CS 5010

Semester Project

Nikki Aaron
Beverly Dobrenz
Amanda West
Joseph Wysocki

CS 5010 Semester Project

Introduction

Our team used this project to explore a large dataset of wine reviews from accomplished and well-respected wine tasters. Starting out, our main objective was to build an interactive wine recommendation algorithm that would pair users with selected wines based on multiple criteria. In addition to this algorithm, we wanted to produce a more user-friendly interface – one that would be built without using text-based terminal inputs. To this end, we aimed to create a Tableau dashboard that would allow users to visually interact with and explore our data set. And finally, to add one more layer to our project, we wanted to analyze each reviewer within our data set. We wanted to produce a set of key metrics for each reviewer that could be used to compare them amongst each other.

The Data

In our search for quality wine data, our team first looked through a few options in the UCI Machine Learning Repository. These sets were extensive, but the columns contained were a lot more technical and scientific than they were descriptive, and we decided that an analysis of this data would not be interesting nor helpful to a lay person.

We shifted to Kaggle, and found a dataset that contained approximately 130,000 unique wine reviews from the WineEnthusiast website (Kaggle, 2020). This set was much more appropriate for the type of analysis that we aimed to complete. The data is stored as a csv file, and contains 10 columns by 130,000 rows, and a quick description of each column is shown below.

- Country - the country that the wine is from
- Description - Taster Name's description of the wine
- Designation - the vineyard within the winery where the grapes that made the wine are from
- Points - the number of points WineEnthusiast rated the wine on a scale of 1-100
- Price - the price in USD for a bottle of the wine
- Province - the province or state that the wine is from
- Region_1 - the wine growing area in a province or state
- Region_2 - sometimes there are more specific regions specified within a wine growing area
- Taster_Name - name of the taster that gave the points and description
- Taster_Twitter_Handle - twitter handle of the taster
- Title - name of wine, includes variety
- Variety - the type of grapes used to make the wine
- Winery - name of the winery where the grapes came from

A quick analysis of the raw data revealed missing values that needed to be cleaned up. All rows with missing values for country, points, price, variety, description, or taster name were removed from the data -- leaving us with 96,400 useful rows.

Pre-Processing - Wine Reviewer Profiles

For the individual reviewer analysis, we outlined several key metrics that we could use to analyze and compare individuals within the data set. The metrics we chose to analyze are listed in the table below:

Metric	Description
Number of Reviews	The total reviews completed by the reviewer
Average Score	The average score given to wines by the reviewer
Highest Score	The highest score given by the reviewer
Lowest Score	The lowest score given by the reviewer
Perfect Score Rate	The ratio of perfect scores to total reviews
Diversity	Captures how many countries a reviewer has sampled wine from. A perfect score of 100% means a reviewer has sampled wine from every wine-producing country
Word Usage	A tiered system that categorizes each reviewer by the average length of their reviews. Tiers include Curt, Average, and Wordy
Status	A tiered system that ranks reviewers by the number of reviews they've completed. Tiers included Novice, Experienced, and Power Reviewer
Scoring Style	A tiered system that ranks reviewers by their average score given in reviews. Tiers included Harsh, Average, and Generous

For this analysis, we needed to add an additional column to our data frame – 'Word Count'. This column counts the amount of words used for each review and is used in calculating the "Word Usage" metric.

We also needed to gather additional data to calculate our Diversity metric. We used the Pandas read_html method to web scrape a table of all wine producing countries in the world (World Population Review, 2020). The length of this list was used as the denominator in the calculation of our Diversity metric.

Experimental Design

Feature Engineering - Wine Type

Despite all the great data we had, there was an important categorization missing from this data -- wine type. People usually have an overall or occasional preference for red versus white wine, and we had nothing to help the recommendations filter for this preference. Some quick research revealed that a good starting point for inferring the wine type would be the grape variety. Any type of wine can be made from red or white grapes simply by processing them differently, but there are some grapes that most often go with a certain wine type, and red wines are usually made from red grapes (Wikipedia, 2020). A very comprehensive list of grape varieties, possible aliases, and their color distinctions was found on Wikipedia (Wikipedia, 2020). This data was parsed by looping through rows and splitting on semicolon and slash delimiters to create a two-column table of grape names and colors.

