Interim Project Report

Wine Recommendation System

Team 1

Team Members

Wonhee Lee

- M.S in Analytics
- B.S in Finance/Int'l Business at NYU
- From South Korea
- wlee364@gatech.edu

Nan Bu

- M.S in Analytics
- B.S in Business at the Univ. of Oregon
- From China
- nbu3@gatech.edu

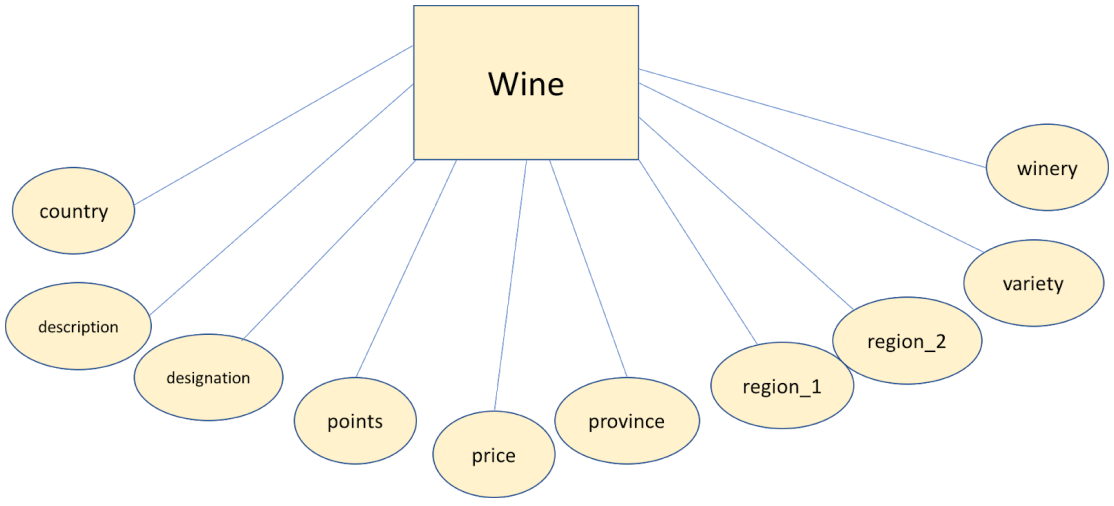
- **Goals / Problem Statement**
  Wine is commonly beloved around the world, yet most wine-lovers are not sommeliers. While people may know a couple brands or products that they drink often, many do not know how to expand their selection due to the lack of professional knowledge. Our database/application intends to help people find new wine products that will suit their taste.

- **Approach**
  In this project, we will attempt to build a wine recommendation system primarily based on wine reviews. The intuition is to use "keywords" that are typically used to describe wine flavors such as "dry", "fruity" or "cassis" to filter the reviews and assign a score to different types of wine based on the user's inputs.

- **Design of Database**
  Below is a preview of the original wine reviews dataset:

country	description	designation	points	price	province	region_1	region_2	variety	winery
US	...	Martha's Vineyard	96	235.0	California	Napa Valley	Napa	Cabernet Sauvignon	Heitz
...									
...									
...				...					...

The ER Diagram would look like the following, with a single entity(Wine) and 10 attributes:



In our analysis, we may elect to add additional columns based on certain keywords that are commonly used to describe wine flavors. For instance, the column "dry" may be a boolean indicating whether the description includes the word "dry". However, depending on the feasibility of making the user application interactive, these columns

may or may not appear on the dataset; they may simply be a concealed part of the interactive tool that is created in the process of running the algorithm.

- **Data**
  The Wine Reviews dataset on Kaggle consists of 280,000 rows of wine products and their features, including their descriptions(or reviews). The columns that we will use from the dataset are country, description, designation, points, price, province, region_1, region_2, variety, and winery. The source URL is as follows: https://www.kaggle.com/zynicide/wine-reviews/data

- **Interface details**
  Our goal is for our application to support the user's selection of the features mentioned above, as well as a number of keywords that the user enters to describe his/her preferences. Then, the application will return the top 5 wine products that our algorithm selects based on the inputs. We will design a REST Web API for wine recommendation applications using the Flask microframework. Users will perform simply CRUD (Create, Read, Update and Destroy) operations as interaction in POSTMAN. For example, users can update their preferences for wine flavors and/or attributes like wine age, winery location and ect. in a JSON formatted template, see an example as below:

  ```
  {
  "datatype": "text",
  "name": "use_input",
  "data": {
  "preference_1": "ripe",
  "preference_2": "fruity",
  "preference_3": "strong",
  "location": "California",
  }
  }
  ```

