# USING EXPLAINABLE RECOMMENDATIONS IN THE WINE INDUSTRY

BRITT MOESKOPS

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
AT THE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
OF TILBURG UNIVERSITY

STUDENT NUMBER

1272910

WORD COUNT

8,794

COMMITTEE

dr. Y. Satsangi
dr. M. Jung
dr. D. Shterionov

LOCATION

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science &
Artificial Intelligence
Tilburg, The Netherlands

DATE

January 13, 2022

# USING EXPLAINABLE RECOMMENDATIONS IN THE WINE INDUSTRY

BRITT MOESKOPS

**Abstract**

Purchasing wine is often a challenging decision for many wine consumers. However, current technology offers new opportunities using recommendation systems. Explanations and reasoning for wine recommendations are still missing, even though it can increase transparency, scrutability and trust. Therefore, this thesis discusses (explainable) wine recommendations models with help of two datasets. The datasets contain information about different wines, users reviews and ratings from the WineEnthusiast and Vivino website respectively.

Two collaborative filtering models and a hybrid recommendation model are discussed in this thesis. The collaborative filtering recommendation model based on the SVD algorithm can only return the predicted rating for a specific user and specific wine. The collaborative filtering recommendation model based on the k-nearest neighbor algorithm returns the most similar users to the test user. The hybrid model uses a content-based analysis combined with the k-nearest neighbor algorithm to get the most similar review. Based on the most similar user and review, an explanation recommendation sentence can be build.

The main finding is that the hybrid recommendation model outperforms the collaborative filtering recommendation model as the explainable recommendations are more accurate. Resulting in more effective recommendations and an easier purchase decision for wine consumers.

## PREFACE

Dear reader,

I am very happy and proud that you are reading my thesis for the master Data Science and Society. Completing this thesis means the ending of an era. After some amazing years, it is time to say goodbye to the student life. It is a pity to end my studies during the worldwide pandemic but I enjoyed every minute in Tilburg. I have learned so much and I met so many nice people who are friends for life. I am looking forward making the next step by starting my professional career.

I would like to take this opportunity to thank my supervisors from Tilburg University, Yash Satsangi and Merel Jung, for guiding me through the entire process. Thank you for your feedback, critical look to improve my thesis and for your time!

Lastly, I would also like to thank my father, mother, sister and boyfriend for always supporting me and believing in me. In addition, thank you to my friends who have motivated me to study but more importantly drink coffees in the library! Those were fun times!

Before you go to the next page, I would recommend to take a glass of wine and enjoy reading this thesis, cheers!

Britt Moeskops

## DATA SOURCE/CODE/ETHICS STATEMENT

Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data and code used in this thesis retains ownership of the data and code during and after the completion of this thesis. The author of this thesis acknowledges that they do not have any legal claim to this data or code. All images used in this thesis are created by the author.

The first dataset, WineEnthusiast, used in this thesis was obtained from Zack Thoutt (2017) and is is publicly available on www.Kaggle.com. It is allowed to share and adapt the dataset as long as appropriate credit to the author is given and the material is not used for commercial purposes.

The second dataset is collected by web scraping www.Vivino.com. All data that is scraped from the website is publicly available.

Lastly, code from Roald Schuring (2019) is used in this thesis. The code is publicly available on www.towardsdatascience.com and www.Github.com.

# 1 INTRODUCTION

Section 1.1 explains the problem statement of this thesis and discusses the relevance of this problem. Besides that, the research questions are developed in section 1.2. Lastly, the main findings of this thesis are summarized in section 1.3.

## 1.1 *Context*

Enjoyment is the most important reason for consuming wine and therefore it is crucial that the consumer likes the purchased wine (Charters & Pettigrew, 2008). However, purchasing wine is a challenging decision for many wine consumers. Imagine you are standing in the supermarket in front of the wine section because you want to buy some wine. Probably, the first question that pops up in your head is "where do I even start with this purchase decision process?". There are thousands of different wines to choose from with different characteristics, such as the grapes that are used or the year of production. Besides the large assortment, wine is also an experiential good which cannot be fully experienced without tasting it (Cooper-Martin, 1991). As a result, the absence of tastings or prior experience with the wine, makes it hard to try new, unknown wines.

Katarya and Saini (2021) address that even though it is possible that your friends taste will not match yours, people use the advice from friends for trying a different wine after some time. However, modern technology has instigated that customers make purchase decisions in a different way (Higgins, Wolf, & Wolf, 2014). The rise of wine applications (apps) offers wine consumers the opportunity to learn more about wine, discover new wines and record their own tastings (Higgins et al., 2014). In addition, because of the current technology, recommendation systems are developed which recommend wine to users based on previous wine history and ratings or similar characteristics (Nagarnaik & Thomas, 2015). But when the recommendation system shows you five different bottles of wine, which one do you choose to try? Argumentation for these recommendations is still missing which results in less effective recommendations and for the consumers it is still hard to make a purchase decision.

Netflix, the online streaming website, is a popular example using an explainable recommendation system (*www.Netflix.com*, n.d.). It recommends movies or series based on your personal history and ratings. More importantly, it also shows why these particular movies or series are recommended. For example: 'Because you watched 'Batman Begins', 'Spider-man' is recommended.' (Lee & Jung, 2018). As a result, the recommendations are more transparent and more effective (Ren, Liang, Li, Wang, & de Rijke, 2017).

The Vivino app has created a matchmaking feature to recommend wine to their users (*www.Vivino.com*, n.d.). However, this recommendation feature lacks reasoning and explanations as it only shows whether a bottle of wine is a great, average or low match with the app user. Cruz, Van, and Gautier (2018) and Katarya and Saini (2021) have looked into wine recommendation systems. However, no research is done on explainable wine recommendation systems. Therefore, this thesis aims to close this literature gap by discussing two explainable wine recommendation systems and one non-explainable wine recommendation system. In addition, to evaluate the performance of the recommendation systems, a new evaluation metric is proposed as it is very hard to evaluate explainable recommendations according to Ekstrand, Riedl, and Konstan (2011).

## 1.2 *Research questions*

As stated above, purchasing wine is a challenging decision and the rise of technology offers new opportunities. Other industries are already working with explainable recommendation systems, yet the wine industry is staying behind. Explanations and reasoning for wine recommendations are still missing while artificial intelligence (AI) can help to make recommendations more effective for users. Therefore, the problem statement that will be pursued in this thesis is:

> *How can a collaborative filtering, an explainable collaborative filtering and an explainable hybrid recommendation model, based upon user preferences, user similarities and wine reviews respectively, be used to make the most accurate explainable recommendations in order to simplify the wine purchasing decision for wine consumers?*

The following research questions follow from the problem statement:

RQ1 *What is the difference between a collaborative filtering recommendation model based on the SVD algorithm and an explainable collaborative filtering recommendation model based on the k-nearest neighbor algorithm in terms of recommending new wines to wine consumers?*

RQ2 *Comparing an explainable user-based collaborative filtering recommendation model and an explainable hybrid recommendation model, which model makes the most accurate recommendations for wine consumers measured in terms of overlapping keywords in the recommendation explanations?*

## 1.3 *Findings*

This thesis discusses three different recommendation models, two collaborative filtering models and a hybrid model. The collaborative filtering recommendation model based on the SVD algorithm only returns the predicted rating for a specific user and specific wine. The user-based collaborative filtering recommendation model based on the k-nearest neighbor algorithm returns the most similar users to the test user. Therefore, the algorithm is more informative as an explanation sentence can be build based on this information.

Based on a content-based filtering approach and an item-based collaborative filtering approach a more robust framework is build for the hybrid model (Das, Sahoo, & Datta, 2017). In short, the content-based approach creates review embeddings from the user's reviews which is in line with Rehurek and Sojka (2011). With help of the tf-idf representation, the item-based collaborative filtering approach searches for the most similar reviews with the k-nearest neighbor algorithm (Felfernig et al., 2014). Recommendations are made based on the most similar reviews. Argumentation for these recommendations are created by personalizing a sentence explanation as stated by Zhang and Chen (2018).

Comparing the user-based collaborative filtering recommendation model and the hybrid recommendation model, the conclusion is that the hybrid model is giving the most accurate recommendations in terms of overlapping keywords in the explanations sentences. With help of the explanation sentences, consumers can make easier purchase decisions because the recommendations are more transparent and effective (Ren et al., 2017).

# 2 RELATED WORK

This section reviews previous research regarding the problem statement. In section 2.1, recommendation systems are explained and the benefits of using them. Followed by the different types of recommendation systems in section 2.2. Previous research regarding wine recommendation systems is discussed in section 2.3 and explainable recommendation systems in section 2.4. Lastly, the evaluation of recommendations is discussed in section 2.5.

## 2.1 *Recommendation systems*

Due to technology, smartphone apps have become essential in our daily life and they are available at all time, even while shopping (Alrumayh, Lehman, & Tan, 2021). For instance, the wine app Vivino makes it able to photograph wine labels in order to receive information about that particular wine (Ginters, 2020). Besides identifying wine, users can rate and review wines as well as share experiences on the Vivino platform (Kotonya, De Cristofaro, & De Cristofaro, 2018). In addition, because of the current technology, recommendation systems are developed. Recommendation systems are software tools and techniques that identify the user's interests and suggest products to customers based on these interests (Das et al., 2017; Nagarnaik & Thomas, 2015; Tan, Guo, & Li, 2008). The recommendation systems use large sets of information to reduce complexity for the users by selecting the relevant information to match the user's needs (Davoodi, Kianmehr, & Afsharchi, 2013). It helps website visitors to obtain relevant information following an easier search process to the destination (Nagarnaik & Thomas, 2015). Therefore, it is much easier to filter through the huge amount of different product or item choices resulting in an increased user satisfaction (Das et al., 2017; Ricci, Rokach, & Shapira, 2015). Another benefit is that personalized marketing advertisements result in achieving maximum profit for the company (Das et al., 2017). For instance, for Netflix it is critical to connect subscribers with movies that they will love because the subscribers will otherwise abandon the service (Bennett, Lanning, et al., 2007). For the wine industry the same applies as consumers can easily choose to switch to another (alcoholic) drink.

