Wine Review Prediction

CSE 258 Assignment 2

Shuwei Liang
A53271181
s1liang@eng.ucsd.edu

Qi Ma
A53263366
qima@eng.ucsd.edu

Zhuo Yue
A53275690
zyue@eng.ucsd.edu

1 INTRODUCTION

Nowadays, more and more people tend to shop online. People can quickly choose their products based on characteristics of the item, other reviewers' descriptions and rating. If we have a recommender system based on information listed above to provide good recommendations, we could effectively notify the customer with new products and make a potential sale. In this paper, we use the dataset on wine reviews to complete three predictive tasks: price prediction, sentimental analysis and classification prediction. With these prediction models, we can determine which wine is more likely to get a high rating, which characteristics of wines may result in higher price, which wines are similar to each other and should be divided into same class. Then we could further establish a recommender system based on these predictions and provide recommendations to customers.

To help us determine appropriate prediction models for future applications and recommender system, we compare the advantages and disadvantages of different models for three predictive tasks mentioned above. For price prediction, we build Random Forest Regression model, Logistic Regression model and Gradient Boosting Regression model with two different features: NLP word features and data category features. For sentiment analysis, we build some predictors based on NLP in five models: Naïve Bayes Classifier, Multinomial Native Bayes Classifier, Bernoulli Naïve Bayes Classifier, Logistic Regression model and Linear SVC. For classification prediction, we use XGBoost classifier with Tf-Idf feature to predict the grape variety for each wine. More details about our models will be introduced in Section 3 and Section 4, prediction results and further discussions will be provided in Section 6 and Section 7. In addition, we analyze the dataset that we used in details in Section 2 and introduce some related works based on the same dataset in Section 5**.**

2 DATA EXPLORATION

2.1 Overall

Our dataset has around 130,000 unique samples. It has 14 columns which describe a wine from almost every aspect including price, rating, description, origin and etc.

There are overall three types of variables, numerical, categorical and textual. 'Numerical' means the values of which are numbers, 'categorical' means the values of which are categories and 'textual' means the values of which are about description. Our team, after carefully analyzing the dataset, decided to divide the prediction into three parts, i.e., price prediction, sentimental analysis and classification prediction. Price prediction uses regression method to predict price of wines specifically and generally. Sentimental analysis uses text mining to predict reviewer's sentiment. And classification prediction part uses natural language processing to predict grape variety.

In the dataset exploration part, there are mainly below questions to discuss about.

- How do we preprocess our dataset to encode categorical features for price prediction?
- How can we use text-based 'description' to analyze the taster's sentiment?
- How are our variables distributed and whether they are appropriate to build the prediction model?
- How do we process dataset with NLP to predict grape variety and price prediction?

2.2 Distribution of Selected Features

To solve the above several questions, our team explored the data in the following aspects. Firstly, our team counts the place of origin of all wines. From the result, we can find that wines from the United States mainly make up the dataset (around 54000), followed by France, Italy, Spain, Portugal, Chile, Argentina and so on. Besides, in order to visualize the geographical distribution, we use 'plotly' to make choropleths shown in figure 2, which gives a global picture.

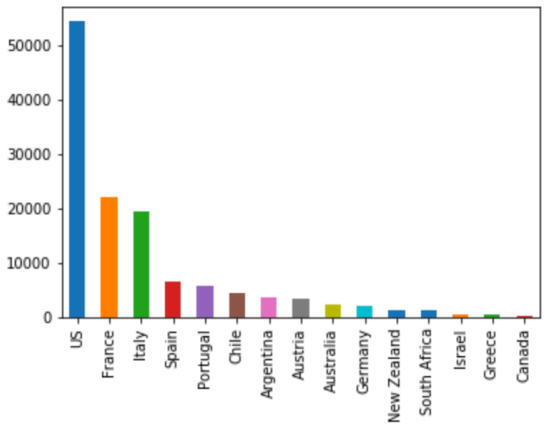

Figure 1: Place of Origin

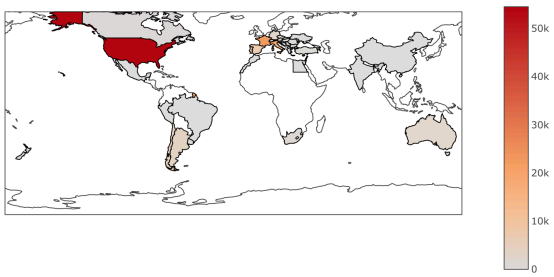

Figure 2: Place of Origin

Besides, in order to predict price, we also obtain the distribution of wine price shown in figure 3. From the plot, it is easy to recognize that the main price of wines focuses between 0 to 250. There is a long tail in this line plot whose counts converge to 0 as price increases.

