Machine Learning Engineer Nanodegree

Capstone Project Report

Durgaprasad Rajani

Jun 28th 2018

Definition

Project Overview

In today life we need everything that meets standards, otherwise we won't consider to take it. So, In this I am going to find the quality of red wine.Good quality of wine makes health good.In some cases the quality does not meet standards.The companies should maintain the belief of the customers.The belief can be achieved from the quality of the product.

Every company invest lot of money for preparing wine.So, it is crucial for them to make a quality of wine.Redwine also preferable by some doctors.There is a belief that it lowers the chances of occurring heartattack or heartstrokes.So,I classify the wine quality as good and bad by giving some quality limit.By using all these features which are involved in wine making helpful for deciding the wine quality. And I explore for the dataset that what features make a good quality of wine.I got the dataset from this Link:https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

This classification problem is cited at the link: https://www.sciencedirect.com/science/article/pii/S0167923609001377?via%3Dihub

Problem Statement

The Redwine quality can be decided by many of the factors that are used for preparing the redwine.Those ingredients proportions can change the wine quality.By considering all those ingredients we can define which of those can significantly effect the wine quality.The main purpose of this is find the most related features with the quality and use the to find the wine quality by saying it is good or bad.And for that we can consider it as a classification problem.

Features and Description:

- fixed acidity – It is non-volatile acid in the wine
- volatile acidity – It is amount of acetic acid in the wine
- citric acid – It is the amount of citric acid in the wine

- residual sugar – It is the amount of sugar remained after the fermentation process
- chlorides – It is the salt content in wine
- free sulfur dioxide – It is amount of free sulfur dioxide present in wine as gas form
- total sulfur dioxide – It is the amount of free and bound forms of sulfur dioxide
- density – It is the density of water that is close to that of water depending on the percent of alcohol and sugar content
- pH – It is the value for representing how much it is acidic or basic nature
- sulphates – It is the amount of sulphates in wine
- alcohol – It is the percent of alcohol content in wine

By observing the above features we can say that alcohol content can effect the quality of wine the higher the alcohol the better the wine.The feature residual sugar can effect majorly because too much sugar content can make wine sweet which is not a good wine.The citric acid feature is also show impact on determining the wine quality.If it increases the taste of wine becomes somewhat sour.So,that wine can be treated as bad wine.The amount of chlorides are also responsible for wine quality. This dataset can be viewed as classification or regression task.By seeing these classes they are not orderd and are imbalanced. Most of the wine quality are normal wines, not poor or excellent.More than one feature can decide the wine quality.So, I transformed the features by using standardscaler.We can use our model to train those data and predict the wine quality.So, I will use classification supervised model for this.I do use different classification models and caluculate the F1score and take the best out of it.If I won't get high accuracy levels then I will use some optimization methods like Grid Search to tune the parameters by taking the k-folds input for getting the better results.The different classification algorithms are RandomForestClassifiers,SVM,SGDC and voting classifier will use for this.And I will compare those algorithms find the best model.

Metrics

I used F1 score as an evalution metric .Between the original test values and predicted values.This score was calculated for every model.After tuning the parameters crossvalidationscore also calculated.Because of data imbalance accuracy doen't give better results .So,I took F1 score as a metric to measure the performance of my models. Accuracy in classification problems is the number of correct predictions made by the model over all kinds predictions made.

$$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN)$$

In the Numerator, are our correct predictions (True positives and True Negatives)(Marked as red in the fig above) and in the denominator, are the kind of all predictions made by the algorithm(Right as well as wrong ones).

Precision is a measure that tells us what proportion of patients that we diagnosed as having cancer, actually had cancer. The predicted positives (People predicted as cancerous are TP and FP) and the people actually having a cancer are TP.

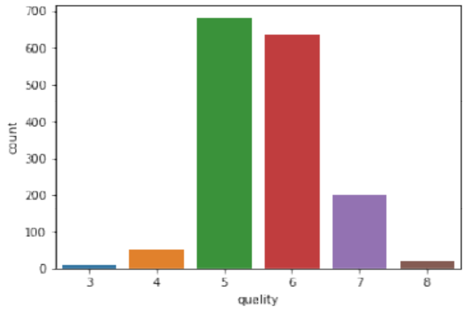
Precision = TP/(TP+FP)

Recall is a measure that tells us what proportion of patients that actually had cancer was diagnosed by the algorithm as having cancer. The actual positives (People having cancer are TP and FN) and the people diagnosed by the model having a cancer are TP. (Note: FN is included because the Person actually had a cancer even though the model predicted otherwise).