## 2.2 *Recommendation methods*

Various approaches for building a recommendation system are developed in the literature (Isinkaye, Folajimi, & Ojokoh, 2015). Figure 1 illustrates that two different recommendation systems - non-personalized and personalized - exist (Khatwani & Chandak, 2016). Non-personalized recommendation systems are relevant to all users as it creates a top N items of popular items. Personalized recommendation systems in contrast, suggest items for users based on their past behavior, their preferences and the relationship to other users (Das et al., 2017). Personalized recommendation systems can be divided into a content-based filtering approach, a collaborative filtering approach and a hybrid approach (Nagarnaik & Thomas, 2015).

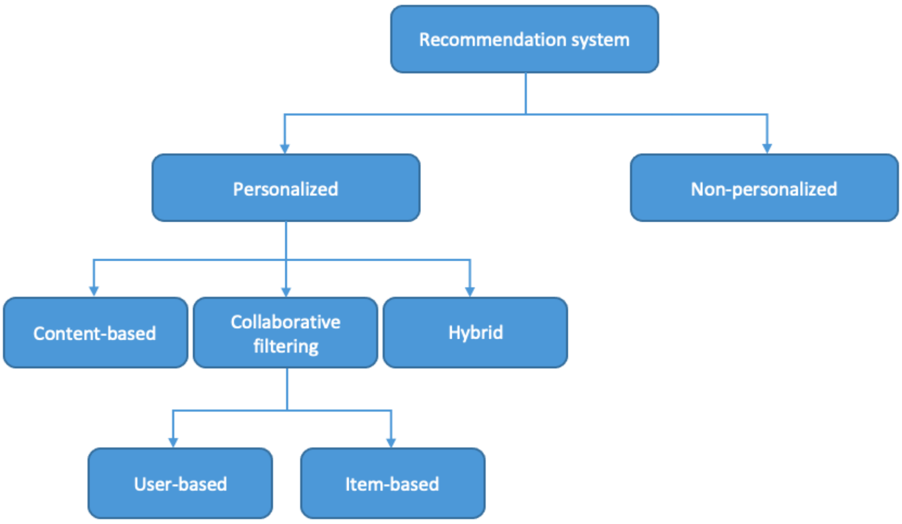


Figure 1: Recommendation methods (Arekar et al., 2015)

Content-based filtering uses the customer's interest and past experience to suggest similar items (Das et al., 2017). According to Das et al. (2017), the benefit of using a content-based filtering approach is that other user's data is not required and there is no data sparsity issue. However, content analysis must be done in order to define the item features (Das et al., 2017). The study of Ricci et al. (2015) states that in most of these cases simple keywords are extracted from item's descriptions. The most used algorithm in a content-based approach is the term frequency - inverse document frequency (tf-idf) representation (Pazzani & Billsus, 2007). This means that words are converted into a numerical feature value based on its term frequency in the given text (Forman, 2008).

The collaborative filtering approach makes use of the user's interest and similar decisions made by other users while evaluating items (Nagarnaik & Thomas, 2015; Pazzani & Billsus, 2007). This can be based on a user-based approach or an item-based approach. "A user-based collaborative filtering approach identifies the k-nearest neighbors of the active user and, based on these nearest neighbors, calculates a prediction of the active user's rating for a specific item." (Felfernig et al., 2014, p.17). The item-based collaborative filtering approach searches for items that received similar ratings. The most used algorithm for collaborative filtering is the k-nearest neighbor according to Thorat, Goudar, and Barve (2015). The k-nearest neighbors method has the advantage of being simple and transparent while giving reliable results (Chemeque Rabel, 2020). Das et al. (2017) state that the benefit of the collaborative filtering approach is that items can be estimated very accurate due to the user's ratings. The drawbacks however, are the cold start problem and the scaleability problem. The cold start problem happens when there is an initial lack of ratings and no reliable recommendations can be made (Bobadilla, Ortega, Hernando, & Bernal, 2012). The scaleability problem occurs when a large number of users and items are being computed (Xue et al., 2005). On the one hand, Van Den Oord, Dieleman, and Schrauwen (2013) conclude that collaborative filtering approaches are outperforming the content-based recommendation systems. On the other hand, Basu, Hirsh, Cohen, et al. (1998) highlight that using explicit data as well as other forms of information can result in a content-based model which outperforms a collaborative filtering model.

The hybrid approach operates with both recommendation systems. Combining both a content-based and a collaborative filtering approach can greatly increase the accuracy of the results (Chemeque Rabel, 2020). Vall et al. (2019) agree that hybrid recommendation systems can outperform collaborative filtering approaches. The hybrid model builds a more robust framework as one method can reduce the weaknesses of the other model (Das et al., 2017).

### 2.3 *Wine recommendation systems*

Chen, Rhodes, Crawford, and Hambuchen (2014) predict whether a wine scored above or below certain points based on wine reviews using different clustering algorithms and association rules. Their paper makes use of wine sensory analysis which means that a consumer is able to describe the wine after tasting it. "This does not only include flavors and aromas but also characteristics such as acidity, tannin, and structure." (Chen et al., 2014, p.2). Multiple studies are using the Wine Aroma Wheel (Chen et al., 2014; Noble et al., 1987). Table 1 illustrates how the Wine Aroma Wheel looks like. This wheel contains unique fragrances which can be found in wine and can therefore perfectly describe a wine in words (Noble et al., 1987). The wine could for instance contain fragrances of fruit, but there are a lot of subsections of what kind of fruit it can be such as red fruit or melon. These subsections have subsections of it own; red fruit could be divided into raspberry or cherry. The Wine Aroma Wheel contains all different flavors, aromas or characteristics which can be found in wine and is therefore very informative.

Table 1: Example of the Wine Aroma Wheel (Noble et al., 1987)

| | | |
|---|---|---|
| **Fruit** | Dark fruit | Plum |
| | | Black currant |
| | Red fruit | Raspberry |
| | | Cherry |
| | Melon | Watermelon |
| | | Honeydrew melon |
| | Citrus fruit | Lemon |
| | | Grapefruit |
| **Caramel** | Honey | Honeycomb |
| | | Beeswax |
| | Chocolate | Dark chocolate |
| | | Cocoa |

Besides predicting wine ratings, only two papers discuss wine recommendation systems. Cruz et al. (2018) process the vocabulary of experts and non-experts to use word embeddings techniques in order to create a wine recommendation system. The study of Cruz et al. (2018) uses Word2Vec to transform the vocabulary of experts and non-experts into a vector space. Word2Vec is the main technique to calculate the similarity and relevance of words meaning and is for instance also used for recommending university courses (Jatnika, Bijaksana, & Suryani, 2019; Pardos, Fan, & Jiang, 2019). Based on a hybrid recommendation model, the recommendation system computes the cosine similarity between the description vector of an item and a user profile (Cruz et al., 2018). An item that is more similar to an item that was preferred, is more relevant to the user.

The other wine recommendation system is the Greedy Clustering Wine recommendation System by Katarya and Saini (2021). This study uses principal component analysis and K-means clustering algorithms. Data including the quality of the wine, which is given by wine experts and wine characteristics, such as amount of percentage of alcohol and amount of acid in the wine are used. The main conclusion of this research is that the personalized suggestions are getting better when the number of users increases (Katarya & Saini, 2021).

## 2.4 *Explainable recommendation systems*

Besides the rise of technology, society is currently also using more AI systems on a daily basis with rapid developments in image recognition, speech analysis and recommendation systems (Adadi & Berrada, 2018; Došilović, Brčić, & Hlupić, 2018). However, such AI systems usually lack transparency; it allows to make predictions but the explanation behind a certain prediction or recommendation is missing (Adadi & Berrada, 2018). Explainable AI proposes a shift towards more transparent AI to produce more explainable models. Explainable recommendations address why particular items are recommended (Zhang & Chen, 2018). Xu et al. (2019) mention that it is important that users understand the underlying reason when AI recommends a decision. Wrong decisions can be costly and dangerous and therefore verifying that the recommendation system is working as expected is needed. In addition, explanations make the recommendations more transparent and effective (Ren et al., 2017). Users can make decisions based on these explanations which are for instance in line with their mood (Ren et al., 2017).

As a result, more research is done on explainable recommendation systems. The study of Hong, Akerkar, and Jung (2019) uses the user's movies history to explain which movie would be recommended and why. Their study builds its explanations on explanation sentences such as: ''You may like the movie 'Call Hell' today because you have enjoyed this kind of horror movies." (Hong et al., 2019, p.7). The recommendations are made based on different elements in the movie domain such as genre. The study concluded that the performance of an explainable recommendation system for movies outperforms other methods. The study of Vig, Sen, and Riedl (2009) uses movie tags as features to explain the recommended movie based on a k-nearest neighbor algorithm. The movie features are ranked between 1-5 stars based on the user's preference, which explains why it is relevant for the user. The model helps to promote the goals of effectiveness and mood compatibility and is therefore very useful (Vig et al., 2009). Next to movie tags, generating (personalized) sentences or visual explanations such as histograms or product images, are the most common recommendation explanations (Zhang & Chen, 2018).

Both research papers highlight that explainable recommendation systems increase transparency, scrutability and trust and ease the purchase decision process.

Based on the benefits of using AI in recommendation systems, research questions 1 and 2 are developed to examine the difference in the recommendations itself and which recommendation model performs best in the wine industry.

## 2.5 *Recommendations evaluation*

Shani and Gunawardana (2011) mention that the success of a recommendation system depends on what the goal is. For instance, discovering similar items could be relevant but discovering new diverse items could be relevant as well. Therefore, the set of relevant properties should be set based on what the goal of the recommendation system is. When the properties are set, the performance of the system can be evaluated (Shani & Gunawardana, 2011). However, evaluating the explanations of recommendations itself is very hard. Ekstrand et al. (2011) even argue that it cannot be measured how users respond to recommendations. Online and offline evaluation approaches are described by Zhang and Chen (2018). However, both approaches depend on the actual feedback of users in experimental groups. To conclude, no standard evaluation for explainable recommendations can be found in the literature. Therefore, this thesis will propose a new evaluation method.

# 3 METHOD

This section discusses the methodology of this thesis. First, the datasets are described in section 3.1. Followed by highlighting the important findings during the performed exploratory data analysis (EDA) in section 3.2. In section 3.3 the preprocessing steps are motivated. Next, the experimental procedure is explained including the evaluation metrics in section 3.4. Lastly, section 3.5 gives an overview of the software used in this thesis.