A first pass at assigning wine color based solely on grape variety revealed a moderately high level of inaccuracy. Many varieties in our data set - like "Red Blend", "Rosé" - did not list any particular grape. There was miscategorization of white wines made from red grapes and vice versa. We also found that some grapes - like "pinot gris" - were contained in both the red and white color lists from Wikipedia, so the script did not know which color to label those wines. More refinement was needed. Some example grape types are as follows:

- Red-wine only grapes → [cabernet, malbec]
- White-wine only grapes → [sauvignon blanc, chardonnay, riesling]
- Ambiguous grapes → [pinot noir (blanc), chardonnay (noir), pinot gris (mendoza)]

Since some wines already had color information in their title or description, we decided to first look for color words in these columns before matching on grape variety. This led to the addition of the "rosé" type that we could not identify by grape variety. A glimpse through the data after this categorization made it apparent that wines from different countries were often labeled with color words in their respective languages -- mainly French, Italian, and German. As such, for each color, a list of possible color words was created. For example, "white" could also be "blanc", "bianco", "bianca", "weißwein", or "weis". Additionally, designators for "Sparkling" and "Blend" were added using a similar process. Our final word list is shown below:

- Red → [red, noir, rotwein, rosso, rouge]
- White → [white, blanc, bianco, bianca, weißwein, weis]
- Rosé → [rosé , rosato, rosado, rosat, roséwein, roséfine]
- Sparkling → [sparkling, champagne, bubbles, bubbly, brut, bruto, sekt, schaumwein, effervescent, spumante, scintillante]
- Blend→ [blend]

Parsing data into these categories was now approaching a level of complexity where we could start to call it an algorithm. Additional refinement was made through logical trial and error, and the final result uses weighted matches. Color matches on the wine title are more reliable, and thus are weighted more heavily than those for the wine description. Words for rosé and words in other languages are usually more reliable indicators of wine type than just "red" or "white" so these are also weighted more heavily. At the completion of our search match loop, the matches are summed, and the wine type with the maximum match count is chosen for each row.

Wine Reviewer Profiles

The experimental design for our wine reviewer profiles was quite straightforward. We took the metrics defined above, created functions to calculate each metric for each reviewer, and then stored those metrics in a nested dictionary. Given this setup, our code is quite nimble as it can easily handle any additions to our original data set. Furthermore, the nested dictionary makes it easy to display all necessary information for any reviewer as needed.

We then converted our nested dictionary into a data frame. This new data frame has a row for each reviewer, and columns corresponding to the different metrics we decided to calculate.

Natural Language Processing - Flavor Profiles

To allow users to select wines based on flavors, we also needed to query based on a wine's distinct flavor profile. To do this, we performed natural language processing (NLP) and text mining techniques on the descriptions for each wine, which were written by a professional wine taster. All 96,400 entries held wine taster descriptions and were parsed for their flavor profile.

To accomplish this, we employed the Python NLTK library (in addition to Pandas, Random, and NumPy) to tokenize and create new knowledge from each wine review (Subramanian, 2020). Using a loop to iterate through one review at a time, we first tokenized these paragraphs into lists. Once

each entry contained a list of words instead of a block of text, we next filtered out stopwords, i.e. connector words such as “as”, “the”, or “and” that held little value to our analysis.

This left us with mainly verbs, adjectives, adverbs, and nouns. For wine descriptors, nouns and adjectives were the most useful (“apple” is a noun; “fruity”, an adjective) at detecting wine flavor, body, and sweetness. To filter out choice words, we employed the pos module in the NLTK library to parse through each word and identify and append that word’s grammar type (Guru99, 2020). Once this was complete, we filtered out words that weren’t identified as nouns or adjectives.

Finally, we created a new “category” column with lists of words that describe each flavor type. For example, “onion” and “rhubarb” are likely to describe a savory-flavored wine, so a beverage with these descriptors would have “savory” appended to the category column. On the other hand, “rosy” or “lavender” would suggest a floral wine. We also parsed to define wine as full-bodied (“rich”, “bold”) or light-bodied (“summery”, “fresh”). Finally, we identified wines as sweet (“sugary”, “sweet”) or dry (“dry”, “brut”). This way a wine could be sweet, light-bodied and floral, or savory, full-bodied and dry all at once, but would not be categorized as both light and full-bodied.

The three categories and their corresponding subsections are listed in the table below.

Flavor Profile Types		
Sweetness	**Flavor**	**Body**
Sweet	Savory	Light-Bodied
Dry	Fruity	Full-Bodied
	Earthy	
	Bitter	
	Floral	

An example result is shown below with its Wine Type and Flavor Profile word matches highlighted.

