a presenter and a moderator. All participants agreed to participate the study with written consent form (Appendix A). The session consisted of the author first describing the research problem and the solution that had been implemented to solve the problem. After this the author demonstrated the use of the solution by sharing his screen and filling the recommendation retrieval form with information about upcoming book. After giving the input to the recommendation system, author explained all the elements that were available on the recommendation landing page.

After the demonstration, situational association and forced relationship techniques (Greenbaum, 1998, pp. 126-132) were used by the author to prompt discussion regarding the pre-planned topics. The topic of how the solution was viewed in general was approached by presenting images of four different foods and asking which food would describe the solution the best. After this, pictures of six different inventions were shown, and the participants were asked that which of the inventions would be the closest when considering the way the solution has approached solving the research problem. To gain information regarding how the solution would be seen from perspective of the selection process, the participants were then asked to select an upgrade part to a car that would be the most similar to how the solution would upgrade the selection process. Lastly, the conversation regarding deploying the solution to real world was started by showing six pictures of different types of situations ranging from snow plowing to a traffic jam. Rest of the time was reserved for open-ended discussion regarding the topic.

After the session, author made notes based on the video recording of the focus group session. These notes were then used as the basis for analysis. By utilizing the video recording it was possible to effectively guide the discussion during the session and make thorough analysis afterwards.

5.2 Automated tests

After the overall recommendation system was ready for final tests, a final test was performed also for machine learning models to compare the baseline models to the developed hybrid recommendation system. In this test all models were trained using both training and validating datasets and tested using the test dataset. Same feature reduction test was done for models using one-hot encoded data as before. The test results in this test were vastly different than in the previous test as seen in Table 9. The reduced feature set tests for each algorithm are marked as RF.

[25] https://www.microsoft.com/en-us/microsoft-365/microsoft-teams/group-chat-software

Table 9. Results for different recommendation system approaches.

Model name	Accuracy (%)	Precision	Recall	F1
Decision Tree	50.32	0.82	0.46	0.59
Decision Tree (RF)	44.26	0.79	0.39	0.52
Support Vector Classification	61.29	0.82	0.64	0.72
Support Vector Classification (RF)	47.74	0.86	0.40	0.55
Random Forest	68.39	0.85	0.72	0.78
Random Forest (RF)	49.03	0.82	0.44	0.57
Xgboost	52.90	0.89	0.45	0.60
Xgboost (RF)	41.55	0.87	0.29	0.44
LightGBM	63.10	0.90	0.60	0.72
LightGBM (RF)	45.16	0.70	0.52	0.60
Deep Feedforward Neural Network	78.32	0.89	0.82	0.85
Hybrid Recommender System	79.79	0.94	0.80	0.86

While lower accuracies were to be expected in the final test from the machine learning models, the drop in accuracies using one-hot encoded data was more immense than anticipated. As the accuracies of off-the-shelf machine learning models decreased more than the deep feedforward neural network models, it could be gathered that the increased dimensionality of the data affected more negatively to these models than to the deep feedforward neural network model. Reducing dimensions using SelectKBest method decreased the performance of the models similar to what was observed in the earlier comparison of different machine learning models done during development phase.

The proposed hybrid recommendation system performed the best in terms of accuracy and achieved total accuracy of 79.79% in automated tests. The system did not make any recommendations for 27.87% of the items. The uncertainty was higher with class 0 samples (35.09%) than with class 1 samples (25.83%). This means that the recommendation system was more hesitant to make recommendations regarding not selecting the book and was fairly sure when giving a recommendation to select a book to be added into the collection.

While precision score seemed to be good across all tested models, the hybrid recommendation system performed best with this metric. With recall metric however there were major differences between the models: the deep feedforward neural metric and the hybrid system performed well when compared to other models. The number of false negatives produced was however considerably high with also the deep feedforward neural network model and with the proposed hybrid recommendation system. From point of F1 score the deep feedforward neural network and the proposed hybrid recommendation system performed both adequately while other models were struggling.

To gain further insights about the performance of proposed hybrid recommendation system, further metrics were gathered regarding the rule-based scoring. An analysis of these metrics was conducted for all the different feature types to see if defined heuristics would result into balanced scoring.

All samples in test dataset had author feature included into them. Out of these authors, 568 were unique values and 135 were new for the collection. The proposed recommender system gave positive score for most of the authors as seen in Figure 16. High count of score of 1 can be partly explained by the recommender system's rule that all new authors are given this score by default. While the rule regarding new author scoring is certainly debatable, it serves as a mean to encourage exploration and novelty in collection development. Overall the author scoring mechanism can be considered as sufficiently balanced based on the results.

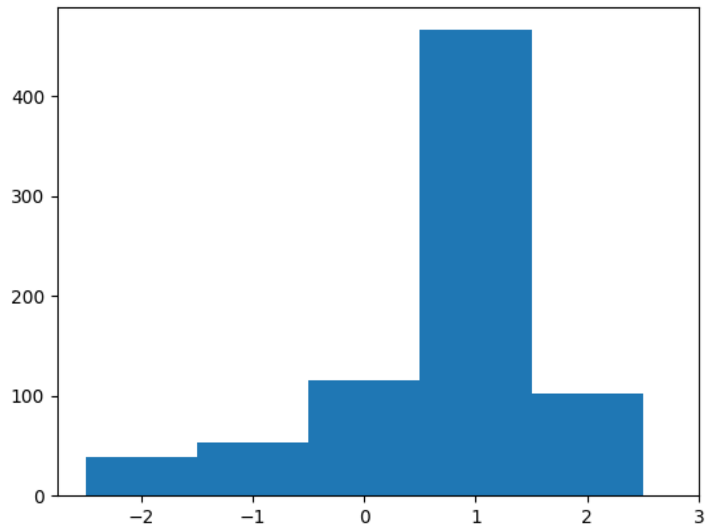

Figure 16. Distribution of author score given by the rule-based system.

There were total of 193 items that had a series feature included. Of these features 127 were unique and 29 were new. Compared to author scores, the scores recommender system assigned for series were using less of the critical score values of -2 and 2 (Figure 17). As new series are neither punished nor rewarded this affects the high count of 0 values in scoring. It can be debated whether new series should receive a negative score as acquiring part of series can be seen as larger commitment than acquiring a book that is not associated with any series. As all possible scores were given regarding series feature, it can be inferred that the heuristics regarding series feature scoring work properly.