## 3.1 *Dataset description*

This thesis is using two different datasets. The first dataset is obtained from www.Kaggle.com. The data was scraped from www.WineEnthusiast.com on November 22nd, 2017 by Zack Thoutt. WineEnthusiast.com is an online magazine which covers wine, food, travel and entertaining topics. Different variables were scraped from the website such as the wine name, item ratings given by a user, the username and the user's review. Before doing EDA, the dataset consists of 129,971 observations containing 10 different variables. Table 2 presents an overview of the dataset with the variable names, the variable types and the description of each variable. For instance, the column 'variety' represents the grapes used in the wine and is an object. The advantage of this dataset is that it contains multiple wine reviews per user. The drawback however, is that all users review different wines in the dataset with only a few exceptions where a wine is rated by two users or more. Because of this limitation, a second dataset is used in this thesis.

Table 2: Variables available in the WineEnthusiast dataset

| **Column** | **Type** | **Description** |
|---|---|---|
| Country | Object | Country where the wine is produced |
| Description | Object | The written review of a customer |
| Points | Int 64 | The rating for a wine given by a unique customer |
| Price | Float 64 | Price of the wine per bottle |
| Province | Object | Province where the wine is produced |
| Region 1 | Object | Region where the wine is produced |
| Taster name | Object | Name of the customer |
| Title | Object | The name of the wine |
| Variety | Object | The grapes used in the wine |
| Winery | Object | The winery which produced the wine |

The second dataset is collected by web scraping the website www.Vivino.com. Vivino is an online wine marketplace which makes use of crowd-sourced data from millions of wine drinkers around the world. The web scraping was done on September 27th, 2021. Just like the first dataset different variables were obtained from the Vivino website. An overview of all variables, variable types and the description of each variable are illustrated in Table 3. Before doing EDA, the dataset consists of 29,064 observations with 14 variables. In contrast to the first dataset, this dataset contains multiple available reviews per wine.

Table 3: Variables available in the Vivino dataset

| **Column** | **Type** | **Description** |
|---|---|---|
| Wine ID | Int 64 | Each wine owns a unique wine id number |
| Wine | Object | The name of the wine |
| Winery | Object | The winery which produced the wine |
| Year | Int 64 | The year wherein the wine was produced |
| Price | Float 86 | Price of the wine per bottle |
| Country | Object | Country where the wine is produced |
| Rating | Float 64 | The average rating of the wine |
| Num review | Int 64 | The total number of different reviews of the wine |
| User rating | Float 64 | The rating for a wine given by a unique customer |
| Note | Object | The written review of a customer |
| Created at | Object | The date and time of which the review was published |
| Language | Object | The language of the review |
| Username | Object | The username of the reviewer on the website |

To avoid confusion regarding the datasets, the first dataset will be called the WineEnthusiast dataset while the second dataset will be called the Vivino dataset in the following sections.

## 3.2 *EDA*

After collecting the datasets, EDA is performed to get familiar with and get a more in-depth understanding of both datasets. It is important to do EDA as ''the more one knows about the data, the more effectively data can be used to develop, test and refine theory." (Hartwig & Dearing, 1979, p.9). The main findings are presented below.

The percentage of missing values for the WineEnthusiast dataset is calculated and plotted in Figure 9, which can be found in the Appendix. One important variable is missing data in the dataset, namely the user column. 20% of the users is unknown, which could lead to the cold start problem in the collaborative filtering approach (Bobadilla et al., 2012). For the content-based approach it would not be a problem as no user's data is required (Das et al., 2017). The Vivino dataset is not missing any values as Figure 10 in the Appendix shows.

The numbers of reviews per user is plotted in Figure 11 in the Appendix for the WineEnthusiast dataset. The most remarkable finding is that the majority of the users reviewed more than 1,000 wines and one user reviewed even over 25,000 wines. In the Vivino dataset, the users did not review as much wines per person. Within the 30,542 users, 28,208 are unique which means that only 7,64% of the users reviewed more than 1 wines.

Furthermore, the number of unique wines and varieties are calculated for the WineEnthusiast dataset. From the 129,971 wines, 118,840 are unique which means that only 8,56% is reviewed more than 1 time. The variety of wines are, in contrast, more often the same as the grapes used in wine are divided into categories such as 'Merlot' and 'Pinot Gris'. Figure 12 in the Appendix shows this distribution plot. In the Vivino dataset it is the other way round; the dataset only contains 23 unique wines. Figure 13 in the Appendix illustrates the number of reviews per unique wine. To conclude, even though the datasets contain the same information about wines, the datasets are different.

The distributions of the ratings are also plotted. The distribution of the frequency of ratings in the WineEnthusiast dataset is plotted in Figure 14 in Appendix. The conclusion of the distribution is that most ratings are between 86 and 91 out of 100. For the Vivino dataset the range is between 1 and 5, however, most ratings are set between 3.2 and 4.0 in Figure 15 in the Appendix. This means that both datasets make use of a different rating scale.

Lastly, a plot based on the reviews is created with help of word cloud. It shows the words who are mentioned in the user's reviews. The bigger the size of the word, the more frequent it is mentioned in the reviews. Figure 16 and Figure 17 in the Appendix show the word cloud of the WineEnthusiast dataset and the Vivino dataset respectively. The figures illustrate that the wines are described in the same way such as 'red fruit' and 'red berry'.

## 3.3 *Preprocessing*

Now that there is a more in-depth understanding of the datasets, the next step is to preprocess the data to be able to run the models in this thesis. The first step is to filter all the English reviews from the Vivino dataset as it contains reviews in multiple languages. The reason behind this is that the language needs to be specified for certain functions that will be used. Besides that, it is very hard to check whether the algorithms work between the different functions if the words are not readable.

Secondly, the ratings for wine items in both datasets are normalized with the Min-Max feature scaling technique. The reason behind this decision is that both datasets use different ranges for rating the wines. In order to be able to compare them with each other, the ratings are set between 0 and 1. Using other scaling techniques would result in an unfair rating distribution.

As explained in the EDA section, the WineEnthusiast dataset is missing data for the user column. This could result in the cold start problem in the collaborative filtering models. Therefore, all rows are deleted when the user is unknown in order to prevent this problem. This will only be done at the collaborative filtering models and not the hybrid model as this model does not require user's data.

Lastly, because this thesis makes use of two datasets, one dataset will be the training set and the other dataset will be the test set. The datasets cannot be merged as the datasets contain different features. The WineEnthusiast dataset will be used as the training dataset. Based on this dataset, the models will be trained and fine tuned. The Vivino dataset is used once the models are fine tuned and parameters are chosen. The Vivino dataset is therefore the test set as the data is unknown to the models. For all users in the Vivino dataset a recommendation will be made based on the information of the Vivino dataset. Followed by evaluating the performance of the model based on these recommendations.

Further required preprocessing steps will be explained in the experimental procedure per specific model.

## 3.4 *Experimental procedure*

This thesis implements the following three models:

1. A collaborative filtering recommendation model
2. An explainable user-based collaborative filtering recommendation model
3. An explainable hybrid recommendation model

The three models are explained and choices are motivated in the following subsections. In addition, the evaluation metric per model is described.

### 3.4.1 *Model 1: A collaborative filtering recommendation model*

The first model is a collaborative filtering recommendation model without explanations. The goal of this recommendation system is to predict the missing ratings for all remaining user-item pairs (Hug, 2020). A collaborative filtering approach is chosen as it can estimate the user's ratings very accurately and make reliable recommendations (Das et al., 2017). In addition, this is the most used recommendation method at the moment (Sánchez-Moreno, González, Vicente, Batista, & García, 2016).

This recommendation system is based on the singular value decomposition (SVD) algorithm which is a collaborative filtering technique (Guan, Li, & Guan, 2017). This is a matrix factorization algorithm which is performed by a stochastic gradient descent (Hug, 2020). The SVD algorithm has gained a lot of attention the past years as it resulted in a more accurate movie recommendation system (Ranjan, Rai, Haque, Lohani, & Kushwaha, 2019). The formula for predicting $\hat{r}_{ui}$ using SVD is as follows:

$$\hat{r}_{ui} = u + b_u + b_i + q_i^T p_u \tag{1}$$

$\hat{r}_{ui}$ represents the prediction that user $u$ likes or dislikes item $i$, $b$ represents the bias and $p_u$ and $q_i$ the interest preferences between users and items.

With help of Figure 2, the pipeline of this model is explained. The first step is to convert the dataset into the right dataset object in order to be able to feed the recommendation system algorithm. This is done by preprocessing the dataset which consists of a few steps. First, a reader object is specified with the range of the ratings in the dataset, which is set between 0 and 1 as a result of normalizing the ratings. Secondly, all usernames and wine names are converted to unique id integers. Thirdly, a pivot table is created which then consists of the username ids as rows and the wine ids as columns with the values being the ratings for each wine by a taster. Now, the dataset is ready to be feed to the recommendation system. The SVD algorithm is applied to the training dataset which means that it is predicting the ratings for items for the users. After running the algorithm, the evaluation score is returned to see how well the model predicted the ratings.

For this model, two different evaluation metrics are chosen. The Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) are used as these metrics are the most popular accuracy measures in the literature of recommendation systems (Yang, Steck, & Liu, 2012). RMSE is the variance of the arithmetic square root by measuring the deviation between the predicted user rating and the actual user rating (Yin, Wang, & Park, 2017). MAE is the deviation between the predicted user ratings and the user real ratings and is used as well to give a broader view of the algorithm (O'Doherty, Jouili, Van Roy, et al., 2012; Yin et al., 2017). The smaller the values of RMSE and MAE, the better the performance (Tang et al., 2016).

In order to get the smallest RMSE and MAE values as possible, the model is fine tuned with help of GridSearchCV. With GridSearchCV different sets of parameters are applied to the model in order to return all optimized parameters. The advantage of GridSearchCV is that it does not fall in local minima (Zhao, Mao, Lin, Yin, & Xu, 2020). In this case, GridSearchCV optimizes the number of epochs, the learning rate and the regularization term.

