Springboard Data Science Intensive Capstone Project
-
Predicting the Quality of Wine by Using Physicochemical Properties

Thanawit Oonseangjan (Tito)

April 31, 2018

Contents

1. Introduction

Problem:

What makes good wine good? I always have this question in my mind when having a glass with friends. Normally, I select my favorite bottle from beautiful label, deepness of the bottom and rating website in the internet. However, it would be much more legit and informative to investigate this problem through physicochemical properties which are the fundamental of flavor and taste of wine. After eliminating all the psychological biases such as price, popularity, year, branding, packaging etc., we can ask something like what physicochemical properties will affect the quality of wine? Is there any magic mixture of acid and sugar that makes wine guru give a thump up?

Client:

By digging deeper, it can be beneficial to both supply side and demand side. For supply side, this is obvious that the wine makers want to brew the best wine for market. Getting known more about physicochemical properties will give them more resource to achieve their goal. In demand side, wine customers will learn more about physicochemical properties and how they affect wine quality. They will have more scientific indicators for selecting their own favorite bottle. In this particular research, we will focus more on supply side because we won't see it

every day that some random guys go to the liquor store with testing equipment to measure residual sugar or pH level of wine before buying them. On the other hand, the one who is fully equipped with all the expensive equipment is the brew house. By making a better wine or higher rating, the brew house will earn higher profit. Therefore, in this sense of getting physicochemical properties, we will target at the supply side. How do we improve the rating? What should be added in wine to make it more appearing to the tester?

2. Data Acquisition and Cleaning

Data set:

The data set is from Kaggle website. This datasets is related to red variants of the Portuguese "Vinho Verde" wine. For more details, consult the reference [Cortez et al., 2009]. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.). Input variables (based on physicochemical tests): 1 - fixed acidity 2 - volatile acidity 3 - citric acid 4 - residual sugar 5 - chlorides 6 - free sulfur dioxide 7 - total sulfur dioxide 8 - density 9 - pH 10 - sulphates 11 - alcohol Output variable (based on sensory data): 12 - quality (score between 0 and 10). For more information, check

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Relevant publication

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Potential data:

For the similar data set in Kaggle, there are some such as

https://www.kaggle.com/akram24/wine-pca/data

https://www.kaggle.com/zynicide/wine-reviews

https://www.kaggle.com/residentmario/most-common-wine-scores/data

https://www.kaggle.com/danielpanizzo/wine-quality.

As we can see from the data, we have some logistic issue with our data. There is no label nor name nor year of the wine in our data set. . It would be much more interesting, if we can investigate more in the relationship between physicochemical properties and price or year, so we might find answer to some myths in winery industry.

Cleaning Process:

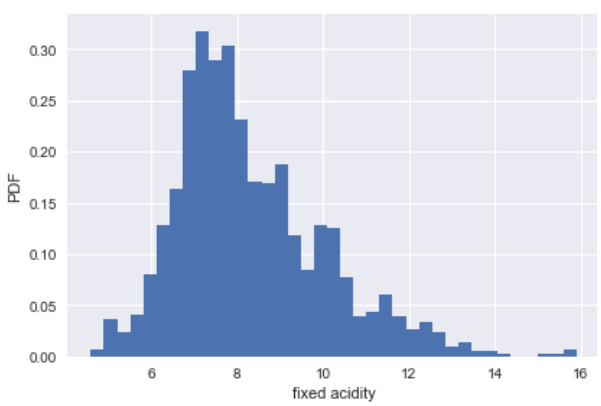
First, we checked the missing value in the data first to make sure that there is no anything missing. After checking, our data looks complete with legit type of information (float 64) which is the type of storing number in python. However, there is some duplicated rows in our data. For purpose of convenience and meaningful result, we decide to delete the duplicated rows since the wine with exactly the same physicochemical properties has a higher chance that it comes from the same origin or even the same tank. In this study, we want to study more about how physicochemical properties affect wine quality, so there is no reason to keep the duplicated rows in our data. Then, we looked for outlier by checking basic statistic parameters such as mean, standard deviation, max, min, etc. The result looks legit. Therefore, we are ready to use this data set in the further process.

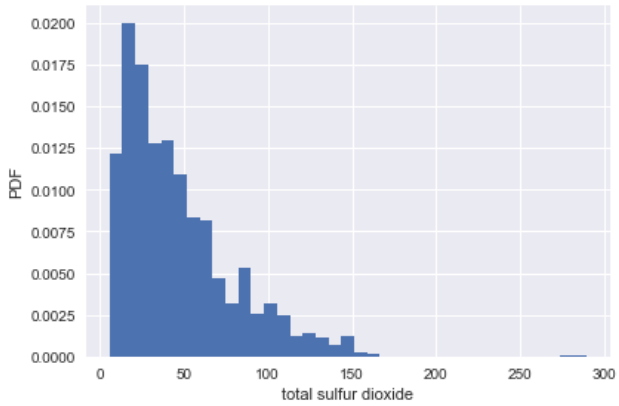
3. Data Exploration

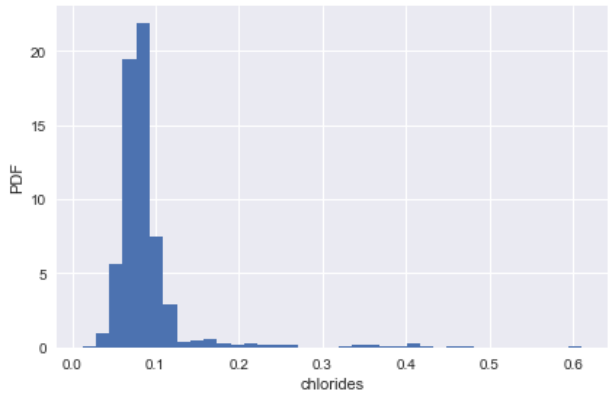
Plot histogram to spot outlier virtually:

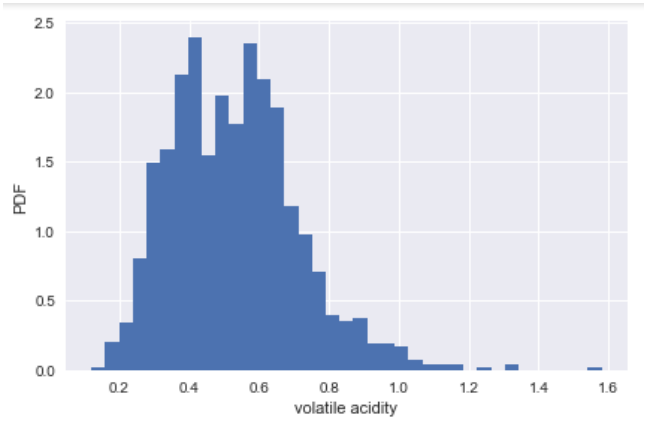
First, the psychochemical properties of wine can be very extreme. For example, total sulfur dioxide level of 270 is still possible; even though the level is almost triple the mean of the feature. However, they are still acceptable according to the meaning of parameters. Second, by

virtually checking, most features are not normally distributed. We investigate this by plotting histogram.