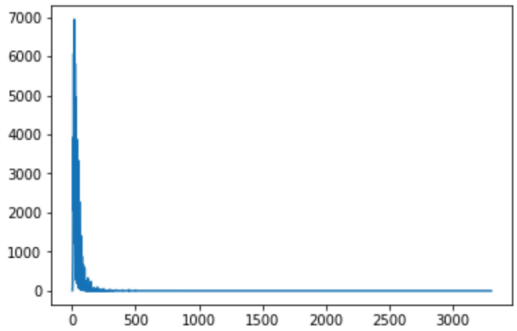

Figure 3: Correlation Heat map

Apparently, the price distributes unevenly so that we need to give up data whose price has only several counts in order to improve model accuracy. In this case, we also plot one kernel density estimation focusing on lower price in figure 4 below. From the figure, most prices frequently appear below 100 and the frequency approaches to 0 when price is higher than 150. Thus, in price prediction part, we choose to make use of data with price lower than 100.

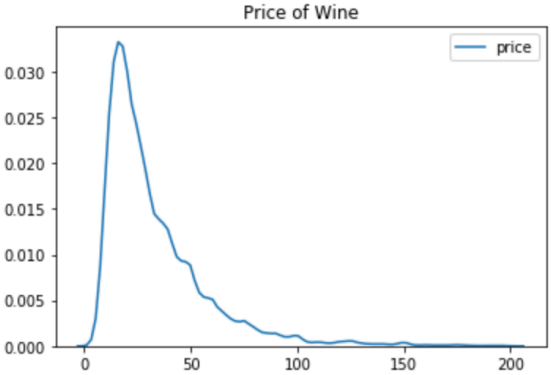

Figure 4: kernel density estimation

2.3 Numerical Correlation

Secondly, we draw a scatter plot of price and points(ratings) seen in figure 5 since there are only two numerical features in this dataset and the remaining is all classification and textual information. Obviously, as the price of wines goes up, ratings tend to increase. Also, from the side histogram, most ratings range from 82.5 to 92.5 corresponding to price extending from 0 to 50.

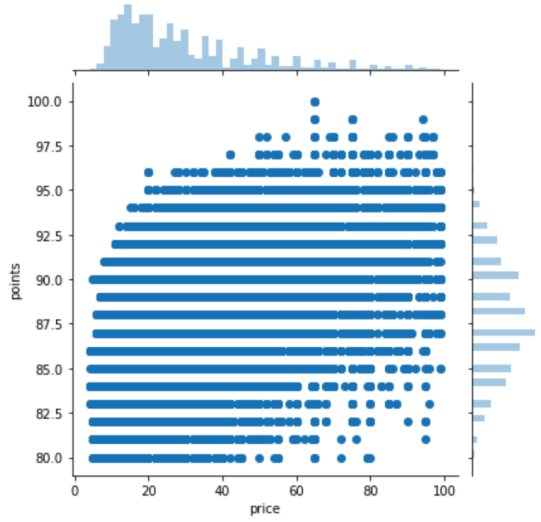

Figure 5: Correlation scatter plot

2.4 Taster

In this dataset, there are 19 tasters tasted and reviewed the wine. Our team plot the relationship of selected features including review count, price, ratings and tasters in below figures.