The combination of parameters with the lowest error is used in the model to execute on the unknown test set. This way the model is optimized and predicts the most accurate ratings for users for items. Based on the outcome of the model on the test set, recommendations are made for specific users. This is done by creating a dataframe with the results.

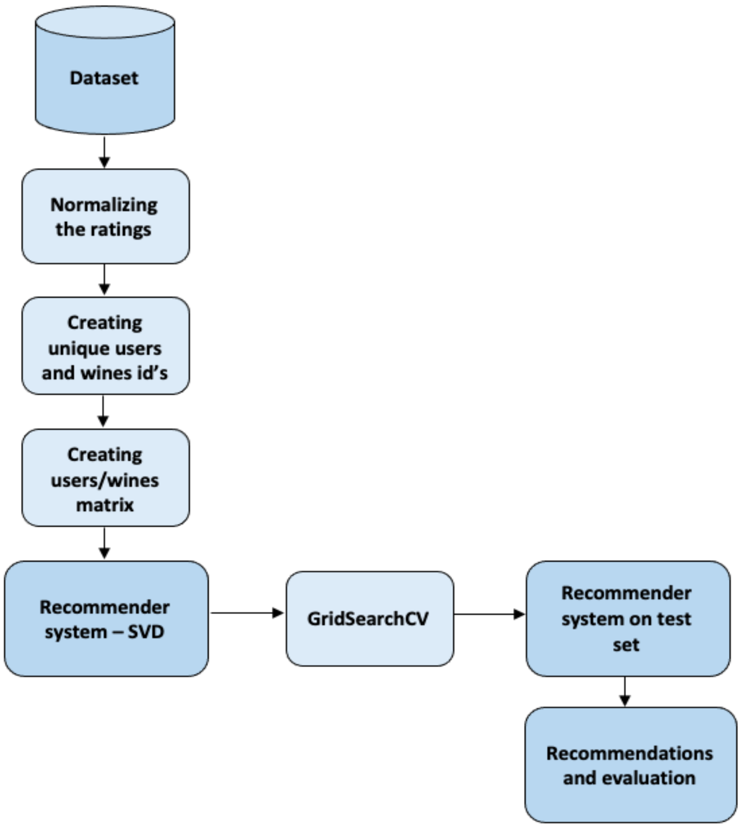


Figure 2: Pipeline collaborative filtering recommendation model

### 3.4.2 *Model 2: An explainable user-based collaborative filtering recommendation model*

The second model is based on an explainable user-based collaborative filtering approach. Based on a collection of users profiles, this approach aims to predict the user's interest for a given item (Wang, De Vries, & Reinders, 2006). The model measures the similarities between a test user and the other users in the dataset. Commonly this is done with explicit data, such as ratings. Figure 3 shows the pipeline of the explainable user-based collaborative filtering recommendation model which is explained step by step.

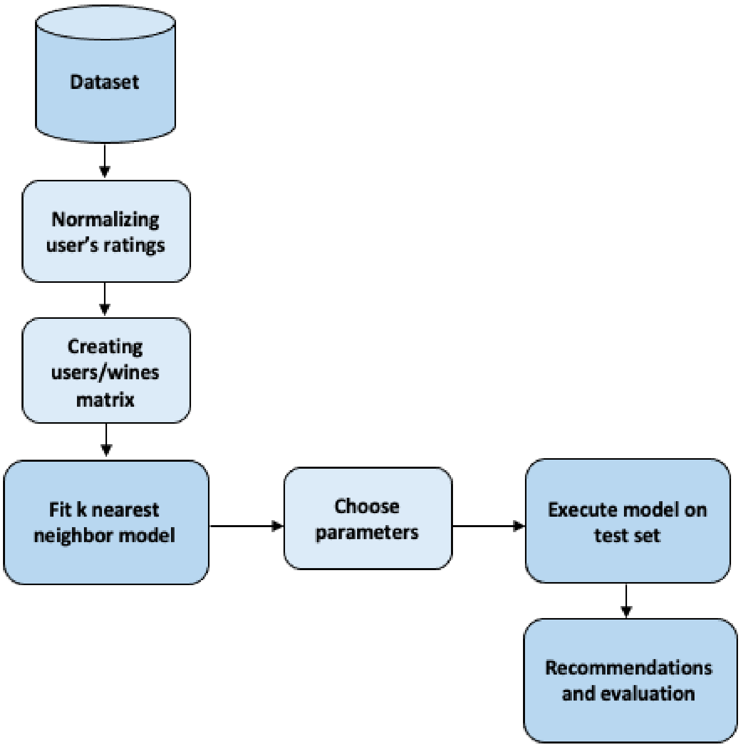


Figure 3: Pipeline explainable user-based collaborative filtering recommendation model

First, to be able to apply the user-based collaborative filtering approach, the dataset needs to be processed into a pivot table again. The principle is to store all data in a users/wines matrix (Chemeque Rabel, 2020). Instead of creating a pivot table with the unique wine id's, the pivot table of the training set is created with the variety of wines. The reason behind this is that the training set only contains ratings about unique wines, with only a few exceptions where the same wine is rated by more than one user. The variety contains more relevant information as the varieties are used for multiple wines.

The algorithm that is used for this model is the k-nearest neighbor algorithm which is one of the most used algorithms (Thorat et al., 2015). The principle behind this algorithm is to find a predefined number of training samples closest in distance to the new point and predict the label of this new point based on the training samples (Pedregosa et al., 2011).

This means that the algorithm is searching for users that are closest to the test user. Based on these users items are recommended.

In order to train the model, different parameters are chosen to apply. Parameters could be changing the number of neighbors, the algorithm used in the model and the distance metric. The parameters resulting in the highest accuracy are chosen. Calculating the accuracy of the model is based on a different evaluation metric compared to the first model. This accuracy evaluation is focused on the explainable part of the recommendations itself. As mentioned in section 2.5, evaluating the recommendation explanations is hard (Ekstrand et al., 2011). As people want to try new wines after some time, the goal of this thesis is to recommend wines that are similar to previous wines (Katarya & Saini, 2021). Therefore, evaluating whether the recommended wine is similar to a wine that the user has liked in the past, is the relevant setting. Based on this, a new evaluation approach is proposed in this thesis. This new evaluation approach is explained below with help of a recommendation example.

The explanations of the recommendations are based on a sentence-level approach. This means that an explanation sentence template will be used which is filled with different keywords to personalize the explanation sentence for different users. The explanation sentence of the user-based collaborative filtering method is:

*'Users most similar to [name of user] are: 1. [Name of most similar user] with distance [cosine distance] 2. [Name of second most similar user] with distance [cosine distance] 3. [Name of third most similar user] with distance [cosine distance]*

*Based on the wine ratings of [Name of most similar user], the following wine variety scores high for both users: [variety]. Therefore, the following wine is recommended: [name of wine].'*

If the recommended name of wine actually falls within the recommended variety, the recommendation is evaluated as correct. This is because the test user will receive a suggestion based on a variety that the test user likes.

The parameters resulting in the highest accuracy on the training set are chosen to apply on the test set. Of all possibilities, the brute algorithm is chosen together with the number of neighbors set at 10. The distance metrics that is used is the cosine similarity. It measures how similar two items are based on their subject (Gunawan, Sembiring, & Budiman, 2018). The test set is used to execute the model on. The pipeline in figure 3 shows that creating recommendations is the next step. In order to make explainable recommendations, a function is build. With k-nearest neighbors the most similar user to the test user is found. The function searches for the wine varieties that correspond to both users. For example, if both users rank the variety 'Cabernet Sauvignon' very high, the best rated 'Cabernet Sauvignon' wine of the most similar user is recommended to the test user. If all recommendations are made, the evaluation score is calculated.

### 3.4.3 *Model 3: Hybrid recommendation model*

The third model is build based on an existing algorithm from Roald Schuring which was posted on May 30, 2019 on www.towardsdatascience.com (Schuring, 2019). The model is a hybrid recommendation model as it combines a content-based approach with an item-based collaborative filtering approach. In contrast to the explainable user-based collaborative filtering model, this model tries to match users with items that are similar to what they have liked in the past (Aggarwal, 2016).

The content-based approach of this model is based on the attributes of the items and is similar to the research of Cruz et al. (2018) regarding processing vocabulary. User's reviews are used to create word embeddings in order to create a wine recommendation system. To be able to create the word embeddings, some pre-process steps should be undertaken as visualized in Figure 4.

All sentences from the reviews are combined into one corpus in order to normalize these sentences. Next, all words are converted into lower cased strings. Followed by mapping different forms of the same word and removing punctuation. Lastly, very common words from the English language such as 'a', 'by' and 'the' are removed as these words do not provide relevant information. Sometimes word pairs are more valuable than words on its own. Therefore, the most relevant bi- and tri-grams are extracted from the normalized sentences. This is done by detecting common phrases from a stream of sentences which consists of multi-word expressions. All n-grams are being counted and compared to the words in the Wine Aroma Wheel as explained in section 2.3. As explained by Chen et al. (2014) and Noble et al. (1987), contains this wheel unique fragrances which can be found in wine and can perfectly describe a wine in different words. Therefore, these words are very informative. The words in the user's reviews that correspond to the words in the Wine Aroma Wheel are used in the following steps. This way the words are also standardized to compare.

Next, all words are mapped to vectors of real numbers with Word2Vec embeddings. This is one of the most popular technique to learn word embeddings using a two-layer neural network (Rehurek & Sojka, 2011).

