IST 736 Text Mining

Final Project Report

June 5, 2019

Steve Alessandrini

Alex Balter

Carlo Mencarelli

Jonah Witt

# Scotch Whisky Reviews

## Introduction

Scotch whiskey production began as early as the 15th century in Scotland. The earliest record of distilling is found in in the late 1400s. There are two main types of scotch whiskey: single malt and single grain. From those two categories, three subcategories exist: blended scotch, blended malt scotch and blended grain scotch.

The popularity of scotch surged in the 1800s when a new type of production process yielded the ability to create a smoother whiskey at a cheaper price. Furthermore, in 1880, the Phylloxera beetle destroyed many vineyards in France affecting wine and cognac production allowing Scotch to take the its place amongst those who enjoy both the pleasant and unpleasant effects of alcohol (Puckette, 2016).

Since then, Scotch has been a popular “upscale” drink across the globe. Bottle prices can vary from $30 to thousands of dollars depending on a variety of factors including age, quality and scarcity. But how does one know which bottle to buy? When looking for a bottle in a given price range, there are countless choices. Scotch aficionados tend to look to experts in the field and the reviews and tasting notes they provide. Each review can provide a unique vocabulary set used to help define a particular bottle of scotch.

## Analysis

### About the Data

The dataset used for this analysis comes from Kaggle and consists of 2,247 scotch reviews. These reviews were originally scraped from a scotch enthusiast website: WhiskeyAdvocate. Each row represents a single bottle of whiskey with the following the attributes:

* (unnamed) Index Column
* Bottle Name
* Category
* Review Score
* Price
* Currency
* Description (text review)

The category column has five distinct classes:

|  |  |
| --- | --- |
| Category | Review Count |
| Single Malt Scotch | 1819 |
| Blended Scotch Whisky | 211 |
| Blended Malt Scotch Whisky | 132 |
| Single Gain Whisky | 57 |
| Gain Scotch Whisky | 28 |

Table Count of scotch whiskey categories found in the dataset.

The Review Score attribute has a normal distribution with a mean review score of **86.7**.

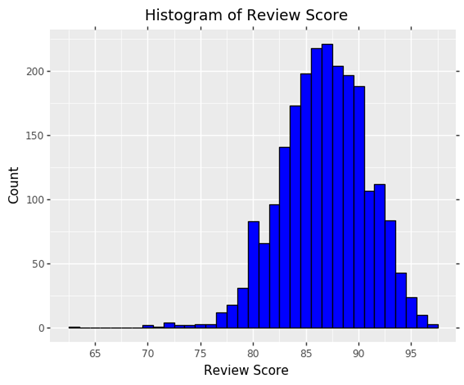


Figure Distribution of 2,247 review scores

The whiskey reviews were usually very descriptive in their language use as seen here in the review for a bottle of Johnny Walker Blue Label, blended scotch whiskey:

“Magnificently powerful and intense. Caramels, dried peats, elegant cigar smoke, seeds scraped from vanilla beans, brand new pencils, peppercorn, coriander seeds, and star anise make for a deeply satisfying nosing experience. Silky caramels, bountiful fruits of ripe peach, stewed apple, orange pith, and pervasive smoke with elements of burnt tobacco. An abiding finish of smoke, dry spices, and banoffee pie sweetness. Close to perfection. Editor's Choice”

### **Cleaning**

#### Price

The price column contained numerous textual features which prevented full numeric conversion.

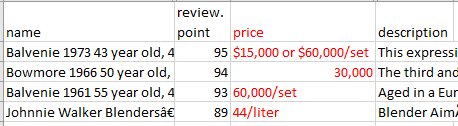


Figure Text features which require cleaning within the price data

Features such as dollar signs, commas and quantity definitions (“/set”) were removed via regular expressions. Cases in which the price was not referencing a single bottle were recalculated to normalize the data. The attribute was then converted to numeric in order to allow for further analyses.

#### Description (whiskey review)

The reviews required extensive cleaning in order to be formatting properly for analysis. Certain reviews contained numbers and unusual characters. These features were removed via regular expressions. In this same step, punctuation was removed, and all alphabetical characters were converted to lowercase. This prevents capitalization from hindering word grouping during vectorization.

After only alphabetical characters remained, stop words were removed. The stop word list chosen was from the default NLTK stop word list:

mustn, own, but, with, until, you'd, up, were, d, you, each, hadn, their, so, she, on, doesn, ve, off, isn't, an, while, just, her, couldn't, she's, that, you're, than, is, did, other, after, o, aren't, below, his, shan't, only, shan, i, to, haven't, being, been, are, didn't, ma, won't, above, wouldn't, out, shouldn, such, because, there, it, theirs, then, yourselves, hasn't, needn, more, no, or, whom, y, wouldn, you've, under, doing, when, t, wasn't, hasn, does, your, they, hers, can, these, very, be, my, down, ours, that'll, won, didn, once, against, through, between, mightn't, weren, hadn't, we, here, nor, shouldn't, yourself, himself, before, those, all, themselves, most, further, its, this, our, should, from, s, if, them, over, during, it's, at, the, myself, how, where, ll, you'll, weren't, couldn, mustn't, herself, was, too, needn't, some, why, do, has, about, what, few, again, for, he, had, in, re, yours, wasn, should've, am, ourselves, by, doesn't, into, of, same, mightn, as, me, don't, m, having, will, aren, which, isn, any, a, who, him, itself, now, have, ain, haven, not, don, both, and

Finally, in order to provide better word grouping during vectorization, the words were stemmed where possible using both the Porter stemmer and the Snowball stemmer from the NLTK package. Both were performed exclusively keeping the two corpora of stemmed text separate as a precaution.