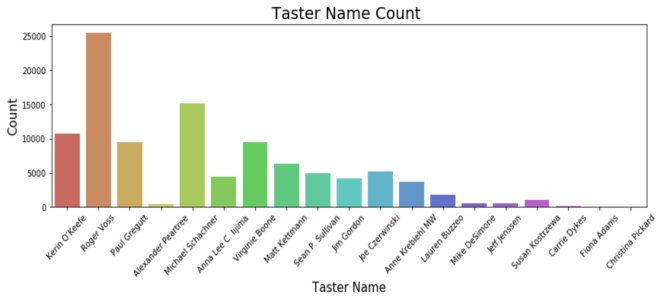

Figure 6: Review counts - Taster plot

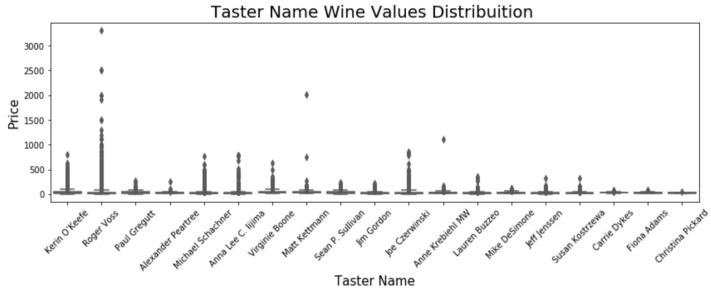

Figure 7: Wine price - Taster plot

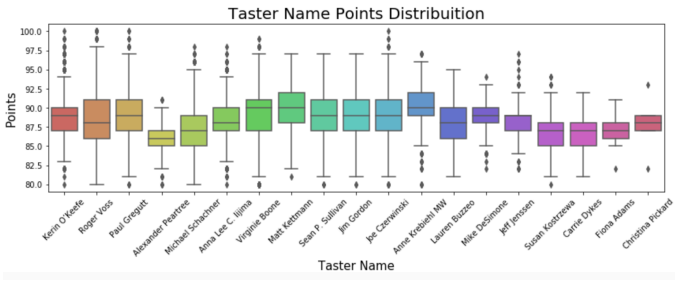

Figure 8: Points(ratings) – Taster

3 PREDICTIVE TASK

3.1 Price Prediction

In this task, our aim is to build a price prediction model by using Supervised Learning. We build a Random Forest Regression model with two different type of features and use Logistic Regression model, Gradient Boosting Regression model and a baseline model which takes average wine price as the price predictor for comparison.

3.1.1 Performance Evaluation

By using 'sklearn. model_ selection. Packet', we obtain our training set and test set, 33% of data for tests and 67% of data for training model.

We initially use MSE to evaluate our model performance, however, all our four models gave a pretty high MSE since there are some extraordinary expensive wines which lead to high standard division of our data and MSE is so sensitive to these outliers. Therefore, we turn to use MAE and R2-score to evaluate our model performance.

3.1.2 Data Preprocessing

When building and testing the prediction model, we found that due to the high standard division of wine price, the model that use the whole data set didn't perform well and data preprocessing was needed for a better prediction. We filter our dataset to only include wine prices which are under 100$. This reduces the standard division of wine price by dropping those extraordinary.

3.2 Sentiment Analysis

Sentiment analysis has applications across a range of industries, especially for some social media text feedback.

Apart from the grape type classification and price-point relation exploration, we consider what else interesting can we dig into. Since there are a lot of text-form "description" in the dataset, we decide to use those content of descriptions, to do the sentiment analysis, by NLP. We attempt to model a baseline, then try to find better models to optimize our result.

3.2.1 Performance Evaluation

The simple baseline is the Naïve Bayes Classifier. And the performance of the predictive model and comparison is mainly evaluated by accuracy, which is related to TP (True positives), TN (True negatives), FP (False positives), FN (False Negatives). Then we consider more complicated Naïve Bayes models and other model with some possibility for better accuracy performance.

3.2.2 Data Preprocessing

Before model building, to keep our reviews clean, the lowercase for all text, extra space, and punctuation stripping has already been completed by 'sklearn package Count Vectorizer'.

After that we need to introduce binary quality classification to partition the text part based on the points they get. Digging into our data, it is not hard to find that the points of all the descriptions range from 80 to 100, and the overall mean is around 87. Hence for binary classification, we start with defining the top quartile is "good" (the value is set to 1) and the bottom quartile (the value is set to 0) is "bad".