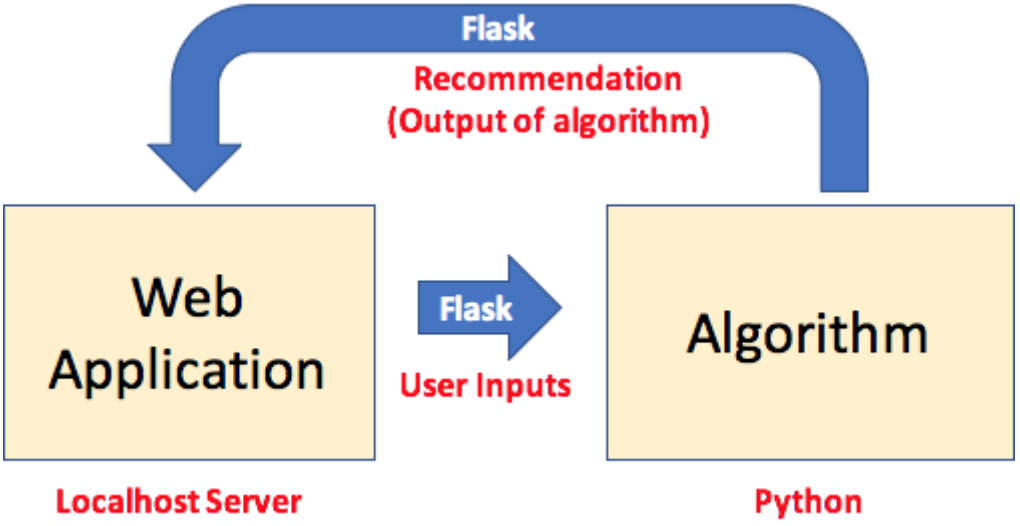
  Users will then send requests to see the recommended wine products displaying in the localhost. We also hope to design our localhost webpage to include some visualizations that provide a general overview of the dataset, to enhance the user's experience.

- **Architectural Overview**
  As of now, our plan is to perform data cleaning/joining/analysis operations on Python using the pandas library, given the size of the dataset and the abundance of machine learning packages. Then, we will aim to employ Flask to translate and depict our discoveries on a localhost server. The main purpose of this divide is to preserve as

much analytical capability as possible while ensuring that the application is interactive. The flow of operations will roughly be as follows:

- User indicates his/her preferences and inputs/update "keywords" describing qualities
- The inputs are sent to the Python environment where the algorithm is run
- The results from the algorithm are returned to the localhost server and displayed in the application

Below is a flow diagram of the aforementioned procedures:



- **Technologies**
  - We will use Flask to build our Web API.
  - We will primarily be using the Python pandas framework and scikit library for data cleaning/analysis.
  - We will use csv files and Python at our backend to handle localhost connection, running recommendation system algorithms.
  - We will design the localhost webpage using HTML to visualize the results.
  - We will use POSTMAN for user interaction.

- **Algorithmic Overview**

  We intend to test a variety of machine learning algorithms including but not limited to regression, classification, decision trees, PCA, etc and incorporate some original features as we see fit. Our big picture is to develop an algorithm that essentially "scores" each wine product based on the keywords that the user selects. As the identification and weighting of these keywords will play an integral role in the success of the recommendation system, we may seek to refer to published algorithms or tools that specialize in this matter. We will consider referring to the method from "Contextual Recommendation based on Text Mining" article. The ideas and step by step approach are listed follow:

- Text mining: we will classify the context from “review” column and/or other attributes into different types.
- Boolean Model: we will use this model to find out the products that match the context.
- Probabilistic Latent Relational Model and EM Parameter Estimation model: we will use these two algorithms incorporating weights assignments to make prediction/recommendation.

- **Demo plan**
  We hope to provide a demonstration of our application and have some users make their selections. We will explain the reasoning and process behind the output.

- **References**
  We referred “Contextual Recommendation based on Text Mining” as to how to build a text-based recommendation model, the link of the article is listed at the bottom. However, we will explore more robust algorithms and doing more research on this topics to aid our understanding and enhance our project, we will list them on our final report.

  References:
  “Contextual Recommendation based on Text Mining”
  http://www.aclweb.org/anthology/C10-2079