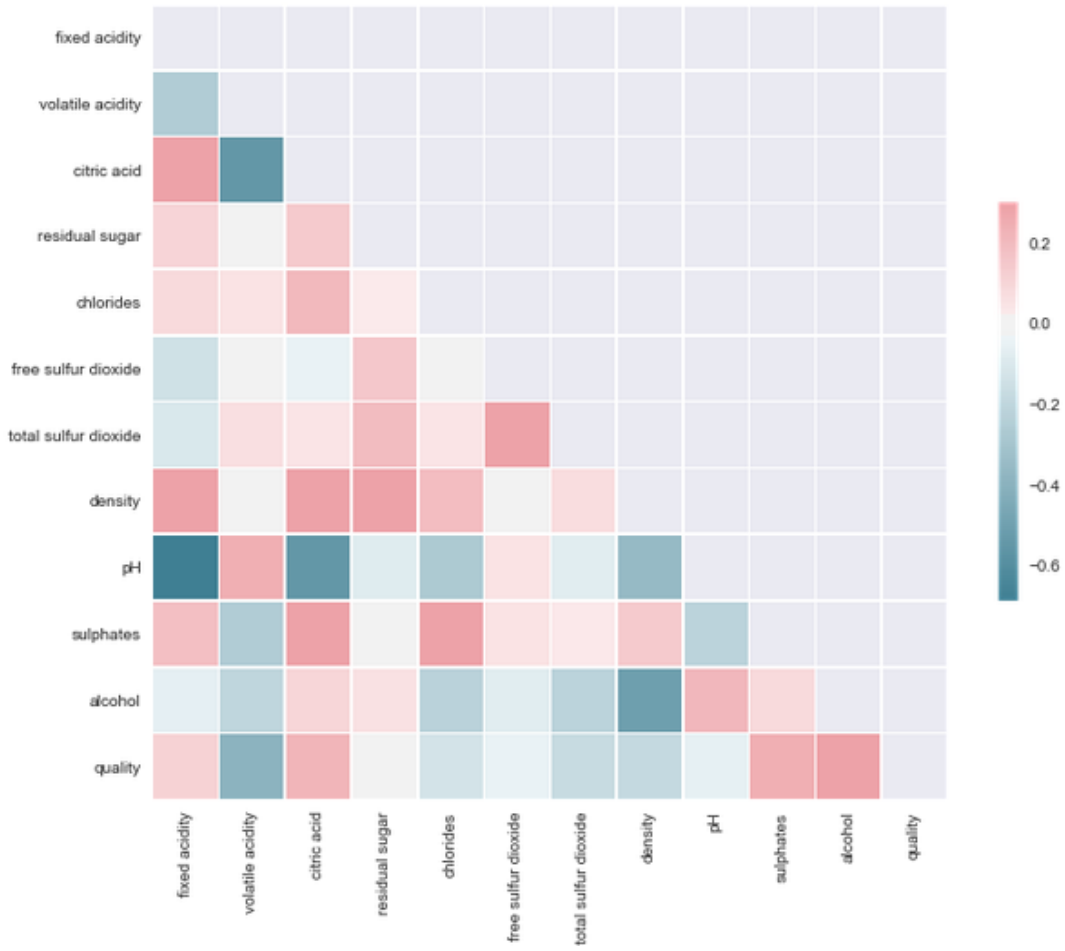






Correlation between variable:

Next, after trying to apply some EDA process to our data, these are the interesting findings. Variables that have correlation more than 0.15 are volatile acidity, citric acid, total sulfur dioxide, density, sulphates, and alcohol.

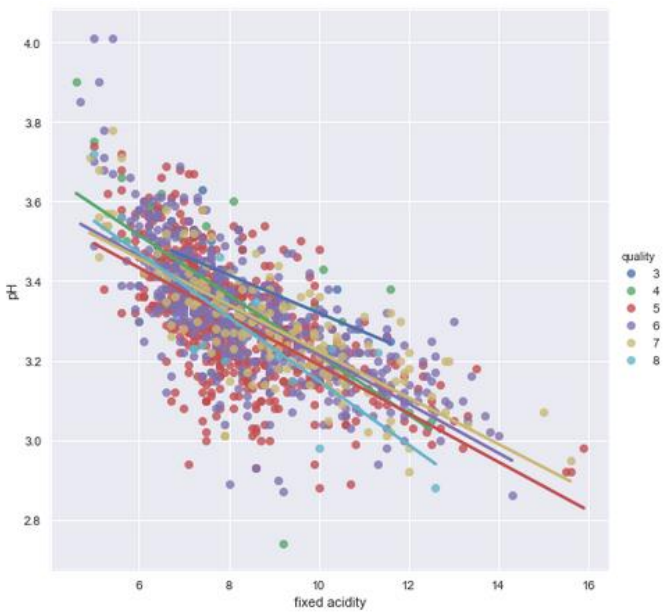


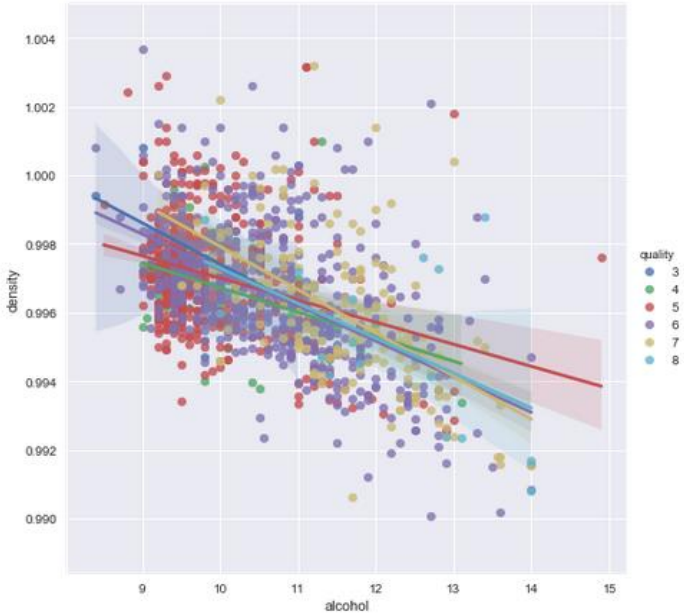
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1.000000	-0.255124	0.667437	0.111025	0.085886	-0.140580	-0.103777	0.670195	-0.686685	0.190269	-0.061596	0.119024
volatile acidity	-0.255124	1.000000	-0.551248	-0.002449	0.055154	-0.020945	0.071701	0.023943	0.247111	-0.256948	-0.197812	-0.395214
citric acid	0.667437	-0.551248	1.000000	0.143892	0.210195	-0.048004	0.047358	0.357962	-0.550310	0.326062	0.105108	0.228057
residual sugar	0.111025	-0.002449	0.143892	1.000000	0.026656	0.160527	0.201038	0.324522	-0.083143	-0.011837	0.063281	0.013640
chlorides	0.085886	0.055154	0.210195	0.026656	1.000000	0.000749	0.045773	0.193592	-0.270893	0.394557	-0.223824	-0.130988
free sulfur dioxide	-0.140580	-0.020945	-0.048004	0.160527	0.000749	1.000000	0.667246	-0.018071	0.056631	0.054126	-0.080125	-0.050463
total sulfur dioxide	-0.103777	0.071701	0.047358	0.201038	0.045773	0.667246	1.000000	0.078141	-0.079257	0.035291	-0.217829	-0.177855
density	0.670195	0.023943	0.357962	0.324522	0.193592	-0.018071	0.078141	1.000000	-0.355617	0.146036	-0.504995	-0.184252
pH	-0.686685	0.247111	-0.550310	-0.083143	-0.270893	0.056631	-0.079257	-0.355617	1.000000	-0.214134	0.213418	-0.055245
sulphates	0.190269	-0.256948	0.326062	-0.011837	0.394557	0.054126	0.035291	0.146036	-0.214134	1.000000	0.091621	0.248835
alcohol	-0.061596	-0.197812	0.105108	0.063281	-0.223824	-0.080125	-0.217829	-0.504995	0.213418	0.091621	1.000000	0.480343
quality	0.119024	-0.395214	0.228057	0.013640	-0.130988	-0.050463	-0.177855	-0.184252	-0.055245	0.248835	0.480343	1.000000