We then split our data into training (with size 33177) and testing part, assign 25% overall to training data, the left for testing.

3.3 Classification Prediction

The task here is to predict the grape variety for each wine. For each wine in the test set, the value of the grape variable based on other properties and wine description.

Since the value of the grape variable is a label, the task should be done with classification. Also, since features in dataset are mostly textual variables except for points(ratings) and price, our team decided to use natural language processing to solve this problem.

3.3.1 Performance Evaluation

One baseline is logistic regression without any data preprocessing. It applies One-Versus-All thinking when clssifying which label current data belongs to.

The performance of the predictive model would be evaluated using accuracy. We give each grape variety a tag and compare predictive results and their real variety.

To assess the validity of the model's predictions, the predictions will be compared with baseline modeled with simple methods.

3.3.2 Data Preprocessing

Since there are lots of uninterested features and null values in the dataset, we have to drop them before modeling. Features in which we are not interested are 'Unnamed: 0', 'designation', 'points', 'region_2'. Apart from that, there are also too many samples with duplicate description which we need to remove. More importantly, capitalization should be ignored which means we should transform all words into lowercase. Furthermore, when counting all grapes variety, there are lots of grape varieties which just appear several times. In this case, we need to delete them from dataset, because we will not be able to train and test on one sample. And many wines are made up of blended grapes, a mixture of different grapes, which contain "blend" and other key words in their names.

It's very hard to predict these kinds of wines with blended grapes and may affect accuracy. In our team's model, we chose to remove this case from consideration. Based on above analysis, data before and after processing is shown in below pictures.

Unnamed: 0	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	variet
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	Whit Blen
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portugues Re
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette Valley)	Pinot Gri
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN	St. Julian 2013 Reserve Late Harvest Riesling ...	Rieslin

Figure 9: Raw data

malvoisie	1
moristel	1
forcalla	1
tinta barroca	1
magliocco	1
bukettraube	1
aidani	1
shiraz-merlot	1
malbec blend	1
garnacha blend	1
groppello	1
franconia	1
carnelian	1
romorantin	1
rufete	1
teroldego rotaliano	1
viura-sauvignon blanc	1
jacquez	1
pinot grigio-chardonnay	1
rebula	1
irsai oliver	1
malvazija	1
garnacha-cabernet	1
cabernet sauvignon and tinta roriz	1
freisa	1
grignolino	1
macabeo-gewurztraminer	1
fruburgunder	1
dafni	1
saperavi-merlot	1

Figure 10: Infrequent grape varieties

	country	description	price	province	region_1	taster_name	taster_twitter_handle	title	variety	winery
2	us	tart and snappy, the flavors of lime flesh and...	14.0	oregon	willamette valley	paul gregutt	@paulgwine	rainstorm 2013 pinot gris (willamette valley)	pinot gris	rainstorm
3	us	pineapple rind, lemon pith and orange blossom ...	13.0	michigan	lake michigan shore	alexander peartree	NaN	st. julian 2013 reserve late harvest riesling ...	riesling	st. julian
4	us	much like the regular bottling from 2012, this...	65.0	oregon	willamette valley	paul gregutt	@paulgwine	sweet cheeks 2012 vintner's reserve wild child...	pinot noir	sweet cheeks
5	spain	blackberry and raspberry aromas show a typical...	15.0	northern spain	navarra	michael schachner	@wineschach	tandem 2011 ars in vitro tempranillo-merlot (n...	tempranillo-merlot	tandem
6	italy	here's a bright, informal red that opens with ...	16.0	sicily & sardinia	vittoria	kerin o'keefe	@kerinokeefe	terre di giurfo 2013 belsito frappato (vittoria)	frappato	terre di giurfo

Figure 11: Processed data

Since wines are from different places of origin, different countries have different names for the same grape. In this case, we should categorize them in the same way. For example ‘trebbiano’ and ‘ugni blanc’ is the same grape. Therefore, we make a mapping between certain names and their common name to improve accuracy of our model. After above processing, we have a total of 639 grape varieties in the end.

After above processing, we have 65 grape varieties in the end.

