University Degree Recommendation System

Islam Ehab Anwar

Faculty of Computer Science

University of Prince Edward Island, Cairo Campus

May 2023

A thesis submitted to University of Prince Edward Island, Cairo Campus in partial fulfillment of the requirements of the degree of Bachelor of Science

Acknowledgements

Words cannot express my appreciation and gratitude towards my professor, project adviser, and dean of the faculty of computer science Dr. Ahmed Elsheikh for his unwavering support and guidance. I would also like to thank Dr. Reham Hossam for her significant help and guidance in the technical specifics of this project. Additionally, this would not have been possible without my research assistants Heba Abdelkader and Ola Galal who without their guidance and experience I would not have been able to achieve this. Furthermore, I would like to thank the university for providing the necessary data for commencing this project.

I am also grateful to Dr. Amal Mohamed, professor at the faculty of computer science for her unconditional moral support and significant impact in inspiring me to pursue this field of study.

Lastly, I would be remiss in not mentioning my family, especially my parents, and brother. Their belief and encouragement have kept my spirits and motivation high during this trying process. I would never have reached this stage without them.

Abstract

Table of Contents

Acknowledgements i

Abstract ii

List of Figures vi

List of Tables vii

1 Introduction 1

1.1 Motivation and Problem Statement 1

1.2 Thesis Outline 2

2 Background 3

2.1 Terminologies and Definitions 3

2.1.1 Data Exploration and Preprocessing 3

2.1.2 Feature Selection and Model Selection 5

2.1.3 Recommendation Systems 5

2.1.4 Supervised Learning Techniques 9

2.1.5 Deep Learning Approaches 12

2.1.6 Model Interpretability 14

2.2 Content-Based Filtering 16

2.3 Collaborative Filtering 16

2.4 Knowledge-Based Approaches 17

2.5 Hybrid Methods 18

3 Methodology **19**

3.1 System Overview 19

3.2 Primary Data 20

3.2.1 University Student Dataset 20

3.3 Data Exploration 22

3.3.1 Univariate Exploration 22

3.3.2 Bivariate Exploration 22

3.4 Data Preprocessing 23

3.4.1 Missing Values 23

3.4.2 Feature Encoding 23

3.4.3 Outlier Detection 24

3.5 Feature Selection 24

3.6 Model Selection 25

3.7 Hyperparameter Optimization 25

3.8 Local Model Interpretations 25

3.8.1 Feature Importance 25

3.8.2 Decision Paths 25

3.8.3 LIME 25

3.8.4 Shapley Values 25

3.9 Data Post-Processing 25

4 Results and Discussion **26**

4.1 Results 26

4.1.1 Data Collection Process 26

4.1.2 Evaluation Metrics 26

4.1.3 System Analysis 26

4.2 Discussion 26

5 Conclusion and Future Work 27

5.1 Conclusion . . . 27

5.2 Future Work . . . 27

List of Figures

2.1 A simple example of outliers in a 2-dimensional data set (Adapted from [1]). . . . 4

2.2 Classification of Recommendation Approaches (Adapted from [2]). 6

2.3 A high-level architecture of content-based recommendation system (Adapted from [3]). 7

2.4 Shows the methods that integrate CB characteristics into the CF approach. Mitigates the cold start problem in collaborative filtering (Adapted from [4]). 9

2.5 Shows the methods that incorporate CF characteristics into a CB approach (Adapted from [4]). 9

2.6 Supervised Learning Process (Adapted from [5]). 10

2.7 Random Forest Procedure (Adapted from [6]). 12

2.8 A categorization of the deep learning methods and their representative works (Adapted from [7]). 13

2.9 Example architecture of Multi-layer Perceptron (Adapted from [8]). 14

2.10 An illustration of how LIME aims to simplify a black box model in a local space (Adapted from [9]). 15

3.1 Architecture of System Overview. 20

List of Tables

3.1 Data Set Features Summary 21

Chapter 1

Introduction

1.1 Motivation and Problem Statement

In recent years, the accelerating advancement of internet technology has left us with an abundant amount of information available. Various sources of data are now publicly accessible to users such as news content, e-commerce websites, as well as digital entertainment [10] [11]. A significant challenge with having an overload of data available is ensuring the quality of the information provided to users from their corresponding searches as well as results that match their interests and preferences. Preferences can vary depending on the background such as educational or entertainment purposes.

As a result, various recent researchers have attempted to create tailor-made solutions to provide users with a series of recommendations catering to specific user preferences. However, there is a lack of research performed in the educational field about the choice of degree or major in which a student enrolls at a university. This provides suitable motivation to pursue this field of study to understand what factors affect the degree chosen by a prospective student.

Recommendation systems are a type of information filtering system that predicts a user's preference for a product or service based on their past behavior, preferences, and interests [12]. These systems have become increasingly popular in recent years due to the growth of e-commerce and online services [12]. They are used in a variety of applications such as movie recommendations, music recommendations, and product recommendations[12]. The goal of this thesis is to explore the different types of recommendation systems and their effectiveness in different domains [12].

A student degree recommendation system is a problem that has not been tackled by many recent publishers. With the vast diversity of university programs available to prospective students, many are left wondering what their real passion is or if a particular degree is the one for them. A recommendation algorithm is suitable to mitigate this issue, by analyzing student academic performance during the latter end of high school as well as other factors such as familial status, personal status, and other daily lifestyle choices that could shape the overall performance of a student and impact the recommended degrees. Furthermore, we dive into the concept of interpretability of our proposed recommendation system and how that can build model trust and reliability by understanding why decisions were made at the local level of our data.

1.2 Thesis Outline

The rest of the thesis is as follows: Chapter 2 describes background research regarding terminologies and definitions, a literature review regarding supervised as well as deep learning techniques used with recommendation systems. Chapter 3 discusses the methodology to develop an effective student degree recommendation system. Chapter 4 offers insight into the results and discussion. Chapter 5 provides the conclusion and future work.

Chapter 2

Background

This chapter provides the necessary background and related research to this project. This project applies a variety of approaches to create a viable student degree recommendation system. The background section will be divided into five sections: Terminologies and Definitions, Content-Based Filtering, Collaborative Filtering Approaches, Knowledge-Based Approaches, and Hybrid Approaches.

2.1 Terminologies and Definitions

This section covers all relevant terminologies and definitions for areas of interest that is directly correlated with a student degree recommendation system. This section is divided into six subsections: Data Exploration and Preprocessing, Feature Selection and Model Selection, Recommendation Systems, Supervised Learning Techniques, Deep Learning Approaches, and Model Interpretability.

2.1.1 Data Exploration and Preprocessing

Data exploration is among one of the vital initial steps to exploring underlying patterns within a data set. Univariate data exploration methods, which analyse a single feature's patterns such as measures of central tendency, measures of dispersion, histogram plots, and kurtosis. Bivariate data exploration techniques such as correlation heat maps, scatter plots, and box plots are used frequently for extracting relations between a pair of numerical variables or numerical-categorical variables respectively [13].