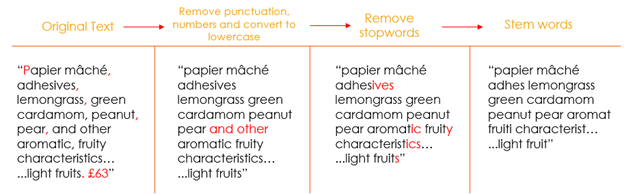


Figure 3 Transformation stages of review cleaning and processing

The distribution of word counts within each review were examined prior to cleaning and after cleaning.

**Original Word Counts Cleaned Word Counts**

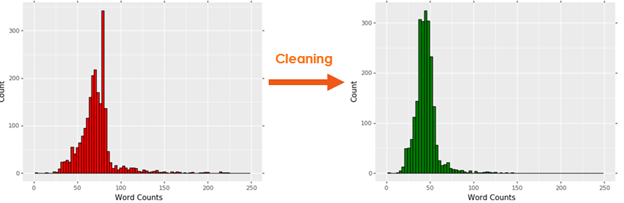


Figure Distribution of word counts in review text before and after cleaning and processing

#### Additional Columns

While the information already contained within the dataset was likely enough to glean valuable insight, additional attributes were created both for the purposes of modeling as well as deeper understanding into the dataset. Not all columns were used in latter stages of analysis, but the additional attributes provided available options for future use.

#### Year

Approximately half of the scotches reviewed by the whiskey advocate website contained an age statement. This age statement represents the number of years the whiskey was aged within an oak barrel. It’s generally accepted that longer aging is better with respect to enhancing the overall flavor and smoothness of whiskey. Examples of names which contained an age statement are:

* Black Bowmore, 1964 vintage, 42 year old, 40.5%
* Bowmore 46 year old (distilled 1964), 42.9%
* The Dalmore, 50 year old, 52.8%

This allowed for very easy extraction using the regular expression: **'(\d+) years? old'**, where the parenthesis allowed for specific extraction of the numeric attribute within the text, which in this case represented the number of years aged. 53.5% of reviews were associated with an age statement following this extraction.

#### ABV

ABV (alcohol by volume) refers the percentage of alcohol within a given whiskey. This information was readily available within the “name” attribute and similar to age statement, was easily extracted with regular expressions. Examples of names containing an ABV are:

* Johnnie Walker Blue Label, 40%
* Balvenie 1973 43 year old, 46.6%
* Ardbeg, 1974 Vintage, Cask #3145, 49.9%

The regular expression: **'(\d+\.?\d+?)%'** was used in this extraction. The examples above provide cases where a decimal place followed by a final digit was not necessarily included within the ABV when the value was a whole number, and therefore optional parameters were included within the expression. Following the extraction of the numeric value, 99.3% of reviews were associated successfully with an ABV.

#### Vintage and Special Edition

Certain names also contained the words “vintage” or “edition” signaling that there was something “special” about the scotch compared to the manufacturer’s normal production. A Boolean attribute of True or False was created for Vintage, Edition, and Vintage or Edition. Once again, regular expressions allowed for easy extraction of this attribute: **'[Vv]intage|[Ee]dition'**. Approximately 19% of reviews were labeled as “True” for Vintage or Edition.

#### Brand

Brand would prove to be the most challenging at extracting from the Name attribute.

|  |  |
| --- | --- |
| Source Name | Desired Label |
| Johnnie Walker Blue Label, 40% | Johnnie Walker |
| The Glenlivet Cellar Collection, 1969 vintage, 50.8% | Glenlivet |
| Bruichladdich *Legacy* 6, 34 year old, 41% | Bruichladdich |
| Highland Park 18 Year Old, 43% | Highland Park |
| Highland Park, 32 year old, 1973 Vintage, Cask #8375, 41.3% | Highland Park |

Table Source name value and the desired label to be used for modelling

Constructing a regular expression to perform this task would be challenging. Many subtleties regarding the structure of the text would make extraction very complicated. The most successful regular expression for extracting brand name would require having the list of brands readily available, but the list of brands could not be gathered from the dataset. The problem was circular.

A different approach was taken. Fortuitously, the whiskey advocate website had four pages with different links to many of the brands reviewed on the website:

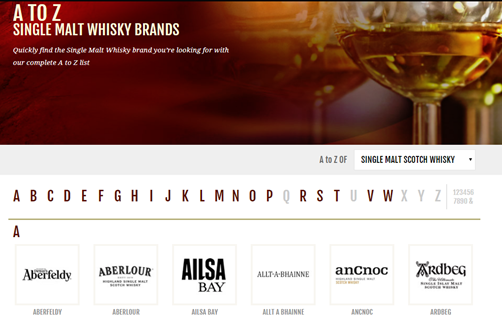


Figure Single Malt whiskey brand listing from WhiskeyAdvocate

A function was created to extract the html from all four brand listing websites (segregated by Single Malt, Blended, Blended Malt and Grain categories). This raw html was then examined the html tags which enclosed each brand name were determined. The html was converted to a string and a regular expression was used to extract the name contained within the html tags.

With the names extracted, a very large regular expression was created by converting all names to uppercase and separating them with ‘|’ which is the ‘or’ symbol within regular expressions. The final production can be represented as **‘BRAND1|BRAND2|BRAND3|…”**. This regular expression was then used to extract brand from the name column. Following this technique, 91.5% of reviews were now associated with a brand. It was determined that cases which failed to extract a brand were the result of the brand either not being listed on the website, or not matching exactly between the Name attribute and the website.