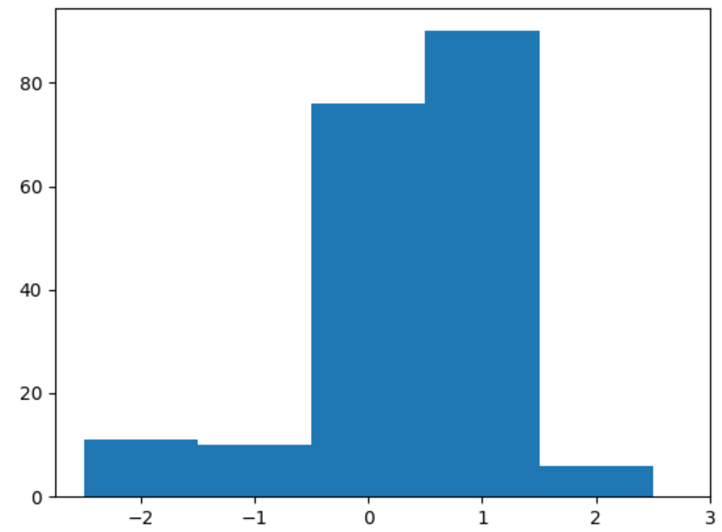

Figure 17. Distribution of series score given by the rule-based system.

Regarding the publisher feature (Figure 18): the feature was present in all samples in the test dataset. There were total of 127 distinct values for the publisher feature of which 25 were considered new as they were not present in the database the recommender system was using. Most of the scores assigned for publishers were positive and the number of +2 positive scores was the highest of all the features. This indicates that most of the books that were acquired were probably published by big publishers as requirement for any feature to receive +2 score was to have item count in collection belonging to the top 10% out of all publishers. Low amount of -1 score with publisher feature indicates that the scoring for publishers could need further tuning. However, to confirm this, further tests should be made with different datasets.

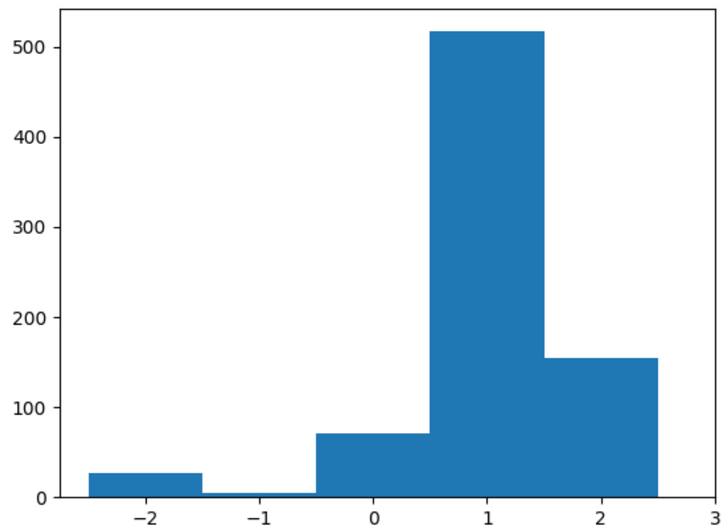

Figure 18. Distribution of publisher score given by the rule-based system.

A list of genre features was available in 557 samples in the test set. Genres had the lowest number of new values with only 23 genres belonging to this category out of 146 unique genres found from the test set. Scores assigned based on genres were the lowest of all features (Figure 19). The reason for low scoring may very well be that the threshold of lower 50% of the feature population receiving a negative score is too high when compared to the number of unique values available. On the other hand, since author, series and publisher features had distributions that lean on the positive side of the scoring table, the genre scoring may very well act as a rule that balances otherwise too optimistic system.

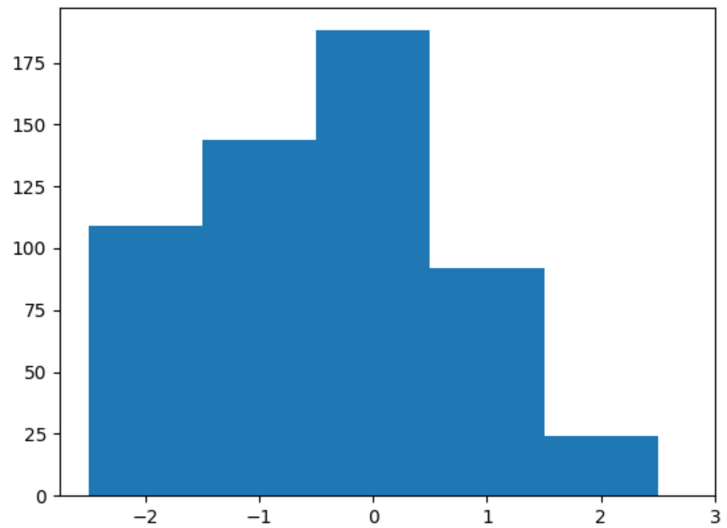

Figure 19. Distribution of genre score given by the rule-based system.

Subject features had the most of new values (n=140) out of all features. At least one subject was included to 734 of the test samples. Total of 2528 unique subjects were found in all test samples. The distribution of scores with this feature was more even than with any other feature as seen in Figure 20. This indicates that the list-based scoring system seems to result in balanced scoring with subject feature.

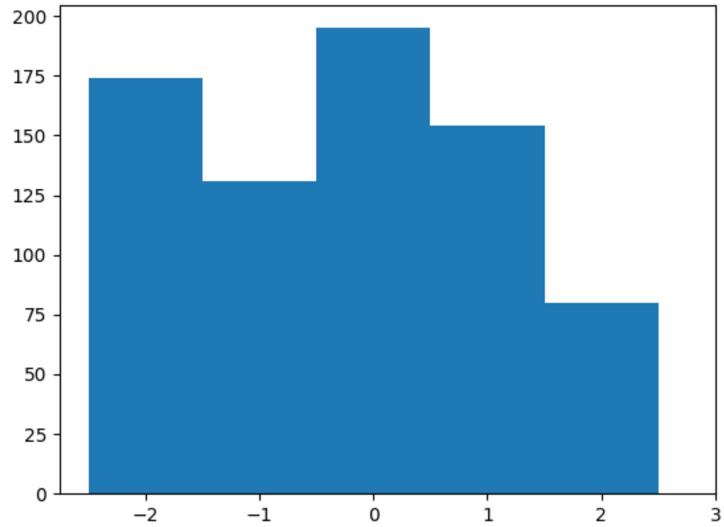

Figure 20. Distribution of subject score given by the rule-based system.

5.3 Focus group evaluation

The findings from the focus group can be summarized under four distinct headings: librarians' interest towards solution, librarians' opinion about solution, librarians' improvement ideas for solution, and problems in solution perceived by librarians. Results regarding each of these four headings are discussed next.

5.3.1 Librarians' interest towards solution

All focus group members were interested in the proposed solution. Furthermore, a general interest towards solutions that help in collection development was observed. The demonstrated artifact was perceived like a light bulb or a bridge by the participants. Group members who compared the solution to light bulb noted that while light bulb can always be perceived to be helpful, its helpfulness depends on its power or the number of light bulbs emitting the light. One member also noted that *"great things can also be achieved in the dark"* that can be interpreted to reflect that it is also possible to succeed in collection development without the use of any additional tools. Similar theme could be found from the answer of another focus group participant, who associated the type of solution to be like bridge which makes crossing a river easier but is not mandatory as you can also swim to the other side. All in all, according to the focus group results librarians perceived this new tool in a positive, although careful, manner.