Real world data can be very messy. In most cases, it can be very difficult to input raw data into a machine learning pipeline and generate predictions or accurate results without data preprocessing. Data preprocessing is responsible for transforming the data set into a favorable format to be input to a model for performing classification. There are a variety of preprocessing steps such as checking for missing values using correlation heat maps. Another valuable transformation is feature encoding. Feature encoding transforms categorical features into numerical features for more effective analysis [14].

In addition to the previous points, outlier detection is another vital aspect of data preprocessing. Outliers are defined as data poitns that do not exhibit normal behaviour relative to the rest of the data [1]. Outliers can sometimes be difficult to identify due to many factors including the nature of the data set or the fact that some outliers can be referred to as meaningful outliers, which provide more information to the rest of the data set then when they are removed. Several techniques including box plots, the Median Absolute Deviation (MAD) are considered effective in identifying the outliers within data.

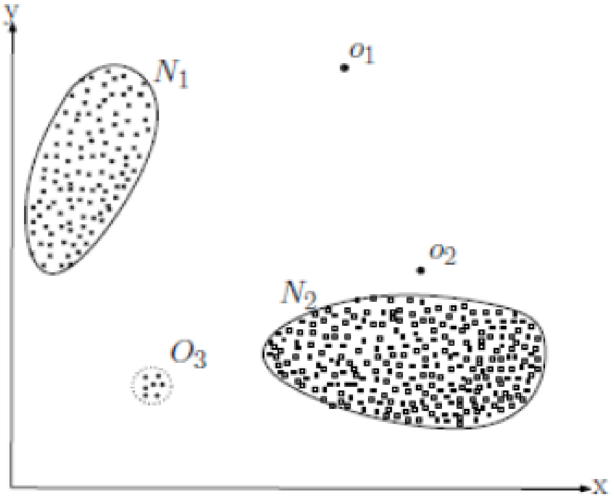

Figure 2.1: A simple example of outliers in a 2-dimensional data set (Adapted from [1]).

2.1.2 Feature Selection and Model Selection

Feature selection is regarded an effective step in preparing data for data mining and machine learning procedures. It's objectives include constructing simpler and more effective models, thus improving performance and interpretability of the data [15]. This in turn provides valuable information for which models are more suitable after performing feature selection.

Feature importance is a common measure used to identify the most important features in the classification performance. Feature importance can be measured in a local or global scale. This can be useful in choosing k number of features to simplify the complexity of the data set for input to a classification model [16]. Correlation heat maps can also be useful. Features that have high correlation with the target variable are considered good variables. The Chi-square test is also a viable option for categorical features in a data set. Chi-square is calculated between the categorical feature and the target variable and best k features are chosen based on their scores [17].

Regarding model selection, there are a variety of subjective metrics that can be used to decide which model is best for a data set. First and foremost is whether a model is overfitting by memorizing the data rather than generalizing. Moreover, the complexity of the model is important to be able to extract interpretable information from the predictions.

2.1.3 Recommendation Systems

Recommendation systems (RS) are sophisticated machine learning algorithms that recommend a set of items or a collection of data to a user based on their specific preferences [10]. The concept of recommendation systems began in the 1990s to help people decide on what products to purchase [18]. Today, there is a plethora of recommender systems that are created and modified to cater to a vast range of fields such as e-commerce, digital entertainment, as well as the educational industry [10].

With the evolution of recommendation systems, they were sub-categorized into various approaches to provide different kinds of recommendations based on different factors.

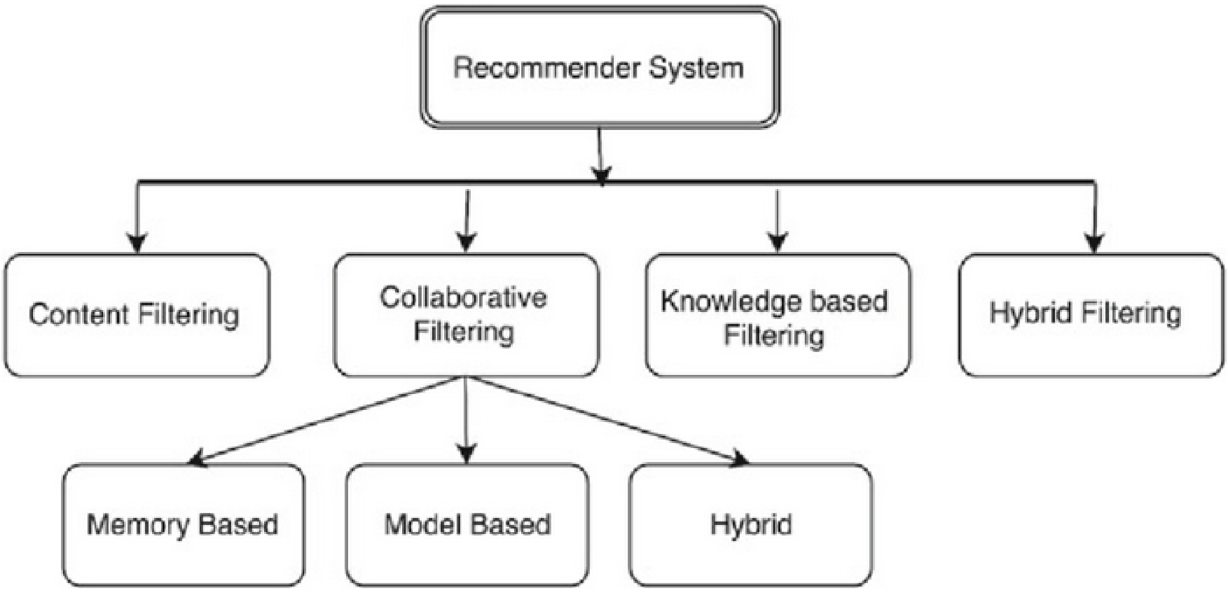

Figure 2.2: Classification of Recommendation Approaches (Adapted from [2]).

A type of recommendation approach is a content-based recommendation system. Content-based filtering works by using a first component known as a content analyzer. A content analyzer is used to structure unstructured data to withdraw relevant information [19]. Its main objective is to represent the relevant information in a matter for further preprocessing steps. Feature extraction methods are utilized to change the representation of the original information. After a content analyzer, the representation is fed into a profile learner.

A profile learner is then used to identify unique user preferences and try to generalize recommendations for each user [19]. Machine learning techniques are then used to produce a model that generates recommendations based on user interactions with certain items.

A filtering component takes advantage of the previous module to suggest relevant recommendations based on matching a user representation to that of the potential recommended items. This results in a list of items based on potential interest to the user [19]. A cosine similarity or some distance metric is deployed to identify the similarity between the resultant vector and the item vector.