4 MODEL

4.1 Price Prediction

4.1.1 Features

We build our models with word features and category features.

Word features is the features for every word in the reviewer’s description for a wine. To obtain word features, we remove stop words in description for each review and calculating the term frequency of each unique words.

Category features are features that we can obtain directly from data set. For each review, we have eight different characteristics: 'country', 'designation', 'province', 'region_1', 'region_2', 'taster_ name', 'variety', ‘winery’. Each characteristic contains multiple categories, we encode these category features into real number by sklearn. preprocessing. LabelEncoder.

4.1.2 Random Forest Regression Model

We build our model use Random Forest Regression with word features and category features.

Random Forest Regression is a supervised learning algorithm that operate by constructing a multitude of decision trees and then combing these decision trees by using bootstrap aggregation to form a single consensus tree. Compared to models which use Gradient Boosting Regression and Logistic Regression, Random Forest Regression model is not easy to have an overfitting problem and has a good generalization ability. However, Random Forest Regression model requires heavier computational resources and has a significant long training time.

We initially performed our tests with the parameters of Random Forest Regression: [n

estimators = 10, min samples split = 2, bootstrap = True, max depth = None, min samples leaf = None] which gave us an MSE of 10.90 with word features and 9.12 with category features. This result is substantially better than baseline performance. With enough time, we can continue optimize our model parameter and engage a much better performance.

Table-1: Model Performance with category features

	MAE	R2-score
Random Forest	**9.12**	**0.54**
Logistic Regression	13.17	0.18
Gradient Boosting	10.18	0.50
Baseline	15.01	-5.73

Table-2: Model Performance with word features

	MAE	R2-score
Random Forest	**10.90**	**0.32**
Logistic Regression	11.92	0.13
Gradient Boosting	12.56	0.25
Baseline	15.01	-5.73

4.2 Sentiment Analysis

At first, we applied Logistic Regression model to train our dataset, and we get the prediction accuracy as 0.9696175. So that we can get the most five positively associated words and the most five negatively ones as shown in Table-3.

Tabel-3: Most five positive and negative words

Positive words		**Negative words**	
sample	4.11366132	simple	-2.77744029
beautifully	3.20801896	lacks	-2.67674258
beautiful	2.79809693	virginia	-1.94905626
opulent	2.66087015	short	-1.84108276
delicious	2.63335301	tad	-1.82714320

And then we consider other classifiers to see if they can contribute to better classifying performance.

We define those descriptions with more than 90 points as the top quartile, and those with points under 87 as the bottom quartile. As many words just show up quite limited times like 1, we would like to take the most 6000 common words into consideration.

a) Naïve Bayes Classifier:

Naïve Bayes method is a supervised leaning algorithm based on the 'naive' assumption of conditional independence.

Since $P(y|x_1, \dots, x_n)$ is constant given the input, we can use the following classification rule:

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

$$\Downarrow$$

$$\hat{y} = arg \max_y P(y) \prod_{i}^{n} P(x_i|y)$$

And the final accuracy percent is 91.57%.

We take the 15 most informative features (words) and get their positive/negative level.

```
Most Informative Features
                everyday = True              neg : pos    =     91.7 : 1.0
                  sugary = True              neg : pos    =     74.6 : 1.0
               easygoing = True              neg : pos    =     65.4 : 1.0
                stunning = True              pos : neg    =     40.9 : 1.0
                  superb = True              pos : neg    =     40.2 : 1.0
           easy-drinking = True              neg : pos    =     40.0 : 1.0
            impressively = True              pos : neg    =     39.5 : 1.0
                  simple = True              neg : pos    =     39.5 : 1.0
                  stalky = True              neg : pos    =     38.2 : 1.0
                    dull = True              neg : pos    =     34.5 : 1.0
             beautifully = True              pos : neg    =     32.8 : 1.0
                   quick = True              neg : pos    =     31.2 : 1.0
              impressive = True              pos : neg    =     31.1 : 1.0
                     odd = True              neg : pos    =     28.6 : 1.0
              incredibly = True              pos : neg    =     28.0 : 1.0
```

Figure 12: Fifteen most informative words and their positive/negative level

b) Multinomial Native Bayes Classifier:

MultinomialNB implements the Naïve Bayes for multinomially distributed data, which is also a classic method in text classification. The distribution is parametrized by vectors $\theta_y = (\theta_{y1}, \dots, \theta_{yn})$ for $P(x_i|y)$ of feature i appearing in a sample belonging to class y.