Recall = TP/(TP+FN)

Due to imbalance data the accuracy is not true accuracy.So,f1_score takes the harmonic mean of precision and recall.

F1_score=(2*Precision*Recall)/(Precision+Recall)

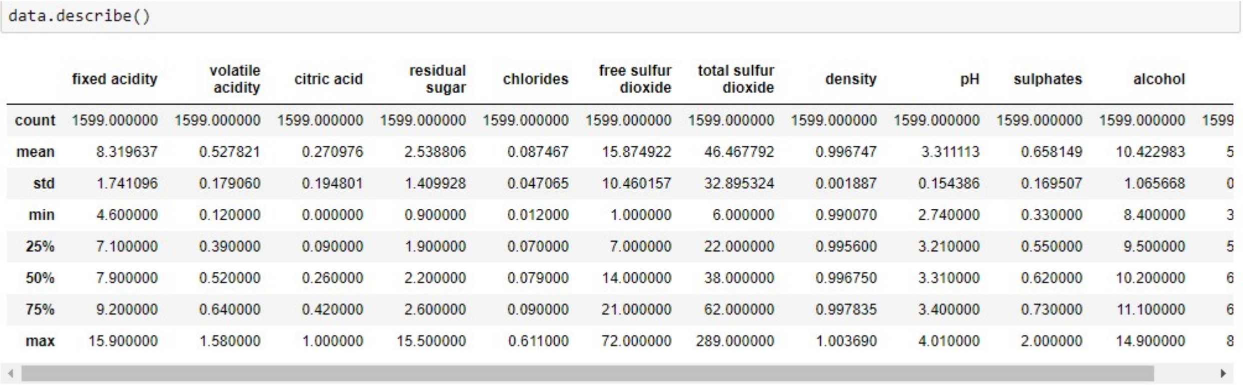


Analysis

Data Exploration

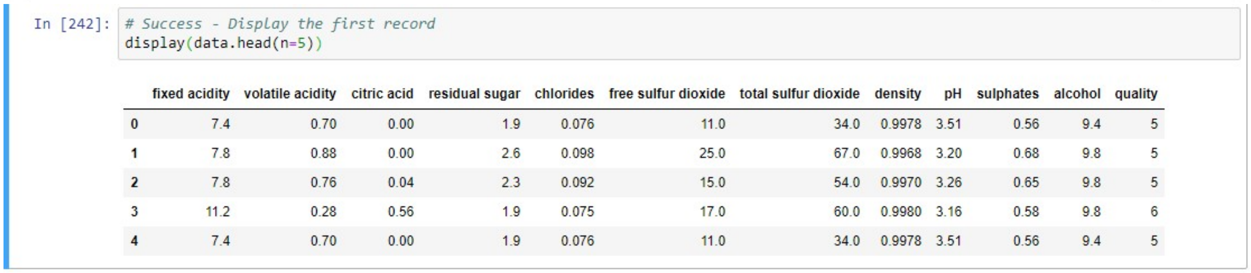
I have taken the dataset from the link: https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009