Another popular recommendation approach is collaborative filtering (CF), which works by representing unique users and items with unique representation vectors. Recommendations are

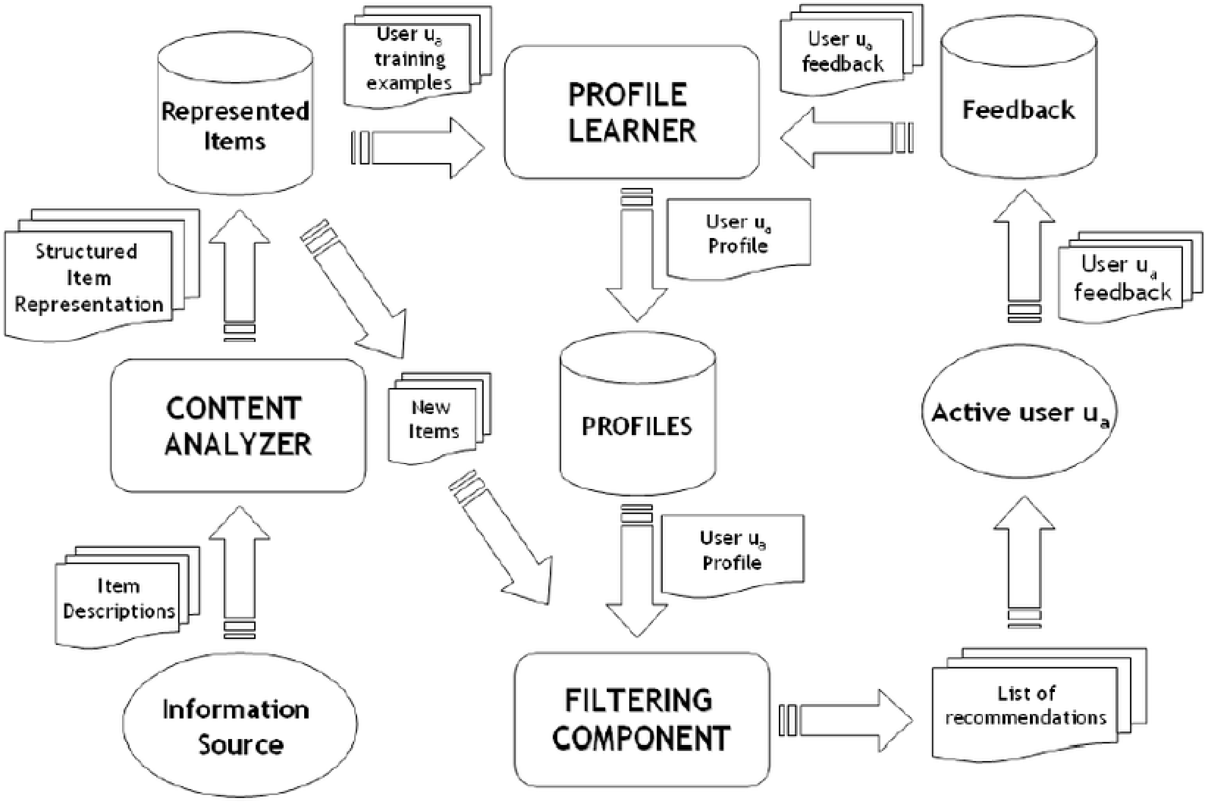

Figure 2.3: A high-level architecture of content-based recommendation system (Adapted from [3]).

considered using a matrix representing the users and their respective relations with all items. A collaborative filtering approach uses the relationships between users and their items through a similarity measure such as cosine similarity [20]. There are various ways to implement a collaborative approach such as neighborhood-based models, and latent-factor models.

Neighborhood-based models can be either user or item-based. User-based approaches take advantage of the user rating in the representation vector as well as other users' similar representations [20]. The item-based approach uses other users' ratings or scores of a certain item to predict a user's relation to an item.

Latent-factor models such as matrix factorization use a matrix of latent factors per user. The objective is to split the original matrix into two matrices S and M such that an approximation can be formed and predictions made for new items.

$$R \approx SM^T \tag{2.1}$$

where:

R = Original orthogonal rating matrix

S = $|U|*$F matrix

U = Matrix of unique users and relations with items

M^T = $|I|*$F matrix

F = Number of factors and is a parameter to be optimized

However, a disadvantage of CF recommendation is suffering from data sparsity and the cold start issue, hence other techniques have been developed to tackle these problems [11].

Another common recommendation approach is knowledge-based systems. Knowledge-based recommendation systems are heterogeneous graphs, whose nodes represent the unique users, and the edges are the relations between the user and the corresponding items. Knowledge-based systems can be designed in a variety of ways, including embedding-based methods, path-based methods, as well as unified methods which combine the former techniques to build the latter [11].

Embedding-based methods are created by building knowledge graphs with several item representations to better model users more accurately. Information such as user-specific graphs can be effective in providing unique representation and hence more accuracy. Entity embedding is the underlying methodology in embedding-based methods to extract key information from graph structures [11].

Path-based methods typically incorporate matrix factorization to enrich the user graph. Furthermore, interpretability is emphasized during the recommendation process when opting for a path-based method. This is performed by matching the similarity of the item or user based on the meta-path level [11]. A unified approach builds on the knowledge graphs and semantic path patterns of both embedding-based and path-based methods respectively as well as inheriting the interpretability feature from path-based methods.

Our last recommendation approach is hybrid recommendation system. They can be implemented in a variety of manners such as individually implementing a content-based and collaborative filter approach and aggregating the results of both. Another technique is generating an intermediate model that fuses characteristics from more than one recommendation approach [4].

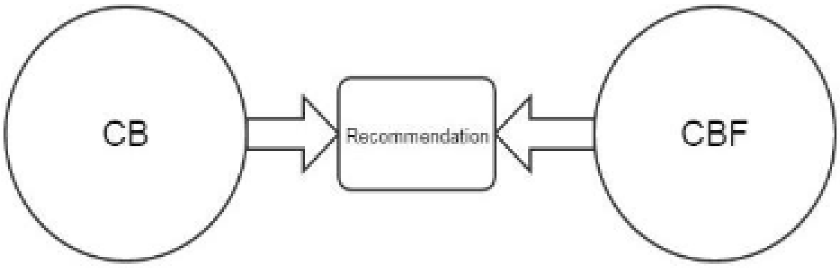

Figure 2.4: Shows the methods that integrate CB characteristics into the CF approach. Mitigates the cold start problem in collaborative filtering (Adapted from [4]).