And the parameters θ_y is estimated by a smoothed version of maximum likelihood, for instance, relative frequency counting as follows:

$$\widehat{\theta_{y_i}} = \frac{N_{y_i} + \alpha}{N_y + \alpha n}$$

where $N_{y_i} = \sum_{x \in T} x_i$ is the number of times feature i appears in a sample of class y in the training set T. And $N_y = \sum_{i=1}^{n} N_{y_i}$ is the total count of all features of class y.

The final accuracy for MultinomialNB is 90.024657.

c) Bernoulli Naïve Bayes Classifier:

BernoulliNB train data in terms of multivariate Bernoulli distributions. The decision rule for this method is based on the following formula:

$$P(x_i|y) = P(i|y)x_i + (1 - P(i|y))(1 - x_i)$$

Which differs from multinomialNB's rule in that it explicitly penalizes the non-occurance of a feature i that is an indicator for class y.

The final accuracy for Bernoulli Naïve Bayes is 91.5435447.

d) Logistic Regression model:

We managed to use Logistic Regression model which we've learned in class as well, to make an optimized decision boundary on the "good" and "bad" word dimensions.

The final accuracy for Logistic Regression is 91.918335.

e) Linear SVC method:

Our data are not really linearly separable, we exploit the soft margin.
The following value is expected to be minimized:

$$[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(\vec{w} \cdot \vec{x_i} - b))] + \lambda||\vec{w}||^2$$

The final accuracy is 90.8649768.

Then we apply the VoteClassifier to get the model (Logistic Regression) with the best performance.

4.3 Classification Prediction

4.3.1 Stop Words

In order to improve accuracy further, we also need to filter out stop words like '.', ',', '`', "", """, '!', ';' , 'wine', 'fruit', 'flavor', 'aromas', 'palate' and words appearing in countries, province, region_1, taster_name, taster_twitter_handle, title, winery from description for they don't make sense for grape variety prediction. By the way, common stop words should also be removed.

```
'my',
'myself',
'we',
'our',
'ours',
'ourselves',
'you',
"you're",
"you've",
"you'll",
"you'd",
'your',
'yours',
'yourself',
'yourselves',
'he',
'him',
'his',
```

Figure 13: Part of filtered stop words

4.3.2 Tf-idf Vectorization

To transform the text-based description column into a number-based object, we use term frequency-inverse document frequency (tf-idf) vectorization. First it counts the occurance of a word in one sample and then down-weighs it with the occurance of the same word over all the samples. It repeats this process for each word and outputs a vector of numbers, where each number represents a word. We decided to analyze the text word by word (1-grams), but choosing 2-grams or n-grams is also allowed, meaning selecting the first n words to represent the first element of the vector.

4.3.3 XGBoost Classifier

XGBoost is a version of gradient boosted decision tree classifier. In boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree. These subsequent trees are called base or weak learners. Each of these weak learners contributes some vital information for prediction, enabling the boosting technique to produce a strong learner by effectively combining these weak learners. The power of XGBoost lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage. We predefine certain input arguments that control the performance of the classifier and use one argument 'subproblem' to optimize.

4.3.4 Grid Search

Finally, we defined the grid search object and used pipe variable serving as an estimator which take into grid parameter. Cross validation fold is defined as 3 although in

theory the more the better. Setting the verbose argument to 0 prevents the algorithm to print our messages to the console. After the grid search object construction, we can use the fit method to train our model.

5 LITERATURE

5.1 Similar Dataset

The dataset comes from the website WineEnthusiastic (a multichannel marketer of a growing line of wine-andspirits-related products):*[https://www.winemag.com/?s=&drink_type=wine]*, during the week of June 15th, 2017. There are four versions revised until the most updated one, collected on November 24th, 2017. And we mainly parse the data from "winemag-data-130k-v2.csv", which contains 130k wine reviews with variety, location, winery, price, point, and description. More specifically, Natural Language Processing is applied to process the "description" related text and the "points".

