Wine Recommendation System

Thibault Bienvenu1 and Jun Heesu2 and Kim Yoonhee3 and Mahdi Tarraf-Kojok4 and Yang Suyeong5 and Thomas Roche6 (Team 11)

Chung-Ang University, South Korea

Epitech, France

**Abstract.** We have developed Wine Recommendation System by using both knowledge-based recommendation and content-based recommendation. We used knowledge-based recommendation system for people who have not experienced wine or not familiar with wine. And we used content-based recommendation system for people who already have experience with wine. In knowledge-based recommendation, price and flavor are used and in content-based recommendation, country, province, description, variety and title of wines are considered as a factor. All the user's recommendations are pre-calculated by the algorithm. It means that we don't have to wait a request from the user to update his recommendations list. The final recommendation system will be provided in website form and people can use this system in following url: <https://bi-wine-recommender.herokuapp.com/>

**Keywords:** Wine Recommendation, Knowledge-based Recommendation, Content-based Recommendation

1. Motivation

There are so many kinds of wine in the world and people want to get the wine list that fits with their own unique preferences. Variety, vintage, country, province, price, and description are typical factors that represent wine. Therefore, we have decided to recommend wines based on the factors for the people who already have experiences with wine by recommending similar wines with the preferred wine list the person have experienced.

However, for the people who have no experience with wines, we cannot consider the factors above because the person does not have enough information with wines. Because those factors can be sound unfamiliar to them. So, we asked some simple questions about their own acceptable price and preferred flavors like fruity or not, sweet or dry, tannin, acidity, and about the body of wine. We thought these kinds of questions are quite intuitive for the beginner than asking about the typical factors.

Therefore, we have decided to implement recommendation for both kinds of people, who have experience to the wine and who do not have experience to wine, content-based recommendation for the former and knowledge-based recommendation for the latter.

1. System Overview
   1. Flow of overall system with website screenshots

We selected a ‘website’ for providing our service. Most of users are friendly to website and because each person has many different types of cell phone, we thought web service is suitable that everyone can easily access.

스크린샷이(가) 표시된 사진

자동 생성된 설명Our main page is like below. We designed most simple UI of the service that is easy to use for our users. Everything can be checked in this url: **“**[**https://bi-wine-recommender.herokuapp.com/index.html**](https://bi-wine-recommender.herokuapp.com/index.html)**”.**

**Fig. 1.** A main page includes menu tab on the left side of top and join & login button on the right side which is linked with sign up or login page. Users can get an information about the current site by clicking ‘get started’ button.

스크린샷이(가) 표시된 사진

자동 생성된 설명**Fig. 2.** Figures above shows the brief description of our service. By scrolling of the page, user can check about who can enjoy our site, what function is the most proper to me, and the methods used in recommendation.

개체이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명After making account, users can enjoy the web site by getting wine recommendation that is satisfying their preferences. When the user logs in, popup icon appears on the top of the banner. This can be confirmed with Fig 3**.** A menu tab of left side shows all specific categories which contains ‘My wine’ and ‘Recommendation’. These are shown for only registered users”.

**Fig. 3.** There is a button of popup icon next to the logout button which is only visible when users make their account at the first time and all menus except ‘Home’ menu are also visible to only logged in users.

After new user logs in our web site, he can see the popup icon and if he clicks it, a message is showed to him like below (see Fig. 4). First, our site asks a question “Have you ever tasted wine?”. According to the answer for the first question, more questions are followed step by step to get information about user. This is for knowledge-based recommendation which is intended to give results that reflect the user's tastes and experiences and more detailed algorithm will be described from behind.

스크린샷이(가) 표시된 사진

자동 생성된 설명

**Fig. 4.** Depending on the user's click event on the question, additional questions will be shown.

Next is ‘My wine’ page. There is an information about specific user’s preference like flavor, kind of fruit, etc. Products(wines) that users liked also displayed in this page. After obtaining information about a user's preferences, an empty page, as shown in Fig. 5, is filled with information about the user, and a list of recommended products is automatically managed (store and delete) in this ‘My wine’ page.