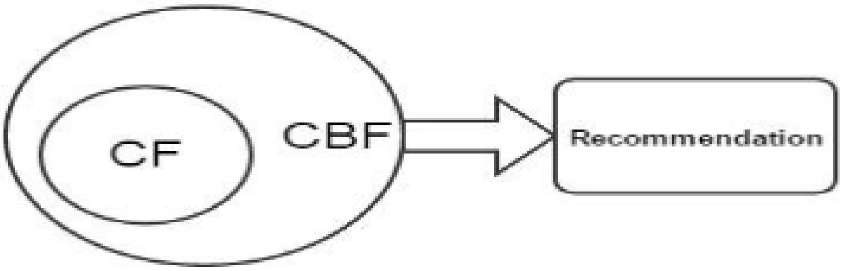

Figure 2.5: Shows the methods that incorporate CF characteristics into a CB approach (Adapted from [4]).

Predictions performed by hybrid systems can vary significantly depending on the original class which they are from. Predictions can be based on a weighted aggregation of prediction scores, or a mixing of recommendations' predictions and picking one out. Feature combination and augmentation also play a role as a characteristic of hybrid systems as that can affect prediction outcome [4].

2.1.4 Supervised Learning Techniques

Supervised learning is a subcategory within machine learning. Supervised learning is defined by labeled datasets that can be used to train algorithms to accurately classify data or predict outcomes [21]. Training data is applied to a supervised algorithm to generate an effective classifier that can correctly classify data as well as form reasonable predictions. Many common algorithms are housed within supervised learning that operates on categorical as well as continuous data such as decision trees (DT) and random forests (RF).

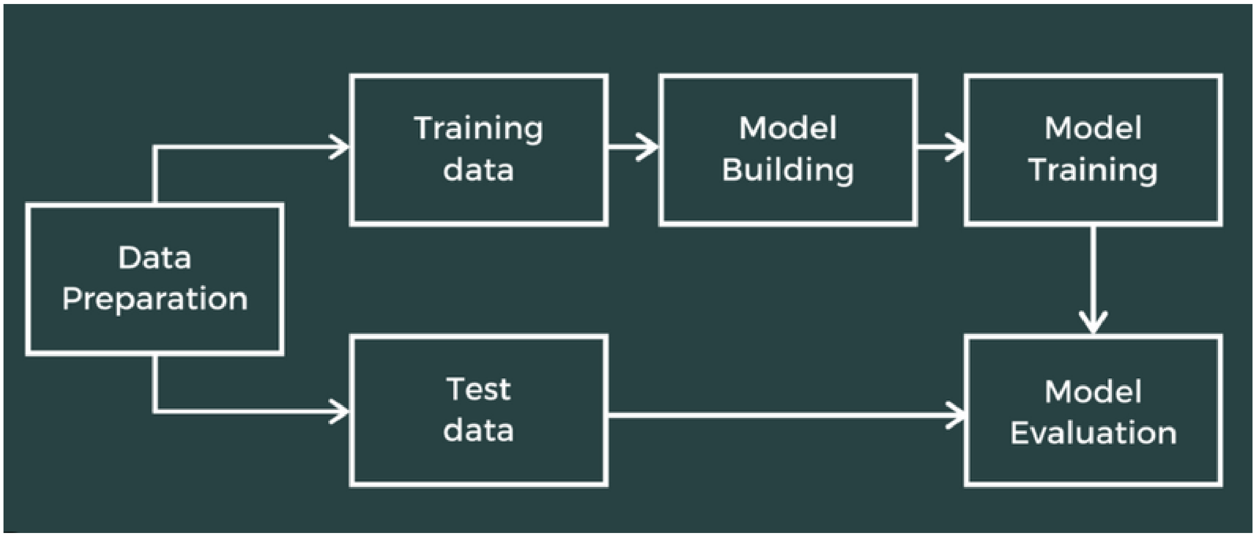

Figure 2.6: Supervised Learning Process (Adapted from [5]).

Decision trees are one of the simplest supervised learning algorithms due to their interpretability and simple execution compared to other algorithms [22]. A series of questions are posed against the set of features input into the model. Each question is contained within a unique node that based on the decision tree algorithm, forms a splitting condition by which to traverse a side of the tree. This repeats iteratively until a stopping condition is met such as a hyper-parameter condition or the tree evaluates a decision path for all possible outcomes of the data [23].

There are various hyper-parameters for decision tree classifiers. However, we will only consider the most impactful ones such as: *criterion*, *max depth*, and *ccp alpha*.

The error criterion is a function that measures the quality of the split in the tree. One of the criteria is entropy, which gives a measure of the impurity of the training items. The entropy is lowest when a single probability equals 1 and the rest are 0. The gini impurity is another criterion measure, which ranges from 0 to 0.5 based on the impurity of a split. The goal is to minimize the impurities of these splits to maximize information gain from all splits in the tree [23].

$$Entropy = -\sum_{i=1}^{m} p_i log(p_i) \tag{2.2}$$

where:

p_j = Probability of the class j.

$$Gini = 1 - \sum_{i=1}^{m} p_i^2 \tag{2.3}$$

where:

p_j = Probability of the class j.

The *max depth* references the maximum depth of the tree. This can be an important hyper-parameter to consider as this affects the number of splits in the tree as well as the quality of splits. If no restriction is given to the max depth, then the tree can continue splitting until every split account for an outcome in the training data. Consequently, this will produce many impure splits and is not a favorable approach. Optimizing this hyper-parameter is crucial to ensure a balance between the number of splits as well as the purity of the splits.

The tree pruning parameter is known as *ccp alpha* which is a regularization parameter that deletes nodes to reduce overfitting. As a result, this will improve the reliability and generalization of a proposed decision tree classifier and eliminate any impure leaf nodes [23].

Decision trees are very useful in classification problems to break down labeled data through effective splits. However, they do have disadvantages such as the potential for overfitting, which is when the algorithm memorizes the data instead of generalization. Furthermore, decision trees struggle to handle large data since a single tree will result in many node splits and lead to overfitting [24].

Another method of supervised learning is random forest (RF). Random forest is an ensemble supervised learning algorithm that uses a collection of decision trees to perform its classification or regression. Its basic functionality relies on a method called bagging, in which subsets of features of the training data are taken and used for training with the decision trees. The final classification or output is based on a majority vote between all the trees in the forest [6].

Regarding random forest hyper-parameters, the most impactful parameters are: *number estimators*, *criterion*, *max depth*, and *ccp alpha*. The criterion, max depth, and ccp alpha have already been discussed. However, the number of estimators in a random forest can significantly impact classification accuracy.

Modifying the number of estimators in a forest can mitigate the overfitting effect of a single decision tree by smoothing the decision boundary. As we increase the number of trees, the decision boundary is smoothed gradually to reduce overfitting and generate strong accuracies.

The advantages of random forests are that they are easy to train and predict, as well as ease of use with high dimensional data. However, they tend to generate bias when dealing with categorical variables [6]. Furthermore, random forests are considered black box models as they are difficult to interpret how predictions are made due to the presence of multiple trees.