5.3.2 Librarians' opinion about solution

Focus group members were united on that the logic of defined heuristics was understandable and the recommendations were reasonable. However, questions were raised regarding the optimal use case of the solution and whether the solution would work in all types of collections: small, big, shared, and floating. Overall, the conclusion that can be made from the observed discussion is that collection development tools need to be more flexible and customizable than the proposed solution.

Each participant associated the solution with different food. One participant reflected the prototype as sushi: *"delicious, yet raw"*. Another positive statement was by participant who reflected the prototype as soup. The justification for this association was that *"while soup keeps you going, often it is not anything mind-blowing"*. The solution was also seen from data-centric point of view by one participant who compared the solution to a vegetable dish: *"it allows person to select what he or she wants to highlight regarding the selection process"*. From the comments it can be gathered that while the solution is seen as a potential step towards right direction by the librarians, it is not yet observed as ready to be deployed.

The solution was associated with visual components regarding car upgrade associations. Parts picked by the participants were either backup camera or headlights. Opinions regarding whether the tool is more about looking backwards or forwards were mixed. One comment regarding this discussion found even a philosophical aspect out of this viewpoint: *"if you look to backup camera while backing up, you are technically looking forward"*. An important note is that no participant chose a luxury upgrade, such as leather seats or aluminum rims, from the available options. From this it can be concluded that the solution is seen as a meaningful upgrade for work instead of a nice-to-have.

Interesting characteristic regarding how the focus group participants received the solution was that there was not a single comment regarding the machine learning, deep learning, or artificial intelligence aspect of the tool. One group member asked a question regarding the meaning of the thumb symbol in the user interface but did not comment further after receiving answer. Lack of comments in this domain may reflect that the topic is still very unknown to librarians. It is also possible that as the machine learning contribution to the overall system was limited, it was not considered to be important by focus group participants.

5.3.3 Librarians' improvement ideas for solution

Members of the focus group came up with multiple ideas regarding how the solution could be enhanced. First, a strong implication was placed on that the rule-based system should be customizable. While the rule-based system was observed to be reasonably well-working and essential for this type of system, all members of the focus group acknowledged that different collections have different needs and for a tool to be useful in them a certain level of customization regarding tool configuration should be included into the solution. During the discussion, the concern regarding narrowing collection accidentally by blindly using tools such as the solution was expressed by one of the participants. From this it can be inferred that librarians are not willing to place trust regarding collection development solely to the hands of a computer system.

Improvement and further options for customization were also hoped regarding data visualization. One participant noted that because of lack of experience in using data visualization tools, it was difficult to point out how the visualization would need to be extended. This comment reinforces the earlier presumption that not all librarians have had access to data visualization tools despite them existing in the market.

5.3.4 Problems in solution perceived by librarians

While the interest towards solutions like the prototype was observed during the focus group, the participants also presented statements and questions regarding problems that may rise when using systems of this type. For example, all participants noted that collections limits go well beyond one library building: multiple libraries may have shared collections and thus collections need to be investigated from broader perspective than what is available in one library. In addition, items located in library archives that are not part of open collections need to be also acknowledged. The main problem of the proposed solution was seen to be that it works only in very limited number of use cases.

Another problem brought up in the discussion was that while the solution does make a recommendation, it is not able to tell how many items should be ordered. Participants saw that the number of items to be acquired is crucial information for bigger libraries or libraries that have shared collections. In context of these larger collections it was questioned if the solution provides any value to the selection process as in most cases it is already known that certain books are selected, and the real question is how many of the selected items should be ordered. Based on the results, item-level approach seems to more desirable than entity-level approach for big libraries and libraries with shared collection when it comes down to making recommendations regarding book selection.

Discussion about the third-party nature of the solution was short but impactful. From the start, it was made clear by one participant that it was a positive thing that the solution was going to be released as free open-source tool. This indicates that cost of software solutions is still a problem for libraries today and reinforces that this may be also the reason behind why artificial intelligence solutions have not been adopted in libraries in large scale. While it was not considered to be an overwhelming problem to have external tools that would need to be introduced to staff, all participants agreed on that collection management and development tools should be included as part of integrated library systems instead of being standalone software if possible. From this it can be deducted that new tools aimed for library collection development should seek to integrate to existing software platforms instead of being developed as standalone software.

6. Conclusion

This thesis sought to answer the research question of *how to implement a recommendation system that can assist in book selection in context of public libraries print collections* using design science research methodology. The following sections discuss the results and limitations of this thesis in addition of the directions for future research. Finally, closing remarks of this thesis are presented.

6.1 Discussion

The proposed hybrid recommendation system performed well when compared against traditional machine learning methods. However, the question how to improve the system further remains. Since the proposed system heavily relies on data, it is beneficial to start the discussion from how different aspects of data affected the proposed system.

Difficulties encountered during data gathering and preprocessing phases suggest that there is room for improvement regarding both the representation and quality of the book metadata. For example, many duplicate values were found from the dataset used in this research and the dataset size can be considered to have been reasonably small. The problems in data quality raises the question of how much the data quality affected the quality of the final solution. The prior belief that having clean data is essential for developing artificial intelligence or machine learning solutions (IBM, 2018, p. 12) suggests that the accuracy of the final solution could have been better if better quality data would have been used.

Problems with metadata quality are not a new problem in the domain of libraries. For example, in Finland there exists multiple expert groups, such as cataloging expert group[26] and descriptive metadata expert group[27], that work towards the unified goal of producing better quality metadata. In addition, controlled vocabularies and authority records have been established as means to unify the data in an effective manner in global context (see e.g. Harper & Tillett, 2007). The findings of this thesis support the already established idea that quality control for metadata is essential and expand this by noting that for machine learning solutions the value lies also in transforming the old data to meet the new standards so that all data would be of near equal quality.

Having good quality metadata however seems to solve only half of the problems related to data processing when building a recommendation system for library book selection. As database building for the proposed solution took nearly 7 hours with a considerably small dataset, it indicates there is a larger problem considering knowledge representation that needs to be addressed. While relational databases can represent the links between entities, slow querying and building times rise a question if other database solutions should be used instead in artificial intelligence solutions. The findings from

[26] https://www.kiwi.fi/pages/viewpage.action?pageId=58493320

[27] https://www.kiwi.fi/pages/viewpage.action?pageId=80155685

this thesis suggest that alternative database solutions, such as graph databases, should be investigated further when solving similar types of problems. In domain of libraries the movement towards using linked data (e.g. Xu, Hess, & Akerman, 2018) may bring a solution to this problem in future. Furthermore, collaborative metadata repositories may become increasingly important in future to ensure same entities could be harvested and used in different applications. Overall, the findings of this thesis suggest that the field of libraries need to overcome the challenges regarding knowledge representation before recommendation system solutions for collection development can be applied effectively.