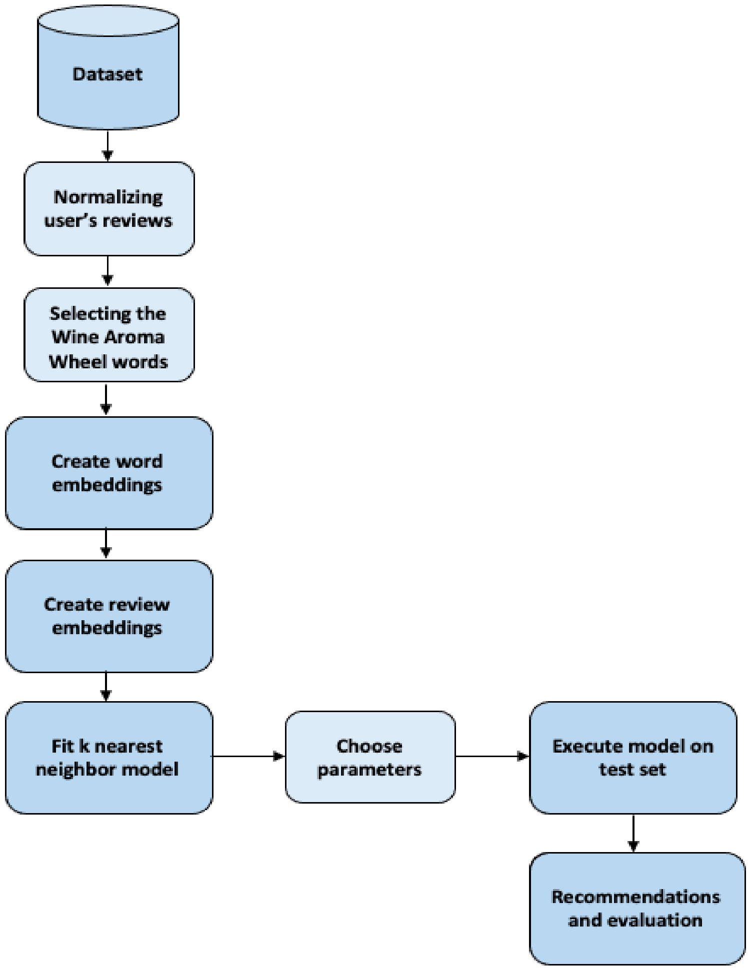


Figure 4: Pipeline hybrid recommendation system

This technique is also commonly used in a content-based model (Cruz et al., 2018; Jatnika et al., 2019; Misztal-Radecka & Indurkhya, 2020). The target is to place similar words close to each other in the same vector space. For instance, the word 'stone fruit' is the most similar to the word 'peach'. After creating word embeddings, review embeddings are created. The TfidfVectorizer function converts the collection of words to a matrix of tf-idf features (Pedregosa et al., 2011). The words that appear more frequently will be weighted less compared to words that appear less frequent. The reason behind this is that the most common words do not give as much information as the less frequent words. Pazzani and Billsus (2007) also state that this is the most used algorithm in content-based models.

Now, the data is ready to be fed to the k-nearest neighbor function. The k-nearest neighbor method has the advantage of being simple and transparent which is the goal of an explainable recommendation system (Chemeque Rabel, 2020). Similar to the explainable user-based collaborative filtering recommendation model, this model searches for the most similar items based on the wine reviews.

The distance metrics that is used in the k-nearest neighbor model is the cosine similarity. The cosine similarity is chosen as this distance measure is often chosen when working with vector space models (Chemeque Rabel, 2020). In addition, the brute algorithm is chosen together with the number of neighbors set to 10.

After running the k-nearest neighbor model on the test set, the recommendations are created. The recommendations within the hybrid model are based on a sentence-level approach, just like the explainable user-based collaborative filtering model. Again an explanation sentence template is used to fill with different keywords to personalize the explanation sentence for different wines. However, the difference between the models is that this model uses an item-item similarity to fill the keywords in the explanation sentence instead of a user-based similarity. Instead of recommending a wine to a specific user, a user can search for the most similar wine by calling a particular wine.

The following sentence template is used for the recommendation:

*'Because you like [name of wine] which contains the following descriptors: [descriptor], [descriptor], [descriptor].*

*The following wines are recommended: suggestion 1 is [name of wine]. This wine has the following descriptors: [descriptor], [descriptor], [descriptor]'.*

This way, the model gives a solid explanation why a certain consumer would like this wine. Zhang and Chen (2018) argue that a user's review can be considered as the ground-truth explanation for purchasing an item. As all wine descriptors in the explainable recommendations are extracted from the user's reviews, the wine descriptors themselves can be seen as ground-truth. Therefore, the evaluation of the explainable recommendations is set differently. Again, recommending the most similar wine is the goal of the recommendation system. Therefore, the recommendation is evaluated as correct when the recommended wine matches with at least one descriptor of the original wine.

## 3.5 *Software*

The experiments in this thesis are done using Python (version 3.8.8) which is an interpreted, high-level, general-purpose programming language (Van Rossum and Drake, 1995). Python contains multiple built-in libraries which are also used. Performing the data exploratory analysis and performing the pre-processing steps is done with the following four libraries. The *NumPy* package provides a multidimensional array object and an assortment of operations on arrays (Harris et al., 2020). The *Matplotlib* library is worldwide used for making visualization (Hunter, 2007). In addition, the *Seaborn* library is used to visualize the data (Waskom et al., 2017). The *Pandas* library is used for data manipulation and analysis (McKinney et al., 2010).

Besides these libraries, other libraries are used in order to perform the recommendation systems. First of all, the *Surprise* package is used for building and analyzing recommendation systems that deal with explicit rating data (Hug, 2020). The *Scikit-learn* library is used for the user-based collaborative filtering model as well as processing data for the other models (Pedregosa et al., 2011). Various algorithms from the *Natural Language Toolkit* library are used to process the language in the wine reviews (Bird, Klein, & Loper, 2009). Lastly, the word embeddings technique from the *Gensim* package is applied (Rehurek & Sojka, 2011). Running the experiments will be performed in the Jupyter Notebook environment in Anaconda (*Anaconda Software Distribution*, 2020).

# 4 RESULTS

The results of the collaborative filtering and the explainable collaborative filtering recommendation model are presented in section 4.1 and section 4.2 respectively. The results of the explainable hybrid recommendation model are discussed in section 4.3.

## 4.1 *A collaborative filtering recommendation model*

After running the SVD algorithm on the training set, the evaluation metrics RMSE and MAE are executed. Table 4 shows the results of the evaluation metrics. The smaller the values of RMSE and MAE, the better the performance which means that the model predicted the ratings already very well. The GridSearchCV was set to apply the following parameters: number of epochs: [5,10, 15], the learning rate: [0.002, 0.005] and the regularization term: [0.4, 0.6, 0.8]. These are the most common parameters to apply on the GridSearchCV stated by Hug (2020). Instead of trying all the different parameters, the GridSearchCV function returns the best parameters. The best combination is setting the number of epochs at 15, the learning rate at 0.005 and the regularization term at 0.4. These parameters are applied to the SVD algorithm on the test set in order to make predictions. The RMSE on the test set is 0.1891 and the MAE is 0.1464. These values are higher compared to the training set. As the training and test set are two different datasets, it is not remarkable that the RMSE and MAE scores are lower. It means that the ratings for the training set are better to predict. All errors are quite low which means that the model predicted the data accurately; however, the ratings are normalized between 0 and 1. This means that in general the predicted rating and the actual rating cannot be too far from each other.

Table 4: Results collaborative filtering recommendation model

| | **RMSE** | | **MAE** | |
|---|---|---|---|---|
| **Model** | *Training set* | *Test set* | *Training set* | *Test set* |
| **SVD** | 0.145 | 0.1891 | 0.117 | 0.1464 |

The recommendations for the recommendation system are illustrated in Figure 5. The recommendations only show the predicted rating for a given item for a given user. No explanations are found why a particular item receives a low or high rating. As no more information is available besides the predicted ratings, the recommendations itself cannot be evaluated besides the RMSE and MAE measures.

| | user_id | wine_id | Real_Rating | Estimated_Rating |
|---|---|---|---|---|
| 0 | 2050 | 4 | 0.375 | 0.581308 |
| 1 | 4208 | 1 | 0.750 | 0.629464 |
| 2 | 3477 | 4 | 1.000 | 0.581308 |
| 3 | 4851 | 2 | 0.500 | 0.539178 |
| 4 | 7079 | 12 | 0.375 | 0.667039 |
| 5 | 14071 | 15 | 0.375 | 0.682612 |
| 6 | 9591 | 18 | 1.000 | 0.665069 |
| 7 | 14073 | 4 | 0.000 | 0.581308 |
| 8 | 13640 | 16 | 0.750 | 0.653511 |
| 9 | 5711 | 6 | 0.750 | 0.577250 |

Figure 5: Recommendations from the collaborative filtering recommendation model

### 4.2 *An explainable user-based collaborative filtering recommendation model*

After fitting the explainable user-based collaborative filtering model to the training data, different parameters are executed in order to fine tune the model. The brute force and auto algorithm are used but the similarity between the users does not change when switching between the algorithms while keeping the other parameters equal. The brute force algorithm is chosen because most recommendation methods in the literature use this algorithm stated by Jaiswal, Kharade, Kotambe, and Shinde (2020). Changing the number of neighbors does also not contribute to a higher accuracy. Trying the last parameter, the distance metrics, makes differences in the outcome of the model.

Applying the cosine distance returns a different sequence of most similar users compared to the euclidean and Manhattan distance. However, this is not surprising as the formulas of distance metrics are quite different. While the cosine distance returns a float between 0 and 1, return the other distance metrics an integer. The cosine similarity is however, one of the most widely used distance metrics for representing the relationship between two sets stated by Liao and Xu (2015). In addition, Dessì, Recupero, Fenu, and Consoli (2019) mention that the cosine distance is more reliable than the Euclidean distance. Therefore, this thesis uses the cosine similarity as parameter in the k-nearest neighbor model. The chosen parameters are the following: number of neighbors = 10, algorithm = 'brute' and metric = 'cosine'.

Next, the explainable recommendations are created by writing a function. This function consists of a few steps. First, the rated wine varieties are ordered descended based on the average rating per variety per user. The next step is to compare the wine varieties between the test user and the most similar user based on the k-nearest neighbor algorithm. When both users rate the same variety high, a wine based on this variety is recommended. To be specific, the best rated wine in that particular variety of the most similar user is recommended. Figure 6 shows how the recommendations look like. This explainable recommendation is evaluated as correct as the recommended wine falls within the 'Pinotage' variety, as explained in section 3.4.2.

```
Users most similar to Carrie Dykes are:

1: Alexander Peartree with distance: 0.6381447691134793
2: Paul Gregutt with distance: 0.6667280688856374
3: Fiona Adams with distance: 0.6724695507429594
4: Anna Lee C. Iijima with distance: 0.6842046442359614
5: Mike DeSimone with distance: 0.6948870790985401

Based on the wine ratings of Alexander Peartree
The following variety scores high for both users: Pinotage
Therefore, the following wine is recommended: Lovingston 2013 Gilberts Vineyard Pinotage

The following variety scores high for both users: Touriga
Therefore, the following wine is recommended: CrossKeys 2013 Touriga (Virginia)
```

Figure 6: Explainable recommendations from the user-based collaborative filtering recommendation model

Applying the model on the test set did unfortunately not give the expected results. Even though that the model is working, the available data in the test set makes it impossible to return recommendations. The reason behind it, is that the users in the dataset only review one wine per person. Even though there are a few exceptions where users review more than one wine, the cosine metrics returns a 0 for all the users that review the same wine. Recommending a different wine for the test user is very hard as the similar users did not review more wines. In order to resolve this problem, more data should be gathered in order to apply the model successfully.