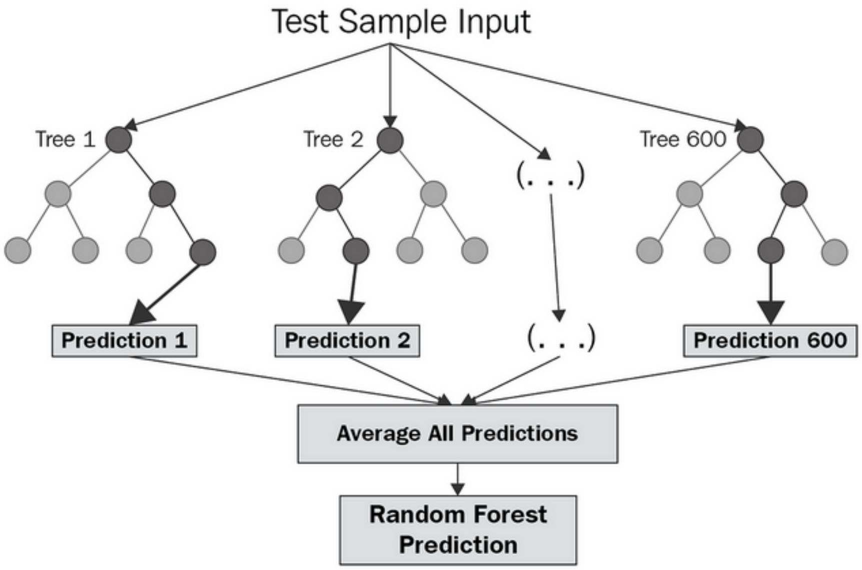

Figure 2.7: Random Forest Procedure (Adapted from [6]).

2.1.5 Deep Learning Approaches

Deep learning (DL) algorithms are another category of machine learning algorithms that aim to find multiple patterns within data using high level architecture. Deep learning has been used extensively in tasks that require high pattern recognition such as computer vision, natural language processing (NLP), as well as other multimedia-centered tasks [7].

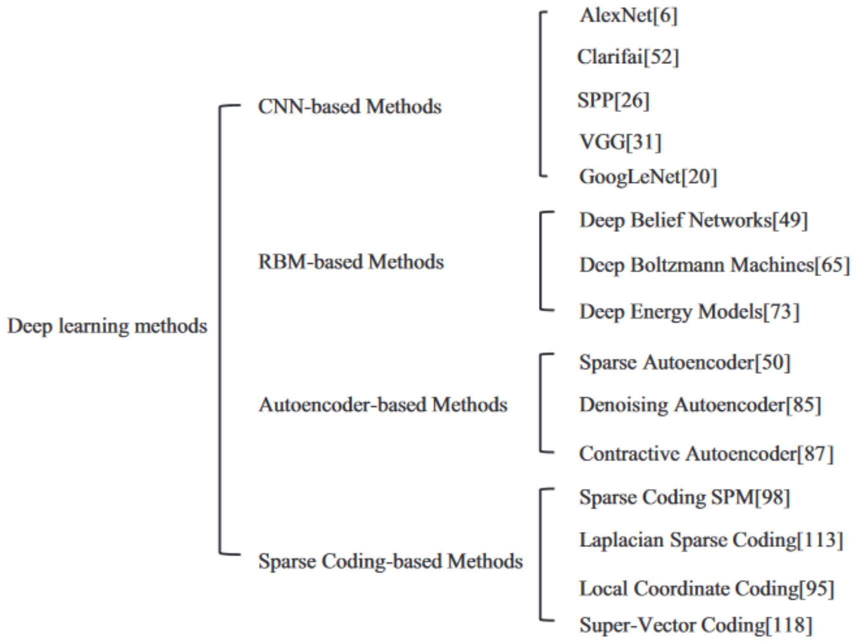

Figure 2.8: A categorization of the deep learning methods and their representative works (Adapted from [7]).

Among the most simple and common neural networks is a multi-layer perceptron (MLP), which works by using back-propagation to update the weights of our neurons during each iteration. This is repeated until a convergence threshold is met or we reach the maximum number of iterations. Multi-layer perceptrons are used in many applications such as prediction and pattern classification [25].

However, one of the most impactful limitations of deep learning algorithms is their hunger for data. Small datasets will render deep neural networks ineffective as they require a large amount of data to continue the learning process. Another problem to consider is poor scalability, which is one of the many reasons supervised learning algorithms are chosen for most tasks that require a scalable model [26].

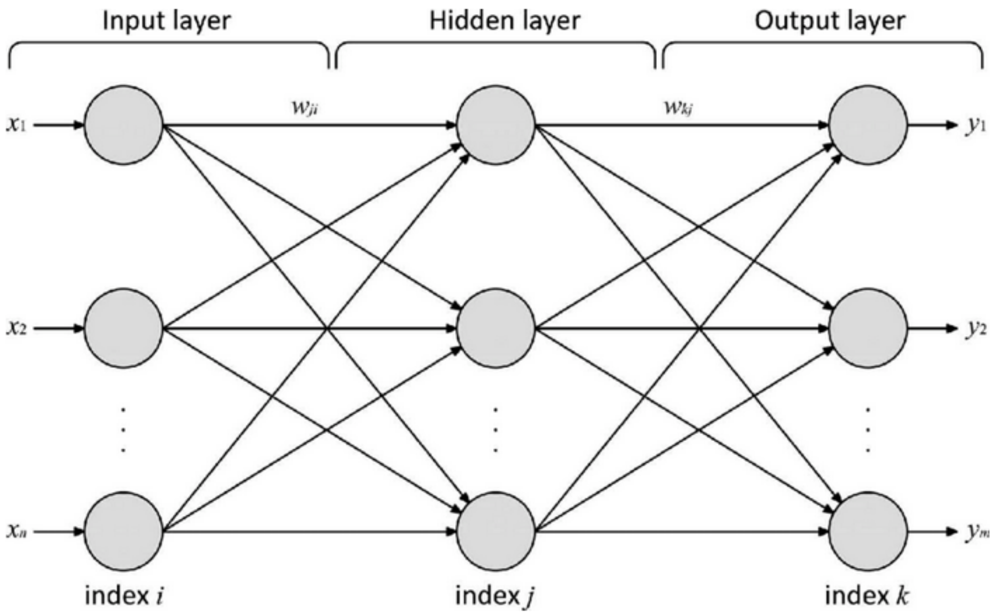

Figure 2.9: Example architecture of Multi-layer Perceptron (Adapted from [8]).

2.1.6 Model Interpretability

Recommendation systems provide personalized recommendations to each user. Therefore, interpreting why a certain output was given is a crucial aspect of recommendation systems. This is where model interpretability is introduced. Some algorithms that are used in recommendation systems such as random forest are classified as black box models that do not provide information regarding their internal functions and why a certain output was produced [27].