Since wine reviews analysis is a really popular and valuable topic in Data Science and Machine Learning, besides the dataset we use from WineEnthusiastic, there are a bunch of similar datasets online, for instance, Red Wine Quality, [https://archive.ics.uci.edu/ml/datasets/wine+quality].

Amazon Reviews for sentiment Analysis, [https://www.kaggle.com/bittlingmayer/amazonreviews/home], and so on. These former similar analyses helped us a lot in this assignment.

5.2 Other Method

5.2.1 Price Prediction task

For the task of price prediction, there are some similar works but not many. Basically, these works use Logistic Regression, Lasso Regression, SVM model with Natural Language Processing (word features) or categorical features. Compare with these works, our models do have a similar result but a little better. Since the high standard division of data set and MSE is so sensitive to outliers, MSE of these price prediction models are extremely high. For example, after remove those more expensive wines (wine price > 1000), MSE of the model which uses Lasso Regression model is 778.95 and MSE of our Random Forest Regression model is 304.52. In addition, there are some works that build model with both words and categorical features, which can be more complex than our model but also give very similar result.

5.2.2 Sentiment Analysis

We also came across a very useful open source python library which performs well on sentiment analysis, name VADER (Valence Aware Dictionary and sEntiment Reasoner). With VADER you can implement sentiment classification very quickly even if you don't have positive and negative text examples to train a classifier.

VADER produces four sentiment metrics from these word ratings: positive, Neutral, Negative, Compound. In other words, we just simply split the dataset into four categories, each with 1299971 elements.

After importing VADER, we load the SentimentIntensityAnalyzer() object from the package and then use polarity_scores() method to get the sentiment metrics for a piece of text. Taken the first 20 lines of data, we can get the following results:

Table-4: The four scores of the first 20 lines of data

	Comp	neg	neu	pos
0	0.1531	0.000	0.935	0.065
1	0.6486	0.000	0.868	0.132
2	-0.1280	0.053	0.947	0.000
3	0.3400	0.000	0.926	0.0740
4	0.8176	0.000	0.805	0.195ßß

Since the compound score is the aggregated score, we can apply it and find the relationship between that and 'points', and 'price'.

- For points:

points	
80	-0.009855
81	0.043990
82	0.102415
83	0.190330
84	0.333103
85	0.406618
86	0.453449
87	0.492951
88	0.515306
89	0.530699
90	0.580637
91	0.600757
92	0.630100
93	0.661940
94	0.693255
95	0.745709
96	0.737685
97	0.796876
98	0.862329
99	0.881230
100	0.889621

Figure 14: The compound scores under each point

- For prices:

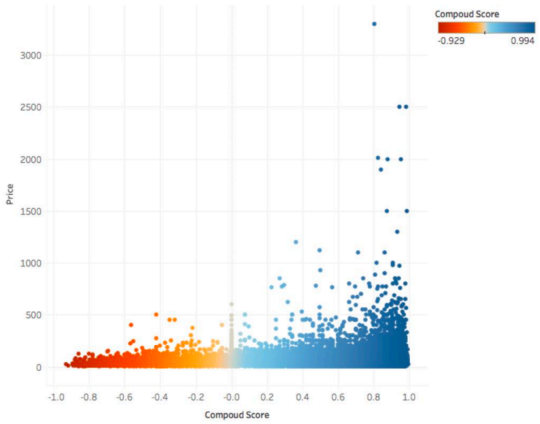

Figure 15: The distribution of compound scores of different prices

From the graph above, we can conclude that:

- wine with price range from 0 to 500 don't have strong association with its quality.
- but for those wine with price higher than 500, they all bear positive sentiment scores.

5.2.3 Classification Prediction

1) POS tagging and Lemma tokenization

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'. In the description text a word might appear in different forms while actually representing the same word, like good and best. To reduce noise, we try to find the basic form (lemma) of each word in the text. We can rely on the nltk package to achieve this. But in our model, we cannot optimize it in this way.