#### Review Class

Much of the analysis would center around the provided numeric review score by the whiskey reviewer. This continuous variable was “discretized” in order to allow for classification techniques. The distribution of the numeric review score was fairly normal and the 25th and 75th percentiles were 84 and 90. These values were used to split the values into three groups:

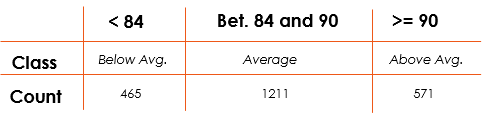
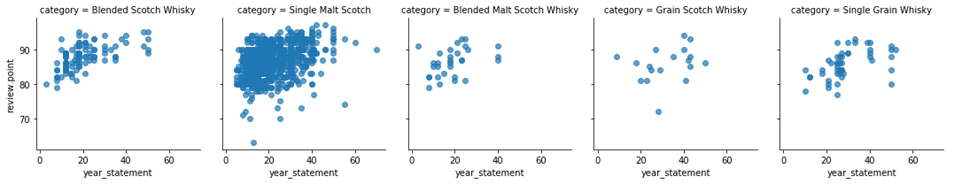


Figure Table displaying Review Class with the chosen splitting values, associated labels and resulting counts of each class.

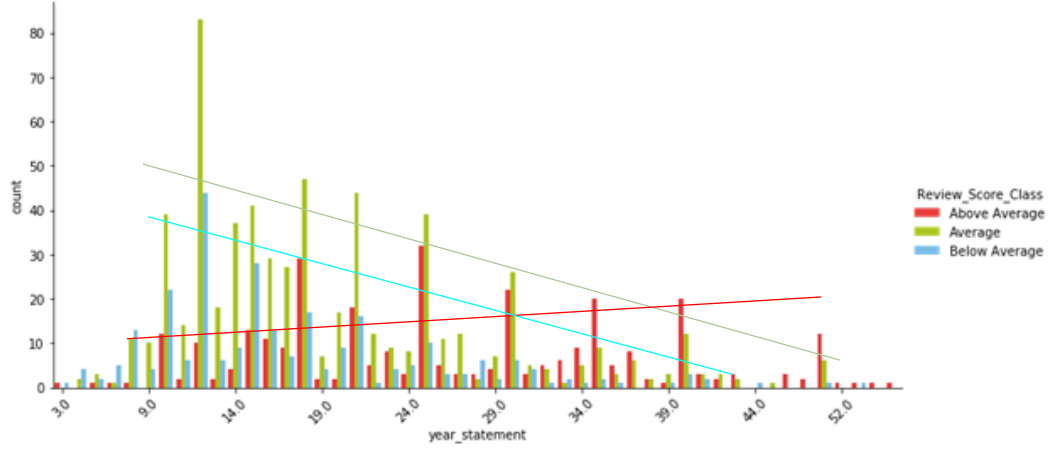
#### Age of Scotch vs. Review Points

The influence of age of scotch on the review points is significant. As is shown in Figure 7 below, there is a positive correlation between the age of scotch and the review points. The correlation holds true for each category.



*Figure 7 - Age of Scotch vs. Review Points graph*

The summarized view of this relationship further emphasizes this correlation as we see not only the positive trend of the review score class with age of scotch, but also the decline of counts for the Average and Below Average score ranges.



*Figure 8 - Age of Scotch vs. Review Score Class graph*

## Models

Multiple models were created and tuned in an effort to have the widest amount of data about which models work well and which do not and where future efforts could be focused for improvement. In the following sections each model was trained and tested on a 90/10 split meaning 90% of the data was used for training and 10% for testing. Unless otherwise specified the parameters for each model were left at the default values, that’s not to say that different options were not tried, it only means that the default options were found to give the best results.

Sentiment was calculated using multiple methods for the review text, both sentiment and price were attempted to be predicted, and topics were attempted to be discovered using several methods.

### Sentiment Analysis

Sentiment analysis can be performed by implementing one of the two different approaches using machine learning — unsupervised or supervised. As it is known sentiments can be either positive or negative. Machine learning algorithms can be used to evaluate if a series of words reflect a positive or negative sentiment (Intellica AI, 2019).

Coming to unsupervised learning, it involves using a rule-based approach to analyze a comment. The supervised approach is a classification model that involves using traditional machine learning or deep learning methods.

The first focus is on the unsupervised approach with pre-built libraries to conduct sentiment analysis. The TextBlob and NLTK-VADER are used open-source, IBM Watson is a paid library but allows you to access the API on trial basis for a few thousand times.

### Word Clouds

A Word Cloud or Tag Cloud is a visual representation of text data in the form of tags, which are typically single words whose importance is visualized by way of their size and color. As unstructured data in the form of text continues to see unprecedented growth, especially within the field of social media, there is an ever-increasing need to analyze the massive amounts of text generated from these systems. A Word Cloud is an excellent option to help visually interpret text and is useful in quickly gaining insight into the most prominent items in a given text, by visualizing the word frequency in the text as a weighted list (Bodapati, 2017).

### Document Clustering

Document clustering involves the use of descriptors and descriptor extraction. Descriptors are sets of words that describe the contents within the cluster. Document clustering is generally considered to be a centralized process. Examples of document clustering include web document clustering for search users (Wikipedia, 2019).

The application of document clustering can be categorized to two types, online and offline. Online applications are usually constrained by efficiency problems when compared to offline applications. Text clustering may be used for different tasks, such as grouping similar documents (news, tweets, etc.) and the analysis of customer/employee feedback, discovering meaningful implicit subjects across all documents.

In general, there are two common algorithms. The first one is the hierarchical based algorithm, which includes single link, complete linkage, group average and Ward's method. By aggregating or dividing, documents can be clustered into hierarchical structure, which is suitable for browsing. However, such an algorithm usually suffers from efficiency problems. The other algorithm is developed using the K-means algorithm and its variants. Generally hierarchical algorithms produce more in-depth information for detailed analyses, while algorithms based around variants of the K-means algorithm are more efficient and provide enough information for most purposes

### K-means clustering

From the cleansed document vector, to get started in K-means document clustering, you must have a TF-IDF matrix. We use the TF-IDF matrix in the SciKit-Learn package. First, each document is normalized to length 1, so there is no bias for longer or shorter documents. This equal staking the relative frequencies instead of the absolute term counts. This is the "TF".