Interpretability is the degree to which a person can understand why a decision was made. It is among one of the most important parts of predictive modelling as understanding why a decision was made by a model implies model reliability and trust. Interpretability has a large scope that pertains to algorithm transparency as well as global or local interpretations of our data [27]. Consequently, interpretability in a recommendation system is vitally important to invoke trust from the user that the predicted output was reliable and provided explanations as to why a certain decision was made by the system. There are a variety of interpretability techniques including Local Interpretable Model-Agnostic Explanations (LIME) and Shapley Values.

The black box problem in machine learning refers to the inability of a model to explain how it arrived at a particular decision or prediction. In other words, it is difficult to understand how a model arrived at its output because the model's internal workings are not transparent. This lack of transparency can make it difficult to trust machine learning models and can limit their usefulness in certain applications. The black box problem is one of the biggest issues facing AI/ML because most out-of-the-box machine learning systems only make the inputs and outputs of your model observable [28].

LIME is a specific type of algorithm mode or technique that can help to address the black box problem in machine learning. When you want to explain an individual prediction, you can use LIME. With LIME, a local surrogate model is trained. This interpretable surrogate model can be used to explain the individual prediction. LIME produces human-understandable explanations of the model. It is used to explain a single observation or record [28].

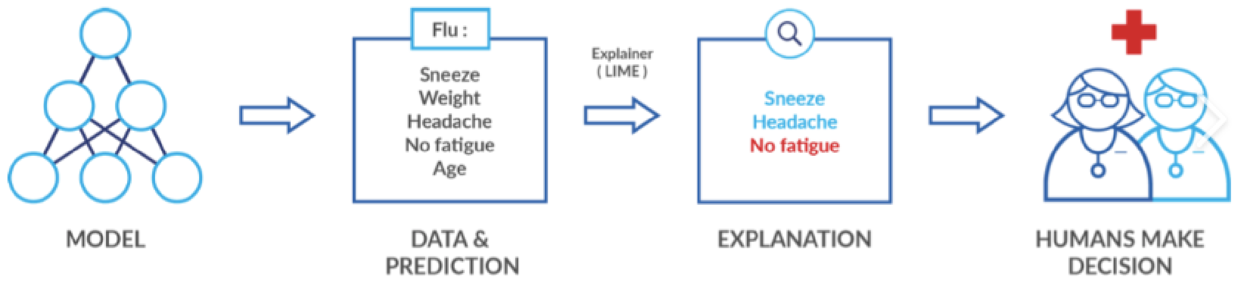

Figure 2.10: An illustration of how LIME aims to simplify a black box model in a local space (Adapted from [9]).

Shapley values are a method for assigning credit to each feature in a machine learning model's prediction. The Shapley value is the only attribution method that satisfies the properties Efficiency, Symmetry, Dummy and Additivity, which together can be considered a definition of a fair payout. The Shapley value can be defined as a function which uses only the marginal contributions of player as the arguments. Shapley values are a widely used approach from cooperative game theory that come with desirable properties [28].

There are several methods for interpreting machine learning models such as partial dependence plots (PDP), permutation feature importance (PFI), rule-based methods, and saliency maps methods. However, these methods can lead to wrong conclusions if applied incorrectly.

2.2 Content-Based Filtering

Content-based recommendation systems are a popular technique used to gather information about user preferences. They are based on past preferences of users and recommendations are suggested with similar items of similar characteristics [29]. Chen et al. [30] used correlation analysis in an experiment regarding the education field to group certain courses. This was performed by segmenting the data into three categories based on a rule-space model using a content-based approach to optimize the learning path for each individual.

Similarly, Shu et al. [31] utilized the historical data of students to create predictions regarding the provided learning materials using a content-based algorithm. The most common learning algorithms used in this domain are fuzzy-based as well as rule-based clustering that relies on a probabilistic methodology, and similarity between neighbors.

The advantages of content-based recommendation systems are that they handle each user as an independent user, meaning that user relations aren't influenced by other user relations. Furthermore, they provide transparency behind decisions based on content features [19]. However, a disadvantage of content-based recommendation is the over-reliance on past data to create predictions. It cannot overcome the cold start problem of having no data in the initial stages as well as data sparsity [29].

2.3 Collaborative Filtering

Another favored technique in recommendation is collaborative filtering, which has been used frequently in recent systems as it mitigates the drawbacks of content-based filtering as we discussed earlier [29]. Liu [32] recommended a collaborative filtering approach that focused on the influence of e-learning group behavior to improve the accuracy of predictions even in presence of data sparsity. Moreover, collaborative filtering has been used with unsupervised learning.

El-Bishouty et al. [33] utilized a k-means algorithm to extract the learning path and objects of interest for each learner. In addition, M. Aljunid et al. [34] proposes a deep learning method for a collaborative filtering recommendation system using the concept of matrix factorization implemented within the deep network. Results demonstrated on the 100k and 1M movie lens datasets show an improved Root Mean Squared Error (RMSE) in comparison to other models such as cosine similarity and the dot product of matrix factorization [34].

Collaborative filtering does improve on content-based systems, but it still comes with its own set of difficulties. It can be difficult to create relations between attributes to their respective items, which can affect recommendation accuracy. It also suffers from cold-start and scalability issues [29].

2.4 Knowledge-Based Approaches

Knowledge-based approaches provide recommendations based on how certain item features meet user needs. H. Wang et al. [10] proposed a knowledge graph convolutional network. The purpose of this architecture was to capture relationships between several items of interest to a user through data mining techniques. Some of the featured techniques involved association rules to identify relations between attributes on a graph. This was performed by sampling data from respective neighbors per entity in a particular graph and fusing the information already gained with the bias to calculate an accurate representation of each entity's graph relations. Three datasets were used for testing including movie, book, and music recommendation datasets. Results indicated that the proposed network outperformed the baseline recommendation techniques with an Area Under the Curve (AUC) of 0.9 or higher with two of the three datasets [10].

Wan and Niu [35] used a knowledge-based approach with an underlying self-organization method to propose learning objects. This improved accuracy but suffered from the increased time of computations and stacking of multiple algorithms.

2.5 Hybrid Methods

Certain hybrid methods have been experimented with including the combination of content-based and collaborative filtering to counter the cold start problem [29]. Hussain et al. [36] opted to use a collection of models including an artificial neural network, decision tree, logistic regression, and support vector machine to predict the troubles students face during an online learning course.

Moreover, P. Kouki et al. [37] deployed a hybrid system based on a probabilistic model to provide personalized recommendations. Furthermore, they pursued explainable artificial intelligence (XAI) from various perspectives including a crowd-sourced approach and mixed model statistical analysis to understand the relations between users and their relations.

