A Case Study of Clustering on Retail Business: Application on Wine Data

Junge Zhang

Supervised by Dr. Maxime Cohen and Dr. Foster Provost

June 17 2019

Table of Contents

1 Problem Introduction

Imagine yourself planning to start your own wine store and you have access to stock any kind of wine in the world, will you hesitate in choosing which types of wines to sell in your wine store? Remember, your access to any type of wines does not mean you have unlimited budget and inventory space. Moreover, how do you know the choices you pick will be those wines expected by your future customers? Will these picks help you differentiate from other wine stores competitors so that you can secure more customers?

In this case study, we want to set a standard to pick wines, considering the limited stock space in the wine store but with potential options of thousands of wines. From the perspective of data, we need to find the similarities and difference among these wines so that we can attribute them into different cluster labels. Under different labels, we can pick representative wines from different clusters based on other evaluations. In addition, although it is expected to cluster the wines based on common and naturally clustered features such as their origins and grape types, such clusters may not be friendly to customers who do not know deeply about wines. In fact, as we expect most of customers are not professionals about wines, we want to find a clustering standard which can direct customers to determine which wines to purchase in a more intuitive way. As we differentiate from other wine stores in using different wine cluster labels, the more convenient and direct clustering format may help the wine store gain further satisfaction from customers when they can find wines they like more conveniently, even though they barely know much about wines. Therefore, in this case, we will focus on clustering different wines purely based on their descriptions, including the information of their colors, smells and tastes to create more informative clustering labels for future customers.

2 Data Description

The wine data is fetched from the wine review data on Kaggle. The dataset contains over 130 thousand samples, each of them evaluated by one professional sommelier. The dataset has totally 14 features including country, description (wine evaluation by a sommelier), designation (the vineyard of grapes), points (numerical rating of the wine), price, province, region1, region2, taster name, taster twitter handle, title (title of the twitter but actually contains information of the wine title), variety (type of grapes), and winery. These features, based on their potential business interpretation in clusters, are further divided into four groups. The first group is normal wine features including country, designation, variety, winery and title (which is used to extract the feature year). These are common features used in a wine store. The second group is a unique feature(s), description, which although has one single column, contains much more information like smell and taste as a text feature. The third group has two numerical features, points and price, that will be used to determine the choices of wines after clustering. The last group contains all other features. These features do not make too much business sense in clustering different wines.

Figure 1, Figure 2 and Figure 3 show the data distribution along with three numerical features (points, price and year) using histogram. The histogram of points shows nearly a normal distribution between 80 and 100. The price histogram is right skewed with the most expensive wine over 200 dollars. Meanwhile, it is expected the price mainly concentrates below 50 dollars. The year histogram shows a relatively left skewed distribution with the most aged wines older than 1995 (notice wines that are older than 1950 have been removed). In the data, most wines are around 2010 to 2015.

Figure 1: Wine Points Distribution

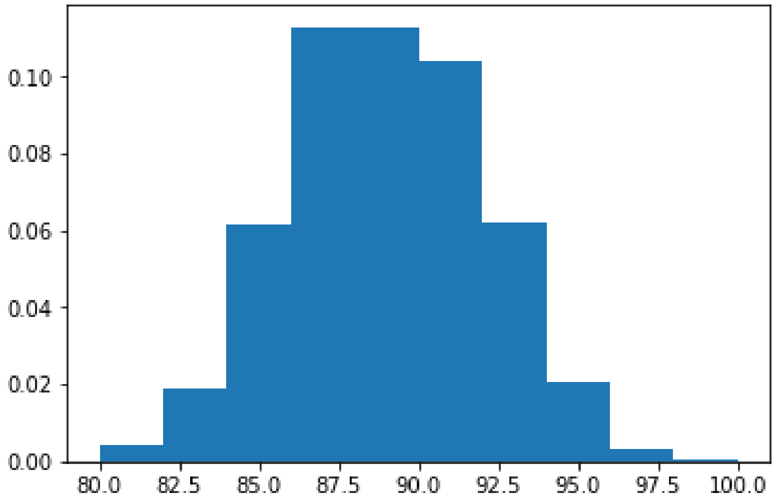


Figure 2: Wine Price Distribution

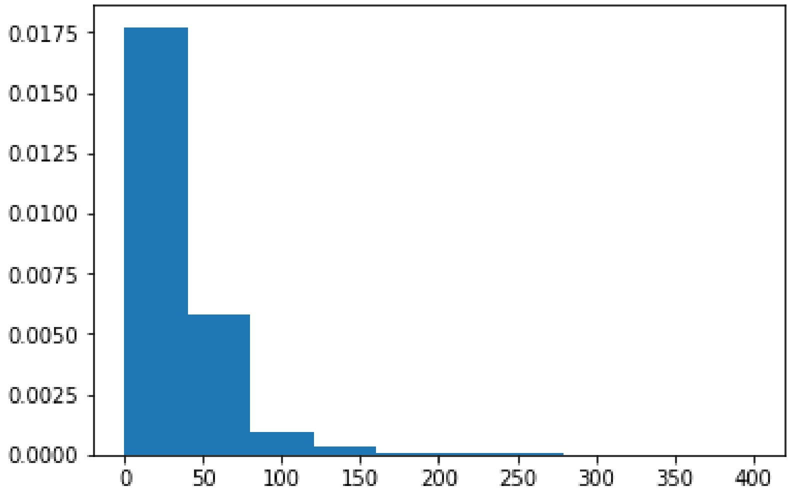
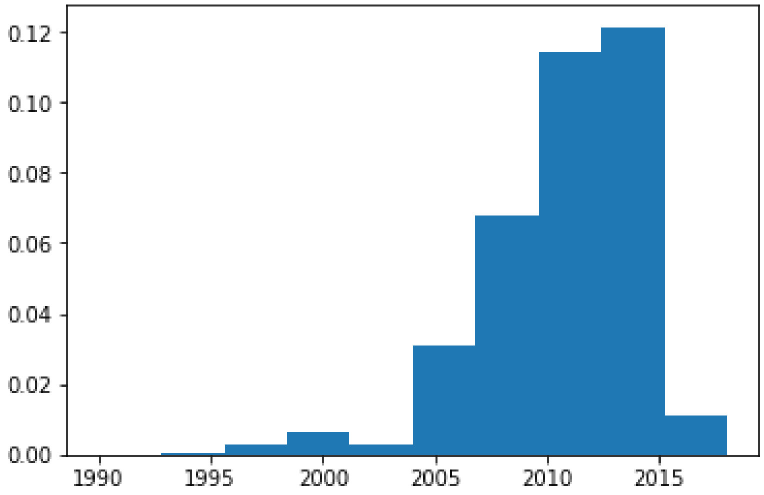


Figure 3: Wine Year Distribution



Because of the property of clustering algorithms, related numerical features will be standardized to reduce the effects of different feature scales. In addition, for related categorical features, since models like Kmeans only accepts numerical input, these features will be converted to multiple binary features by OneHotEncoder. To avoid unnecessary large feature input, only pick top three frequent categories in each feature and make other categories as “others” before applying OneHotEncoder. For the text feature, it will be treated using TF-IDF method, which will be explained in detail in the clustering with NLP sections.