Second, IDF then is a cross-document normalization, that puts less weight on common terms, and more weight on rare terms, by normalizing (weighting) each word with the inverse in-corpus frequency. Here it does not matter whether you use the absolute or relative frequency, as this amounts just to a constant factor across all vectors, so you will get different distances, but only by a constant factor (the corpus size).

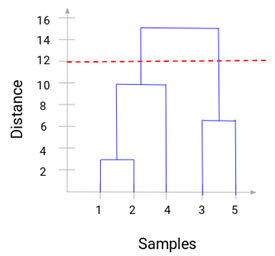
Cosine similarity is measured against the TF-IDF matrix and can be used to generate a measure of similarity between each document and the other documents in the corpus (each synopsis among the synopses). Subtracting it from 1 provides cosine distance which I will use for plotting on a Euclidean (2-dimensional) plane.

Now onto the clustering piece. Using the TF-IDF matrix, you can run a slew of clustering algorithms to better understand the hidden structure within the synopses. K-means initializes with a pre-determined number of clusters. Each observation is assigned to a cluster (cluster assignment) to minimize the within cluster sum of squares. Next, the mean of the clustered observations is calculated and used as the new cluster centroid. Then, observations are reassigned to clusters and centroids recalculated in an iterative process until the algorithm reaches convergence.

### Hierarchical Cluster Analysis

Agglomerative hierarchical clustering differs from k-means in a key way. Rather than choosing a number of clusters and starting out with random centroids, we instead begin with every point in our dataset as a “cluster.” Then we find the two closest points and combine them into a cluster. Then, we find the next closest points, and those become a cluster. We repeat the process until we only have one big giant cluster.

Ward's method is a criterion applied in hierarchical cluster analysis. Ward's minimum variance method is a special case of the objective function approach originally presented by Joe H. Ward, Jr. Ward suggested a general agglomerative hierarchical clustering procedure, where the criterion for choosing the pair of clusters to merge at each step is based on the optimal value of an objective function. The algorithm starts at the bottom and calculates the merged distances This objective function could be "any function that reflects the investigator's purpose." Many of the standard clustering procedures are contained in this very general class. To illustrate the procedure, Ward used the example where the objective function is the error sum of squares, and this example is known as Ward's method or more precisely Ward's minimum variance method.

The results between k-means and hierarchical clustering may in some cases be pretty similar. This is not always the case, however. In general, the advantage of agglomerative hierarchical clustering is that it tends to produce more accurate results. The downside is that hierarchical clustering is more difficult to implement and more time/resource consuming than k-means.

*Figure 9 – Example Dendogram*

A dendrogram is a tree-like diagram that records the sequences of merges or splits. Whenever we merge two clusters, a dendrogram will record the distance between these clusters and represent it in graph form. Let’s see how a dendrogram looks like:

We can clearly visualize the steps of hierarchical clustering. More the distance of the vertical lines in the dendrogram, more the distance between those clusters.

### Topic Modeling

Beginning with the cleaned and stemmed data set, a duplicate is made with all columns removed with the exception of the brand, review score classification, and the review text that was stemmed using the Porter stemmer. The documents are extracted and tokenized using the NLTK regular expression tokenizer. A document term matrix is created from these tokens and used to create a bag of words. The document term matrix is a collection of all the words in the corpora and whether its presence is represented in a document while a bag of words is a collection of all of the words in the corpora with the actual number of occurrences of each word. After these two items are created, they are plugged into the Gensim LDA model function along with the number of topics to search for.

For this dataset, without a clear number of known topics prior to the analysis 2, 3 and 5 topics were search for in the review text. After each model was run it was saved for later user and visualized using the pyLDAvis package.

### Naïve Bayes

Naïve Bayes is a method of classification based on Bayes Theorem. This theorem assumes that each predictor in a dataset is independent of one another. The model is considered “Naive” because of this assumption. Even if it is well known that the predictors are related to one another, the model still treats every predictor as entirely independent. These predictors are then used to predict the posterior probability. This can be shown in the equation below.

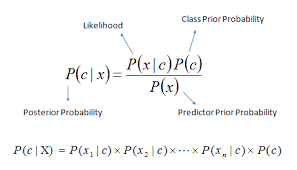


Figure 10: Posterior Probability Using Bayes Theorem

The assumption of independence allows the model to be run very quickly since the model does not have to calculate the probability for every single combination of predictors. However, this speed is a tradeoff for a more accurate model that takes the relationships of predictors into account. The Naïve Bayes model requires the data to be labeled. However, the data set can only have one label. Since the restaurant review data has two labels (sentiment and lie), the data set had to be duplicated and the labels separated. Including the label in the data set, even if predicting the other label, could skew the results inaccurately.

### Bernoulli

The Bernoulli Model is a method of classification using Bayes Theorem. This theorem assumes that each predictor in a dataset is independent of one another. The model is considered “Naive” because of this assumption. Even if it is well known that the predictors are related to one another, the model still treats every predictor as entirely independent. These predictors are then used to predict the posterior probability. This can be shown in the equation below.

The difference between the Bernoulli Model and the Multinomial Naïve Bayes Model is that the Bernoulli Model is used for classifying binary labels. Multinomial Naïve Bayes records the probability that a feature is one of many possible options from the counts of the other predictors. However, Bernoulli uses each feature independently to classify a label as one or the other using the binary value of the predictors. The assumption of independence allows the model to be run very quickly since the model does not have to calculate the probability for every single combination of predictors. However, this speed is a tradeoff for a more accurate model that takes the relationships of predictors into account.