For convenient, we will categorize these features to have a direct effect to wine quality. However, for other variables, the correlation between them and wine quality is weak, but they have a relationship with the direct effect features. For example, pH is strongly related with acid properties (0.67, 0.25,-0.55). Also, density is strongly related with alcohol (-0.5). Hence, we will categorize our features into two categories direct features and indirect features.

Behavior of the direct effect features and the indirect effect features based on wine quality:

Then, we investigate more in the behavior of direct features and indirect features for each rating of wine. After plotting scatter plot, for each quality level, the behavior of feature is the same among different features.

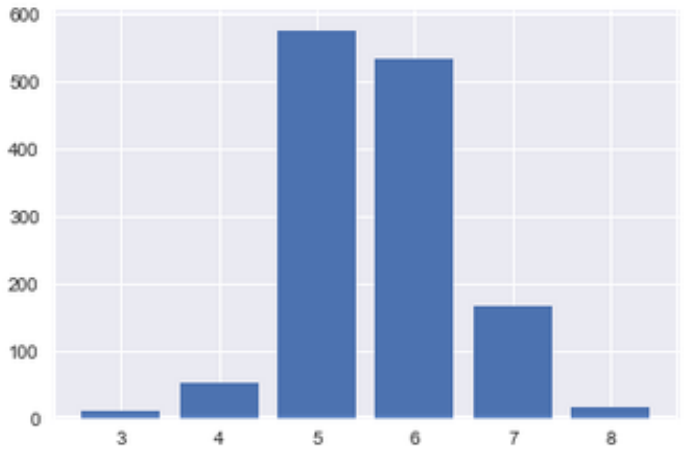




According to the graphs, for each quality level, the behavior of feature is the same among different features. For example, the slope of correlation between fixed acidity and pH is negative no matter what quality level it is. The same behavior happens in other scatter plot that we illustrated too.

Crate new rating variable:

Therefore, to make it easier for applying logistic regression to our data, we decide to create other variable called rating. The wine quality will be split into two categories (0 and 1). Zero parameter represents lower quality of wine range from 3 to 5. One parameter represents 6 to 8 quality rating. By doing this, we will equally separate our data into above average and below average category (aka good wine and bad wine).



The cutting point between 5 and 6 looks like a good point to separate data, so the data will be spitted into two almost equaled dummy variables. This separation will bring meaning to our

result at the end. If the wine is classified as good wine, it means that the wine is at least above average. On the other hand, the wine that is classified as bad wine is below average. Also, we can apply logistic regression directly to the data. We will talk about the reason of using logistic regression in the next procedure.

Correlation between variable after creating new variable:

Next, we run the same EDA process to our new variable to see whether there is any change in direction of relationship or not. The result is that the degree of relationship is somehow weaker than using original rating variable, but the direction of relationship remains the same.

:

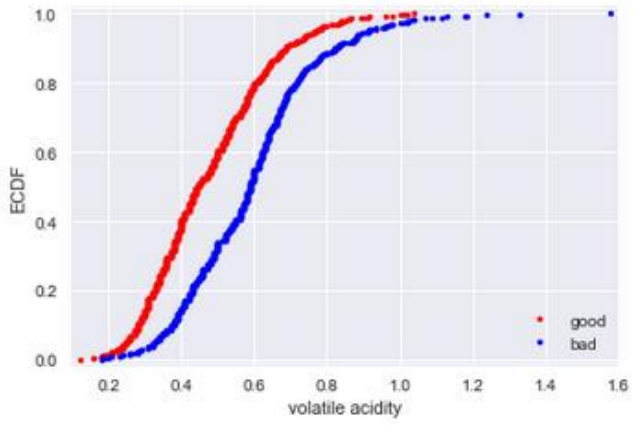
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality	rating
fixed acidity	1.000000	-0.255124	0.667437	0.111025	0.085886	-0.140580	-0.103777	0.670195	-0.686685	0.190269	-0.061596	0.119024	0.091761
volatile acidity	-0.255124	1.000000	-0.551248	-0.002449	0.055154	-0.020945	0.071701	0.023943	0.247111	-0.256948	-0.197812	-0.395214	-0.327195
citric acid	0.667437	-0.551248	1.000000	0.143892	0.210195	-0.048004	0.047358	0.357962	-0.550310	0.326062	0.105108	0.228057	0.167903
residual sugar	0.111025	-0.002449	0.143892	1.000000	0.026656	0.160527	0.201038	0.324522	-0.083143	-0.011837	0.063281	0.013640	-0.002371
chlorides	0.085886	0.055154	0.210195	0.026656	1.000000	0.000749	0.045773	0.193592	-0.270893	0.394557	-0.223824	-0.130988	-0.115071
free sulfur dioxide	-0.140580	-0.020945	-0.048004	0.160527	0.000749	1.000000	0.667246	-0.018071	0.056631	0.054126	-0.080125	-0.050463	-0.069207
total sulfur dioxide	-0.103777	0.071701	0.047358	0.201038	0.045773	0.667246	1.000000	0.078141	-0.079257	0.035291	-0.217829	-0.177855	-0.235046
density	0.670195	0.023943	0.357962	0.324522	0.193592	-0.018071	0.078141	1.000000	-0.355617	0.146036	-0.504995	-0.184252	-0.168958
pH	-0.686685	0.247111	-0.550310	-0.083143	-0.270893	0.056631	-0.079257	-0.355617	1.000000	-0.214134	0.213418	-0.055245	0.004693
sulphates	0.190269	-0.256948	0.326062	-0.011837	0.394557	0.054126	0.035291	0.146036	-0.214134	1.000000	0.091621	0.248835	0.211365
alcohol	-0.061596	-0.197812	0.105108	0.063281	-0.223824	-0.080125	-0.217829	-0.504995	0.213418	0.091621	1.000000	0.480343	0.446176
quality	0.119024	-0.395214	0.228057	0.013640	-0.130988	-0.050463	-0.177855	-0.184252	-0.055245	0.248835	0.480343	1.000000	0.844955
rating	0.091761	-0.327195	0.167903	-0.002371	-0.115071	-0.069207	-0.235046	-0.168958	0.004693	0.211365	0.446176	0.844955	1.000000