The high variance in accuracy between the tests regarding machine learning models can be explained with the problem in learnability when there are thousands of different attributes as presented by Kubat (2017, p. 204). As the data that was used to make recommendations consisted of thousands of different attributes, it can be asked if it became difficult for a tree-based machine learning algorithm to construct a reliable model after a certain threshold regarding number of attributes was reached. Since restricting used features to the ones gathered using dimensionality reduction technique did not yield better results, it leaves an unanswered question whether such techniques can be successfully applied in domain of library book selection when using book metadata.

Since neural networks can solve increasingly complicated problems by utilizing the simplified representation (Goodfellow et al., 2016, pp. 5, 12), it explains the lack of observable threshold regarding number of attributes in the tests that were conducted. The deep feedforward neural network model was able to produce very consistent accuracy despite the data being high dimensional and low in sample size. The results support the observations from previous research showing deep neural networks can be used effectively with high dimensional low sample size data (Liu, Wei, Zhang & Yang, 2017). The question regarding how larger dataset size would affect the deep learning model’s performance in this context remains unsolved. However, the consensus is that scaling the training dataset improves accuracy (Hestness et al., 2017, p. 11). Overall, results obtained in this thesis are not sufficient to make generalizable conclusions whether tree-based approached are universally worse than using deep neural networks when using high dimensional low sample size data. Furthermore, no whitepapers comparisons in this context were found. For book selection recommendations, deep feedforward neural networks seem to be much promising than traditional supervised machine learning algorithms based on the results of this thesis.

Classifying the final recommendation system under any one of the types that were presented in Chapter 2 is problematic. It can be argued that the system’s base logic is closest to content-based filtering: if library patrons have liked items with certain features in the past, the system will recommend items with these features given that the interest is greater than the number of items containing the feature. However, since additional points are given for certain new features, the system actively encourages exploration which is a known problem in recommendation systems using content-based filtering.

There also exist hybridization in the system as the total score is calculated based on the score given by total of six components. If the machine learning models output is thought as an additional feature, the way of building the recommendation is in line with the description of feature combination archetype of hybrid recommendation system. Alternatively, it is possible to view the recommendation score calculations for each feature as individual recommendation systems. This view is supported as the logic for scoring each feature type is different. If the overall recommendation system is seen like this, it can be argued that the type of system would fit the mixed type of hybrid recommendation systems. System architecture description from this perspective would include six content-based recommendation systems that have been parallelized. The difficulty regarding decisively classifying the recommendation system under any of the classes presented by Burke (2007), Jannach et al. (2011) and Aggarwal (2016) rises a question whether the definitions have too much overlap between the different classes.

An additional benefit of developing the recommendation system using hybridization of rule-based approach and machine learning approach is that the recommendations can be explained clearly to the user. Since data visualization was seen as a positive feature by the librarians, it can be suggested based on the results that even in cases where machine learning model is solely used for making recommendations for library book selection, visualization aspect of the data used to train the machine learning model should be considered. As librarians are hesitant to place trust solely to a recommendation system, the role of system's transparency should not be left unnoticed when developing systems for this domain.

Regarding the challenges observed in recommendation systems it can be said that the final artifact heavily suffers from cold start problem. As both the rule-based component and machine learning component rely heavily on both available metadata and available circulation data it is very plausible that the system would be unusable in situation where there is only little data available. This raises an important note regarding the importance of preserving library's circulation log data as well as book metadata. This type of data is crucial for development of systems similar to what was proposed in this thesis.

Additional problems rise when the memory usage and sustainability of the proposed hybrid recommendation system is considered. Since the rule-based system heavily utilizes memory when calculating scores and both components responsible of contributing to the total score of the recommendation are dependent on new builds when data is updated, the system suffers from problems of both memory-based and model-based recommendation solutions. The impact of having these downsides comes largely down to how many times the data should be updated in this type of recommendation system to keep the system's recommendations relevant. To answer this question more research would be needed.

The usefulness of the artifact in real-world was challenging to evaluate. The results of the automatic tests show that there clearly is room for improvement. Still, the value for the system comes also partly from the transparency of the recommendations. The evaluation done by the focus group tells a story regarding the interest towards using tools to enhance processes in collection development. However, there are multiple challenges that must be addressed before new tools, such as the artifact, can be deployed to libraries. All in all, when recommendation systems are developed for collection development, attention should be placed also to user perspective.

The results of this study partly reinforce the results from Ex Libris (n.d., p. 12) study: budget continues to be a challenge when it comes down to libraries adopting new technologies. Since open-source solutions in domain of integrated library systems have started to gain more and more market share (see e.g. Breeding, 2020), there are now more integration opportunities for new tools in libraries than ever before. For example, new modular open-source library system FOLIO[28] has an ecosystem build around integrating apps to the core system. Question remains how integrations with artificial intelligence solutions should be arranged with these systems as they all use traditional

[28] https://www.folio.org/

means to represent data which was observed to be problematic for scaling artificial intelligence solutions effectively in the results of this study.

According to the focus group results tools developed to help librarians in collection development processes should aim to be customizable as possible. This raises a question whether machine learning systems are good system type for enhancing collection development process as making them customizable is not an easy task. With rule-based system it is easier to make system more configurable, but recommendations might miss some links between features that machine learning system are able to recognize. Hybrid recommendation systems, such as the proposed system, offer a way to balance these two aspects but along comes a lot of design decisions that have impact on the system performance, transparency, and accuracy. Since there are multiple ways to construct a hybrid system, more research is needed before one type of hybrid system can be said to perform better than another in the context of library book selection.

6.2 Limitations

There are number of limitations to the research that was conducted and the results that were reported in this thesis should be interpreted while acknowledging these limitations. First, the motivation and the research approach rise from the author's own interest and expertise in the field of libraries. Therefore, it can be questioned if the author's own viewpoints have affected the perspective of where the solution to the problem was approached.

Second, the accuracy of the final artifact was evaluated using only one part of one public library's collection. Furthermore, only physical collections and loans were considered when building the artifact and statistics such as loans for electronic resources or renewals for physical resources were not considered. These aspects limit the ability to generalize the tools performance and more automated testing with different types of collections would be needed to draw conclusive conclusions.

Finally, the focus group evaluating the tool consisted of three members and as such, the sample size of user evaluation does represent only a small fraction of the real userbase. While the focus group considered the tool to be interesting and worth developing further, the majority of the userbase may very well think of the opposite. Not only the differences between individual collection librarians may affect the result of evaluation, but also the context where the tool is used. In this thesis, the tool was only evaluated from point of view of Finnish public libraries which excludes other library types and other nationalities.