스크린샷이(가) 표시된 사진

자동 생성된 설명

**Fig. 5.** Before users give their information about their own preferences or experiences of wine, ‘My wine’ page shows nothing.

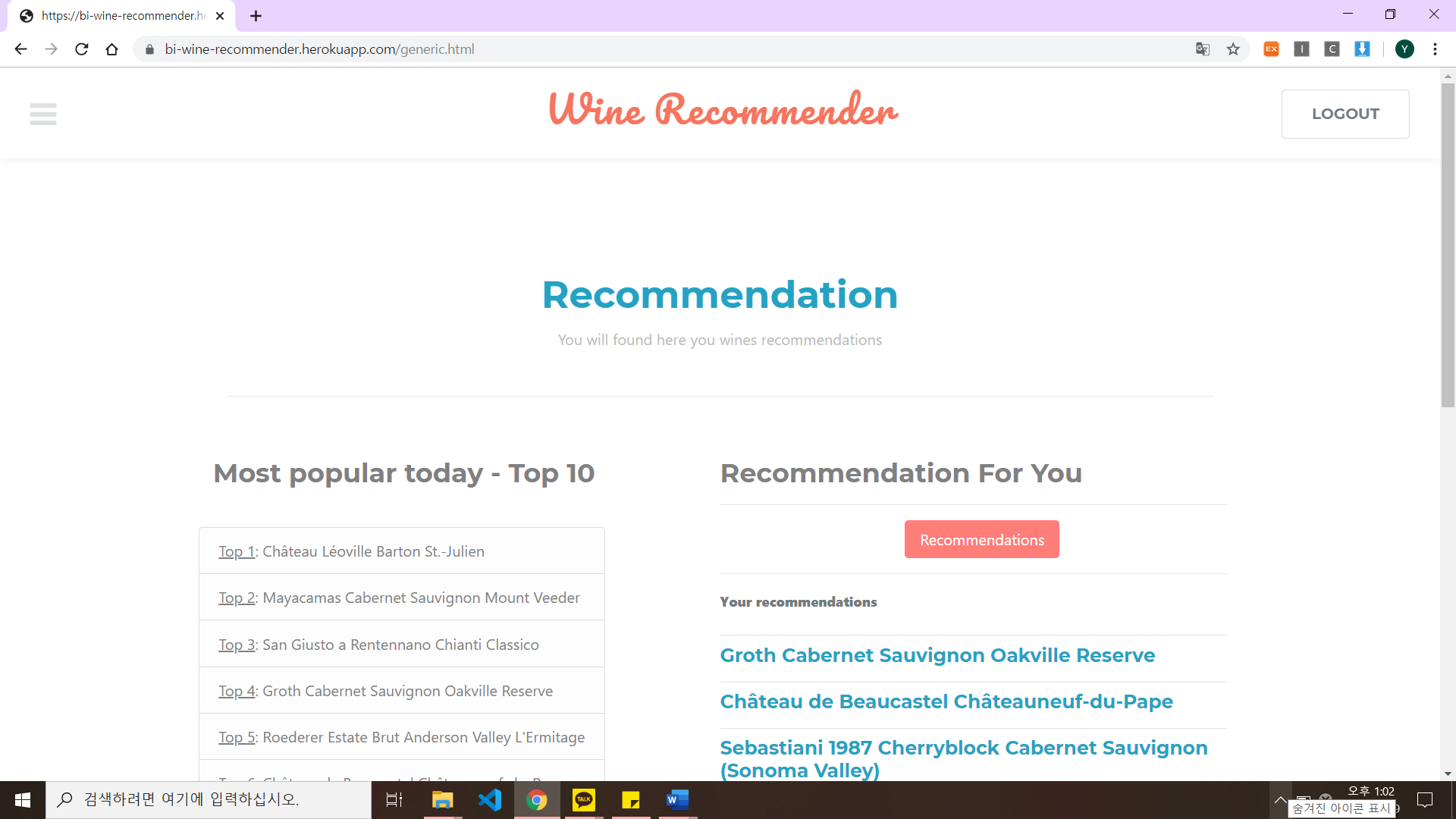
As shown in Fig. 6**,** after user answers every question from popup and our system gets information about the user, the corresponding item will be recommended and user preferences and product information are saved. At first, we plan to show recommended wine only on recommendation page (Fig. 7), but we change to show my wines and recommendation list together on my wine page for convenience. All of wines can be deleted by the user and wines in ‘my wine section’ offer the information of preferences for content-based recommendation. It can be added either by the user's own registration of his favorite wine, or by the result with the recommendation and users have not deleted. (I like it button).