Similarly, Karga and Satratzemi [38] used a similarity matrix to create the relations between learners and their respective learning paths according to their needs and preferences. This methodology allows both content-based and collaborative filtering methods to complement each other's weaknesses while improving prediction accuracy [29].

Chapter 3

Methodology

This chapter provides a comprehensive review of the methods used to create a functioning student degree recommendation system. This chapter is divided into the following eight sections: System Overview, Primary Data, Data Exploration, Data Preprocessing, Feature Selection, Model Selection, Hyperparameter Optimization, and Local Model Interpretations.

3.1 System Overview

The primary objective of this thesis is to create a student degree recommendation system. That primary objective is comprised of a series of sub-goals to ensure we have a reliable and interpretable system. An emphasis of importance is placed on model accuracies but also on interpretation of predictions generated by our proposed system. An explanation for an outcome can be a very effective feature to provide users with reasoning behind the systems' recommendations and to potentially mitigate the black box effect or lack of interpretability of some of the proposed supervised classifiers and deep learning models.

This section illustrates the overall system overview of our proposed recommendation system. Various sequential steps are taken in order to obtain a reliable and trustworthy system. The first stage of the system is the data collection process. This includes the methodology by which we obtained our data and a brief description of the nature of the data set. Following data collection is data exploration, which is comprised of a series of techniques and strategies to obtain a more transparent understanding of our data.

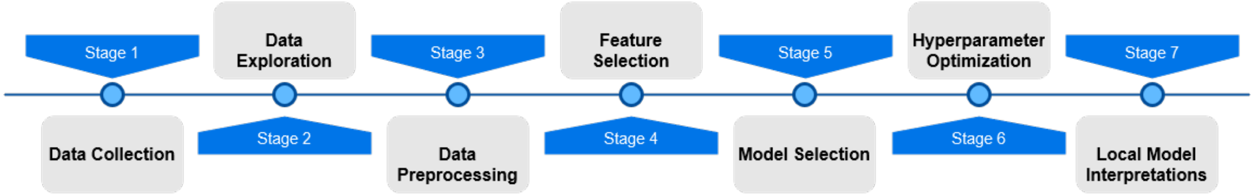

Figure 3.1: Architecture of System Overview.

Shortly after data exploration, several preprocessing steps will be undertaken to prepare the data for training during the later stages of our proposed system. Moreover, feature selection will be performed to observe the importance of certain features on our evaluation metrics, interpretability of our proposed model, as well as impact on decisions made by the models. Upon exploring feature selection, model selection is the next stage to observe performance comparisons as well as comparisons of model interpretability in a local space. Hyperparameter optimization is the next step to maximize the effectiveness of our proposed classifiers and deep learning models. Post optimization, we perform local model interpretations using a plethora of techniques on our models to gain further insight into the inner workings of how an outcome or prediction was evaluated for each individual user.

3.2 Primary Data

This section discusses the chosen data set used for training our proposed recommendation system as well as explanations regarding the nature of the data set and its features are also discussed.

3.2.1 University Student Dataset

Our data set of choice is an entire record of information regarding a single undergraduate-level university student. Several features are collected to illustrate the performance of an individual student during his final years of high school such as their A-level mathematics or English grades.

A vast amount of features regarding academic performance are collected with care to ensure that our proposed model can effectively identify patterns and extract meaningful insights.

Furthermore, we collected the actual major and specialization of each student, with specialization acting as the dependent variable. Other factors are also considered including whether they are a transfer student or pursued a different curriculum of high school education such as IB or IG. Other features are illustrated for the purpose of potential post processing and identifying the most impact or significant features relative to the predictions.

Table 3.1: Data Set Features Summary

Feature	**Explanation**
school id	Stores the initials of each student's high school name
school type	The curriculum of education obtained during high school e.g. IB, IG
alevel math	Stores a numerical value between 0 and 100
olevel math	Stores a numerical value between 0 and 100
chem	Stores a numerical value between 0 and 100
phy	Stores a numerical value between 0 and 100
bio	Stores a numerical value between 0 and 100
english	Stores a numerical value between 0 and 100
ap course	A binary feature, whether student took any AP courses
cs ig	A binary feature, whether student took a computer science IG course
adv math	A binary feature, whether student took an advanced mathematics course
international student	A binary feature, a student had foreign high school education
transferred courses	Stores number of courses transferred
major	Describes student's chosen major e.g. BUS, CS, ENG
specialization	Describes student's chosen specialization e.g. Finance, Video Game
cgpa	Cumulative GPA of each student using 4.3 scale
credit hours completed	Describes number of completed credit hours of each student

3.3 Data Exploration

A variety of data exploration techniques are utilized in this section. We will first get an idea of the data set by extracting the first 5 observations as well as the shape of our data set to see how many student observations we have at our disposal.

3.3.1 Univariate Exploration

Regarding univariate data exploration, our first plan of action is to describe our numerical features obtaining basic statistics such as the mean and standard deviation. We perform this to develop an understanding of our numerical variables and how they are reflected in this data set. Furthermore, we will plot histograms to find the distributions of our numerical features as well as the kurtosis to potentially find outliers in our data.

With respect to any categorical features, we will be assessing different things, such as how many categories are present in a certain feature, the most common category, value counts for each category, and a plot visualizing the proportions of categories as a percentage.

3.3.2 Bivariate Exploration

In bivariate exploration, we will explore relationships between certain numerical variables with other numerical variables as well as numerical with categorical. Regarding the former, scatter plots will be used to understand relationships between two numerical features. As for the latter, box plots will be the primary tool to understand the relations between a numerical and categorical variable.

One of the most useful measures of data exploration is a correlation heat map. It can serve as a great indicator for feature selection through visualizing relationships between a pair of numerical variables

3.4 Data Preprocessing

In this section, we highlight the data preprocessing steps to prepare the raw data to be trained for a machine learning model.

3.4.1 Missing Values

The first step in the preprocessing stage will be to check missing values, which can occur through human error or failures of measurement.

To check for missing values, we will use several visualizations to guide us including a nullity matrix that shows striped lines indicating missing values for certain features of our data. Another technique is using a variation of the correlation heatmap. A correlation heatmap will calculate the nullity correlation between pairs of features in the data set, representing how strongly the presence of one feature impacts the other. In addition, a simple numerical summary is an effective method to identify percentages of outliers present in each feature of our data.

We will mitigate the issue of missing values using imputation methods such as forward fill and backward fill if needed. However, we will not delete any observations or modify values of 0 in our data as that has meaningful information to contribute to our data set.

3.4.2 Feature Encoding

We will transform categorical features including our target specialization feature into numerical features using one-hot encoding. One-hot encoding transforms the categorical feature into an orthogonal vector space with each category being assigned a unique value in the vector. However, it can be a hindrance when dealing with a high-feature space, as that can invoke the curse of dimensionality and make visualizations difficult to create.