6.3 Future research

The solution proposed in this research helps librarians only with one of the multiple processes in managing library's collections. Furthermore, this research just scratches the surface of recommending books for purposes of developing library's collection. Further research regarding how to implement intelligent systems that can produce value out of the data stored into library systems is needed in addition to validating the proposed solution with help of varying datasets. As the final artifact of this thesis can be enhanced and extended, research regarding user-interface design, knowledge representation, and tool deployment is suggested in order to reinforce the current solution and gain information on how these topics should be approached in the context of collection development.

Given that the field of libraries is looking to linked data in the future, research regarding use of data models such as BIBFRAME[29] as starting point for artificial intelligence or machine learning solution is suggested. Also, a lot of value remains in the data cataloged tens of years ago. Future research could therefore investigate computationally effective ways of transforming old data to meet the new standards. Design science research is proposed for future research regarding data transformations as produced artifacts could support real-world work tasks such as feature deduplication.

6.4 Closing remarks

This research showed that new tools to enhance collection development processes can be developed using the data saved in integrated library systems and that librarians are willing to try out these newly developed tools. Implementing tools for the domain of libraries however comes with special requirements. With recommendation systems, this seem to mean that new types of old problems need to be solved. For example, while scalability issues are mainly associated with collaborative filtering systems (Jain et al., 2015, p. 957) in context of libraries the scalability issues may rise from the need of transforming data to a form that provides access to connections between data points. In addition, there is the problem of data sparsity (e.g. Khusro et al., 2016, pp. 1184-1185) but it is observed differently than in collaborative filtering setting: first, the circulation data is sparse and second, the data regarding item features is sparse. The feature sparsity which results into book metadata being high-dimensional data is especially problematic.

Finally, data privacy offers an interesting and important viewpoint when considering the development of systems like what was produced as output artifact in this thesis. By default the circulation data of libraries is a well-guarded secret within the library systems. However, if all identification factors of individual library patrons are deleted from the circulation data, it suddenly becomes just an individual portion of the library circulation statistics, that at least in Finland are made publicly available[30]. Since open-source library system vendors are encouraging developers to develop new features for the systems (e.g. The Open Library Foundation, 2020), it is worth questioning how big impact it would make if the circulation data would be made publicly available?

[29] https://www.loc.gov/bibframe/docs/index.html

[30] https://tilastot.kirjastot.fi/intro.php?lang=fi

References

Adams, B., & Noel, B. (2008). Circulation Statistics in the Evaluation of Collection Development. *Collection Building 27*(2), 71-73. https://doi.org/10.1108/01604950810870227

Aggarwal, C. C. (2016). *Recommender Systems*. Springer. https://doi.org/10.1007/978-3-319-29659-3

Ali, Z., Khusro, S., & Ullah, I. (2016). A Hybrid Book Recommender System Based on Table of Contents (ToC) and Association Rule Mining. *Proceedings of the 10th International Conference on Informatics and Systems - INFOS '16. the 10th International Conference*, 68-74. https://doi.org/10.1145/2908446.2908481

Alpaydin, E. (2014). *Introduction to Machine Learning (3rd edition)*. MIT Press.

Alpaydin, E. (2016). *Machine Learning: The New AI*. MIT Press.

Antoniou, A., Storkey, A., & Edwards, H. (2018). *Data Augmentation Generative Adversarial Networks*. arXiv preprint arXiv:1711.04340.

Azaria, A., Hassidim, A., Kraus, S., Eshkol, A., Weintraub, O., & Netanely, I. (2013). Movie Recommender System for Profit Maximization. *Proceedings of the 7th ACM Conference on Recommender Systems - RecSys '13. the 7th ACM conference*, 121-128. https://doi.org/10.1145/2507157.2507162

Baba, K., Minami, T., & Nakatoh, T. (2016). Predicting Book Use in University Libraries by Synchronous Obsolescence. *Procedia Computer Science, 96*, 395-402. https://doi.org/10.1016/j.procs.2016.08.082

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, D., & Blundell, C. (2020). *Agent57: Outperforming the Atari Human Benchmark*. arXiv preprint arXiv:2003.13350.

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design Science Research Contributions: Finding a Balance between Artifact and Theory. *Journal of the Association for Information Systems, 19*(5), 358-376. https://doi.org/10.17705/1jais.00495

Borrego, P., & Lewellen, R. (2014). Collection Development and Data Visualization: How Interactive Graphic Displays Are Transforming Collection Development Decisions. *Proceedings of the Charleston Library Conference*, 549-556. http://dx.doi.org/10.5703/1288284315637

Breeding, M. (2020). Map of Libraries: the Library Automation System Used is Koha. Retrieved November 3rd, 2020, from https://librarytechnology.org/libraries/map.pl?ILS=Koha

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, 43-52.

Burke, R. (2000). Knowledge-based Recommender Systems. *Encyclopedia of library and information systems, 69*(Supplement 32), 175-186.

Burke, R. (2002). Hybrid Recommender Systems: Survey and Experiments. *User modeling and user-adapted interaction, 12*(4), 331-370.

Burke, R. (2007). Hybrid Web Recommender Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web* (pp. 377-408). Springer.

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends. *Neurocomputing, 408*, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. https://doi.org/10.1145/2939672.2939785

Claesen, M., & De Moor, B. (2015). *Hyperparameter Search in Machine Learning*. arXiv preprint arXiv:1502.02127.

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *Proceedings of the 10th ACM Conference on Recommender Systems. RecSys '16: Tenth ACM Conference on Recommender Systems*, 191-198. https://doi.org/10.1145/2959100.2959190

Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google News Personalization: Scalable Online Collaborative Filtering. *Proceedings of the 16th international conference on World Wide Web*, 271-280.

Dietterich, T. (1995). Overfitting and Undercomputing in Machine Learning. *ACM Computing Surveys, 27*(3), 326-327. https://doi.org/10.1145/212094.212114

Drachsler, H., Hummel, H., & Koper, R. (2007). Recommendations for Learners Are Different: Applying Memory-based Recommender System Techniques to Lifelong Learning. *Proceedings of the 1st Workshop on Social Information Retrieval for Technology-Enhanced Learning & Exchange*, 18-26.

Doshi-Velez, F., & Kim, B. (2017). *Towards a Rigorous Science of Interpretable Machine Learning*. arXiv preprint arXiv:1702.08608.

Dou, Y., Yang, H., & Deng, X. (2016). A Survey of Collaborative Filtering Algorithms for Social Recommender Systems. *2016 12th International Conference on Semantics, Knowledge and Grids (SKG)*, 40-46. https://doi.org/10.1109/skg.2016.014

Eaton, M. (2016). Seeing Library Data: A Prototype Data Visualization Application for Librarians. *Journal of Web Librarianship, 11*(1), 69-78. https://doi.org/10.1080/19322909.2016.1239236

Edelman, H. (1979). Selection Methodology in Academic Libraries. *Library Resources & Technical Services, 23*(1), 33-38.