스크린샷이(가) 표시된 사진

자동 생성된 설명

**Fig. 6.** After our web site gets information about users, users can check their own informations on ‘My wine’ page.

In ‘Recommendation’ page, as shown in Fig. 7, a user can check out the 10 most popular wines and he can get recommendations based on his result of answers to the popup questions(knowledge-based-recommendation) and also based on the products that have recommended to the user and he liked . As shown in Fig. 8, the user can check the information of name, province, price, variety, flavors, price of the wine that the user clicked on.



스크린샷이(가) 표시된 사진

자동 생성된 설명**Fig. 7.** The most popular items today and the result of recommendations are shown in this page and the user can get another recommendation by clicking the tab “Similar wine(CB)”.

스크린샷이(가) 표시된 사진

자동 생성된 설명**Fig. 8.** If the user click products of the list, informations collapsed within the box appears.

스크린샷이(가) 표시된 사진

자동 생성된 설명스크린샷이(가) 표시된 사진

자동 생성된 설명If the user has experienced some wines recently, he can search them with their title and register them to the list in ‘My wine’ page and because we provide recommendations based on contents in ‘My page’, that will be the resource for content-based recommendations. We will explain about the more detailed procedure about this algorithm. As shown in Fig. 9,there is input area, where the user can search for the wines and register them.

**Fig. 9.** If the user clicks the box and enters part of wine title, he can see the full title of the wine. If he chooses it and clicks the button “Add”, it will be registered.

1. Datasets
   1. Data processing

We implement our recommendation system based on two data files. First one from “Kaggle” has about 130.000 wines including country, description, designation, points, price, province, regions, taster name, title, variety, and winery. And second one from data world has about 25.000 wines including vintage, country, county, designation, points, province and title.

Although first one has 5 times more wines than second one, it has many missing information. Also, vintage is one of the most important part to determine a taste of wine. But first one doesn’t have. So, we think that it’s better to combine both files by copying the wine taster’s description from first one to second one when both files have same wine. Then we erase some unnecessary factors like county, designation, province and points. Here is the final form of wine data that we make.

**Table 1.** information format of wine.

|  |
| --- |
| 1. Title : title of wine  2. Variety : variety of grapes ( affect taste )  3. Vintage : the year in which the grapes were harvested ( greatly affect taste and quality )  4. Country : producing country  5. Province : producing province ( = detailed region )  6. Price : price of wine  7. Description : wine taster’s description of taste |

Considering the goal of our project, recommendation system for all newbies of wines, factors like variety, vintage or province cannot be familiar to them. So, we try to extract words about taste from descriptions to help intuitive understanding of wines for newbies. For example, wine from Washington, US named ‘hoodsport 1998 charnonnay’, made by Chardonnay grape in 1998. It’s really hard to know the taste of wine for newbies. But, if we extract words like apples, grapefruits, not much depth and residual sugar, it can helps imagine the taste. Thus, we implement tokenizing frequent words about taste from description by using NLTK(natural language toolkit).