3 Clustering without Natural Language Processing

In this part, we only consider clustering with the features from the first group. As normal features are either numerical or categorical, the preprocessing steps only include scaling the numerical features and convert categorical features into binary numerical features as described in the above section.

3.1 Kmeans Clustering

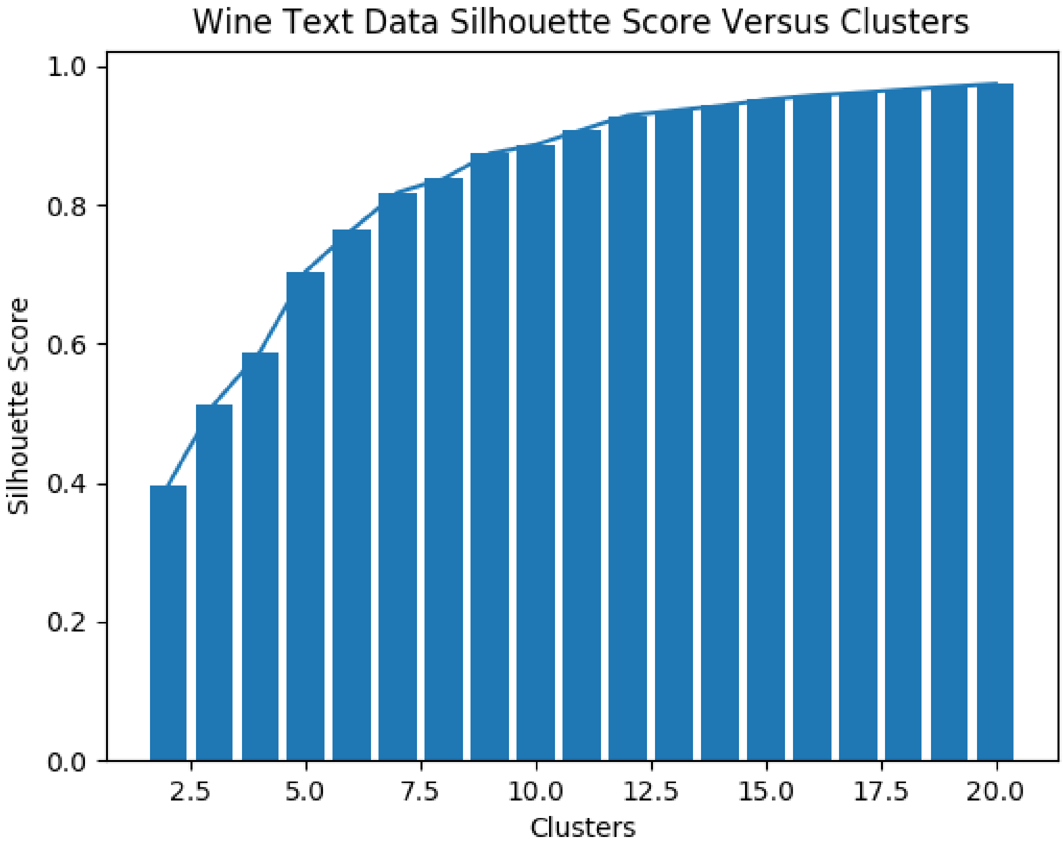
We first apply Kmeans clustering algorithm on the dataset. Kmeans algorithm is an iterative algorithm that tries to minimize the inertia, sum of square based on the centroid within each cluster, at every step. The algorithm first randomly picks centroids in the data from the chosen cluster number. Then it computes distance, commonly Euclidean distance that is computed by the l2 norm between two points, from each point to each centroid and decides its cluster based on the shortest distance to a certain centroid. After finding clusters by computing distances, the algorithm will re-compute the centroid of each cluster. Then the algorithm repeats the above steps until the centroid of the clusters converges. Kmeans algorithm is simple, flexible and easy to interpret but may have the risk of failing to find the minimizer because of the randomness of initial centroids.

3.1.1 Kmeans Application on Wine Data

The main Python package we refer is to sklearn.cluster.Kmeans, which can easily implement the Kmeans algorithm by specifying the number of clusters and methods of picking the initial centroids in the dataset. Regarding how we tune the parameter, we use both quantitative and qualitative metrics to evaluate the performance of the clustering algorithm.

Quantitatively, since we have no true labels of wines in our dataset, the unsupervised evaluation method is Silhouette score. The way Silhouette score defines the performance of the clusters by computing two distances of each point in the dataset. The first distance is the mean distance of one sample from all other samples within the same cluster. The second distance is the mean distance of one sample from all other samples in its nearest cluster. The Silhouette coefficient is computed by the difference of the two distances divided by the larger distance between these two. The range of the Silhouette coefficient is between -1 to 1 and higher Silhouette score indicates high dense clusters. In this evaluation, we choose to use cosine distance, which is computed based on the inner product of two samples. We choose cosine distance because most columns are binary features, which are similar to text features that are suited to be measured by cosine distance. Figure 4 shows Silhouette score for different chosen clusters. From the graph, we can observe the Silhouette score is relatively high for this case, which implies the normal features from the first group naturally cluster wines.

Figure 4: Silhouette Score in Kmeans Clustering



However, while Silhouette score quantitatively defines the well separated clusters, it does not necessarily mean the clusters are correct from a qualitatively perspective. In another word, in the clustering algorithm, it is significant to examine whether data samples in each cluster make sense in real world. In our data, this means we need to review whether wines in one cluster are similar to each other and fit the new label of the cluster. To find the new labels of different clusters, we treat the dataset now as a supervised classification problem while our cluster numerical label is the target feature. Since it is a multiclass problem, the idea is to build a One-VS-All logistic regression model for the data to find top features, determined by the value of their coefficients from the logistic regression model, to describe different clusters. The top features can be viewed as the typical characteristics in each cluster, and that is where we need the domain knowledge of wine to review whether these wines are correctly fit under the description by typical characteristics.

Since the Silhouette score among clusters are similar, we pick 5 clusters and 10 clusters and build the One-VS-ALL logistic regression models on these two choices. Table 1 shows number of data points in each cluster. Figure 5 shows the centroid distribution based on different features on the five cluster Kmeans algorithm. Figure 6 and 7 show the typical characteristics of different clusters for 5 cluster case and 10 cluster case.