### Support Vector Machine

The Support Vector Machine (SVM) is a supervised machine learning method used to classify binary data. This means that the SVM is used to split data into two groups with either “Label A” or “Label B”. This is achieved by creating a plane to separate the data. For data that is linearly separable, this plane is simply a line that maximizes the distance between both groups (the line that is evenly between the two groups).

**Plane of Separation with Linearly Separable Data**



Figure 11: Plane of Separation with Linearly Separable Data

While this seems simple for data that is linearly separable, this method becomes much more complicated when the data cannot be separated this way. In this case, the data will need to be transformed into a higher dimension. For example, consider the following example where the data appear to be separated by a circle.

**Example Non-linearly Separable Data**



Figure 12: Non-linearly Separable Data

In this example, the third-dimension *z* can be defined as z = x2 + y2 (This is the equation for a circle). A slice of this three-dimensional data can be shown below.

**Transformed Example Data**



Figure 13: Z vs X Plot of Transformed Data

The linear plane can now be created to separate the data. When transformed back to the two-dimensional data, the plane of separation is shown below.

**Plane of Separation in Original Dimension**



Figure 14: Plane of Separation Transformed Back to Original Dimensions

### However, the method to define the third dimension is not often obvious. To accommodate this, different kernel functions can be used to train the model. These kernel functions can be linear, radial, polynomial, etc.

### Decision Tree

Decision Tree Analysis is a method of classifying data by splitting it into different label groups based on the values of various features. The data is broken off into “branches” from the main data set. These branches are determined by whether the data point meets certain criteria for a particular feature. For example, these decision nodes could represent whether the data record represents a male or female, a child or an adult, or any other split based off of a single feature. After the data is split, the data points reach the “terminal node” which classifies the data point based on the label in question. These trees can be tuned using various control measures. The two measures that were used to tune the trees in this analysis were minimum size of each split and the complexity parameter. The minimum size of each split indicates the minimum number of data points that must be in each “leaf” of the tree. The complexity parameter indicates how much the data point must improve the fit of the tree for the split to be included.

## Results

### Sentiment Analysis

After classification using both the VADER and the TextBlob unsupervised sentiment analysis methods, it was discovered that both methods have wildly different interpretations of the review text.

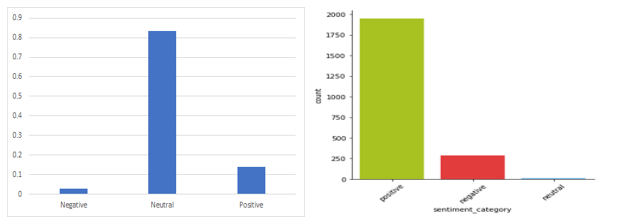


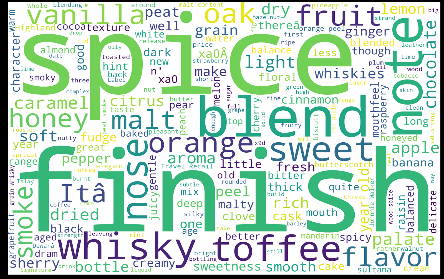
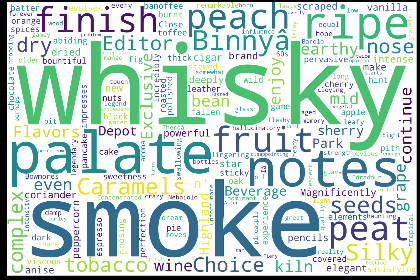
Figure 15 VADER (left) and TextBlob (right) sentiment classification distributions

The VADER analysis learned much more heavily on not classifying the reviews as positive or negative. This could prove useful to get the very strong negative or positive reviews to potentially pull out the words that may make them favor either end of the spectrum so strongly.

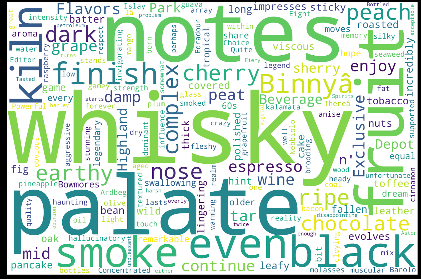
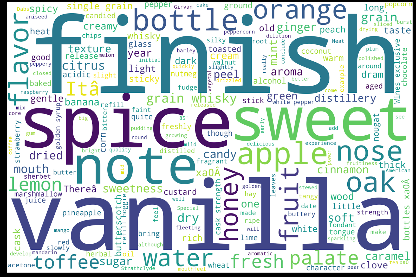
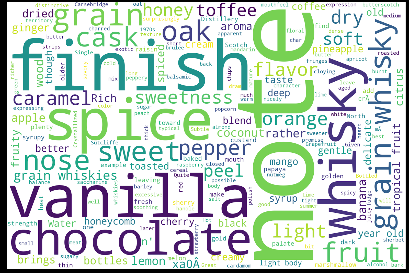
The TextBlob analysis on the other hand classified positive a whole 80% of the time. This may mean that when looking at the data with a looser view of the meaning of each word, the reviews are generally more positive.

Though the unsupervised approach is a good start for our sentiment prediction, using the open source unsupervised approach doesn’t produce consistent results given our requirements. While the unsupervised approach is ideal for generic use, supervised approach is better suited to analyze large amount of labeled data for a specific domain.

### Word Clouds



*Figure 16 Overall Word Cloud Figure 16a Blended Malt Scotch Whisky Figure 16b Blended Scotch Whisky*



*Figure 16c Blended Scotch Whisky Figure 16d Single Grain Whisky Figure 16e Single Malt Scotch*

For the Overall Word Cloud, the interesting words are: palate, smoke, finish, and obviously “whisky” dominates the cloud.