## 4.3 *Hybrid recommendation system*

The hybrid recommendation system model is fine tuned on the training set by changing the parameters. The parameters that are analyzed are the following: the number of neighbors is set on [5, 10, 15], the algorithm of the k-nearest neighbor model is set on [brute and auto] and the distance is compared between [cosine and euclidean]. However, changing any of the parameters did not result in a lower or higher accuracy. Because the parameters did not change the accuracy, the same parameters that are applied to the explainable user-based collaborative filtering model are chosen, which are: number of neighbors = 10, algorithm = 'brute' and metric = 'cosine'.

After the parameters are chosen, the model is fitted on the test set and the explainable recommendations are created. The wine recommendations itself are based on the k-nearest neighbor model which returns the three wines that are most similar to the test wine based on the wine reviews. The explanation sentences are returned with help of a formula by filling in the wine descriptors in the explanation sentence. Based on this, the user can see why the wines are recommended based on the wine characteristics. Figure 7 illustrates an example of an explainable recommendation. This example shows that at least one wine descriptor of the recommended wines is equal to the wine descriptor of the original wine. As explained in section 3.4.3, this recommendation is evaluated as correct. Figure 8 shows an explainable recommendation which is evaluated as incorrect as the wine descriptors of the recommended wines do not correspond with the wine descriptors of the original wine.

```
Because you like Chronic Cellars 2015 Stone Fox White (Paso Robles)
Which contains the following descriptors: ['crisp', 'apple', 'peach', 'wet_rocks', 'crisp', 'grippy', 'melon',
'lime_pith']
__________
The following wines are recommended:
Suggestion 1 is Quinta da Aveleda 2001 Trajadura (Vinho Verde) with a cosine distance of 0.047
This wine has the following descriptors: ['peach', 'apricot', 'apple', 'citrus', 'crisp', 'refreshing', 'tart',
'chalk']

Suggestion 2 is Nebla 2012 Verdejo (Rueda) with a cosine distance of 0.049
This wine has the following descriptors: ['minerality', 'apple', 'citrus', 'zesty', 'citrus', 'orange', 'nectar
ine', 'crisp', 'chalk']

Suggestion 3 is Tangent 2015 Paragon Vineyard Albariño (Edna Valley) with a cosine distance of 0.050
This wine has the following descriptors: ['pith', 'grapefruit', 'lemon', 'crisp', 'pear', 'chalk', 'light_bodie
d', 'crisp', 'apple', 'pith']
```

Figure 7: Correct explainable recommendation from the hybrid recommendation system

```
Because you like Quinta dos Avidagos 2011 Avidagos Red (Douro)
Which contains the following descriptors: ['ripe', 'fruit', 'smooth', 'firm', 'juicy', 'berry', 'fresh']
_________
The following wines are recommended:
Suggestion 1 is Château Fondarzac 2012 Bordeaux Supérieur with a cosine distance of 0.066
This wine has the following descriptors: ['crisp', 'citrus', 'snappy', 'racy', 'citrus', 'lime', 'nettle', 'tan
gerine', 'green', 'tart']

Suggestion 2 is Château Bertinerie 2011 Grande Cuvée (Blaye Côtes de Bordeaux) with a cosine distance of 0.069
This wine has the following descriptors: ['off-dry', 'green', 'apple', 'lime', 'zesty', 'crisp']

Suggestion 3 is Château Garrineau 2012 Bordeaux with a cosine distance of 0.071
This wine has the following descriptors: ['tangy', 'crisp', 'sharp', 'kiwi', 'lime', 'grapefruit']
```

Figure 8: Incorrect explainable recommendation from the hybrid recommendation system

For each wine in the dataset, a recommendation is made based on the k-nearest neighbor algorithm. The name of the original wine, the wine descriptors of the original wine as well as the recommended wines and its wine descriptors are saved into a dataframe. The accuracy is calculated checking each row in this dataframe whether at least one wine descriptor of the original wine is equal to at least one wine descriptor of the recommended wines.

The accuracy of the explainable recommendations on the test set is 100%. It is remarkable that each recommendation is evaluated correctly but the test set contains only a small number of unique wines which makes it more understandable. To conclude, the hybrid model is performing very well on the training set as well as on the test set.

# 5 DISCUSSION

As explained in the introduction, purchasing wine is a challenging decision for many wine consumers as it is an experiential good (Cooper-Martin, 1991). However, due to the development of recommendation systems, tools and techniques can identify the user's interests and based on these interests suggest products to customers (Das et al., 2017). Argumentation for these recommendations are still missing which results in less effective recommendations. In addition, for the consumers it is still hard to make purchase decisions. Therefore, this thesis aims to discuss how recommendation models can be used to make explainable wine recommendations for wine consumers.

Several recommendation models can be found in previous research (Chemeque Rabel, 2020; Das et al., 2017; Forman, 2008; Hug, 2020; Nagarnaik & Thomas, 2015; Pazzani & Billsus, 2007; Ricci et al., 2015; Vall et al., 2019). This thesis chooses to discuss a collaborative filtering recommendation model, an explainable collaborative filtering recommendation model and an explainable hybrid recommendation model.

The collaborative filtering model is based on the SVD algorithm and predicts the ratings of items for users. With help of the available explicit data, the missing ratings for items and users are predicted. In order to return a recommendation for a specific user and item, the user and item should be called directly. The recommendation returns a number which represents whether the user will like or will not like the item, which is in this case the wine.

The explainable user-based collaborative filtering model based on the k-nearest neighbor algorithm searches for the users that are most similar to the test user based on the historical wine ratings. Based on the highest rated items of the most similar user a recommendation is returned. With help of a formula, an explanation can be returned why this particular wine is recommended by filling in keywords in a recommendation explanation sentence template.

Both recommendation models use the collaborative filtering approach, however the recommendation itself are different. While the SVD algorithm only returns the predicted rating for a specific user and specific wine when calling the pair, returns the k-nearest neighbor model the most similar users to the test user. The most important difference is the amount of information the algorithm return. Based on the k-nearest neighbor information, an explanation sentence can be build to show why the recommendation system recommended a particular wine. This is in contrast with the SVD algorithm as this algorithm only predicts the missing ratings. There is no information available to build a sentence to explain why the recommended wine received a low or high rating. In other words, the SVD algorithm is in this case a black box that is unable to explain it outputs (Haghighi, Seton, & Nasraoui, 2019).

The third recommendation model discussed in this thesis, does not use explicit data. Based on a content-based filtering approach and an item-based collaborative filtering approach a more robust framework is build (Das et al., 2017). In short, the content-based approach creates review embeddings from the user's reviews which is in line with Rehurek and Sojka (2011). With help of the tf-idf representation, the item-based collaborative filtering approach calculates the most similar reviews with the k-nearest neighbor algorithm. The most similar wines obtain the similar wine describing keywords. This means that the wines have similar wine characteristics. Argumentation for these recommendations are created by personalizing a sentence explanation as stated by Zhang and Chen (2018).

The evaluations of the explainable recommendations are based on the goal of the recommendation system as there is no standard evaluation for recommendations (Ekstrand et al., 2011; Shani & Gunawardana, 2011; Zhang & Chen, 2018). The explainable recommendations in both the user-based collaborative filtering model and the hybrid model are based on a sentence-level approach. This means that keywords in the sentences are added to make them personalized. The proposed evaluation metrics is based on analyzing these keywords in the recommendations because for both recommendation models the goal is to recommend the most similar wines. For the collaborative filtering recommendation model the explanation sentence can be evaluated as correct if the recommended name of wine actually falls within the recommended variety. For the hybrid model the recommendation can be evaluated as correct when at least one wine description in the recommended wine is equal to one wine description in the test wine.

When comparing the explainable user-based collaborative filtering recommendation model with the explainable hybrid recommendation model, the conclusion is that the hybrid recommendation model is the best model to use as it gives the most accurate recommendations in terms of overlapping keywords in the explanations. Both models worked quite well based on the k-nearest neighbor algorithm just like Chemeque Rabel (2020) stated. However, the expected results for the explainable collaborative filtering model were not achieved. Due to a data limitation, the accuracy score on the test dataset cannot be measured. In order to solve this problem, more data should be gathered. The results on the training set state that the model is working and can therefore be seen as valid. The hybrid model in contrast, does not need ratings in order to make recommendations as it uses product features (Das et al., 2017). Therefore, the hybrid model can be applied to both datasets as the model does not require the users profiles. As a result, the data limitations do not play a role. Besides that, the hybrid recommendation model acquired a 100% accuracy. The hybrid model outperforming the collaborative filtering model is in line with the findings of Chemeque Rabel (2020) and Vall et al. (2019).

To summarize, several recommendations models can be build in order to suggest (new) products to customers. Recommendations including explanations improve transparency and reliability and are more effective as stated by Ren et al. (2017) and Vig et al. (2009). For instance, wine consumers can try a new wine which is in line with their mood with help of recommendation explanations. This thesis concludes that the hybrid recommendation model in this thesis can be used to make the most accurate explainable recommendations in order to simplify the wine purchasing process for wine consumers. Discussing two different explainable wine recommendation models is the contribution of this thesis as no research have been done before on this topic.