In this Data Exploration section,I described the data like mean,count,min,max,std,boundaries for boxplot graph by using the describe() method in pandas library.After that I printed the first five records.The describe() method describes every feature in the data.Then I came to the conclusion that the data doesn’t have any null values. The datset consists of 1599 rows and 11 columns.In this no missing values are present in any column.All these are positive values and no negative values are involved in this dataset.Because of having all the skwed distributions of features.I transformed the feature values using standardscaler .Only physiochemical inputs are involved in this and no sensory inputs(like wine brand, wine price etc.) are involved.Here,we will consider two potential classes good/1 and bad/0.The imbalances can be solved by taking the wine good which has more than or equal to 7 and bad as below 7.I will split the data by using the train_test_split in cross_validtion of scikitlearn.



```
data.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8



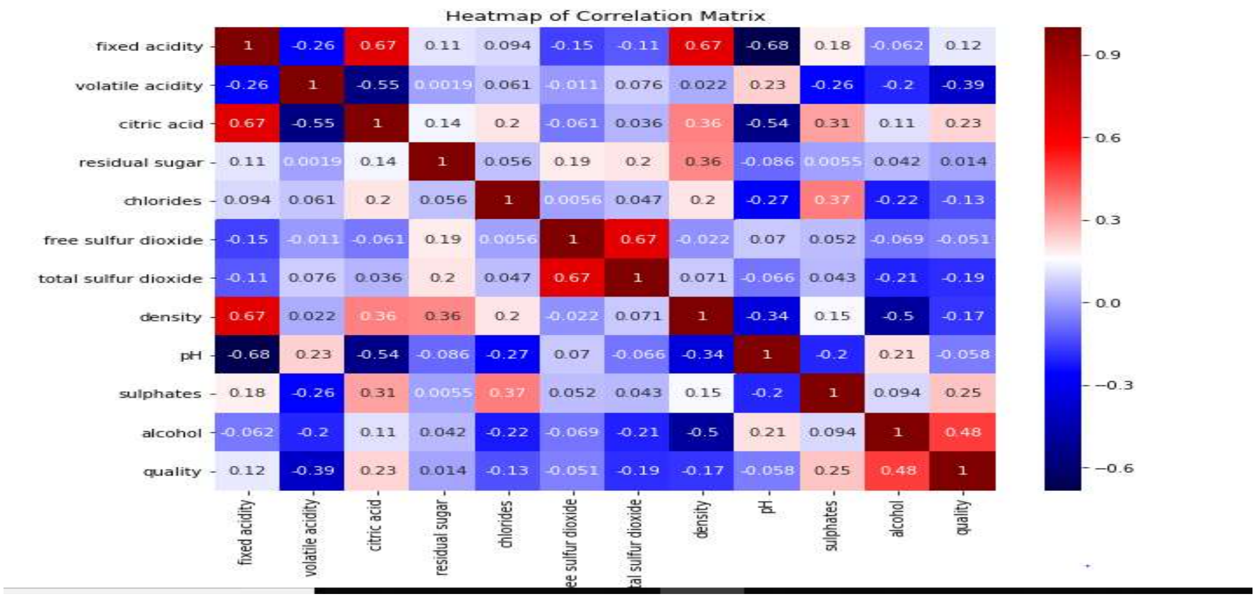
In [242]:

```
# Success - Display the first record
display(data.head(n=5))
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

Visualization

At first I plotted the correlation heatmap for the features in the data.To know how the features are correlated.Then,I came to know that alcohol,suphates,free sulfur dioxide are more likely be correlated.The heatmap is shown below and u can observe that alcohol and free sulfur dioxide have high correlation values.



Heatmap of Correlation Matrix

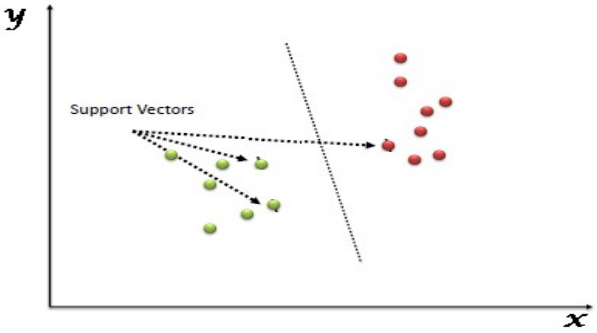
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	ee sulfur dioxide	tal sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.12
volatile acidity	-0.26	1	-0.55	0.0019	0.061	-0.011	0.076	0.022	0.23	-0.26	-0.2	-0.39
citric acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.23
residual sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.014
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.13
free sulfur dioxide	-0.15	-0.011	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.051
total sulfur dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.066	0.043	-0.21	-0.19
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.17
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.066	-0.34	1	-0.2	0.21	-0.058
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.25
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.48
quality	0.12	-0.39	0.23	0.014	-0.13	-0.051	-0.19	-0.17	-0.058	0.25	0.48	1

Then I plot distplots() for every feature in the data.So that I can see that how the values are distributed by the particular feature.Volatile acidity,density,pH are normally distributed. Fixed acidity, Citric acid, Free sulfur dioxide, Total sulfur dioxide, Sulphates, Alcohol.Remainig features have long tailed graph.