3.4.3 Outlier Detection

Outlier detection is a vital aspect of data preprocessing. Outliers can occur due to many reasons such as data entry errors, and measurement errors, or the observation can be a natural outlier within our data.

Regarding univariate outlier detection, we will use the box plot to visualize any potential outliers in a single feature of our data. Furthermore, we will use the Median Absolute Deviation (MAD) to test for outliers. This is considered a robust z-score since the mean and standard deviation values can be affected by outliers. Median absolute deviation uses the median and absolute deviation from the mean to find outliers.

Another measure is the winsorization method that works similarly to the inter-quartile range but with a percentile range. If a certain observation exceeds a set percentile then it is considered an outlier.

3.5 Feature Selection

here

3.6 Model Selection

3.7 Hyperparameter Optimization

3.8 Local Model Interpretations

3.8.1 Feature Importance

3.8.2 Decision Paths

3.8.3 LIME

3.8.4 Shapley Values

3.9 Data Post-Processing

Chapter 4

Results and Discussion

4.1 Results

4.1.1 Data Collection Process

Student Data Input GUI

4.1.2 Evaluation Metrics

4.1.3 System Analysis

Synthetic Data

Real World Data

4.2 Discussion

Chapter 5

Conclusion and Future Work

5.1 Conclusion

5.2 Future Work

Bibliography

[1] Karanjit Singh and Shuchita Upadhyaya. Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*, 9(1):307, 2012.

[2] Sandeep Raghuwanshi and R. Pateriya. *Recommendation Systems: Techniques, Challenges, Application, and Evaluation: SocProS 2017, Volume 2*, pages 151–164. 01 2019.

[3] Ansgar Koene, Elvira Perez, Chris Carter, Ramona Statache, Svenja Adolphs, Claire O'Malley, Tom Rodden, and Derek McAuley. Ethics of personalized information filtering. pages 123–132, 05 2015.

[4] Poonam B Thorat, Rajeshwari M Goudar, and Sunita Barve. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4):31–36, 2015.

[5] The Click Reader. Introduction to supervised machine learning, Oct 2021.

[6] Cristhiana Albert, O. Isgor, and Ueli Angst. Exploring machine learning to predict the pore solution composition of hardened cementitious systems. *Cement and Concrete Research*, 162:107001, 12 2022.

[7] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016. Recent Developments on Deep Big Vision.

[8] André Honorato, Gustavo Silva, and Celso Santos. Monthly streamflow forecasting using neuro-wavelet techniques and input analysis. *Hydrological Sciences Journal*, 63:1–16, 01 2019.

[9] Kaja Kaja Polachowska·and Polachowska, Sep. 04, and Like(5)Comment. The challenges of ai adoption - dzone, Sep 2019.

[10] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. In *The world wide web conference*, pages 3307–3313, 2019.

[11] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568, 2020.

[12] Pradeep Singh, Pijush Dutta Pramanik, Avick Dey, and Prasenjit Choudhury. Recommender systems: An overview, research trends, and future directions. *International Journal of Business and Systems Research*, 15:14–52, 01 2021.

[13] Zhongheng Zhang. Univariate description and bivariate statistical inference: the first step delving into data. *Annals of translational medicine*, 4(5), 2016.

[14] A Famili, Wei-Min Shen, Richard Weber, and Evangelos Simoudis. Data preprocessing and intelligent data analysis. *Intelligent data analysis*, 1(1):3–23, 1997.

[15] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

[16] Mirka Saarela and Susanne Jauhiainen. Comparison of feature importance measures as explanations for classification models. *SN Applied Sciences*, 3:1–12, 2021.

[17] Todd Michael Franke, Timothy Ho, and Christina A Christie. The chi-square test: Often used and more often misinterpreted. *American journal of evaluation*, 33(3):448–458, 2012.

[18] Richa Sharma and Rahul Singh. Evolution of recommender systems from ancient times to modern era: a survey. *Indian Journal of Science and Technology*, 9(20):1–12, 2016.

[19] Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, pages 73–105, 2011.

[20] Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

[21] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168, 2006.

[22] Anuja Priyam, Rahul K. Gupta, Anju Rathee, and Saurabh Kr. Srivastava. Comparative analysis of decision tree classification algorithms. 2013.

[23] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature biotechnology*, 26(9):1011–1013, 2008.

[24] Bahzad Taha Jijo and Adnan Mohsin Abdulazeez. Classification based on decision tree algorithm for machine learning. *evaluation*, 6:7, 2021.

[25] M.W Gardner and S.R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric Environment*, 32(14):2627–2636, 1998.

[26] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, pages 372–387, 2016.

[27] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[28] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Ben-

jamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[29] Shristi Shakya Khanal, PWC Prasad, Abeer Alsadoon, and Angelika Maag. A systematic review: machine learning based recommendation systems for e-learning. *Education and Information Technologies*, 25:2635–2664, 2020.

[30] Yung-Hui Chen, Chun-Hsiung Tseng, Ching-Lien Huang, Lawrence Y Deng, and Wei-Chun Lee. Recommendation system based on rule-space model of two-phase blue-red tree and optimized learning path with multimedia learning and cognitive assessment evaluation. *Multimedia Tools and Applications*, 76:18237–18264, 2017.

[31] Jiangbo Shu, Xiaoxuan Shen, Hai Liu, Baolin Yi, and Zhaoli Zhang. A content-based recommendation algorithm for learning resources. *Multimedia Systems*, 24(2):163–173, 2018.

[32] Xiuju Liu. A collaborative filtering recommendation algorithm based on the influence sets of e-learning group's behavior. *Cluster Computing*, 22(Suppl 2):2823–2833, 2019.

[33] Moushir M El-Bishouty, Ahmed Aldraiweesh, Uthman Alturki, Richard Tortorella, Junfeng Yang, Ting-Wen Chang, Sabine Graf, et al. Use of felder and silverman learning style model for online course design. *Educational Technology Research and Development*, 67(1):161–177, 2019.

[34] Mohammed Fadhel Aljunid and Manjaiah Dh. An efficient deep learning approach for collaborative filtering recommender system. *Procedia Computer Science*, 171:829–836, 2020.

[35] Shanshan Wan and Zhendong Niu. An e-learning recommendation approach based on the self-organization of learning resource. *Knowledge-Based Systems*, 160:71–87, 2018.

[36] Mushtaq Hussain, Wenhao Zhu, Wu Zhang, Syed Muhammad Raza Abidi, and Sadaqat Ali. Using machine learning to predict student difficulties from learning session data. *Artificial Intelligence Review*, 52:381–407, 2019.

[37] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. Personalized explanations for hybrid recommender systems. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 379–390, 2019.

[38] Soultana Karga and Maya Satratzemi. A hybrid recommender system integrated into lams for learning designers. *Education and Information Technologies*, 23:1297–1329, 2018.