For the “by category” clouds: Blended Malt Scotch fans place an emphasis on “fruit” which seems to be an influencing word focusing on the blend of fruity flavors. Blended Scotch admirers use the word “Spice” frequently as it jumps out as a significant inspiration for the reviewers. Grain Scotch lovers are consumed by “vanilla” and “chocolate” which dominates along with “note” seemingly making a metaphor of peoples love of scotch with their favorite soothing melody. Single Grain Scotch drinkers focus on the “Finish”, not on the start but how it closes. And finally, Single Malt Scotch supporters emphasize the “palate”. So, the roof of the mouth makes major decisions with single malt scotch.

### Document Clustering

### K-Means Clustering

The first step in K-Means clustering is finding the optimal number of clusters. To find the optimal number of clusters the “Elbow Method” (see Figure 15 below) is used. The graph below depicts the squared distance between clusters vs. The optimal # of clusters. The optimal clusters is at the “elbow” of the graph. This graph did not show a clear “elbow”, so a number of 5 clusters was selected.

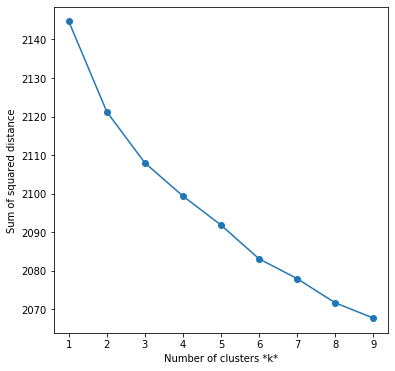
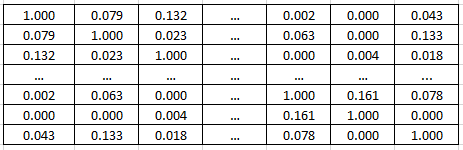


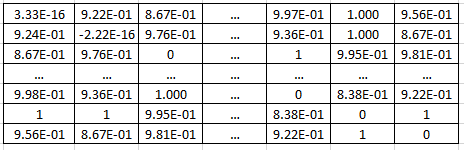
Figure 16 K-Means Elbow Method

Once the 5 clusters are selected, the TF-IDF Document for each document is normalized to length 1, so there is no bias for longer or shorter documents. This equals taking the relative frequencies instead of the absolute term counts. This is "TF". Then, IDF then is a cross-document normalization, that puts less weight on common terms, and more weight on rare terms, By normalizing (weighting) each word with the inverse in-corpus frequency. This is “IDF”.

Then the Cosine Similarity matrix is created



From the matrix above, the Euclidean distance Matrix is created



And finally, the major words for each cluster are derived

**Top terms per cluster:**

Cluster 0: whiski note vanilla dri fruit

Cluster 1: sherri cask finish year old

Cluster 2: sweet water fruit light smoke

Cluster 3: nose cask bottl finish oak

Cluster 4: peat smoke whiski seawe smoki

**How do we categorize these clusters:**

0 Vanilla Fruit

1 Sherry Cask

2 Sweet Water

3 Scent and Finish

4 Peat Smoke

From our clusters, a 2D representation is created of all the documents. This graph makes you wonder how the purple documents, which are those in “sweet water” cluster could be related, but if you look at this graph as if it were a 3D cylinder, you could see how these items would be at the top of the 3D cylinder and related across that specific slice.

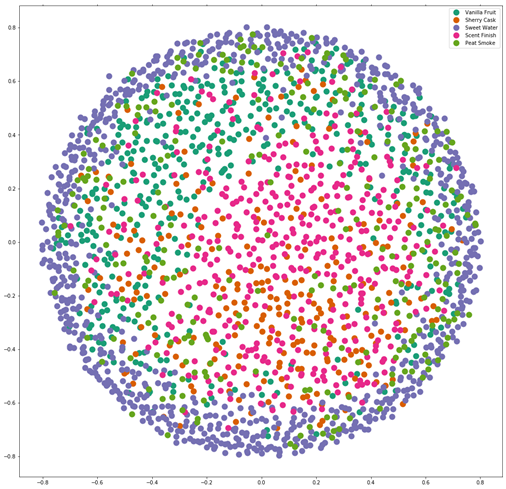


Figure 17 Visualization of the K Means Cluster model

Finally, we see how we can predict with the cluster of a document through our model. By running a sentence or paragraph through the model, a prediction of the document cluster can then be derived.

Document Cluster Predictions

Entry 1

marriag three differ singl malt age american french oak whiski show advantag marri whiski one distilleri proper done vibrant complex array fruit orchard fruit sultana sweet light toffe marzipan honey malt spice creami vanilla mocha warm pepper smoke tar smoke oliv coal lesser note toast

*Predict* - Peat Smoke

Entry 2

Aromatic sherri flavors are good

*Predict*- Sherry Cask

### Hierarchical Document Clustering

Agglomerative hierarchical clustering is used here which differs from k-means in a key way. Rather than choosing several clusters and starting out with random centroids, instead begin with every point in the dataset as a “cluster.” Then find the two closest points and combine them into a cluster. Then, find the next closest points, and those become a cluster. Then repeat the process until we only have one big giant cluster.

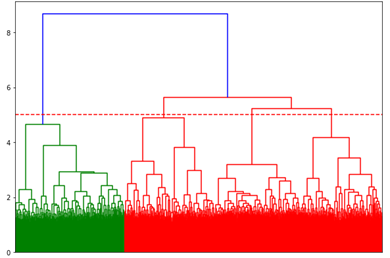


Figure 18 Dendogram created during document clustering

Ward’s method is used and a very dynamic package from SciPy called cluster.hierarchy. A dendrogram is created, which works it’s way from the bottom matching related document and building up the cluster.