Algorithm and techniques

In machine learning model, the initial task is to take the model which is suitable for our problem. So,I took a model as benchmark model and calculate it's performance. Then I took different models and calculate their performances and at last compare them and found the best out of it. Here, the benchmark model I used was RandomForestCassifier. It is one of the model that performs well for these type of problems. Actually Randomforestclassifier works well even without parameter tunning. Here I tried different values of estimators and train it by using the train_test_split() and calculated the F1_score for this model. Here estimators are used to build the trees for getting the better predictions.

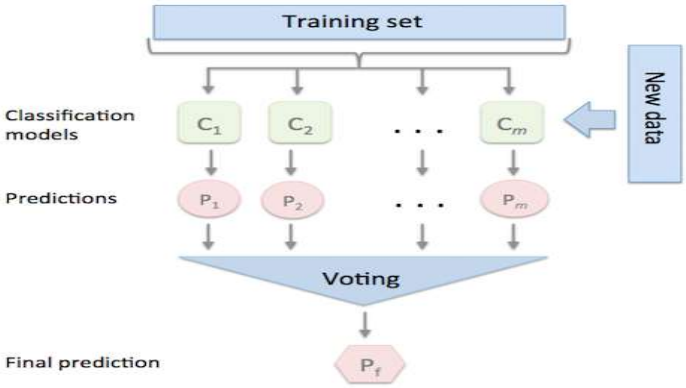
And after the benchmark model I took the model named svm there I trained the model by using those splits and find out the F1_score for the model. Where each dataitem is represented in a n_dimensional space and classify the points according to their classes by using a hyperplane.



This problem is binary classification and this model also well suited for the problem.

Stochastic Gradient Descent classifier is another model I used for this problem and trained the model with this train and test set. Then, In order to increase the F1_score for the models I used parameter tuning technique called GridSearchCV for the svm estimator and I used best_params_ for taking the best parameters.

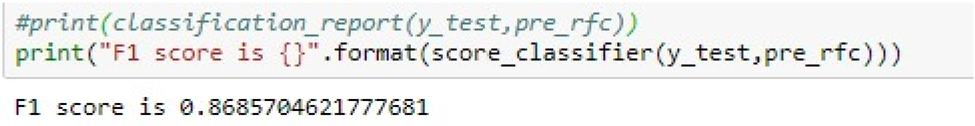
At first I used some random cross validation size it didn't perform well. After that I used k-fold for taking different train and validation sets . As it is binary classification , it classifies the classes better. Then again I train the svm with parameter tuning and calculated the F1_score for the model. At last I used ensemble model for getting the better F1_score .The ensemble learning method is Voting classifier by taking the above estimators as params for the model. Stochastic Descent classifier needs a lot more tuning than the other model and descent is sometime slow and very steep. Stochastic takes the sample to find the gradient and continue through it.SVM is one of the best option for binary classification and provide different features. This increases the score by taking all the predictions of the model and make the best predictions from it.



So, as I used different models and I take predictions from those and applied to voting classifier. By using those predictions and it analyse and give better predictions finally. This is one of the method that enhances our score.

Benchmark Model

I used RandomForestClassifier model as my benchmark model.F1 score taken as reference to compare the results with other models. If it alone gives the best metric scores than other models then I will consider this as my model. The model F1_score u can see below



```
#print(classification_report(y_test,pre_rfc))
print("F1 score is {}".format(score_classifier(y_test,pre_rfc)))
```

```
F1 score is 0.8685704621777681
```

Methodology

Data Pre-processing

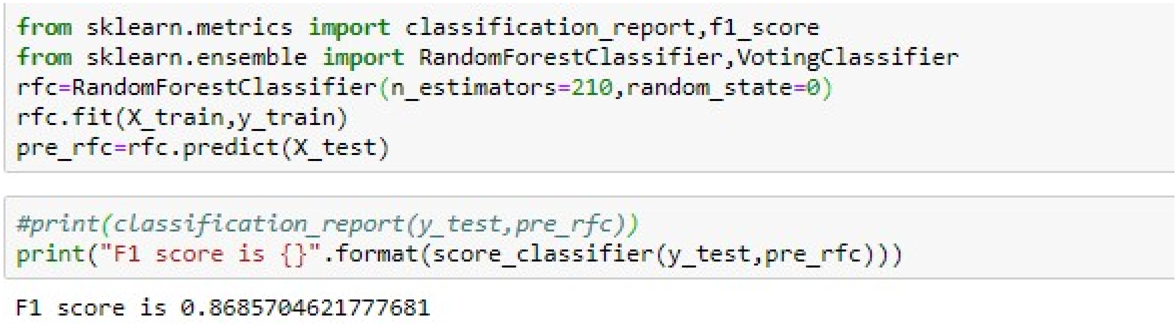
In this section I explored the data and found out no null values and missing values in the data. And coming to feature description I normalized the data by using standardscaler as the data values are well distributed as we saw in the exploration section. And I converted the continuous target data into categorical data as bad and good quality and make numerical for it. Then in the below section I split the data as 80% training and 20% testing by taking the stratify in the train_test_split suggested by the reviewer. It enhances the score by evenly split the data.