2) Introduction of New Features

During blind tasting, wine experts are allowed to look and expect the wine, but the wine label is not exposed to them. Therefore, introducing the color feature of the selected grapes can be accepted which can further improve accuracy of the model. Since we don't have this dataset, this method is not applied to our model.

6 RESULTS

Since we have three parts for predictive task, price prediction, sentiment analysis, and classification of grape type. We will discuss the results of these three parts respectively.

6.1 Price Prediction

At first, we tried to parse all the samples in the dataset by MSE. However, the MSE shows a relatively poor performance since the price of the samples are separated a lot, so we may change the measurements into MAE and R2-score, as well as strip some samples with high prices but possess little show times.

Taking word features and category feature into account, we compare the performance of four models: Random Forest, Logistic Regression, Gradient Boosting, and also the baseline model, as for both words feature s and category features, Random Forest is the best performer.

6.2 Sentiment Analysis

The baseline model for sentiment analysis is Logistic Regression. We use accuracy as the metric to evaluate the model performance in terms of the samples' 'description'. We looked up for several methods to optimize our model, but since the baseline model has already achieved rather high accuracy, which is 91.918335, there's not so much significant improvement when applying new models. Since Logistic Regression is also a very simple and straight-forward way, we may directly use Logistic Regression model.

Apart from the typical models hackneyed in Natural Language Processing, we found a new open source library VADER, which provides more convenience for us to grasp the correlation between points and compound score, and prices and compound score.

6.3 Classification of Grape Type

6.3.1 Baseline Model

The baseline model using logistic regression without any data processing and other advanced model-building method can achieve 0.151 accuracy on train set and test set.

6.3.2 NLP and XGBoost model

After tuning parameters, the model can achieve 0.583 accuracy. The part of specific results of each grape are as follows.

Classification report:

	precision	recall	f1-score	support
aglianico	0.09	0.57	0.15	14
albarino	0.50	1.00	0.67	17
albariño	0.19	0.50	0.27	54
barbera	0.13	0.43	0.20	56
blaufränkisch	0.19	0.71	0.30	17
cabernet franc	0.33	0.72	0.45	178
cabernet sauvignon	0.72	0.59	0.65	3561
cabernet sauvignon-merlot	0.00	0.00	0.00	5
cabernet sauvignon-syrah	0.07	0.27	0.12	11
carmenère	0.23	0.40	0.29	98
chardonnay	0.91	0.53	0.67	5814
chenin blanc	0.25	0.75	0.37	53
corvina, rondinella, molinara	0.47	0.70	0.56	100
dolcetto	0.08	0.36	0.13	11
fiano	0.15	0.38	0.22	16
g-s-m	0.05	0.30	0.09	10

Figure 16: Part of results

7 CONCLUSION

Whenever we make some prediction or classification, we do need to collect a bunch of usable data. And then we will do some data exploratory analysis to deeply understand the components of the dataset, such as the head, the size, and the correlation between different columns of the given samples, etc. We may need to learn something about the wines' origin distribution, the price distribution, also like the correlation between 'points' and 'prices'. Afterwards, we can determine what is the potential value for us to predict, is that an available way if we attempt to find the relationship between 'description' and 'prices'? Should it's a better predictive topic if we cover 'points' and 'region' as well?

After data mining part, we try to do something like predictor, classifier, or recommender. As we had no idea which model would definitely perform better, we tried several models and choose some metrics like MAE and R2-score for price prediction, Compound scores and accuracy for sentiment analysis, accuracy for classification of grape types. Ultimately choose the best performer.

In this assignment, each member in the team contribute a lot and we discussed a lot on task implementing, topic choosing, and model selection. The process we attempt to find a better model, is the process that we overcome obstacles. We learned a lot by self-learning and team work, and finally gained deeper understanding on data mining and recommender system building.

8 REFERENCES

- http://t-redactyl.io/blog/2017/04/using-vader-to-handle-sentiment-analysis-with-social-media-text.html
- http://cs229.stanford.edu/proj2017/final-reports/5244216.pdf
- https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a
- http://webdropin.com/wordpress99/projects/wine-exploration-in-python/