4. Inferential Statistics

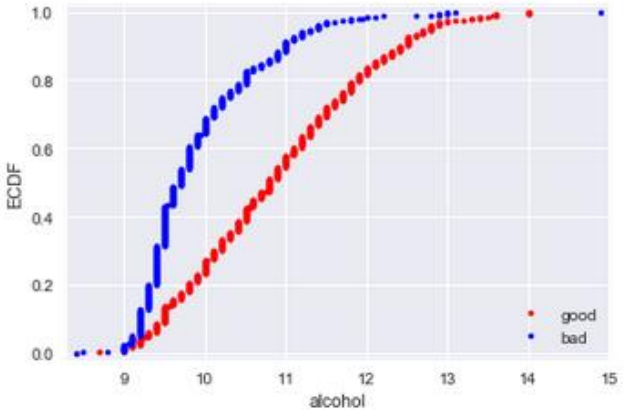
After creating new variable 'Rating', we separate wine into two categories which have more than 5 rating and which have less than 5. We will refer them as good wine and bad wine in the following procedure. Then, we try to run some test related to independence variables. The variables that have direct effect on rating are volatile acidity, citric acid, total sulfur dioxide, density, sulphates, and alcohol. Our test will be consisting of two steps:

1. Plotting ECDF of each group to see whether the distribution of each group (good wine, bad wine) is the same or not

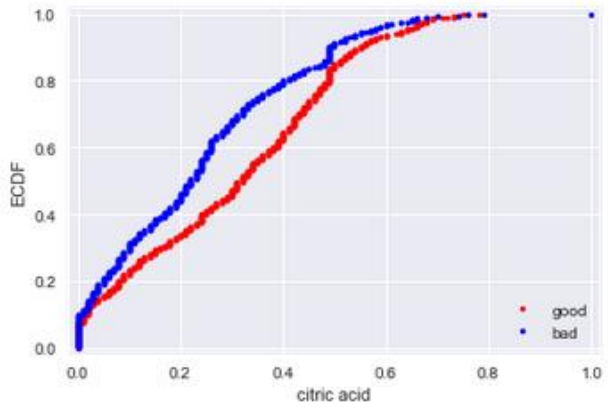
2. Using bootstrapping method to test if there is any significant difference in mean of feature between bad wine and good wine



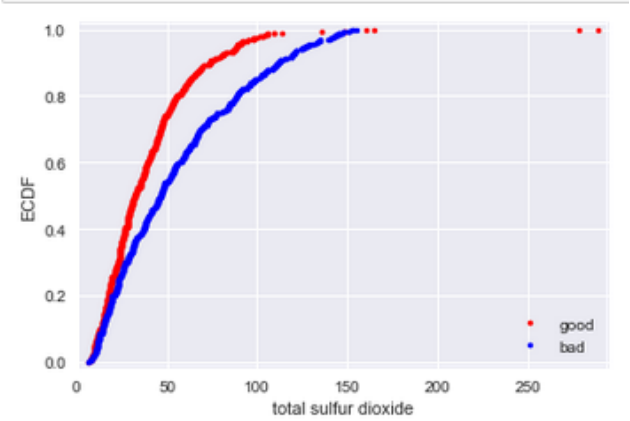

```
p-value = 1.0
```




```
p-value = 0.0
```



p-value = 0.0



p-value = 1.0

As we can see from the results, all tests that we conducted by using bootstrapping method has significant result meaning that the mean of each feature of each group is statistically significant. However, we want to dig deeper on every feature. By doing that, we use t-test the distribution of each feature is unknown. Moreover, it will be much faster and easier to use t-test to test on a lot of feature.

```
fixed acidity Ttest_indResult(statistic=3.429690129057006, pvalue=0.0006224652259994661)
volatile acidity Ttest_indResult(statistic=-12.663112642586356, pvalue=1.0270784781602082e-34)
citric acid Ttest_indResult(statistic=6.301952761680176, pvalue=3.966613371490847e-10)
residual sugar Ttest_indResult(statistic=-0.08743750672371305, pvalue=0.9303367824350479)
chlorides Ttest_indResult(statistic=-4.175824466776456, pvalue=3.203283265747779e-05)
free sulfur dioxide Ttest_indResult(statistic=-2.5432543553431044, pvalue=0.011097105019784335)
total sulfur dioxide Ttest_indResult(statistic=-8.766046924165353, pvalue=6.337983150190238e-18)
density Ttest_indResult(statistic=-6.403466171052481, pvalue=2.098721635166707e-10)
pH Ttest_indResult(statistic=0.17297062036867633, pvalue=0.8627006248387413)
sulphates Ttest_indResult(statistic=7.888647833741197, pvalue=6.597118478126576e-15)
alcohol Ttest_indResult(statistic=18.742690953986735, pvalue=1.8012380348945214e-69)
quality Ttest_indResult(statistic=59.24050921487324, pvalue=0.0)
rating Ttest_indResult(statistic=inf, pvalue=0.0)
```

From the results, only residual sugar and pH fail to reject the test which means that the difference in mean of those features is not statistically significant. To sum up, after using ECDF, bootstrapping, and t-test, the difference of mean of features between good wine and bad wine

is statistically significant. Only two variables (residual sugar and pH) are not statistically significant.

5. Modeling

Knowing the labels for wine rating, i.e. 0 for bad wine and 1 for good wine, we use supervised machine learning algorithms to build a predictive model. Furthermore, since there are only two outcomes (or classes) in the data (0 and 1), we use binary classification algorithms. The models are trained using the 70% of the data and the remaining 30% is used to evaluate the performance of the models.

Evaluation Metric:

Once the data is ready, we feed them to classification algorithm to build the model. In order to evaluate the performance of the model, we test the model on the test dataset. For tuning hyper-parameters, we use 5-fold cross validation with grid search method in scikit learn. And, we also plot the accuracy of each parameter to get the best parameter because of limitation of computational speed and time management. In this project, we keep in mind all elements of a confusion matrix. There are two popular metrics such as area under the curve (AUC) of

Receiver operating characteristic (ROC) curve and AUC of precision recall (PR) curve. Theoretically, ROC curves are useful in an algorithm that optimizes PR AUC. Also we will stick with PR AUC as our scoring parameter for tuning hyper parameters.

Logistic regression:

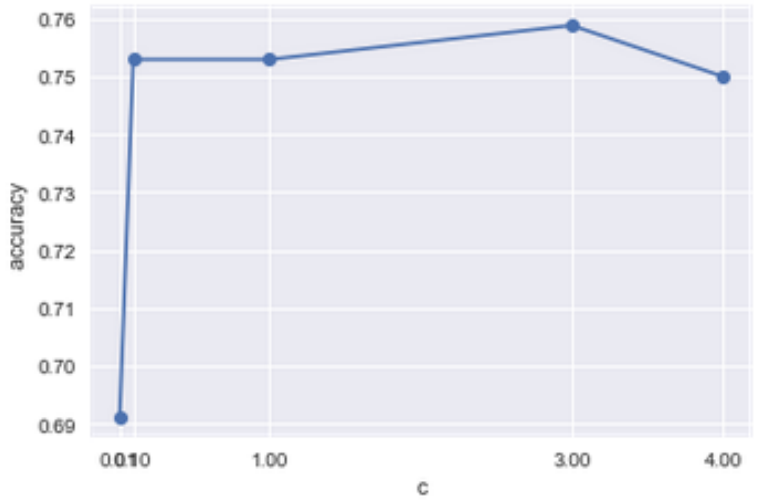
For this particular research, we decide to use logistic regression as a main machine learning model because of its interpretation of coefficient since we do care about how to improve rating of wine by modifying physicochemical properties. By using other classifiers such as SVM, Random Forest, etc., it won't be any meaningful interpretation. However, we will still imply some of those in the very end of this research in order to see how other models improve the predictability of the data. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor. In practical sense, the benefit of logistic regression is that it will indicate quantitatively change in probability of being classified as good wine when the physicochemical properties in wine change. For example, if we want to improve quality of wine, the coefficients received by logistic regression can be used to identify which feature should be added or drop. We will discuss about this later. We will try to fit the logistic model with training data set which is separated from test data set in order to preventing contamination and measure how well our model classify unseen data.