Evans, G. E., & Saponaro, M. Z. (2012). *Collection Management Basics (6th edition)*. ABC-CLIO, LLC.

Ex Libris. (n.d.). Artificial Intelligence in the Library: Advantages, Challenges and Tradition. Retrieved June 29, 2020, from https://cdn2.hubspot.net/hubfs/2909474/Ex%20Libris%20Artificial%20Intelligence%20White%20Paper.pdf

Fieldhouse, M. (2012). The Processes of Collection Management. In Fieldhouse, M., & Marshall, A. (Eds.), *Collection Development in the Digital Age* (pp. 27-43). Facet Publishing.

Finch, J. L., & Flenner, A. R. (2016). Using Data Visualization to Examine an Academic Library Collection. *College & Research Libraries, 77*(6), 765-778. https://doi.org/10.5860/crl.77.6.765

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining*. Springer. https://doi.org/10.1007/978-3-319-10247-4

Geyik, S. C., Guo, Q., Hu, B., Ozcaglar, C., Thakkar, K., Wu, X., & Kenthapadi, K. (2018). Talent Search and Recommendation Systems at LinkedIn: Practical Challenges and Lessons Learned. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1353-1354. https://doi.org/10.1145/3209978.3210205

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *Communications of the ACM, 35*(12), 61-70.

Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System. *ACM Transactions on Management Information Systems, 6*(4), 1-19. https://doi.org/10.1145/2843948

Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). Combining Collaborative Filtering with Personal Agents for Better Recommendations. *Proceedings of AAAI'99 conference*, 439-446.

Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press. Retrieved August 22nd, 2020 from https://www.deeplearningbook.org

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Nets. *Advances in Neural Information Processing Systems 27*.

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. *MIS quarterly, 37*(2), 337-355. https://doi.org/10.25300/MISQ/2013/37.2.01

Gregory, V. L. (2019). *Collection Development and Management for 21st Century Library Collections: An Introduction (2nd edition)*. American Library Association.

Guo, C., & Berkhahn, F. (2016). *Entity Embeddings of Categorical Variables*. arXiv preprint arXiv:1604.06737.

Guo, G., Zhang, J., & Thalmann, D. (2014). Merging Trust in Collaborative Filtering to Alleviate Data Sparsity and Cold Start. *Knowledge-Based Systems, 57*, 57-68. https://doi.org/10.1016/j.knosys.2013.12.007

Hammais, E., Ketamo, H., & Koivisto, A. (2019). *Virtual Information Assistants on Mobile App to Serve Visitors at Helsinki Central Library Oodi*. World Library and Information Congress: 85th IFLA General Conference and Assembly, Athens, Greece. http://library.ifla.org/2536/

Harper, C. A., & Tillett, B. B. (2007). Library of Congress Controlled Vocabularies and Their Application to the Semantic Web. *Cataloging & Classification Quarterly, 43*(3–4), 47-68. https://doi.org/10.1300/j104v43n03_03

Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., ... & Zhou, Y. (2017). *Deep Learning Scaling is Predictable, Empirically*. arXiv preprint arXiv:1712.00409.

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems, 19*(2), 87-92.

Hevner, A. R., March, S.T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research, *MIS Quarterly, 28*(1), 75-105. https://doi.org/10.2307/25148625

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural computation, 9*(8), 1735-1780.

Horava, T., & Levine-Clark, M. (2016). Current trends in collection development practices and policies. *Collection Building, 35*(4), 97-102. https://doi.org/10.1108/cb-09-2016-0025

Hussein, T., Linder, T., Gaulke, W., & Ziegler, J. (2014). Hybreed: A software framework for developing context-aware hybrid recommender systems. *User Modeling and User-Adapted Interaction, 24*(1-2), 121-174. https://doi.org/10.1007/s11257-012-9134-z

IBM. (2018). Beyond the hype: A guide to understanding and successfully implementing artificial intelligence within your business. Retrieved October 15th, 2020, from https://www.ibm.com/downloads/cas/8ZDXNKQ4

Iivari, N., & Kuutti, K. (2017). Towards Critical Design Science Research. *Proceedings of International Conference on Information Systems 2017*.

Iivari, J. & Venable, J. (2009). Action research and design science research - Seemingly similar but decisively dissimilar. *Proceedings of the 2009 European Conference on Information Systems*.

Iqbal, N., Jamil, F., Ahmad, S., & Kim, D. (2020). Toward Effective Planning and Management Using Predictive Analytics Based on Rental Book Data of Academic Libraries. *IEEE Access, 8*, 81978-81996. https://doi.org/10.1109/access.2020.2990765

Jabbar, H., & Khan, R. Z. (2015). Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study). *Computer Science, Communication and Instrumentation Devices*, 163-172.

Jain, S., Grover, A., Thakur, P. S., & Choudhary, S. K. (2015). Trends, problems and solutions of recommender system. *International Conference on Computing, Communication & Automation*, 955-958. https://doi.org/10.1109/ccaa.2015.7148534

Jannach, D., Zanker, M., Felfernig, A. & Friedrich, G. (2011). *Recommender systems: an introduction*. Cambridge University Press.

Johnson, P. (2018). *Fundamentals of Collection Development and Management (4th edition)*. American Library Association.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science, 349*(6245), 255-260. https://doi.org/10.1126/science.aaa8415

Kaggle. (n.d.). Kaggle's State of Data Science and Machine Learning 2019. Retrieved August 23rd, 2020, from https://www.docdroid.com/qzyxCr4/kaggle-state-of-data-science-and-machine-learning-2019-pdf

Kansalliskirjasto. (n.d.). Julkaisutilasto Kirjat 2000- [Statistics of published books 2000-]. Retrieved November 16th, 2020, from https://www.doria.fi/bitstream/handle/10024/123264/Julkaisutilasto%20Kirjat%202000-.pdf?sequence=10&isAllowed=y

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 3149-3157.

Kelleher, J. D. (2019). *Deep Learning*. MIT Press.

Khusro, S., Ali, Z., & Ullah, I. (2016). Recommender systems: issues, challenges, and research opportunities. *Information Science and Applications (ICISA) 2016*, 1179-1189. Springer Singapore. https://doi.org/10.1007/978-981-10-0557-2_112

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised Learning with Deep Generative Models. *Advances in Neural Information Processing Systems*, 3581-3589.

Kirjastot.fi. (n.d.). Suomen yleisten kirjastojen tilastot [Statistics of public libraries in Finland]. Retrieved November 16th, 2020, from https://tilastot.kirjastot.fi/?orgs=1&years=2010%2C2019&stats=33%2C101#results

Knievel, J. E., Wicht, H., & Connaway, L. S. (2006). Use of Circulation Statistics and Interlibrary Loan Data in Collection Management. *College & Research Libraries, 67*(1), 35-49.