The number of document clusters is the obtained by selecting the number of clusters on the longest vertical line prior to when the vertical line crosses the horizontal line. Therefore, the selection here is 5 clusters which actually lines up with our K-means selection on the y-axis for the longest vertical line at the point prior to the first horizontal line for the cluster group in this hierarchy.

### Latent Dirichlet Allocation

The choice of aiming for two topics in the review corpora was found to have the best results with the least amount of overlap between the keywords and the widest spread of topics. When specifying two topics ideas about taste and ideas about appearance surface.

Three topics also had favorable results, though one of the topics focused heavily on the Glenlivit brand while the other two topics focused on fruits and non-fruit tastes.

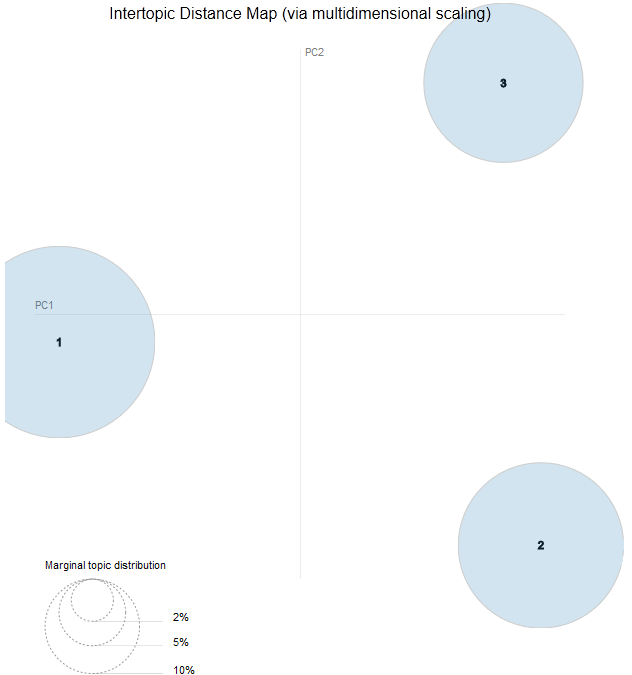


Figure 19 Visualization of the topic distance when topics=3

### Naïve Bayes

The results for using Naïve Bayes to classify price are shown below:

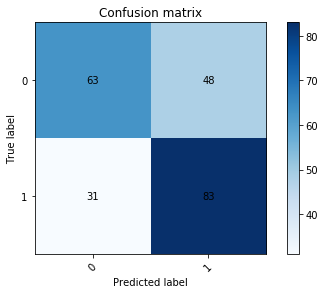


Figure 20: Confusion Matrix for Naïve Bayes – Price

For price, the Naïve Bayes model produced output that classified price with 64.89 % accuracy against the test data.

The results for using Naïve Bayes to classify sentiment are shown below:

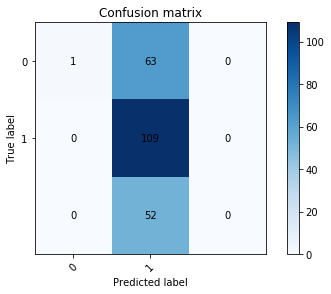


Figure 21: Confusion Matrix for Naïve Bayes – Sentiment

For sentiment, the Naïve Bayes model produced output that classified price with 48.89 % accuracy against the test data.

### Bernoulli

The results for using Bernoulli to classify price are shown below:

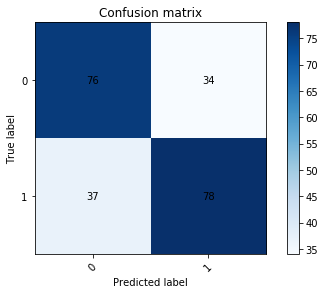


Figure 22: Confusion Matrix for Bernoulli – Price

For price, the Bernoulli model produced output that classified price with 68.44 % accuracy against the test data.

The results for using Bernoulli to classify sentiment are shown below:

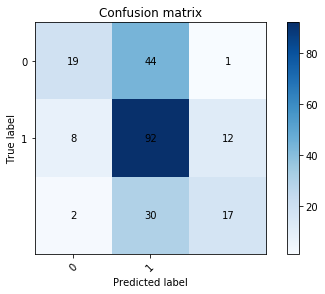


Figure 23: Confusion Matrix for Bernoulli – Sentiment

For sentiment, the Bernoulli model produced output that classified price with 56.89 % accuracy against the test data.

### Support Vector Machine

The results for using SVM to classify price are shown below:

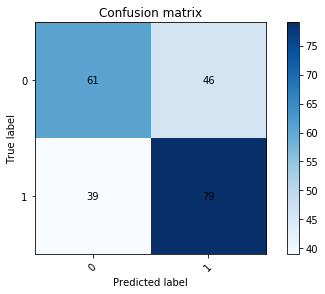


Figure 24: Confusion Matrix for SVM – Price

For price, the SVM model produced output that classified price with 62.22 % accuracy against the test data.

The results for using SVM to classify sentiment are shown below:

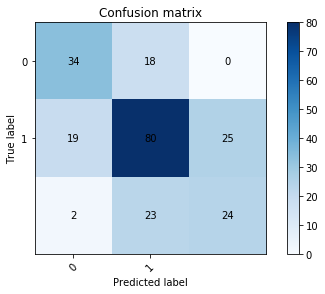


Figure 25: Confusion Matrix for SVM – Sentiment

For sentiment, the SVM model produced output that classified price with 61.33 % accuracy against the test data.