Table 1: Number of Data in Each Cluster

Cluster Index	Number of Data in Five Clusters	Number of Data in Ten Clusters
0	16072	3579
1	23789	29447
2	30876	27209
3	16343	16072
4	29447	9823
5		12764
6		6724
7		7242
8		1917
9		1750

Figure 5: Centroid Distribution in Kmeans Clustering

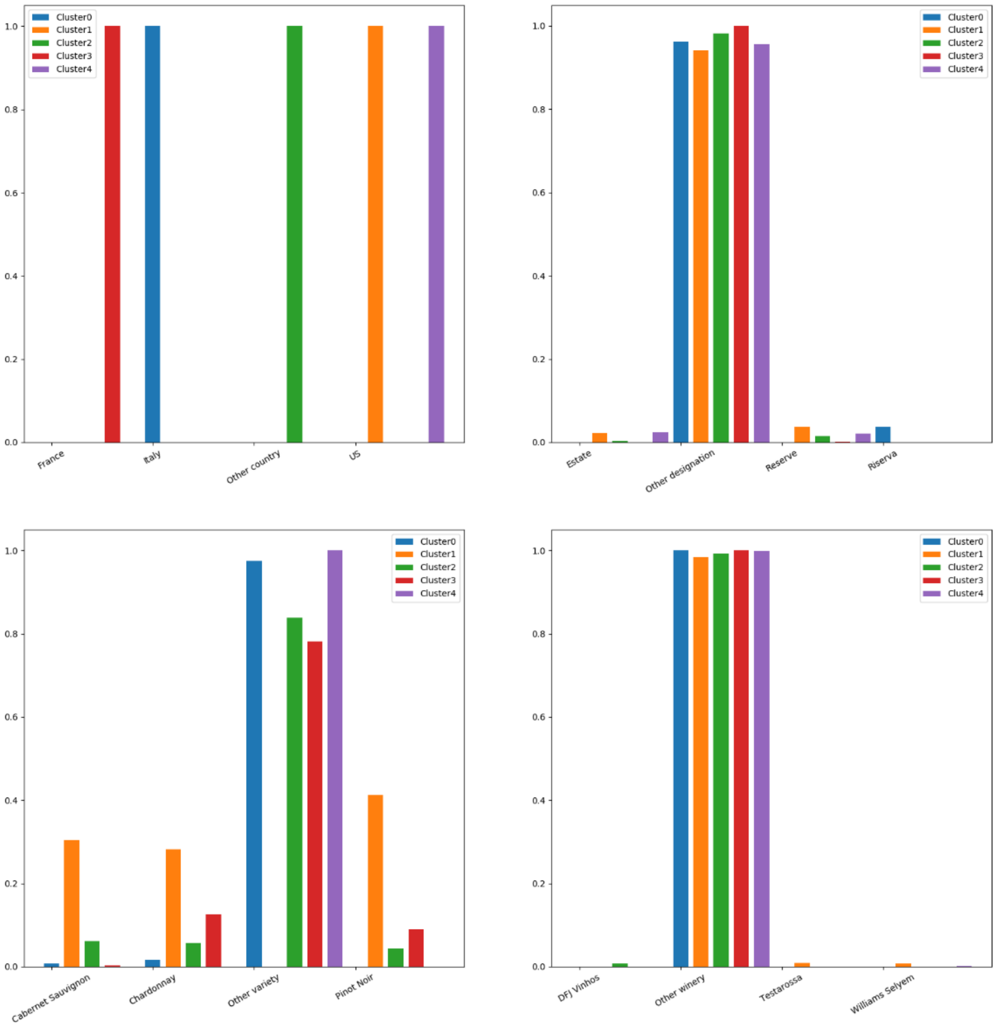
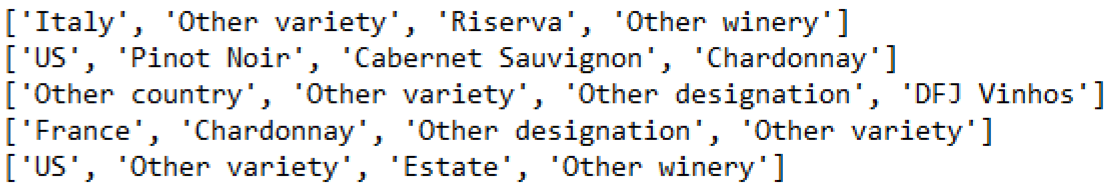
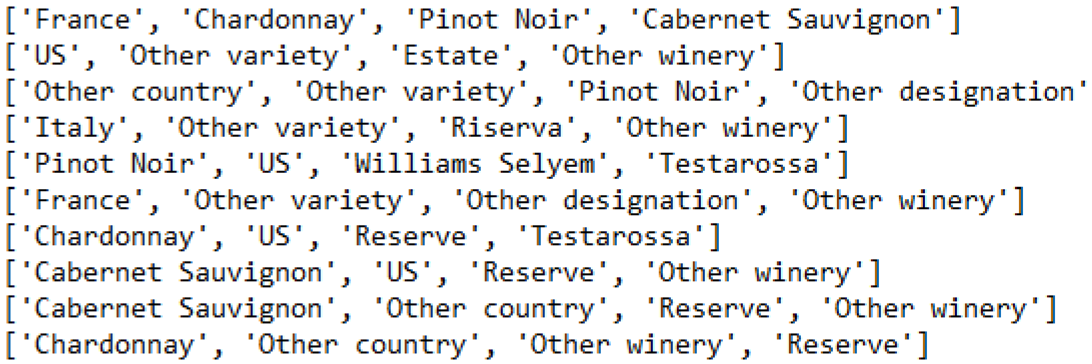


Figure 6: Typical Characterisitics for 5 Clusters



```
['Italy', 'Other variety', 'Riserva', 'Other winery']
['US', 'Pinot Noir', 'Cabernet Sauvignon', 'Chardonnay']
['Other country', 'Other variety', 'Other designation', 'DFJ Vinhos']
['France', 'Chardonnay', 'Other designation', 'Other variety']
['US', 'Other variety', 'Estate', 'Other winery']
```

Figure 7: Typical Characteristics for 10 Clusters



```
['France', 'Chardonnay', 'Pinot Noir', 'Cabernet Sauvignon']
['US', 'Other variety', 'Estate', 'Other winery']
['Other country', 'Other variety', 'Pinot Noir', 'Other designation'
['Italy', 'Other variety', 'Riserva', 'Other winery']
['Pinot Noir', 'US', 'Williams Selyem', 'Testarossa']
['France', 'Other variety', 'Other designation', 'Other winery']
['Chardonnay', 'US', 'Reserve', 'Testarossa']
['Cabernet Sauvignon', 'US', 'Reserve', 'Other winery']
['Cabernet Sauvignon', 'Other country', 'Reserve', 'Other winery']
['Chardonnay', 'Other country', 'Other winery', 'Reserve']
```

3.1.2 Impact of Adding Features

In this section, we experiment the Kmeans clustering algorithm after adding a new feature, year, into the dataset. As stated above, the year feature is extracted from the title column. After scaling and merging, we did similar things as the above section to review the performance change in the clustering and classification model. Figure 8 shows the new computed Silhouette score. Table 2 shows number of data points in each cluster with the added feature. Figure 9 shows the new centroid distribution based on different features. Figure 10 and 11 show the typical characteristics of different clusters for 5 cluster case and 10 cluster case.