After removing some rare and unusual words, we can extract list of flavor terms like acidity, buttery, chewy, tannin, cigar, creamy, and so on. Then we distinguish all of these words in 5 categories. Here are five categories.

**Table 2.** Categories of wine flavor.

|  |
| --- |
| 1. Acidity : an element of taste that wine gives in the mouth. High acidity makes a wine taste too sour of tart, low acidity makes the wine taste flat and flabby.  2. Tannin : an element of taste that adds both bitterness and astringency.  3. Body : a tasting term that refers to the weight of a wine, mouthfeel in terms of heavy or light. It is commonly referred to in three ways, full bodied, medium bodied and light bodied.  4. Sweetness : amount of sugar in the wine.  5. Flavor of fruit : wine taste similar with fruits. Cherry, berry, peach, lime, apple and etc. |

After extracting depends on five categories, we add 5 more words which are frequently occurred in description. Its spicy, earth, smoky, flowery, and herb. Except of flavor of fruit. We mark all wines 1 if description of that wine mention words in flavor word list. At the end of data parsing, we make one excel file for convenience.

Here is the final form of excel file with example

**Table 3.** Final format of processed dataset of wine.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Variety | Vintage | Country | Province |
| La Bete 1998 Knight's Gambit Vyd Pinot Noir (Willamette Valley) | Pinot Noir | 1998 | US | Oregon |
| Sweet | Dry | Acidity | Body | Fruity |
| 0 | 0 | 0 | 0 | 1 |
| Fruit name | Earthy | Spicy | Smoky | Flowery |
| Cherry | 1 | 1 | 0 | 0 |
| Herb | Tannin |  | | |
| 0 | 1 |

At first, we thought this file is sufficient for our recommendation system. But we start to worry about that result can ignore the popularity. In other words, we need additional data like top famous 100 wine to recommend wine list which is taste of user and famous. We can’t find complete dataset of famous wines. So, by searching lots of website and crawling, we make one more data file including 100 top famous wine list in 2019. Form of data is same with first datafile we made before.

1. Recommendation System
   1. Knowledge-based Recommendation

Knowledge based recommendation is a specific type of recommender system that is based on explicit knowledge about the item assortment, user preferences, and recommendation criteria. A major strength of knowledge-based recommender system is the non-existence of cold start problem. Because we can’t get any information of user’s taste at first, in our recommendation system we use knowledge-based recommendation when users visit our recommendation site and sign up.

Here is our process of recommendation based on knowledge. Right after users finish to make their account. We ask some questions to users. First question is “have you ever tasted wine before?”. This question is to know whether user is newbie of wine or not. It’s very important question because it decides the next question. if user answers yes, then the next question will ask three favorite wines. Depends on favorite wines of user, we find other wines that is similar with them. For example, if user’s favorite wine is full-bodied, sweet and taste like berry. We try to find other wines that satisfied that options.

If user answers no. then we ask additional five questions to users. First question asks preference in between sweet, dry, and tannin. Second question asks if user like acidity. Third one asks if user like fruit taste and favorite fruits. Fourth one asks preference between light body and full body. Fifth one askes preference of aroma like smoky and flowery. Last questions ask limit price of wine that user wants. Regardless of that user is newbie or not, we can work preferences out of users depends on questions. When we try to find recommended wine list in our complete dataset, we will select three wines in dataset which have 25000 wines. And select two wines in 100 top famous wines dataset. This it the reason for avoiding ignorant of popularity of wines that we already mentioned in data processing part.

* 1. Content-based Recommendation

Content-based recommendation uses data that user provides and based on that data, a user profile is generated. In our case, we have users’ ‘My Wines’ list, which is generate d by each user’s likes after tasting the wines and our system considers country, province, description, variety, and title of the wine. After data vectorization, we compute matrix cosine similarity and compute TF-IDF score between the description of the data. And then, we can retrieve wines with the highest score.

The score is calculated in this way.