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
features = wine.drop(['rating','quality'],axis = 1)
y = wine.rating
Xlr, Xtestlr, ylr, ytestlr = train_test_split(features.values, y.values,random_state=5)
clf = LogisticRegression()
clf.fit(Xlr, ylr)
print("accuracy score: {}".format(accuracy_score(clf.predict(Xtestlr), ytestlr)))
```

```
accuracy score: 0.7529411764705882
```

Turning hyper parameter:

Next, in order to increase accuracy score, we will try different value of C. C in logistic regression is a penalty parameter called as Regularization. Regularization is applying a penalty to increasing the magnitude of parameter values in order to reduce over fitting. When you train a model such as a logistic regression model, you are choosing parameters that give you the best fit to the data. This means minimizing the error between what the models predicts for your dependent variable given your data compared to what your dependent variable actually is.



From the graph, using C = 3 will pump up accuracy score to 0.76 which is a bit higher than using default C

Coefficients:

-1.74560594 - (0.03258015*_fixed acidity)_ - _(3.11894759*_volatile acidity) -(0.45769148*_citric acid)_ - _(0.01674858*_residual sugar) -(2.79987165*_chlorides)_ + _(0.01363415*_free sulfur dioxide) - (0.01327637*_total sulfur dioxide)_ - _(1.72874178*_density) -(1.28555826*_pH)_ +_(2.03273057*_sulphates) + (0.88810736*alcohol)

The coefficients can be interpreted in this way.

- Increase in fixed acidity by 1 decrease the probability of classifying as a good wine by 3.31 percentage point ceteris paribus
- Increase in volatile acidity by 0.01 decreases the probability of classifying as a good wine by 3.168 percentage point ceteris paribus
- Increase in citric acid by 0.1 decreases the probability of classifying as a good wine by 4.68 percentage point ceteris paribus
- Increase in residual sugar by 1 decrease the probability of classifying as a good wine by 1.69 percentage point ceteris paribus
- Increase in chlorides by 0.01 decreases the probability of classifying as a good wine by 2.8 percentage point ceteris paribus
- Increase in free sulfur dioxide by 1 increase the probability of classifying as a good wine by 1.36 percentage point ceteris paribus
- Increase in total sulfur dioxide by 1 decrease the probability of classifying as a good wine by 1.34 percentage point ceteris paribus

- Increase in density by 0.01 decreases the probability of classifying as a good wine by 1.744 percentage point ceteris paribus

- Increase in pH by 0.01 decreases the probability of classifying as a good wine by 1.29 percentage point ceteris paribus

- Increase in sulphates by 0.01 increases the probability of classifying as a good wine by 2.05 percentage point ceteris paribus

- Increase in alcohol by 0.1 decreases the probability of classifying as a good wine by 9.29 percentage point ceteris paribus.

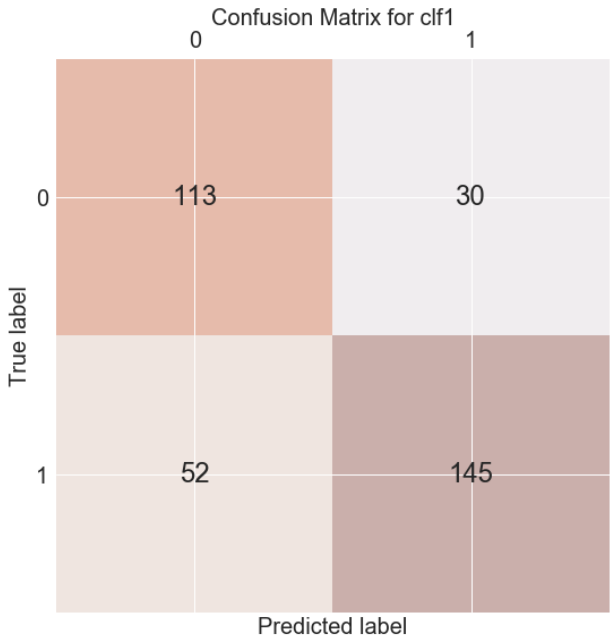
According to the model above, there are only three variables that will increase probability of classifying as a good wine such as free sulfur dioxide, sulphates, and alcohol.

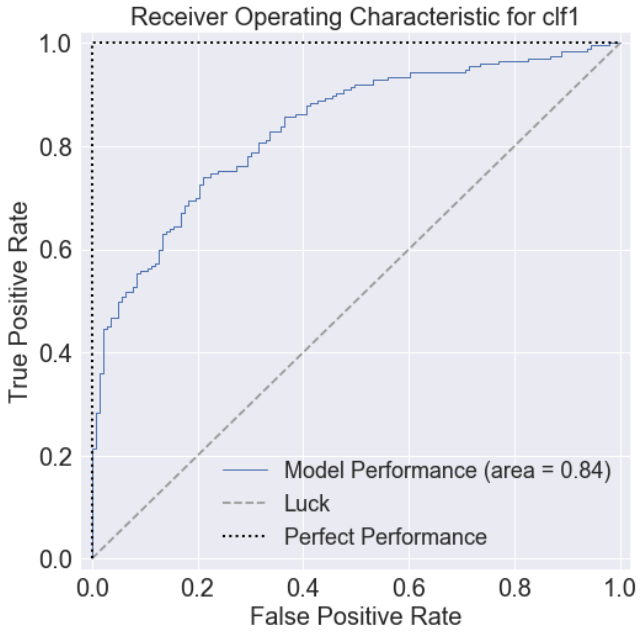
Performance Metrics:

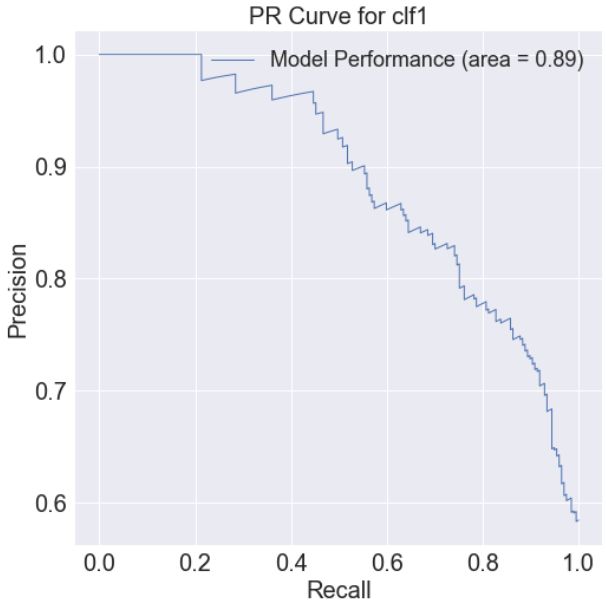
```
####################
Test data
####################
F1:  0.7795698924731183
Cohen Kappa:  0.5153832782895881
Brier:  0.2411764705882353
LogLoss:  0.4999267436385303
             precision    recall  f1-score   support