Cavas Hill NV 1887 Rosado Sparkling

(Sparkling Rosé Blend from Spain)

$13 | 82 Points | Sweet, Light-Bodied

Red in color, with berry and apple aromas, this is a sweet blend, with a light body and nose tingling effervescence.

Beyond the Original Specifications

Wine Reviewer Profiles

To complete the wine reviewer profiles, we utilized the get_html function from Pandas to scrape data from the web. We scraped the data, stored it in a data frame, and queried it. The information we gathered from this new data allowed us to calculate the “Diversity” metric for each reviewer.

Additionally, the Jupyter notebook for the wine reviewer profiles includes a section for user interaction. We created a function (get_info) that accepts one argument (reviewer name) and returns the metric values for that reviewer. Finally, the script prints a list of every reviewer, asks the user which one they would like to know more about, takes that input, and returns the specified information.

Tableau Data Visualization

The original specifications of the project started and ended with Python. However, to maximize the utility of the finalized data for the wine recommender, we decided to take our visualizations and data interactivity outside of the Python visualization world. We accomplished this by creating a Tableau dashboard.

Unit Testing

Most of the tagging code involved functions that use regular expressions to check for delimiters, whole words, or whole phrase matches in large strings. To ensure that our grape variety color matching and wine type word matching functions were performing as expected, two unit test classes were created. Proper test-driven design would have these files created before the final product, but in this case they were created afterwards for validation. The first class focused on validating the parsing of grape variety lists by delimiter. A sample function and test is shown below.

```
# Function to split a comma, forward slash, or 'and' separated string to a list
def splitNames(namesStr):
    nameLst = re.split(r",|\/| and ", namesStr, flags=re.IGNORECASE)
    nameLst = [name.strip() for name in nameLst]
    return nameLst
```

```
def test_is_splitNames_by_forwardslash_working(self):
    names = 'Douce noir/Charbono/Bonarda/Turca'
    nameLst = splitNames(names)
    self.assertEqual(nameLst, ['Douce noir', 'Charbono', 'Bonarda', 'Turca'])
```

The next class focused on validating regex word matching. The following code tests that a match is returned True because the search list contains ‘sauvignon blanc’ and that term is contained in the target phrase ‘sauvignon blanc blend’.

```
# Function to see if any searchTerm from list equals any whole word in the searchTarget string
def partialMatchPhrase(searchTarget, searchTerms):
    for term in searchTerms:
        matches = re.findall(r'\b' + re.escape(term) + r'\b', searchTarget)
        if len(matches) > 0:
            return True
    return False
```

```
def test_is_partialMatchPhrase_true_working(self):
    self.assertTrue(partialMatchPhrase('sauvignon blanc blend',
    | ['pinot grigio', 'sauvignon blanc', 'chardonnay']))
```

All tests ran successfully.

```
.........
-------------------------
Ran 9 tests in 0.003s

OK
```

Results

Wine Recommender Application

With our data sufficiently cleaned and enhanced, we were ready to create an interactive terminal app. This app is designed to take in user responses to various filtering criteria via “input()”. There are 5 filter types, and a while loop ensures that the user has the opportunity to add additional filters until either they have chosen criteria for every option, the results list becomes less than 10, or they enter “0” to display results based on the current filters. Their choices are then used to subset the full wine list using Pandas data frame boolean filtering. The final list of results is displayed in a formatted list and output to the user in the terminal. The following images show the recommender app in use.