Kotsiantis, S, B., Kanellopoulos, D., & Pintelas, P. E. Data Preprocessing for Supervised Leaning[31]. *International Jounral of Computer Science 1*(1), 111-117.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica 31*(3), 249-268.

[31] The published article includes this typo.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM, 60*(6), 84–90. https://doi.org/10.1145/3065386

Krulwich, B. (1997). Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data. *AI magazine, 18*(2), 37-46.

Kubat, M. (2015). *An Introduction to Machine Learning*. Springer.

Kubat, M. (2017). *An Introduction to Machine Learning (2nd edition)*. Springer.

Kuncheva, L. I. (2014). *Combining Pattern Classifiers: Methods and Algorithms (2nd edition)*. John Wiley & Sons, Inc.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature 521*, 436-444.

Li, P., & Yamada, S. (2004). A Movie Recommender System Based on Inductive Learning. *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, 318-323. https://doi.org/10.1109/iccis.2004.1460433

Library Journal. (2018). Academic Library Collection Development Survey. Retrieved August 21, 2020, from https://s3.amazonaws.com/WebVault/research/2018_AcademicLibraryCollectionDevelopment.pdf

Liu, B., Wei, Y., Zhang, Y., & Yang, Q. (2017). Deep Neural Networks for High Dimension, Low Sample Size Data. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2287-2293.

Lops, P., De Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State ofthe Art and Trends. In Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.), *Recommender Systems Handbook* (pp. 73-105). Springer. https://doi.org/10.1007/978-0-387-85820-3_3

Lyngsoe Systems. (2019). Intelligent Material Management System. Retrieved June 29, 2020, from https://lyngsoesystems.com/intelligent-material-management-system/

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*(4), 251-266. https://doi.org/10.1016/0167-9236(94)00041-2

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Mitchell, T. M., (2006). The Discipline of Machine Learning. Carnegie Mellon University. Retrieved August 23rd, 2020, from http://ra.adm.cs.cmu.edu/anon/usr0/ftp/anon/ml/CMU-ML-06-108.pdf

MIT Technology Review. Machine Learning: The New Proving Ground for Competitive Advantage. Retrieved August 22nd, 2020, from https://s3.amazonaws.com/files.technologyreview.com/whitepapers/MITTR_GoogleforWork_Survey.pdf

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). *Playing Atari with Deep Reinforcement Learning*. arXiv preprint arXiv:1312.5602.

Mohammed, M., Khan, M. & Bashier, E. B. M., (2017). *Machine Learning: Algorithms and Applications*. Auerbach Publications.

Mohri, M., Rostamizadeh, A. & Talwalkar, A. (2012). *Foundations of Machine Learning*. MIT Press.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics, 7*. https://doi.org/10.3389/fnbot.2013.00021

Netflix, Inc. (2009). Netflix Prize. Retrieved June 16, 2020, from https://www.netflixprize.com/

OCLC. (2020a). GreenGlass: Manage print monographs to free library space. Retrieved July 3rd, 2020, from https://www.oclc.org/en/greenglass.html

OCLC. (2020b). WorldShare Collection Evaluation. Retrieved August 21, 2020, from https://www.oclc.org/content/dam/oclc/services/brochures/215024-WWBE_WorldShare-Collection-Evaluation-Flier.pdf

Offermann, P., Blom, S., Schönherr, M., & Bub, U. (2010). Artifact Types in Information Systems Design Science - A Literature Review. *International Conference on Design Science Research in Information Systems*, 77-92. https://doi.org/10.1007/978-3-642-13335-0_6

Okon, E. U., Eke, B. O., & Asagba, P. O. (2018). An Improved Online Book Recommender System using Collaborative Filtering Algorithm. *International Journal of Computer Applications, 179*(46), 41-48.

Padilla, T. (2019). Responsible Operations: Data Science, Machine Learning and AI in Libraries. OCLC Research. https://doi.org/10.25333/XK7Z-9G97

Pazzani, M. J. (1999). A Framework for Collaborative, Content-Based and Demographic Filtering. *Artificial Intelligence Review, 13(5-6)*, 393-408.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2008). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-77. https://doi.org/10.2753/MIS0742-1222240302

Perrier, A. (2017). *Effective Amazon Machine Learning*. Packt Publishing.

Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications, 97*, 205-227. https://doi.org/10.1016/j.eswa.2017.12.020

Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. arXiv preprint arXiv:1811.12808.

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

Rendle, S., & Schmidt-Thieme, L. (2008). Online-Updating Regularized Kernel Matrix FactorizationModels for Large-Scale Recommender Systems. *Proceedings of the 2008 ACM Conference on Recommender Systems*, 251-258.

Resnick, P., & Varian, H. R. (1997). Recommender Systems. *Communications of the ACM, 40*(3), 56-58.

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender Systems: Introduction and Challenges. In Ricci, F., Rokach, L., & Shapira, B. (Eds.), *Recommender systems handbook* (pp. 1-34). Springer. https://doi.org/10.1007/978-1-4899-7637-6_1

Robillard, M., Maalej, W., Walker, R. J. & Zimmermann, T. (2014). Recommendation Systems in Software Engineering. Springer. https://doi.org/10.1007/978-3-642-45135-5

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386-408.

Rudin, C. (2019). *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*. arXiv preprint arXiv:1811.10154.

Russell, S. & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach (3rd edition)*. Pearson Education Inc.

Safoury, L., & Salah, A. (2013). Exploiting User Demographic Attributes for Solving Cold-Start Problem in Recommender System. *Lecture Notes on Software Engineering, 1*(3), 303-307. https://doi.org/10.7763/lnse.2013.v1.66

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the 10th international conference on World Wide Web*, 285-295.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In Brusilovsky, P., Kobsa, A., & Nejdl, W. (Eds.), *The Adaptive Web* (pp. 291-324). Springer.

Schafer, J. B., Konstan, J., & Riedl, J. (1999). Recommender Systems in E-Commerce. *Proceedings of the 1st ACM conference on Electronic commerce*, 158-166.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Sharma, R., Gopalani, D., & Meena, Y. (2017). Collaborative Filtering–Based Recommender System: Approaches and Research Challenges. *3rd IEEE International Conference on Computational Intelligence and Communication Technology*, 1-6. https://doi.org/10.1109/ciact.2017.7977363

Shaw, W. (2011). Collection development policies for the digital age. In Fieldhouse, M., & Marshall, A. (Eds.), *Collection Development in the Digital Age* (pp. 165-180). Facet Publishing.