### Decision Tree

The results for using Decision Tree to classify price are shown below:

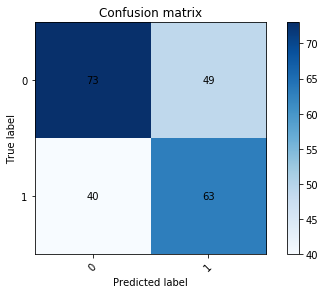


Figure 26: Confusion Matrix for Decision Tree – Price

For price, the Decision Tree model produced output that classified price with 60.44 % accuracy against the test data.

The results for using Decision Tree to classify sentiment are shown below:

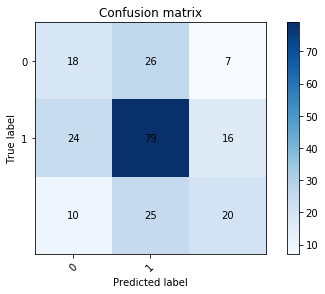


Figure 27: Confusion Matrix for Decision Tree – Sentiment

For sentiment, the Decision Tree model produced output that classified price with 52.00% accuracy against the test data.

### Summary of Results

Overall, the models show that it is possible to analyze the text of the review to classify the sentiment and price of the review of the whiskey in question. However, some models performed better than others. For sentiment, the best model was the Support Vector Machine with an accuracy of 61.33%. For price, the best model was Bernoulli with an accuracy of 68.44% against the test data. While the price models performed better, this is likely due to the lower number of possible classes (2) for price as opposed to the larger number of classes (3) for sentiment. A breakdown of the most influential words for positive and negative sentiment are shown below:

**Feature Importance**

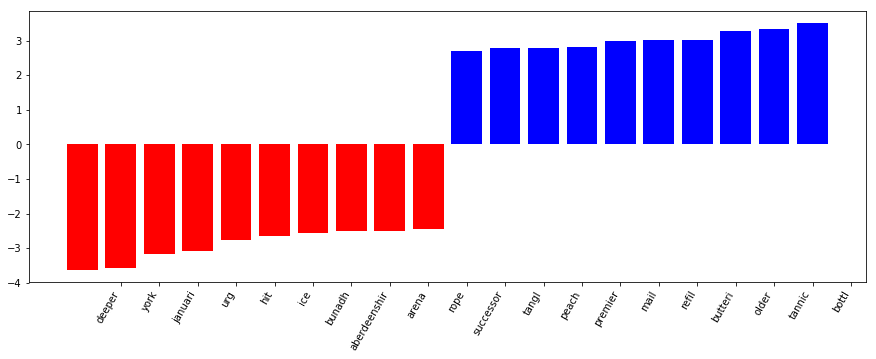


Figure 28: Feature Importance – Sentiment

These feature importance values show the subjectivity in the reviews of the different whiskeys. The most influential words for negative sentiment appear to be technical words such as “jamuari” or “aberdeenshir”. However, the most influential words for positive sentiment are more laymans terms such as “peach” or “rope”. This lends one to believe that the more negative reviews are coming from educated critics, and the positive reviews are coming from more casual whiskey drinkers. This subjectivity appears to be the main obstacle in predicting the sentiment and price from the whiskey reviews.

## Conclusion

Scotch will continue to remain popular in the United States and many countries for the foreseeable future. There are many different varieties of scotch, at many different price points, and at many different qualities.

Based upon this analysis, there is significance in the age of a scotch and its review score. Older scotches have the advantage. However, there is currently discussion on the accuracy of age statements on labels. Some distillers may be inaccurately stating age. It seems that consumption of the aged scotch is greater than the production. This may impact future pricing and also the continued decline in age accuracy in the future. There is also discussion of taking the age statements off the labels. However, if it becomes clear that consumers react negatively to eliminating age statements, presumably that will lead to some distillers using the age statement to make a quality claim.

Experts can offer their opinions on what constitutes a good scotch, but in the end the choice is left up to the consumer. It may be possible to use the scotch reviews of experts in meaningful ways to help classify a scotch based on quality, absent any other review system. Words like “finish”, “palate”, “oak” and other descriptive words show high prevalence in scotch reviews and one would assume reviews which praise the quality of a bottle would have distinguishing vocabulary from one which condemns it.

Using the choices in vocabulary of a review shows some potential in distinguishing the different varieties of scotch: “Single Malt”, “Blended”, “Blended Malt”, “Single Grain” and “Grain”. It is not unreasonable to assume that different varieties of scotch would be associated with different vocabulary describing them. There may be more useful methods within the realm of data science to help with this assignment that involve more focused direction when attempting to make this classification.

The love of scotch is a deep one with its loyal aficionados as can be exhibited by one of the most famous scotch drinkers, Mark Twain. Twain said, “Too much of anything is bad, but too much good whiskey is barely enough.”

# References

Ando, K. (2018, June 13). *2.2k+ Scotch Whiskey Reviews*. Retrieved from Kaggle: https://www.kaggle.com/koki25ando/22000-scotch-whisky-reviews

Bodapati, N. (2017, December 18). *Visualizing Text Analysis Results with Word Clouds*. Retrieved from Dundas BI: https://www.dundas.com/support/blog/visualizing-text-analysis-results-with-word-clouds

Intellica AI. (2019, June 1). Retrieved from Intellica AI: https://intellica.ai/

Puckette, M. (2016, January 23). *There's Still No Cure for Grape Phylloxera*. Retrieved from Wine Folly: https://winefolly.com/review/no-cure-for-grape-phylloxera/

Whiskey Advocate. (2019, June 01). *Scotch Whiskey Brands*. Retrieved from Whiskey Advocate: https://www.thewhiskyexchange.com/brands/scotchwhisky/40/single-malt-scotch-whisky.html

Wikipedia. (2019, June 6). *Document Clustering*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Document\_clustering