Table 2: Number of Data in Each Cluster

Cluster Index	Number of Data in Five Clusters	Number of Data in Ten Clusters
0	25142	11020
1	42430	13658
2	13458	20339
3	20949	17142
4	14548	6427
5		12281
6		11674
7		12228
8		3362
9		8396

Figure 8: Silhouette Score in Kmeans Clustering with Additional Feature

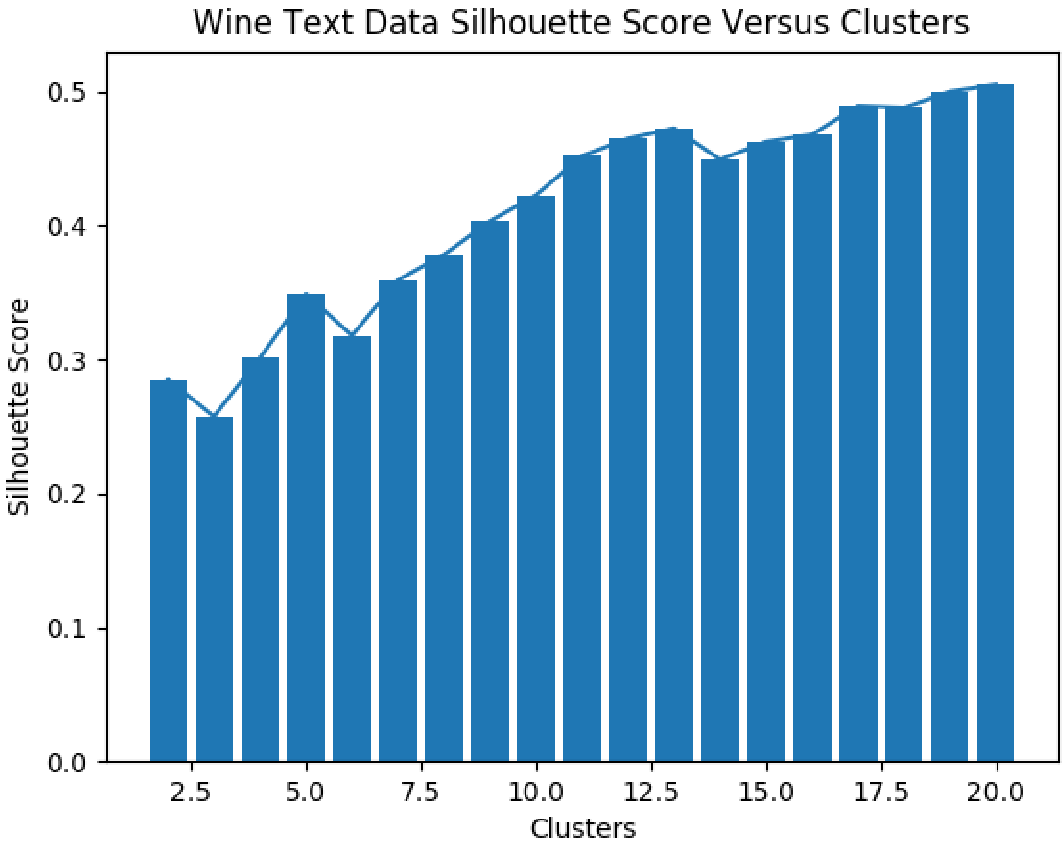


Figure 9: Centroid Distribution in Kmeans Clustering with Additional Feature

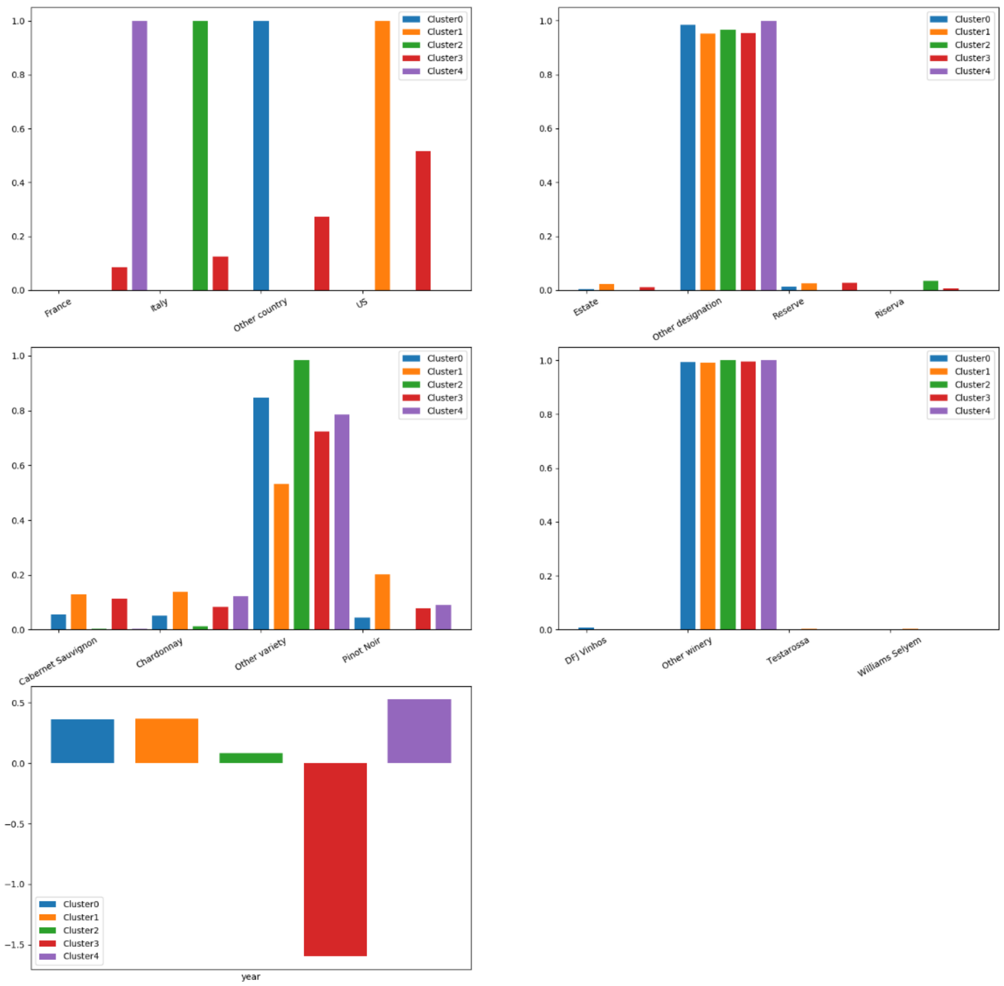
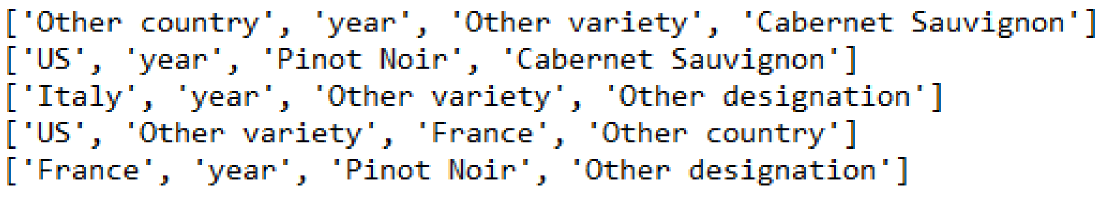
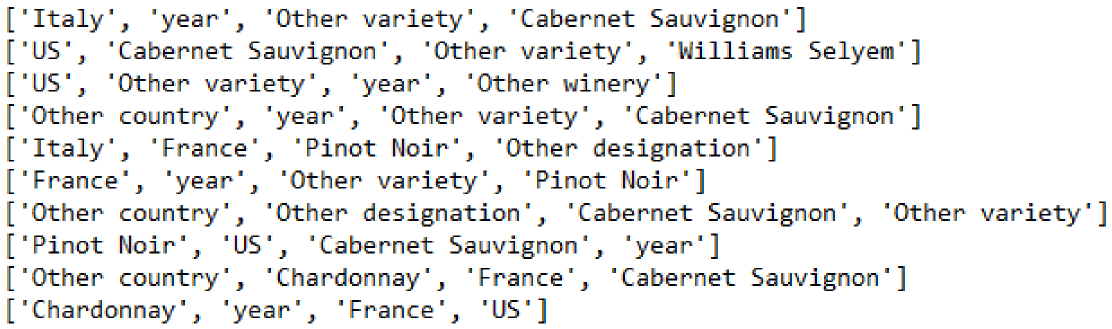