# 6 CONCLUSION

The objective of this thesis was to study how recommendation models can be used to make the most accurate explainable recommendations in order to simplify the wine purchasing decision for wine consumers. The answer is that the hybrid recommendation model is the best model to use as it gives the most accurate recommendations in terms of overlapping keywords in the explanations. Based on a content-based filtering approach and an item-based collaborative filtering approach a more robust framework is build (Das et al., 2017). In short, the content-based approach creates review embeddings from the user's reviews which is in line with Rehurek and Sojka (2011). With help of the tf-idf representation, the item-based collaborative filtering approach searches for the most similar reviews with the k-nearest neighbor algorithm. Recommendations are made based on the most similar reviews. Argumentation for these recommendations are created by personalizing a sentence explanation as stated by Zhang and Chen (2018). With help of the explanation sentences, consumers can make easier purchase decisions. In addition, the recommendations are more transparent, effective and reliable as consumers know how the recommendations are made (Ren et al., 2017).

An explainable collaborative filtering recommendation model can be used as well, however, due to dataset limitations the model did not performed as expected in this thesis. Therefore, future work should be focusing on improving the model by collecting more data. The dataset should contain multiple wine ratings per user and multiple users rating the same wine. Resulting in preventing the cold start problem which leads to making reliable recommendations. In addition, because of time limitation, future work should also include other feature attributes. Interesting examples could be the year in which wine was produced, the country of production and the price of the wine. These features could for instance highlight a different relationship regarding the recommendation system compared to the wine reviews.

## REFERENCES


Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, *6*, 52138–52160.

Aggarwal, C. C. (2016). Content-based recommender systems. In *Recommender systems* (pp. 139–166). Springer.

Alrumayh, A. S., Lehman, S. M., & Tan, C. C. (2021). Emerging mobile apps: Challenges and open problems. *CCF Transactions on Pervasive Computing and Interaction*, *3*(1), 57–75.

*Anaconda software distribution.* (2020). Anaconda Inc. Retrieved from https://docs.anaconda.com/

Arekar, T., Sonar, M., & Uke, N. (2015). A survey on recommendation system. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, *2*, 1–5.

Basu, C., Hirsh, H., Cohen, W., et al. (1998). Recommendation as classification: Using social and content-based information in recommendation. In *Aaai/iaai* (pp. 714–720).

Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of kdd cup and workshop* (Vol. 2007, p. 35).

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Bobadilla, J., Ortega, F., Hernando, A., & Bernal, J. (2012). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-based systems*, *26*, 225–238.

Charters, S., & Pettigrew, S. (2008). Why do people drink wine? a consumer-focused exploration. *Journal of Food Products Marketing*, *14*(3), 13–32.

Chemeque Rabel, M. (2020). *Content-based music recommendation system: A comparison of supervised machine learning models and music features.*

Chen, B., Rhodes, C., Crawford, A., & Hambuchen, L. (2014). Wineinformatics: applying data mining on wine sensory reviews processed by the computational wine wheel. In *2014 ieee international conference on data mining workshop* (pp. 142–149).

Cooper-Martin, E. (1991). Consumers and movies: Some findings on experiential products. *ACR North American Advances*.

Cruz, C., Van, C. N., & Gautier, L. (2018). Word embeddings for wine recommender systems using vocabularies of experts and consumers. *Open Journal of Web Technologies (OJWT)*, *5*(1), 23–30.

Das, D., Sahoo, L., & Datta, S. (2017). A survey on recommendation system. *International Journal of Computer Applications*, *160*(7).

Davoodi, E., Kianmehr, K., & Afsharchi, M. (2013). A semantic social network-based expert recommender system. *Applied intelligence*, *39*(1), 1–13.

Dessì, D., Recupero, D. R., Fenu, G., & Consoli, S. (2019). A recommender system of medical reports leveraging cognitive computing and frame semantics. In *Machine learning paradigms* (pp. 7–30). Springer.

Došilović, F. K., Brčić, M., & Hlupić, N. (2018). Explainable artificial intelligence: A survey. In *2018 41st international convention on information and communication technology, electronics and microelectronics (mipro)* (pp. 0210–0215).

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011). *Collaborative filtering recommender systems*. Now Publishers Inc.

Felfernig, A., Jeran, M., Ninaus, G., Reinfrank, F., Reiterer, S., & Stettinger, M. (2014). Basic approaches in recommendation systems. In M. P. Robillard, W. Maalej, R. J. Walker, & T. Zimmermann (Eds.), *Recommendation systems in software engineering* (pp. 15–37). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-45135-5_2 doi: 10.1007/978-3-642-45135-5_2

Forman, G. (2008). Bns feature scaling: an improved representation over tf-idf for svm text classification. In *Proceedings of the 17th acm conference on information and knowledge management* (pp. 263–270).

Ginters, E. (2020). Digital technologies acceptance/adoption modeling respecting age factor. In *World conference on information systems and technologies* (pp. 621–630).

Guan, X., Li, C.-T., & Guan, Y. (2017). Matrix factorization with rating completion: An enhanced svd model for collaborative filtering recommender systems. *IEEE access*, *5*, 27668–27678.

Gunawan, D., Sembiring, C., & Budiman, M. A. (2018). The implementation of cosine similarity to calculate text relevance between two documents. In *Journal of physics: conference series* (Vol. 978, p. 012120).

Haghighi, P. S., Seton, O., & Nasraoui, O. (2019). An explainable autoencoder for collaborative filtering recommendation. *arXiv preprint arXiv:2001.04344*.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. doi: 10.1038/s41586-020-2649-2

Hartwig, F., & Dearing, B. E. (1979). *Exploratory data analysis* (No. 16). Sage.

Higgins, L. M., Wolf, M. M., & Wolf, M. J. (2014). Technological change in the wine market? the role of qr codes and wine apps in consumer wine purchases. *Wine Economics and Policy*, *3*(1), 19–27.

Hong, M., Akerkar, R., & Jung, J. J. (2019). Improving explainability of recommendation system by multi-sided tensor factorization. *Cybernetics and Systems*, *50*(2), 97–117.

Hug, N. (2020). Surprise: A python library for recommender systems. *Journal of Open Source Software*, *5*(52), 2174.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian informatics journal*, *16*(3), 261–273.

Jaiswal, S., Kharade, T., Kotambe, N., & Shinde, S. (2020). Collaborative recommendation system for agriculture sector. In *Itm web of conferences* (Vol. 32).

Jatnika, D., Bijaksana, M. A., & Suryani, A. A. (2019). Word2vec model analysis for semantic similarities in english words. *Procedia Computer Science*, *157*, 160–167.

Katarya, R., & Saini, R. (2021). Enhancing the wine tasting experience using greedy clustering wine recommender system. *Multimedia Tools and Applications*, 1–34.

Khatwani, S., & Chandak, M. (2016). Building personalized and non personalized recommendation systems. In *2016 international conference on automatic control and dynamic optimization techniques (icacdot)* (pp. 623–628).

Kotonya, N., De Cristofaro, P., & De Cristofaro, E. (2018). Of wines and reviews: measuring and modeling the vivino wine social network. In *2018 ieee/acm international conference on advances in social networks analysis and mining (asonam)* (pp. 387–392).

Lee, O.-J., & Jung, J. J. (2018). Explainable movie recommendation systems by using story-based similarity. In *Iui workshops.*

Liao, H., & Xu, Z. (2015). Approaches to manage hesitant fuzzy linguistic information based on the cosine distance and similarity measures for hfltss and their application in qualitative decision making. *Expert Systems with Applications*, *42*(12), 5328–5336.

McKinney, W., et al. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (Vol. 445, pp. 51–56).

Misztal-Radecka, J., & Indurkhya, B. (2020). Getting to know your neighbors (kyn). explaining item similarity in nearest neighbors collaborative filtering recommendations. In *Adjunct publication of the 28th acm conference on user modeling, adaptation and personalization* (pp. 59–64).

Nagarnaik, P., & Thomas, A. (2015). Survey on recommendation system methods. In *2015 2nd international conference on electronics and communication systems (icecs)* (pp. 1603–1608).

Noble, A. C., Arnold, R. A., Buechsenstein, J., Leach, E. J., Schmidt, J. O., & Stern, P. M. (1987). Modification of a standardized system of wine aroma terminology. *American journal of enology and viticulture*, *38*(2), 143–146.

O'Doherty, D., Jouili, S., Van Roy, P., et al. (2012). Trust-based recommendation: an empirical analysis. In *Sixth acm workshop on social network mining and analysis (sna-kdd 2012).*

Pardos, Z. A., Fan, Z., & Jiang, W. (2019). Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance. *User modeling and user-adapted interaction*, *29*(2), 487–525.

Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web* (pp. 325–341). Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830.

Ranjan, A. A., Rai, A., Haque, S., Lohani, B. P., & Kushwaha, P. K. (2019). An approach for netflix recommendation system using singular value decomposition. *Journal of Computer and Mathematical Sciences*, *10*(4), 774–779.

Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, *3*(2).

Ren, Z., Liang, S., Li, P., Wang, S., & de Rijke, M. (2017). Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of the tenth acm international conference on web search and data mining* (pp. 485–494).

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender systems handbook* (pp. 1–34). Springer.

Sánchez-Moreno, D., González, A. B. G., Vicente, M. D. M., Batista, V. F. L., & García, M. N. M. (2016). A collaborative filtering method for music recommendation using playing coefficients for artists and users. *Expert Systems with Applications*, *66*, 234–244.

Schuring, R. (2019, 05). *Wine Embeddings and a Wine Recommender.* Retrieved from https://towardsdatascience.com/robosomm-chapter-3-wine-embeddings-and-a-wine-recommender-9fc678f1041e

Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, & P. B. Kantor (Eds.), *Recommender systems handbook* (pp. 257–297). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/978-0-387-85820-3_8 doi: 10.1007/978-0-387-85820-3_8

Tan, H., Guo, J., & Li, Y. (2008). E-learning recommendation system. *2008 International Conference on Computer Science and Software Engineering*, *5*, 430-433.

Tang, J., Wang, S., Hu, X., Yin, D., Bi, Y., Chang, Y., & Liu, H. (2016). Recommendation with social dimensions. In *Thirtieth aaai conference on artificial intelligence.*

Thorat, P. B., Goudar, R., & Barve, S. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, *110*(4), 31–36.

Vall, A., Dorfer, M., Eghbal-Zadeh, H., Schedl, M., Burjorjee, K., & Widmer, G. (2019). Feature-combination hybrid recommender systems for automated music playlist continuation. *User Modeling and User-Adapted Interaction*, *29*(2), 527–572.