*score\_cosine\_similarity \* 0.5 + score\_TFID \* 0.5* (1)

After calculation, we apply additional filters by price and points. To be specific, score\_cosine\_similarity is computed for each wine keywords using CountVectorizer. The keywords considered are description, country, province, and variety. Nextly, score\_TFID is used for measuring the description similarity for each wines. The descriptions contain some basic information like taste, flavor, year and other major characteristics. Thus, by using TfidVectorizer, we eliminate stopwords and vectorize the description. And then by calculating the similarity, we can compare wines by the description, too.

In website, this is implemented with each page. First, the user can enter the wine that have tasted in ‘Recommendation’ page, or by the pop-up in the first page. And then, in the ‘My Wines’ page, user can check the updated wines and can delete the wines if the wine is not the user’s taste. And finally, the user can get the recommended wine list in ‘Recommendation’ page, by selecting ‘similar wines (CB)’ button. The results will appear based on the highest score we have calculated.

1. Evaluation
   1. 80/20 Validation

Our system is mainly composed of recommendation system. Thus, the evaluation for the recommendation itself is essential. Due to the short period of implementing time and lack of users of our service, we implemented 80/20 validation by our data. By knowledge-based recommendation or content-based recommendation, user can finally get the wine list which the user likes. And then, we can split the wines by 80% and 20%. For the 80% of wines, we will calculate the similarity between the wines as we did in content-based recommendation. After calculation, we can compare the similarity of the wines and the left 20% of wines.

If the recommendation works well, we can say the user’s own taste have uniqueness and it will appear by the preferred wine’s consistency. By the 80%/20% validation, we can compare the similarity and consistency of the wines. The evaluation is carried out in this way and if the score(score\_cosine\_similarity) is high, we can say the system works well.

1. Limitation
   1. Lack of User Data

At first, we want to implement hybrid recommendation system including knowledge, content-based and collaborative filtering. But our project result only contains knowledge and content-based part. The major reason that we can’t implement collaborative filtering is that we can’t secure many information of user data.

Collaborative filtering is a method of making automatic predictions about the interests of a user by collecting preferences or taste information from many users. So, we need to have many user data, specially user’s taste of wines for comparing between users and make meaningful recommendation. But our recommender system is one of the team project of course. So, it’s impossible to get real users. That’s the reason why we can’t check the accuracy of recommendation using collaborative filtering, although we already implement it in python codes.

* 1. UX/UI

Because of the short-term of implementations and we mainly focused on the core function of recommendation system, we couldn’t considered UI/UX part so much of the website clearly and it can be connected with weak usability.

* 1. Time Optimization

After finishing development of recommendation system, we found a big problem of result. Because we use dataset contains about 25.000wines and for content-based recommendation, we have to compare cosine similarity with all of 25.000 wines to make meaningful recommendation. Thus, it takes lots of time to get result. We have to solve this problem by enhancing algorithm for recommendation or using another better server API instead of ‘herokuapp’.

1. Conclusion

As a result, we have developed wine recommendation system not only for dksl who knows well about wine but also who wants to try wines for the first time. By using knowledge-based recommendation and content-based filtering together, our website will give opportunity to get recommendation for anyone who wants to know best wine suits their taste and preference. Everything can be checked in this url: **“**[**https://bi-wine-recommender.herokuapp.com/index.html**](https://bi-wine-recommender.herokuapp.com/index.html)**”.**

References

1. Wikipedia hompage <https://en.wikipedia.org/wiki/Knowledge-based_recommender_system>
2. Wikipedia hompage <https://en.wikipedia.org/wiki/Collaborative_filtering>
3. Datasets for wine review Kaggle <https://www.kaggle.com/zynicide/wine-reviews>
4. Datasets for wine <https://data.world/login?next=%2Fmarkpowell%2Fglobal-winepoints%2Fworkspace%2Ffile%3Ffilename%3DWines.xlsx>