Figure 10: Typical Characterisitics for 5 Clusters



```
['Other country', 'year', 'Other variety', 'Cabernet Sauvignon']
['US', 'year', 'Pinot Noir', 'Cabernet Sauvignon']
['Italy', 'year', 'Other variety', 'Other designation']
['US', 'Other variety', 'France', 'Other country']
['France', 'year', 'Pinot Noir', 'Other designation']
```

Figure 11: Typical Characterisitics for 10 Clusters



```
['Italy', 'year', 'Other variety', 'Cabernet Sauvignon']
['US', 'Cabernet Sauvignon', 'Other variety', 'Williams Selyem']
['US', 'Other variety', 'year', 'Other winery']
['Other country', 'year', 'Other variety', 'Cabernet Sauvignon']
['Italy', 'France', 'Pinot Noir', 'Other designation']
['France', 'year', 'Other variety', 'Pinot Noir']
['Other country', 'Other designation', 'Cabernet Sauvignon', 'Other variety']
['Pinot Noir', 'US', 'Cabernet Sauvignon', 'year']
['Other country', 'Chardonnay', 'France', 'Cabernet Sauvignon']
['Chardonnay', 'year', 'France', 'US']
```

By comparison, the Silhouette is relatively low but it can only indicate that the new algorithm forms less dense clusters instead of worse clusters. Note that better or worse can be compared horizontally only when the Silhouette score is applied on the exact same dataset. Consequently, it means the qualitative metrics are better metrics to see whether the additional feature improves the performance. By comparing the typical characteristics, it can be observed that the new added year feature dominates some of the clusters. Generally, this means a new added feature may provide more information within the clustering itself. However, more information does not mean better performance. In our case, although the year feature becomes the top descriptive label, the clustering samples do not make better sense than the previous clustering without the year feature.

3.1.3 Interpretation

Be clear about the relationship and significance of the Silhouette coefficient and the qualitative metrics, the interpretation of the cluster is established on the top descriptive features for each cluster. With the descriptive cluster features, we can explicitly say which type of wines are in each of the cluster.

Other than the new descriptive labels, let’s not forget the business purpose behind this case study: to pick wines from each cluster for your wine store. With over 130 thousand samples, we can expect there are still a huge number of choices within each cluster. To pick representative wines from these clusters, the idea is to find wine samples that are nearer to the centroid of each cluster. Since Kmeans algorithm finds the clusters to minimize the sum of square distance within in every cluster, samples with less distance to the centroid will be the representatives of the clusters and that is why we extract 10% wine samples from each cluster.

Beyond that, don’t forget we have a group of related features that are not used in the model training. The third group contains points and price of the wine, which can be alternative dimensions we can use to select wines from the top 10% wine samples. This idea results in the final wine selection plan for the wine store. First sort the top 10% wine samples by their points in each cluster and then choose the 20 highest rating wines. Then, create the price bin for price region of below $10, between $10 and 20$, between $20 and $40, and above $40 for the 20 wines so that the top 20 wines can be further divided by their prices.

3.2 Hierarchical Clustering

In this section, we apply hierarchical clustering with 5 clusters on the dataset with the additional feature year, to compare the results with 5 Kmeans clusters. Hierarchical clustering is an algorithm that first treats each sample as a leaf in the tree and successively merge the nodes based on certain standard in a bottom-up way until the nested clusters fulfill a complete tree. In our application, we use “ward” option, which tends to minimize the sum of square distances within each cluster to approach it more similar to Kmeans algorithm. Unlike Kmeans, it does not include any randomness but it will be time consuming for large dataset like that in our case.

3.2.1 Hierarchical Clustering Application on Wine Data

The main package we refer is sklearn.cluster.AgglomerativeClustering, which can easily implement the hierarchical clustering algorithm by specifying the linkage criterion.

In hierarchical clustering algorithm, there is no need to tune the parameter. Although we specify 5 as the cluster number, it only means that, in a bottom up fashion, we find 5 cluster partition for the samples. The hierarchical tree will not change itself even if you change the cluster number here.

3.2.2 Interpretation

The interpretation part uses the similar procedure as that in Section 3.1.3.

4 Clustering with Natural Language Processing

In this part, we only consider clustering with the features from the second group. Although there is only one single column in this group, it hides huge information as a text feature. From either sample, the evaluation contains specific properties of the wine evaluated by the sommeliers, such as the smell, taste and texture. This will give a completely different yet far more informative clustering result than using the normal features from the first group after we extract the word information from the text feature.

4.1 Brief Introduction to Natural Language Processing

While a feature containing words or sentences covers a huge potential of information, it cannot be guaranteed to be a type of valid input in a machine learning algorithm most of the time. To convert the languages into valid input format and learn them as normal data is the core target for natural language processing.

Tokenizing is one common processing for natural language processing. Tokenizing normally splits the text data into one word or a set of words. Then these tokens can be parsed into numerical features as the input dataframe. The two packages referred in this case are sklearn.feature_extraction.text.CountVectorizer and TfidfVectorizer. The CountVectorizer tokenizes the text and produces occurrence of the word in each document as the feature columns. On the other hand, the TfidfVectorizer tokenizes the text and parses the word into TF-IDF in each document as the feature columns, while TF-IDF is computed as the product the term frequency and the inverse document frequency. In this formula, the term frequency refers to the occurrence of the term in a document while the inverse document frequency is the inverse fraction of documents containing the term in the corpus in a log scale. TF-IDF uses inverse document frequency as the weight to import the importance of a word in the whole corpus into the single document. In addition, TF-IDF is naturally standardized and can be directly used as an input in the Kmeans algorithm.

Some details in the CountVectorizer and TfidfVectorizer used in the case include stop_words and ngram_range. Stop words mean those words that are ignored during the tokenization. For example, words like “the”, “a” are normally meaningless in the machine learning process and should be removed as stop words. In our case, we include both a common stop word list, which includes words like “the”, “a’” and a customized stop word list. The customized stop word list is acquired by ranking the tokenized words based on their inverse document frequency. Typically, we don’t want words with small inverse document frequency (high frequency in the corpus) to be meaningless in the clustering model. For instance, the word “wine” itself can be a word with small inverse document frequency. However, “wine” itself cannot be some characteristic of wine in a cluster so that it is added into the customized stop word list. Ngram_range means whether we want to combine individual words into a set of words to create more meaningful features. For instance, while two words “black” and “berry” describe different properties of a wine, if these two words are adjacent, “black berry” actually means completely different from two split words. In our case, we tried bi-gram but the result is not good enough,so that the bi-gram result is abandoned in this report.

Other than the stop_words and ngram_range, there are other attributes that can be tried during the tokenizing process and they can be experimented to see whether the clustering performance can change for a further study.