        0.0       0.68      0.79      0.73       143
        1.0       0.83      0.74      0.78       197

avg / total       0.77      0.76      0.76       340
```







According to the result, ROC AUC of the logistic model is quite high at 0.84. Also, PR AUC is high at 0.89. We will do more investigation and comparison with other supervised machine learning models.

KNeighbors Classifier:

In this particular machine learning model, we tent to apply preprocessing procedure before we try to fit the model with the data because the different in scale of each feature. For classification algorithms like KNN, we measure the distances between pairs of samples and these distances are also influenced by the measurement units also. For example, we are applying KNN on a data set having 3 features. First feature ranging from 1-10, second from 1-20 and the last one ranging from 1-1000. In this case, most of the clusters will be generated based on the last feature as the difference between 1 to 10 and 1-20 are smaller as compared to 1-1000. To avoid this miss classification, we should normalize the feature variables. In the data, there is a big difference in scale.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000	1359.000000
mean	8.310596	0.529478	0.272333	2.523400	0.088124	15.893304	46.825975	0.996709	3.309787	0.658705	10.432315
std	1.736990	0.183031	0.195537	1.352314	0.049377	10.447270	33.408946	0.001869	0.155036	0.170667	1.082065
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996700	3.310000	0.620000	10.200000
75%	9.200000	0.640000	0.430000	2.600000	0.091000	21.000000	63.000000	0.997820	3.400000	0.730000	11.100000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000

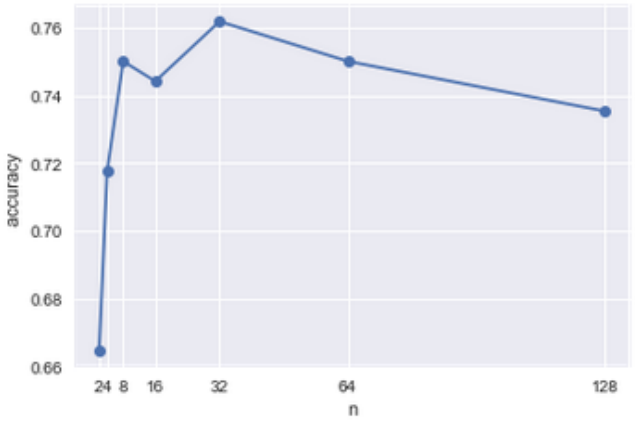
As we can see from the table, the mean of free sulfur dioxide is 46.8225975 ,meanwhile the mean of citric acid is only 0.2723. This suggests the imbalance in scale of features. We will investigate more on this problem and how it affects the model.

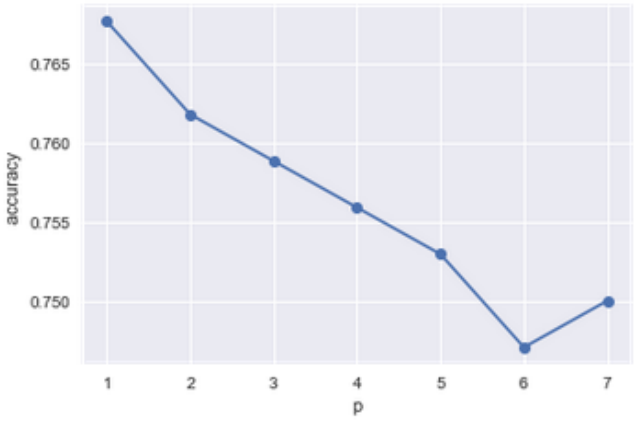
```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
# Setup the pipeline steps: steps
steps = [('scaler', StandardScaler()),
         ('knn', KNeighborsClassifier())]
# Create the pipeline: pipeline
pipeline = Pipeline(steps)
# Create train and test sets
Xlr, Xtestlr, ylr, ytestlr = train_test_split(features.values, y.values,random_state=5)
# Fit the pipeline to the training set: knn_scaled
knn_scaled = pipeline.fit(Xlr, ylr)
# Instantiate and fit a k-NN classifier to the unscaled data
knn_unscaled = KNeighborsClassifier().fit(Xlr, ylr)
# Compute and print metrics
print('Accuracy with Scaling: {}'.format(accuracy_score(knn_scaled.predict(Xtestlr), ytestlr)))
print('Accuracy without Scaling: {}'.format(accuracy_score(knn_unscaled.predict(Xtestlr), ytestlr)))
```

```
Accuracy with Scaling: 0.7323529411764705
Accuracy without Scaling: 0.6323529411764706
```

According to the result, the accuracy of the model jump around 0.1 with preprocessing procedure. Therefore, we included the scaling procedure before applying the model.

Turning hyper parameter:





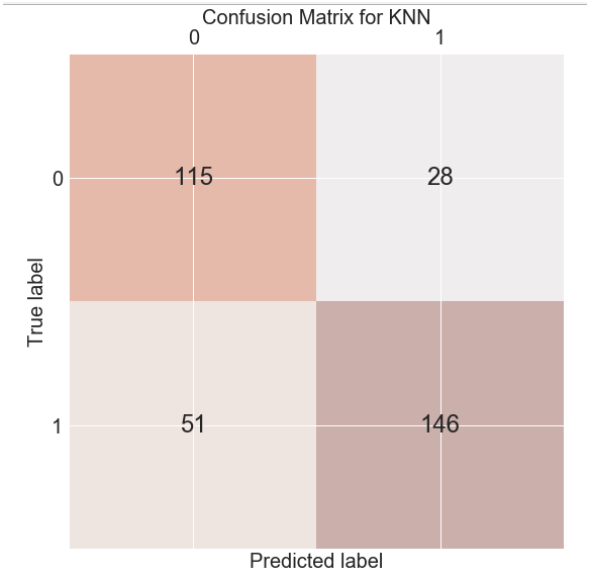
From the graph, using n_neighbors = 32 and p = 1 will pump up accuracy score to 0.77 which is a bit higher than using default setting.

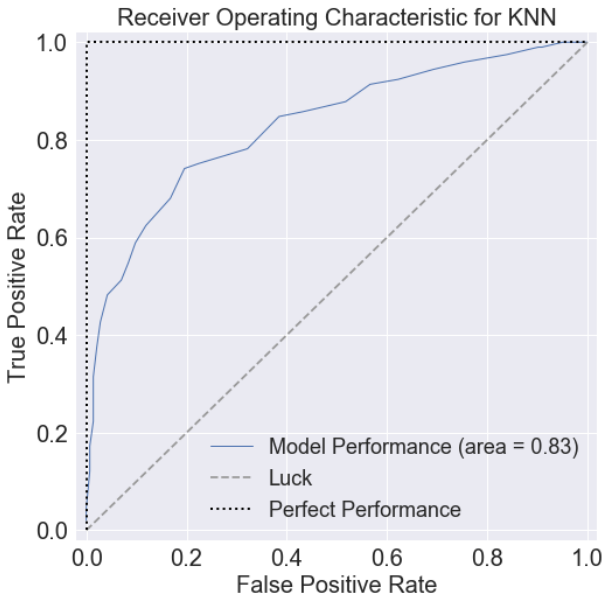
Performance Metrics:

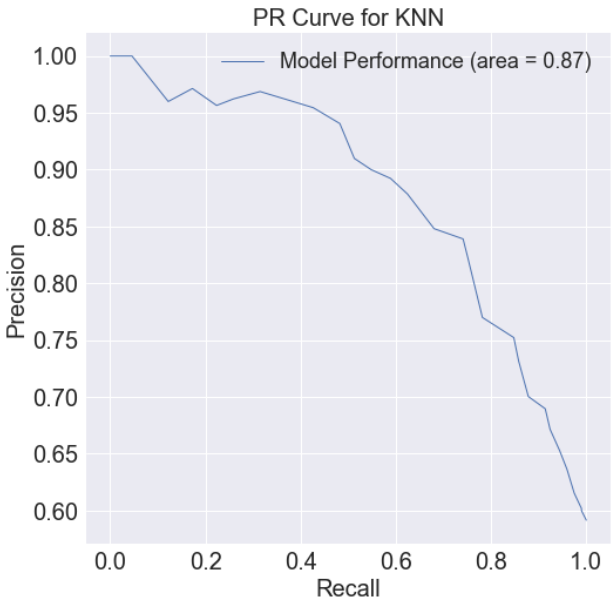
```
####################
Test data
####################
F1:  0.7870619946091645
Cohen Kappa:  0.533550986385107
Brier:  0.2323529411764706
LogLoss:  0.5085826325168699
             precision    recall  f1-score   support