Schedl, M., Zamani, H., Chen, C. W., Deldjoo, Y., & Elahi, M. (2018). Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval, 7*(2), 95-116. https://doi.org/10.1007/s13735-018-0154-2

Springshare. (2020). LibInsight - analyze library services and make more informed service decisions. Retrieved August 21st, 2020, from https://www.springshare.com/libinsight/

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence 2009*, 1-19. https://doi.org/10.1155/2009/421425

Suominen, O. (2019). Annif: DIY automated subject indexing using multiple algorithms. Retrieved November 5th, 2020, from https://www.doria.fi/bitstream/handle/10024/169004/Annif%20DIY%20automated%20subject%20indexing%20using%20multiple%20algorithms.pdf?sequence=1

Tilastokeskus. (2007). Kirjan menestystarinoita [Success stories of books]. Retrieved November 16th, 2020, from https://www.stat.fi/artikkelit/2007/art_2007-06-15_001.html?s=0

The Open Library Foundation. (2020). Welcome to FOLIO software development. Retrieved November 3rd, 2020, from https://dev.folio.org/

The Steam Team. (2019). Introducing the Interactive Recommender. Retrieved June 25, 2020, from https://steamcommunity.com/games/593110/announcements/detail/1612767708821405787

Turing, A. (1950). Computing Machinery and Intelligence. *Mind, 49*, 433-460.

Venable, J. (2006). The Role of Theory and Theorising in Design Science Research. *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology*, 1-18.

Wagstaff, K. L., & Liu, G. Z. (2018). Automated Classification to Improve the Efficiency of Weeding Library Collections. *The Journal of Academic Librarianship, 44*(2), 238-247.

Wang, Y., Chan, S. C. F., & Ngai, G. (2012). Applicability of Demographic Recommender System to Tourist Attractions: A Case Study on TripAdvisor. *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 97-101. https://doi.org/10.1109/wi-iat.2012.133

Wheatley, A., & Hervieux, S. (2019). Artificial intelligence in academic libraries: An environmental scan. *Information Services & Use, 39*(4), 347-356. https://doi.org/10.3233/ISU-190065

Xu, A., Hess, K., & Akerman, L. (2018). From MARC to BIBFRAME 2.0: Crosswalks. *Cataloging & Classification Quarterly, 56*(2-3), 224-250. https://doi.org/10.1080/01639374.2017.1388326

Yang, Z., Wu, B., Zheng, K., Wang, X., & Lei, L. (2016). A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications. *IEEE Access, 4*, 3273-3287. https://doi.org/10.1109/access.2016.2573314

Yelton, A. (2018). HAMLET: Neural-Net-Powered Prototypes for Library Discovery. *Library Technology Reports, 55*(1), 10-15.

Zednik, C. (2019). Solving the Black Box Problem: A Normative Framework for Explainable Artificial Intelligence. *Philosophy & Technology*, 1-24. https://doi.org/10.1007/s13347-019-00382-7

Zhang, Y., & Chen, X. (2018). *Explainable Recommendation: A Survey and New Perspectives*. arXiv preprint arXiv:1804.11192.

Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. *National Science Review, 5*(1), 44-53. https://doi.org/10.1093/nsr/nwx106

Appendix A. Consent form for study participants

Oulun yliopisto
Tieto- ja sähkötekniikan tiedekunta
PL 4500
90014 Oulun yliopisto

Master's Thesis: Recommender System for Library Book Selection
TIEDOTE TUTKITTAVILLE JA SUOSTUMUS TUTKIMUKSEEN OSALLISTUMISESTA

1 Tutkijoiden yhteystiedot

Vastuullinen tutkijaopiskelija
Aatu Nykänen
Vanhantullinkatu 6 F93
90100 Oulu
aatu.nykanen@student.oulu.fi
p. 050 493 0511

Tutkielman valvojat:
Minna Isomursu
Tietojenkäsittelytiede
90014 Oulun yliopisto
minna.isomursu@oulu.fi

Umar Farooq
Tietojenkäsittelytiede
90014 Oulun yliopisto
umar.farooq@oulu.fi

2 Tutkittavien oikeudet

Osallistuminen tutkimukseen on täysin vapaaehtoista. Tutkittavilla on tutkimuksen aikana oikeus kieltäytyä tutkimukseen liittyvästä tiedonkeruusta. Tutkimuksen tulokset julkaistaan tutkimusraportissa, joka on julkinen dokumentti. Tutkittavilla on oikeus saada lisätietoa tutkimuksesta tutkijaryhmän jäseniltä missä vaiheessa tahansa.

3 Tutkittavan suostumus

Olen perehtynyt tämän tutkimuksen tarkoitukseen ja sisältöön. Suostun osallistumaan työkalun arviointiin liittyvään fokusryhmähaastatteluun annettujen ohjeiden mukaisesti. Voin halutessani peruuttaa tai keskeyttää osallistumiseni missä vaiheessa tahansa. Tutkimustuloksiani saa käyttää tieteelliseen raportointiin.

Päiväys Tutkittavan allekirjoitus

Päiväys Tutkijan allekirjoitus

Appendix B. Consent form for library data used in research

Oulun yliopisto
Tieto- ja sähkötekniikan tiedekunta
PL 4500
90014 Oulun yliopisto

Master's Thesis: Recommender System for Library Book Selection
SUOSTUMUS TUTKIMUSDATAN KÄYTTÄMISEEN JA AVOIMEEN JULKAISEMISEEN

1 Tutkijoiden yhteystiedot

Vastuullinen tutkijaopiskelija
Aatu Nykänen
Vanhantullinkatu 6 F93
90100 Oulu
aatu.nykanen@student.oulu.fi
p. 050 493 0511

Tutkielman valvojat:
Minna Isomursu
Tietojenkäsittelytiede
90014 Oulun yliopisto
minna.isomursu@oulu.fi

Umar Farooq
Tietojenkäsittelytiede
90014 Oulun yliopisto
umar.farooq@oulu.fi

2 Tutkimusdata

Tutkimusdata koostuu Vaara-kirjastoissa käytössä olevasta Koha-kirjastojärjestelmästä SQL-kyselyillä haetusta datasta. Data sisältää sekä Joensuun pääkirjaston aikuisten kauno -osaston kirjaniteisiin liittyviä bibliografisia metatietoja että niteisiin kohdistuvia lainatietoja. Lainatiedot sisältävät päivämäärätiedon, nidenumeron ja tiedon lainan tyypistä. Lainatietoihin ei sisälly asiakkaita yksilöiviä tietoja.

3 Suostumus tutkimusdatan käytöstä ja avoimesta julkaisemisesta

Olen perehtynyt tämän tutkimuksen tarkoitukseen ja sisältöön. Tutkimuksessa saa hyödyntää osiossa 2 (Tutkimusdata) määriteltyä dataa. Tutkimuksessa hyödynnetty data on avointa julkista tietoa ja sitä saa edelleenvälittää vapaasti.

Päiväys — Datan omistajan allekirjoitus