4.2 Kmeans Clustering with Natural Language Processing

After processing the description column as using TfidfVectorizer and stop_words as described in the above section, we retain a 1000 word feature dataset as the input into the Kmeans clustering algorithm.

4.2.1 Kmeans with Natural Langue Processing Application on Wine Data

Similar to Section 3.1.1, Table 3 shows number of data points in each cluster with natural language processing. Figure 12 shows the Silhouette score for different clusters. The Silhouette score is relatively low maybe because we remove many words that contain much information, although not the information we want to interpret the cluster. And it can only indicate the final cluster is not high dense but not enough to say it is a bad cluster as long as the Silhouette score is above 0. Figure 13 and 14 show the typical characteristics of different clusters with natural language processing for 5 cluster case and 10 cluster case.

Table 3: Number of Data in Each Cluster

Cluster Index	Number of Data in Five Clusters	Number of Data in Ten Clusters
0	6770	8334
1	35671	17617
2	13537	16280
3	52147	5870
4	8402	5585
5		5334
6		4207
7		6427
8		33179
9		13694

Figure 12: Silhouette Score in Kmeans Clustering with Natural Language Processing

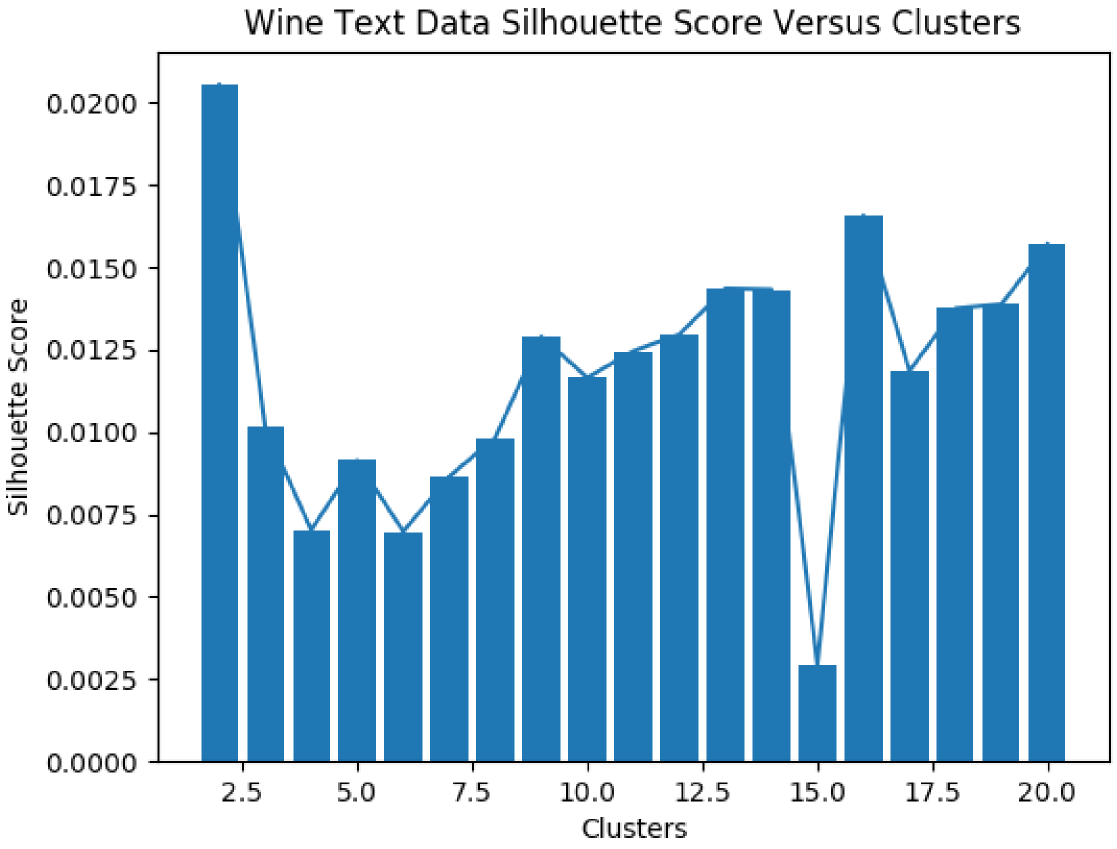
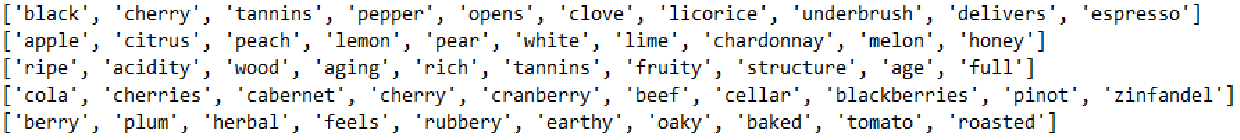
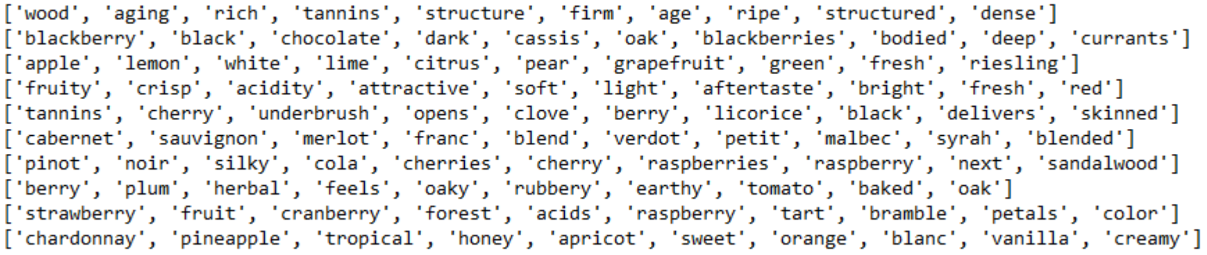


Figure 13: Typical Characterisitics for 5 Clusters



```
['black', 'cherry', 'tannins', 'pepper', 'opens', 'clove', 'licorice', 'underbrush', 'delivers', 'espresso']
['apple', 'citrus', 'peach', 'lemon', 'pear', 'white', 'lime', 'chardonnay', 'melon', 'honey']
['ripe', 'acidity', 'wood', 'aging', 'rich', 'tannins', 'fruity', 'structure', 'age', 'full']
['cola', 'cherries', 'cabernet', 'cherry', 'cranberry', 'beef', 'cellar', 'blackberries', 'pinot', 'zinfandel']
['berry', 'plum', 'herbal', 'feels', 'rubbery', 'earthy', 'oaky', 'baked', 'tomato', 'roasted']
```