Van Den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Neural information processing systems conference (nips 2013)* (Vol. 26).

Vig, J., Sen, S., & Riedl, J. (2009). Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on intelligent user interfaces* (pp. 47–56).

Wang, J., De Vries, A. P., & Reinders, M. J. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval* (pp. 501–508).

Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Lukauskas, S., Gemperline, D. C., . . . Qalieh, A. (2017, September). *mwaskom/seaborn: v0.8.1 (september 2017).* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.883859 doi: 10.5281/zenodo.883859

*www.netflix.com.* (n.d.). Retrieved from http://www.netflix.com/

*www.vivino.com.* (n.d.). Retrieved from http://www.vivino.com/

Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., & Zhu, J. (2019). Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Ccf international conference on natural language processing and chinese computing* (pp. 563–574).

Xue, G.-R., Lin, C., Yang, Q., Xi, W., Zeng, H.-J., Yu, Y., & Chen, Z. (2005). Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th annual international acm sigir conference on research and development in information retrieval* (pp. 114–121).

Yang, X., Steck, H., & Liu, Y. (2012). Circle-based recommendation in online social networks. In *Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1267–1275).

Yin, C., Wang, J., & Park, J. H. (2017). An improved recommendation algorithm for big data cloud service based on the trust in sociology. *Neurocomputing*, *256*, 49–55.

Zhang, Y., & Chen, X. (2018). Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*.

Zhao, S., Mao, X., Lin, H., Yin, H., & Xu, P. (2020). Machine learning prediction for 50 anti-cancer food molecules from 968 anti-cancer drugs. *International Journal of Intelligence Science*, *10*(1), 1–8.

## APPENDIX

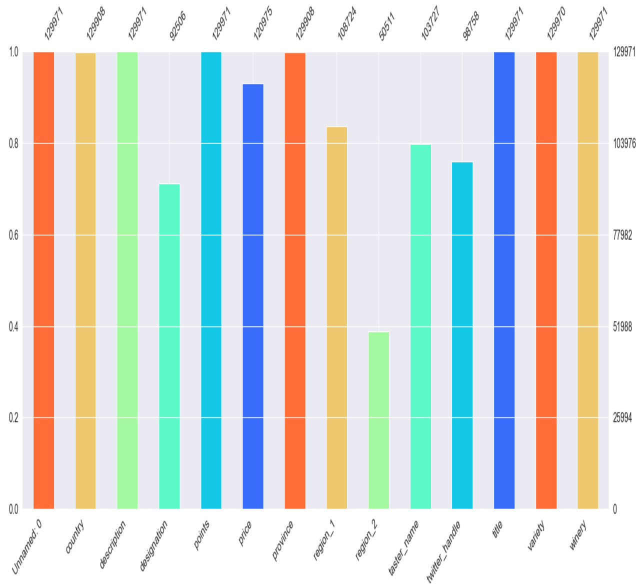


Figure 9: Visualization of missing data in the WineEnthusiast dataset

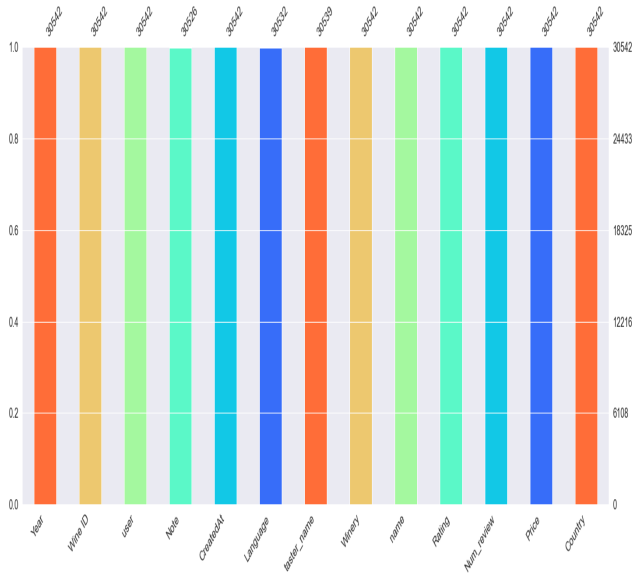


Figure 10: Visualization of missing data in the Vivino dataset

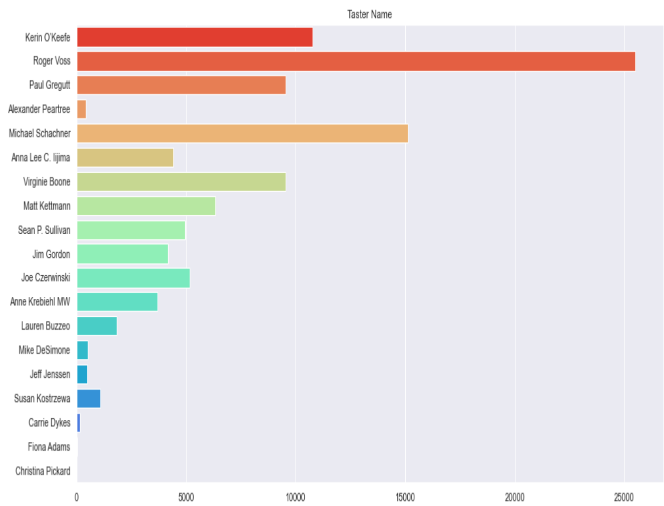


Figure 11: Number of reviews per user in the WineEnthusiast dataset

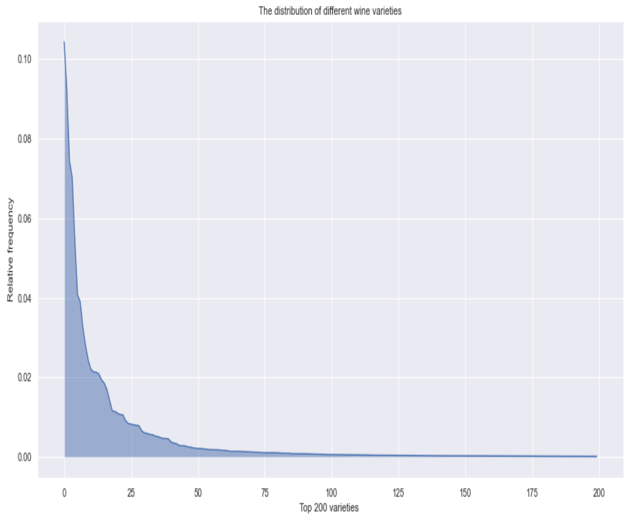


Figure 12: Variety distribution in the Vivino dataset

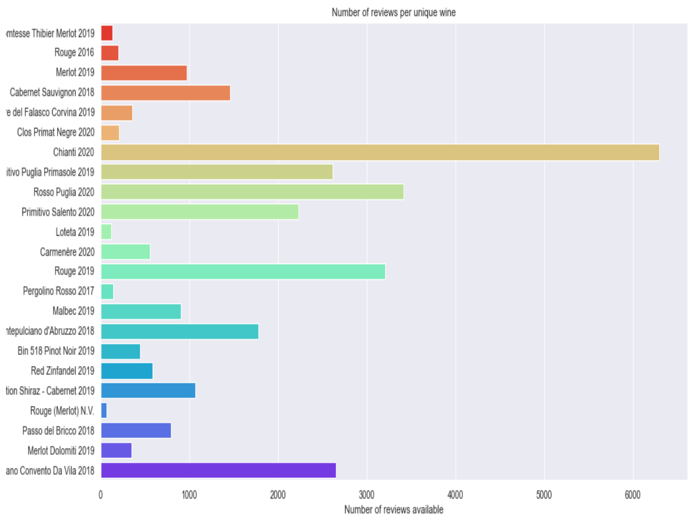


Figure 13: Number of reviews per unique wine in the Vivino dataset

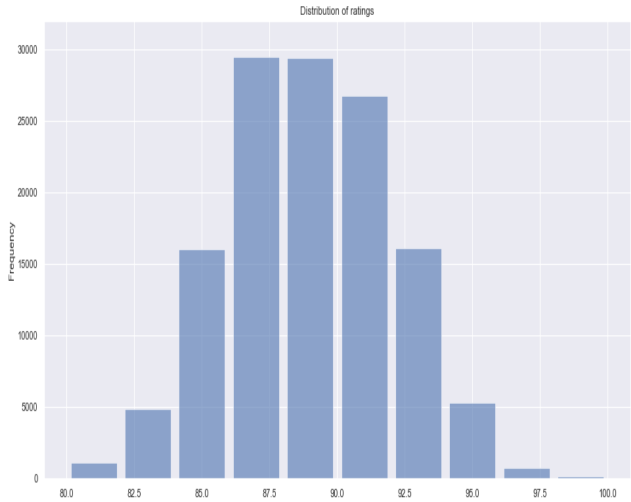


Figure 14: Rating distribution in the WineEnthusiast dataset

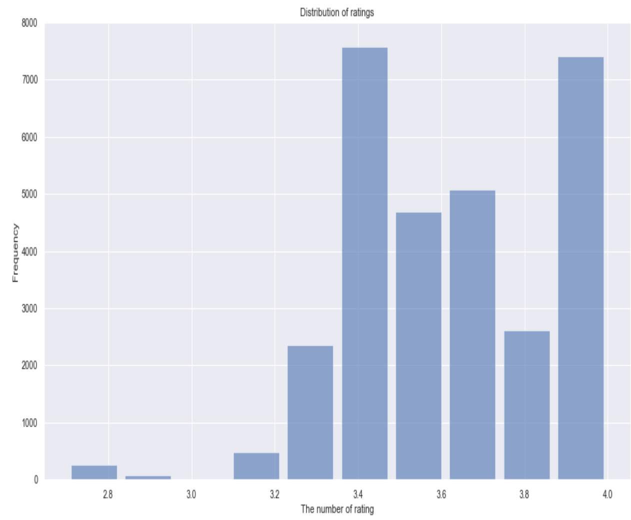


Figure 15: Rating distribution in the Vivino dataset


Figure 16: Visualization of review words in the WineEnthusiast dataset


Figure 17: Visualization of review words in the Vivino dataset