        0.0       0.69      0.80      0.74       143
        1.0       0.84      0.74      0.79       197

avg / total       0.78      0.77      0.77       340
```







According to the result, ROC AUC of the KNeighbors model is a bit lower than the previous one at 0.83. Also, PR AUC is lower at 0.87. Therefore, this model is not quite good to apply to the data since it yield a lower score on performance matric and has no interpretation meaning.

Random Forest Classifier:

Final RandomForestClassifier model

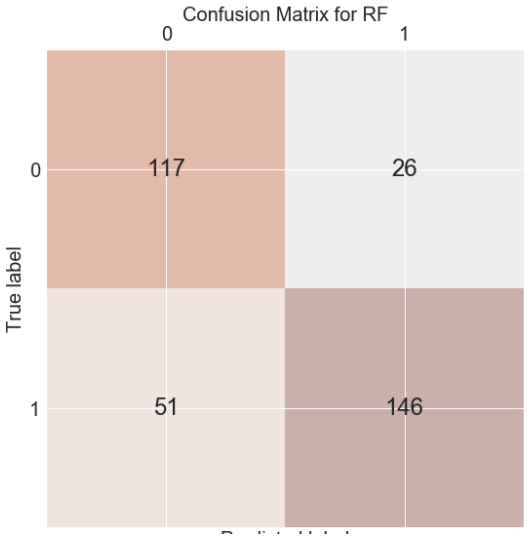
```
RF = RandomForestClassifier(n_estimators = 1000, max_depth= 30, min_samples_split=10 ,min_samples_leaf=1 )
RF.fit(Xlr, ylr)
print("accuracy score: {}".format(accuracy_score(RF.predict(Xtestlr), ytestlr)))
```

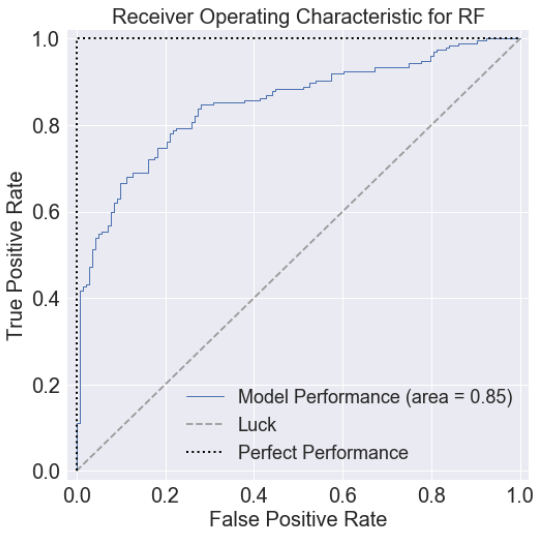
```
accuracy score: 0.7735294117647059
```

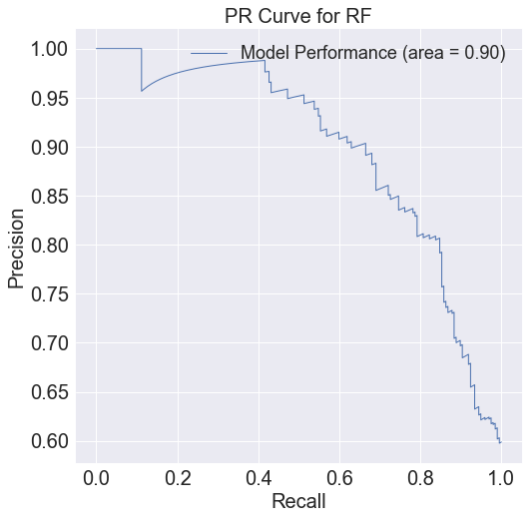
```
######################
Test data
######################
F1:  0.7913279132791328
Cohen Kappa:  0.54621091312487
Brier:  0.22647058823529412
LogLoss:  0.4870998471237059
             precision    recall  f1-score   support

        0.0       0.70      0.82      0.75       143
        1.0       0.85      0.74      0.79       197