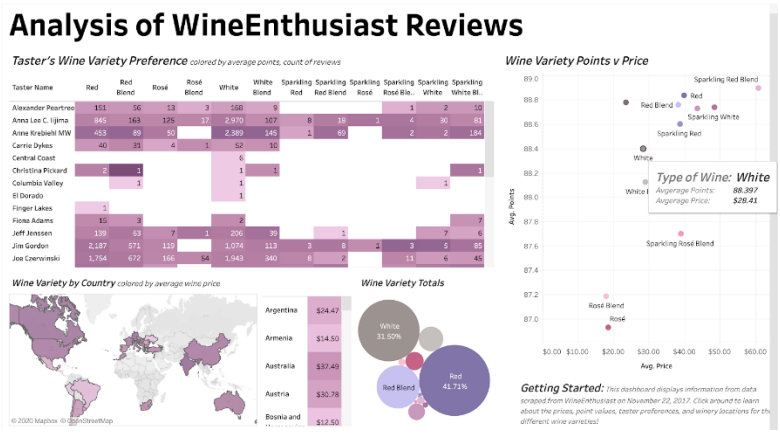
```
(nltk) nadev@na5zn:/mnt/c/Users/Nikki/Sites/MSDS-Coursework/CS 5010/Project 01$ python filters.py

So you are interested in selecting a new wine to try? You can apply various filters, then select 0 to see your recommendations.
What should I call you during this process?
Nikki

What type of filter would you like to apply? Enter the number for one of these options:
0: None, just show my recommendations
1: Wine Type
2: Flavor Profile
3: Taster Rating
4: Country of Origin
5: Price Range
1

Alright Nikki, we have a several types of wine to choose from. Enter the number for one of these options:
0: None, just show my recommendations
1: Red (43493 wines)
2: White (28915 wines)
3: Red Blend (13781 wines)
4: White Blend (2861 wines)
5: Sparkling Rosé Blend (682 wines)
6: Rosé (3717 wines)
7: Sparkling White Blend (2069 wines)
8: Sparkling White (666 wines)
9: Sparkling Rosé (144 wines)
10: Rosé Blend (13 wines)
11: Sparkling Red (76 wines)
12: Sparkling Red Blend (3 wines)
8

What type of filter would you like to apply? Enter the number for one of these options:
0: None, just show my recommendations
2: Flavor Profile
3: Taster Rating
4: Country of Origin
5: Price Range
5

What price range would you like to limit this search to? Enter the number for one of these options:
0: None, just show my recommendations
1: Everyday $1-$25  (258 wines)
2: Occasional $26-$75  (307 wines)
3: Premium $76-$100  (63 wines)
4: Luxury $101-$200  (30 wines)
5: Iconic $201-$3300  (8 wines)
```

```
Nikki's Top 5 Wine Recommendations
----------------------------------------
----------------------------------------

Schramsberg 2004 Brut Late Disgorged Blanc de Noir Pinot Noir
----------------------------------------
Sparkling White (US)
$95.0  |  94 points  |  Dry, Fruity, Savory, Earthy
Gorgeous aromas of toasted bread and caramel give way to a rich, buttery texture and flavors that layer ripe apple, honey, toasted almond
and walnut notes on a smooth, softly bubbling background. It's showing the benefits of extra aging through the slightly nutty character, f
ine bubbles and seemingly sweet feel even though it's elegant and dry.

Delamotte 1999 Blanc de Blancs Brut Chardonnay
----------------------------------------
Sparkling White (France)
$95.0  |  90 points  |  Dry, Savory
An already mature, toasty Champagne, with yeast, dry toast and almond flavors. The grapefruit edge from primary fruits is there, but this
wine is filling out, becoming deliciously open and certainly entering maturity. Enjoy as a food Champagne.

Delamotte NV Blanc de Blancs Brut Chardonnay
----------------------------------------
Sparkling White (France)
$79.0  |  93 points  |  Dry, Fruity, Bitter
Produced exclusively from Grand Cru vineyards in the Côte des Blancs, this is a beautiful, dry wine. It has a chalky texture and ample aci
dity that frames the tight fruit. The pure line of citrus and crisp green apples goes right to the heart of this balanced, bright wine tha
t will age well. Drink now for the forward fruit, or wait for another two to three years.

Louis Roederer 2010 Blanc de Blancs Brut Chardonnay
----------------------------------------
Sparkling White (France)
$86.0  |  94 points  |  Dry, Full-Bodied
This latest Blanc de Blancs vintage from Roederer is well poised and balanced, already offering a hint of toast, while the minerality is p
rominent. It is in a dry style, relying on the ripe fruit to soften the impact of the acidity. A beautiful wine, ready to drink.

Lombard et Cie NV Brut Nature Grand Cru Blanc de Blancs Chardonnay
----------------------------------------
Sparkling White (France)
$80.0  |  93 points  |  Dry
A blend of wine from four Côte des Blancs villages, this is crisp, very dry in the Lombard style while also smooth from the short wood agi
ng. Mineral and tangy, it is a bottling (disgorgement January 2016) that could age a bit longer. Drink from 2017.
```

Tableau Dashboard

The Tableau dashboard is a beginner friendly way to dive deeper into the wine data without the possible intimidation of a text-based terminal input. In addition, the flexibility of the Tableau dashboard allows users to ask their own questions of the data - questions that they might have gleaned from the output of the wine recommender. For example, if a user is interested in the wines from a specific country, they can select the country from the map widget and view the tasters, wine types, price distributions, and points distributions of the wines from the chosen country. Or, if a user approaches the dashboard with a wine type in mind, the user can use the same dashboard to filter the data to only show information about the specified wine type.