Implementation

I tried different models for the problem to get better results.F1_score is used to measure the performance.

RandomForestClassifier:

Random Forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because it's simplicity and the fact that it can be used for both classification and regression tasks. It is used as benchmark model here. Here I got the score of 0.87.



```
from sklearn.metrics import classification_report,f1_score
from sklearn.ensemble import RandomForestClassifier,VotingClassifier
rfc=RandomForestClassifier(n_estimators=210,random_state=0)
rfc.fit(X_train,y_train)
pre_rfc=rfc.predict(X_test)
```

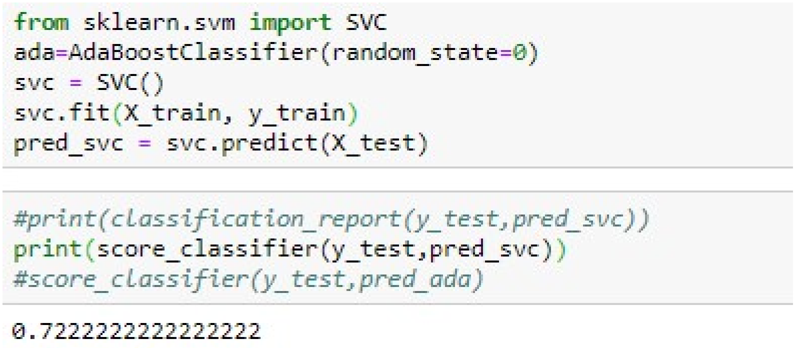
```
#print(classification_report(y_test,pre_rfc))
print("F1 score is {}".format(score_classifier(y_test,pre_rfc)))
```

```
F1 score is 0.8685704621777681
```

Support Vector Machine:

Support vector machines can be used as classification and regression problems. It is more popular for their multiple features here I took the simple 'svc' for the problem and calculated the performance.

later I tuned the parameters using grid search and find out the best parameters from it. By using those parameters I took svc and calculated the performance of the model. By using the grid search the performance of the model increased.

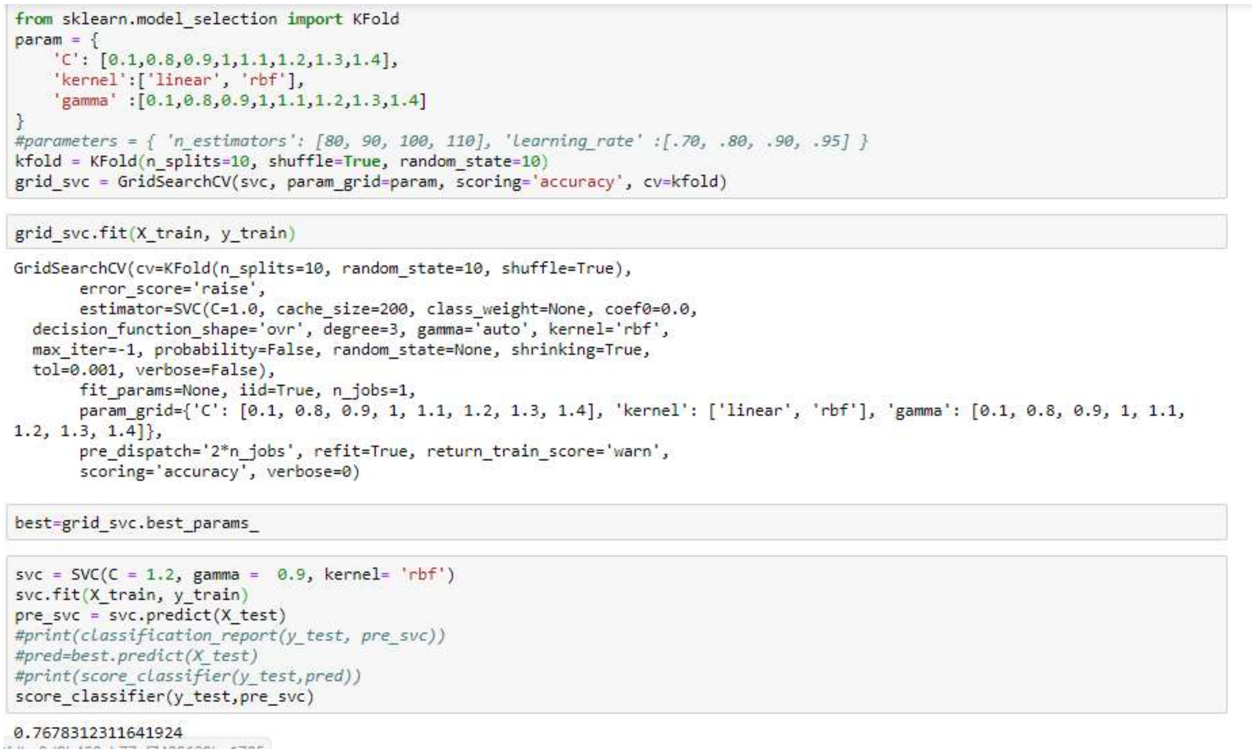


```
from sklearn.svm import SVC
ada=AdaBoostClassifier(random_state=0)
svc = SVC()
svc.fit(X_train, y_train)
pred_svc = svc.predict(X_test)
```

```
#print(classification_report(y_test,pred_svc))
print(score_classifier(y_test,pred_svc))
#score_classifier(y_test,pred_ada)
```

```
0.7222222222222222
```