Figure 14: Typical Characterisitics for 10 Clusters



```
['wood', 'aging', 'rich', 'tannins', 'structure', 'firm', 'age', 'ripe', 'structured', 'dense']
['blackberry', 'black', 'chocolate', 'dark', 'cassis', 'oak', 'blackberries', 'bodied', 'deep', 'currants']
['apple', 'lemon', 'white', 'lime', 'citrus', 'pear', 'grapefruit', 'green', 'fresh', 'riesling']
['fruity', 'crisp', 'acidity', 'attractive', 'soft', 'light', 'aftertaste', 'bright', 'fresh', 'red']
['tannins', 'cherry', 'underbrush', 'opens', 'clove', 'berry', 'licorice', 'black', 'delivers', 'skinned']
['cabernet', 'sauvignon', 'merlot', 'franc', 'blend', 'verdot', 'petit', 'malbec', 'syrah', 'blended']
['pinot', 'noir', 'silky', 'cola', 'cherries', 'cherry', 'raspberries', 'raspberry', 'next', 'sandalwood']
['berry', 'plum', 'herbal', 'feels', 'oaky', 'rubbery', 'earthy', 'tomato', 'baked', 'oak']
['strawberry', 'fruit', 'cranberry', 'forest', 'acids', 'raspberry', 'tart', 'bramble', 'petals', 'color']
['chardonnay', 'pineapple', 'tropical', 'honey', 'apricot', 'sweet', 'orange', 'blanc', 'vanilla', 'creamy']
```

4.2.2 Interpretation

The interpretation uses similar procedure of 3.1.3. However, notice this time, the descriptive labels become words extracted from text instead, and it can represent more of the wine itself instead of the wine background.

4.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a topic model that generates several topics based on the mixture of words of the text and attribute the data sample to the topics. Based on the Dirichlet distribution, the model will generate two matrices. One matrix consists the probability of selection a term for one topic. In another word, the topic can be viewed as a linear combination of words, which is the sum of the product of coefficients and words. The other matrix tells the probability of the topic for a sample document. The second matrix implies a clustering property of Latent Dirichlet Allocation as the topics can be viewed as clusters and this matrix gives the weight of cluster for each sample so that we can pick the topic with the largest probability as the cluster of the sample document.

4.3.1 Latent Dirichlet Allocation Application on Wine Data

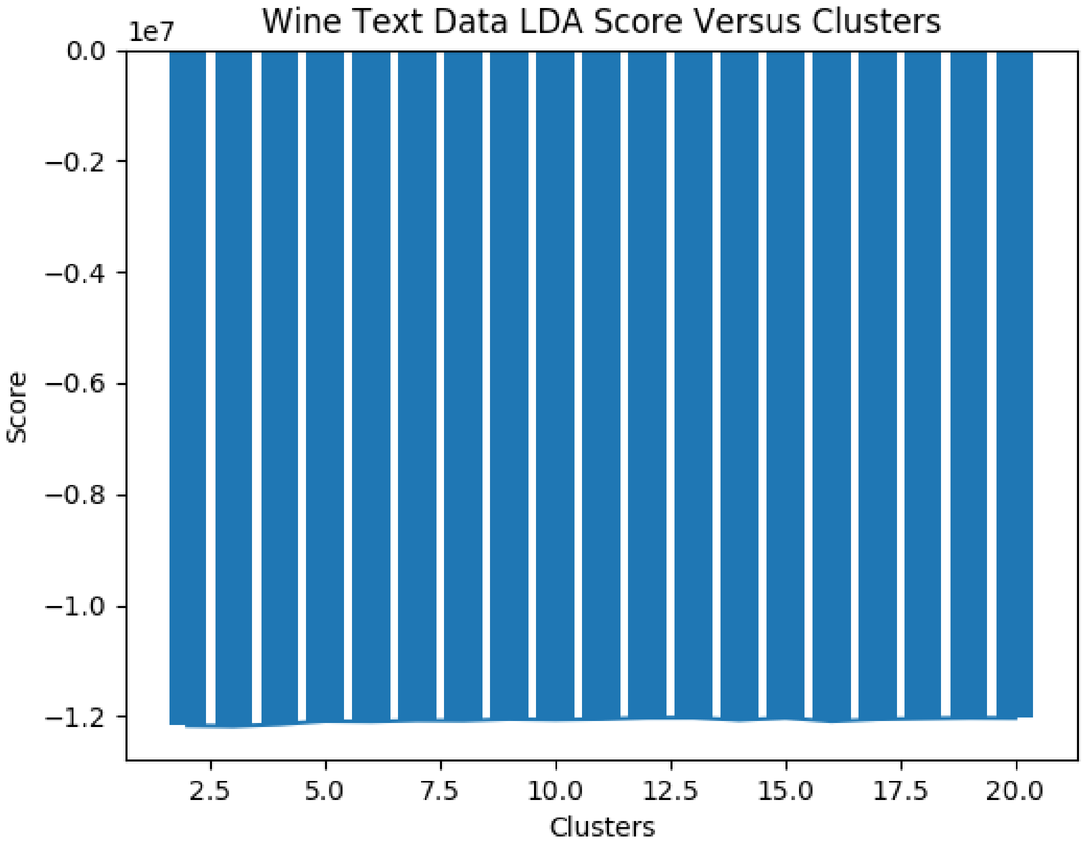
In this part, the main package we refer to is sklearn.decomposition.LatentDirichletAllocation. This package can implement Latent Dirichlet Allocation algorithm by picking n_components, which is the number of topics expected from the model. Table 4 shows number of data points in each cluster with Latent Dirichlet Allocation.

The quantitative evaluation metric we select for Latent Dirichlet Allocation is the score attribute within the package, which computes the log likelihood of the data matrix. While it is expected to select the maximized log likelihood parameter, we choose 5 and 10 topics to model to align with the above clustering models. Figure 15 shows the log likelihood scores for different topic number choices.

Table 4: Number of Data in Each Cluster

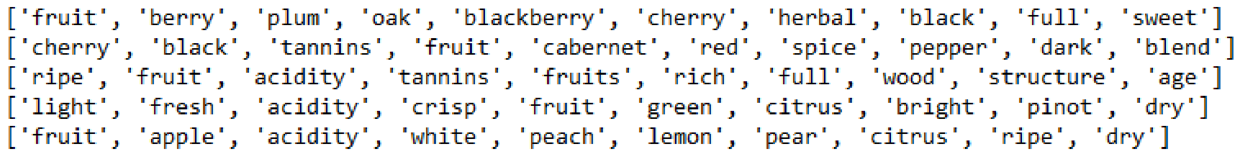
Cluster Index	Number of Data in Five Clusters	Number of Data in Ten Clusters
0	19399	16548
1	37338	9887
2	18020	10810
3	15072	13768
4	26698	7836
5		11855
6		13154
7		5339
8		17118
9		10212

Figure 15: Log Likelihood in Latent Dirichlet Allocation



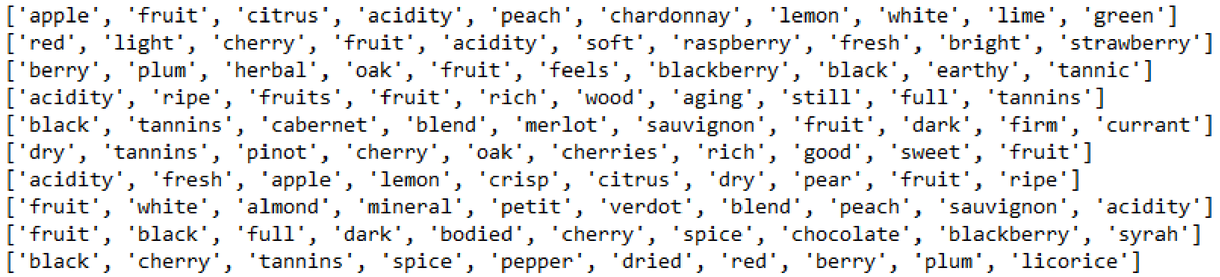
After building the model, for each wine sample, we pick the topic with the highest probability as the final cluster for the wine sample. Then we do the similar One-VS-All logistic regression process for it to get the characteristics labels composed of top 10 words. Figure 16 and 17 show the typical characteristics of different clusters with Latent Dirichlet Allocation for 5 cluster case and 10 cluster case.