Creating the Tableau dashboard ensures that the work we did ingesting, cleaning, and manipulating the original wine dataset puts additional analytical power into the hands of the user to perform their own exploratory analysis.



Wine Reviewer Profiles

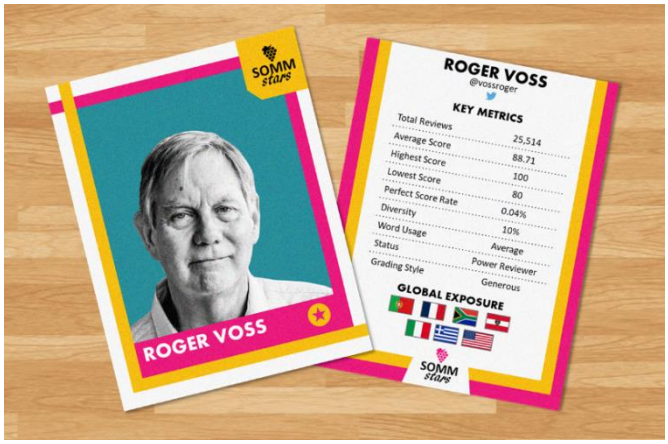
A screenshot of the output from our wine reviewer profiles is shown below. This output is a clean way to extract lots of important information for each reviewer within our data set. This information is relevant to anyone who wants more context behind the people who provided the reviews.

```
1 get_info('Lauren Buzzeo')
```

```
Twitter handle:  @laurbuzz
Number of Reviews:  1711
Average Score:  87.57
Highest Score:  95
Lowest Score:  81
Perfect Score Percent:  0.0
Diversity Percent:  10.0
Word Usage:  Wordy
Status:  Experienced
Scoring Style:  Generous
Review Count by Country:  {'South Africa': 905, 'Israel': 198, 'France': 586,
'US': 19, 'Canada': 1, 'Portugal': 1, 'Spain': 1}
Average Price of Wine Reviewed:  24.49
```

We learned quite a lot about our reviewers through this analysis. Some individuals – such as Roger Voss – are *extremely* prolific when it comes to reviewing wines. He alone contributed over 25,000(!) reviews to our data set. Additionally, we learned a lot about the geography of wine production. It was surprising to see how many different countries grow wine.

For added fun, we created a fake brand of trading cards – known as "Somm Stars" – and featured each reviewer. The cards organize the information gathered through our work in Python and present it in a friendly format. An example card for Roger Voss is shown below.



Conclusion

The entirety of our project was created with others in mind. As such, our work is set up perfectly for others to use. The wine recommender system can supply any user with a set of wines for them to try, the wine reviewer analysis can help individuals learn more about each reviewer, and the Tableau dashboard is perfect for those who are less technically-inclined.

Looking forward, there are many ways that we could improve this product. First, while our data set is large, it is still quite dated. The information comes from 2017. Thus, in future enhancements, we would like to scrape more review data and add it to our data set. Additionally, the data set consists

of only 19 reviewers. We think the system would benefit from a more diverse set of individuals and, as such, would like to add more data from new reviewers.

Finally, we would like to move from a Python-based interaction to a web-based interaction. While functional, our current setup is limited to those who are comfortable operating within a Python terminal. If we could make this entire process web-based, we would greatly expand our potential user base.

References

Guru99. (2020, August 5). *POS (Part-Of-Speech) Tagging & Chunking with NLTK*. Retrieved from Guru99: https://www.guru99.com/pos-tagging-chunking-nltk.html

Kaggle. (2020, August 4). *Wine Reviews*. Retrieved from Kaggle: https://www.kaggle.com/zynicide/wine-reviewsss

Subramanian, D. (2020, August 5). *Text Mining in Python: Steps and Examples*. Retrieved from Medium: https://medium.com/towards-artificial-intelligence/text-mining-in-python-steps-and-examples-78b3f8fd913b

UCI Center for Machine Learning and Intelligent Systems. (2020, August 4). *Wine Data Set.* Retrieved from Center for Machine Learning and Intelligent Systems: http://archive.ics.uci.edu/ml/datasets/Wine/

Wikipedia. (2020, August 4). *List of Grape Varieties*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/List_of_grape_varieties

Wikipedia. (2020, August 4). *Wine Color*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Wine_color

World Population Review. (2020, August 4). *Wine Producing Countries 2020*. Retrieved from World Population Review: https://worldpopulationreview.com/country-rankings/wine-producing-countries