avg / total       0.78      0.77      0.77       340
```







According to the result, ROC AUC of the Random forest model is slightly higher than the logistic one at 0.85. Also, PR AUC is slightly higher at 0.90. Therefore, this model is a little bit better than logistic regression model in term of predictability, but there is no any meaningful interpretation of coefficient.

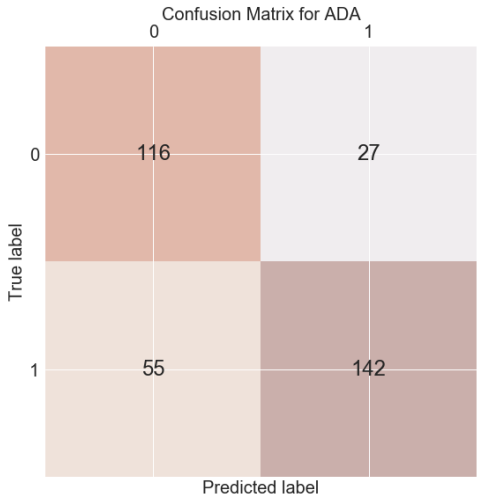
AdaBoost Classifier:

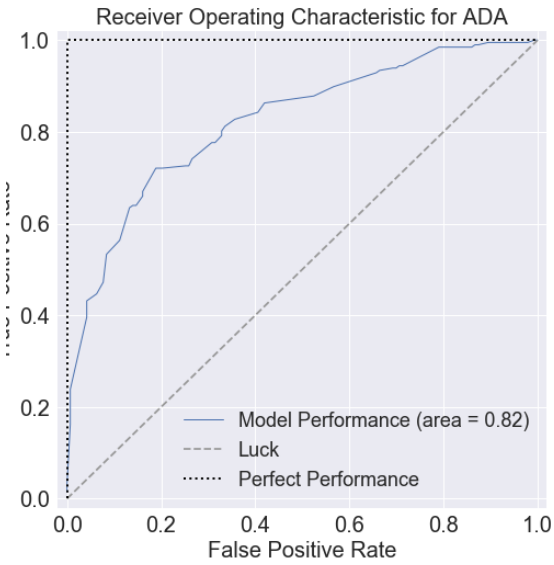
In [165]:
```
ADA = AdaBoostClassifier(n_estimators=10, learning_rate=1)
ADA.fit(Xlr, ylr)
print("accuracy score: {}".format(accuracy_score(ADA.predict(Xtestlr), ytestlr)))
```

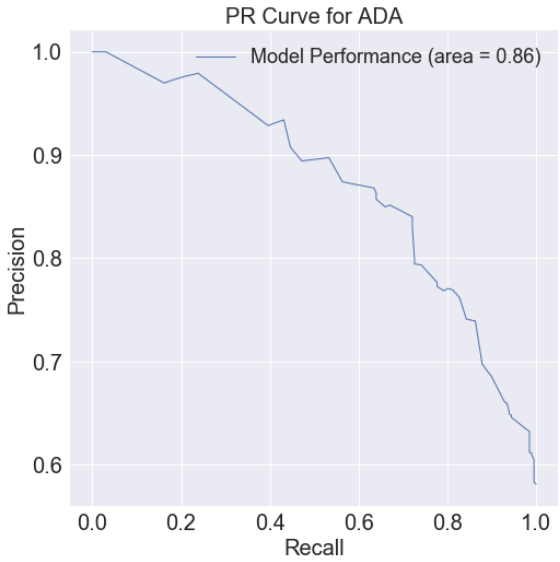
accuracy score: 0.7588235294117647

```
######################
Test data
######################
F1:  0.7759562841530055
Cohen Kappa:  0.5180972793583849
Brier:  0.2411764705882353
LogLoss:  0.6382432848936095
```

	precision	recall	f1-score	support
0.0	0.68	0.81	0.74	143
1.0	0.84	0.72	0.78	197
avg / total	0.77	0.76	0.76	340



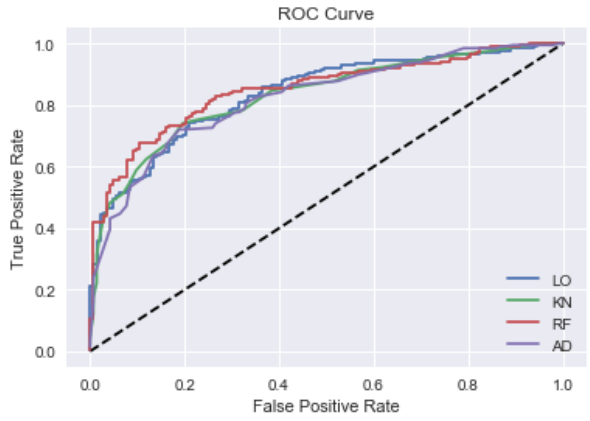




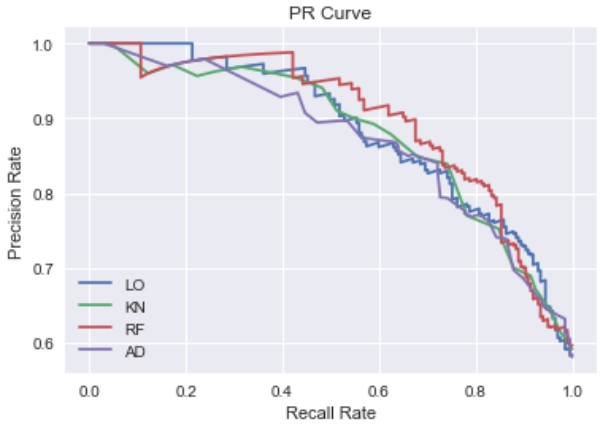
According to the result, ROC AUC of the ADA boost model is the lowest compared to other model at 0.82. Also, PR AUC is also the lowest at 0.86. Therefore, Ada boost model is the worst model among the others.

Model Comparison:

We have used Logistic Regression, KNeighbors, Random Forest, Ada boost classifiers to build a model to predict rating of wine from physicochemical properties. Based on testing the models on the holdout dataset (70% of the whole data), we found different performance of all models. The results of various evaluation metrics scores are shown below for all models such as ROC AUC and PR AUC. These metrics are suitable for comparing models when we are interested in predicting probabilities or likelihood.



roc score: 0.8358240744027545
roc score: 0.8301977210606653
roc score: 0.8476447410457562
roc score: 0.822122040396152



PR score: 0.8860298237126274
PR score: 0.8719587583823536
PR score: 0.8968161609220517
PR score: 0.8629936637082314

The plot shows that Random Forest has the highest value of both ROC AUC and PR AUC followed by logistics regression, Ada boost, and KNeighbors. Therefore, the random forest model is the best one in this study in term of predictability. We can also look at the ROC and PR curves for all models which corroborates the fact that the random forest model performs the best. Anyway, the performance of random forest model does not out preform logistics

regression model much both in ROC AUC and PR AUC. As we mentioned about this in the very beginning of the modeling section, logistic regression has a very important and useful properties because of its interpretation of coefficient. Since the performance of random forest model is just slightly better than logistic regression, we still use logistic regression model as the one that we apply to the wine dataset.

6 Assumptions and Limitations

First, our data set is not included wine from other country and different kind of wine. This can be critical drawback when we want to generalize our finding. Second, there is only one rating in our dataset. Adding more rating from different source can increase more credibility of the result. On the other word, the result heavily depends on just one rating system. Last, lack of price, name and year limits the application of the finding. It would be much more interesting, if we can investigate more in the relationship between physicochemical properties and price or year, so we might find answer to some myths in winery industry.

7 Conclusions

In this project, we first explored the wine dataset which has been cleaned. Then, we applied some cleaning techniques to see whether the data is really clean. After that, we did data exploratory analysis and applied supervised machine learning model to fit the data. This is what we found from the research.

Data Exploration Conclusions:

1. Most variables is not normally distributed
2. Some features have a stronger relationship to wine rating than other features such as volatile acidity, citric acid, total sulfur dioxide, density, sulphates, and alcohol.
3. The behavior between direct effect variables and indirect effect variables is similar among each rating

Modeling Conclusions:

After exploring all datasets, we used four different supervised classification algorithms (Logistic Regression, KNeighbors, Random Forest, Ada boost) to train the predictive model by using 70% of the whole data. The remaining 30% was used to evaluate the model.

1. Using various performance evaluation metrics, we found that the Random forest classifier gives the best model performance in term of predictability.

2. We used the logistic regression model to apply to the data because of the combination of its performance and interpretability.

3. We achieved the ROC AUC to be about 0.85. And, The AUC for PR was great about 0.90

4. In terms of features, we found that we do not need do any feature modification since it didn't improve any predictive score