Figure 16: Typical Characterisitics for 5 Clusters



```
['fruit', 'berry', 'plum', 'oak', 'blackberry', 'cherry', 'herbal', 'black', 'full', 'sweet']
['cherry', 'black', 'tannins', 'fruit', 'cabernet', 'red', 'spice', 'pepper', 'dark', 'blend']
['ripe', 'fruit', 'acidity', 'tannins', 'fruits', 'rich', 'full', 'wood', 'structure', 'age']
['light', 'fresh', 'acidity', 'crisp', 'fruit', 'green', 'citrus', 'bright', 'pinot', 'dry']
['fruit', 'apple', 'acidity', 'white', 'peach', 'lemon', 'pear', 'citrus', 'ripe', 'dry']
```

Figure 17: Typical Characterisitics for 5 Clusters



```
['apple', 'fruit', 'citrus', 'acidity', 'peach', 'chardonnay', 'lemon', 'white', 'lime', 'green']
['red', 'light', 'cherry', 'fruit', 'acidity', 'soft', 'raspberry', 'fresh', 'bright', 'strawberry']
['berry', 'plum', 'herbal', 'oak', 'fruit', 'feels', 'blackberry', 'black', 'earthy', 'tannic']
['acidity', 'ripe', 'fruits', 'fruit', 'rich', 'wood', 'aging', 'still', 'full', 'tannins']
['black', 'tannins', 'cabernet', 'blend', 'merlot', 'sauvignon', 'fruit', 'dark', 'firm', 'currant']
['dry', 'tannins', 'pinot', 'cherry', 'oak', 'cherries', 'rich', 'good', 'sweet', 'fruit']
['acidity', 'fresh', 'apple', 'lemon', 'crisp', 'citrus', 'dry', 'pear', 'fruit', 'ripe']
['fruit', 'white', 'almond', 'mineral', 'petit', 'verdot', 'blend', 'peach', 'sauvignon', 'acidity']
['fruit', 'black', 'full', 'dark', 'bodied', 'cherry', 'spice', 'chocolate', 'blackberry', 'syrah']
['black', 'cherry', 'tannins', 'spice', 'pepper', 'dried', 'red', 'berry', 'plum', 'licorice']
```

4.3.2 Interpretation

With the word feature, the interpretation is similar to Section 4.2.2

4.4 Comparison between Clustering with or without Natural Language Processing

While we definitely want to compare the clustering performance with or without natural language processing, two things need to be kept in mind for the mathematical evaluation metric, the Silhouette score. First is the Silhouette score comparison is only valid for the same dataset. Secondly, as long as the Silhouette score is above 0, even it is relatively low, it can only indicate that the cluster is low dense, but we cannot say it is bad cluster. The cluster accuracy should be referred to the qualitative metrics instead.

Mathematically speaking, in our case, clustering without natural language processing has higher Silhouette scores while clustering with natural language processing has lower Silhouette scores and the potential reason is discussed in Section 4.2.1. However, this only indicates that cluster without natural language processing is high dense while cluster with natural language processing is not very well separated.

The more important evaluation metric for clustering, especially in our case, is the qualitative metric. It is important to check whether the new created descriptive labels for each cluster appropriately fit wines within the cluster. With necessary domain knowledge, both clustering with normal features (without year), and clustering with natural language processing do well for most sample wines attributed in each cluster. The clustering with normal features (with year), however, makes less sense.

4.5 Prescriptive Analysis

Here we need to go back to the most important purposes in our case study: to pick representative wines from each cluster to fulfill the future expectations of customers. While mathematical clustering standard is one guideline, it is more significant to start from customers' perspective in real world business. That is why qualitative metrics, the new descriptive labels, are the center for our final decision. As a wine store, we expect most customers are not professionals about wines. To leave deep impressions as well as differentiate from other wine stores, we want our customers have a direct view of wines before they even taste them so that they can better pick what they want.

Following the above argument, we believe clustering with natural language processing is better in our case. Recall that clustering with common features from the first group are about background of wines, such as country and winery. These are used in most wine stores as standard description settings of wines, but they are too strict to show any property within the wine. However, features extracted from clustering with natural language processing can vividly exhibit the wine for its taste, smell, texture, etc. These descriptions can lead customers to instantly know the wine almost like they have tasted a drink sample. It may be argued customers will lose mystery of tasting wines if they pick wines based on these new descriptive labels. However, taste is a completely different thing from the perception based on these labels. We use these labels to describe wines to help customers choose what they like and avoid what they do not like. Tasting process will be another adventure starting from what customers like.

5 Conclusion

In this case study, we explore how clustering techniques can be applied to retail data. Specifically, we explore how different clustering algorithm, Kmeans (with or without natural language processing), hierarchical clustering and Latent Dirichlet Allocation can be trained and evaluated on the wine dataset to help our wine store to pick different types of representative wines for our potential customers.

We divide our data features into four groups while the first group is used for clustering without natural language processing, the second group is used for clustering with natural language processing and the third group is used for supporting the final wine selections from each cluster. We apply quantitative evaluation metrics including Silhouette Coefficient for Kmeans and Hierarchical clustering, and log likelihood for Latent Dirichlet Allocation, as well as qualitative evaluation metric, which combines the descriptive label of clusters and domain knowledge. Quantitatively, we propose that Silhouette score only indicates the dense and separable degree of the cluster instead of the accuracy and we can only compare the Silhouette score within the same dataset. Qualitatively, the descriptive labels are collected from further classification on the data to check whether wines in the clusters are good fit.

With our final business target to create clusters and pick wines that can be more friendly for potential customers who are relatively new to the wine, we prefer the clustering with natural language processing, which can create descriptive cluster labels as the characteristics of wines like taste, smell and texture rather than simple background of wine used in most wine stores. We then gather top centered wines with top points and price bins to further cluster the wines as our final selections of wines in our wine store to fulfill expectations of potential future customers.

Appendix

All of codes, graph visualizations, descriptive labels and cluster result csv files can be referred to the following GitHub page:

https://github.com/d1s0rder/Wine-Case-Study

Reference

Provost, F and Fawcett, T. (2013). *Data Science in Business*. California: O'Relly.

Molnar, C. (2019). *Interpretable Machine Learning*.

Srivastava, A. (2009). *Text mining : classification, clustering, and applications*. Florida: CRC Press.

Scikit-learn. *2.3 Clustering*. [Online] Available at:

https://scikit-learn.org/stable/modules/clustering.html [Accessed 17 May 2019]

Scikit-learn. *2.3 Working with Text Data*. [Online] Available at:

https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html [Accessed 17 May 2019]