Optimized:



```
from sklearn.model_selection import KFold
param = {
    'C': [0.1,0.8,0.9,1,1.1,1.2,1.3,1.4],
    'kernel':['linear', 'rbf'],
    'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
#parameters = { 'n_estimators': [80, 90, 100, 110], 'learning_rate' :[.70, .80, .90, .95] }
kfold = KFold(n_splits=10, shuffle=True, random_state=10)
grid_svc = GridSearchCV(svc, param_grid=param, scoring='accuracy', cv=kfold)
```

```
grid_svc.fit(X_train, y_train)
```

```
GridSearchCV(cv=KFold(n_splits=10, random_state=10, shuffle=True),
       error_score='raise',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'C': [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4], 'kernel': ['linear', 'rbf'], 'gamma': [0.1, 0.8, 0.9, 1, 1.1,
1.2, 1.3, 1.4]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring='accuracy', verbose=0)
```

```
best=grid_svc.best_params_
```

```
svc = SVC(C = 1.2, gamma =  0.9, kernel= 'rbf')
svc.fit(X_train, y_train)
pre_svc = svc.predict(X_test)
#print(classification_report(y_test, pre_svc))
#pred=best.predict(X_test)
#print(score_classifier(y_test,pred))
score_classifier(y_test,pre_svc)
```

```
0.7678312311641924
```

Stochastic Gradient Descent:

The use of SGD In the neural network setting is motivated by the high cost of running back propagation over the full training set. SGD can overcome this cost and still lead to fast convergence. Generally each parameter update in SGD is computed w.r.t a few training examples or a minibatch as opposed to a single example. The reason for this is twofold: first this reduces the variance in the parameter update and can lead to more stable convergence, second this allows the computation to take advantage of highly optimized matrix operations that should be used in a well vectorized computation of the cost and gradient.

Code snippet:

```
from sklearn.linear_model import SGDClassifier

sgd = SGDClassifier(penalty=None,random_state=0)

sgd.fit(X_train, y_train)

pred_sgd = sgd.predict(X_test)

score_classifier(y_test,pred_sgd)
```

output:

```
F1-score is 0.52
```

Voting Classifier:

It is an ensemble classifier used to increase the performance of the model.It can use multiple models and combine them to make predictions that’s why it also got some high performances.Taking the best prediction by taking all the models and boost them to get the higher results.

Code snippet:

```
eclf1 = VotingClassifier(estimators=[('lr', rfc), ('rf', svc), ('gnb', sgd)],
voting='hard')

eclf1.fit(X_train,y_train)

eclf1_pred=eclf1.predict(X_test)

score_classifier(y_test,eclf1_pred)
```

output:

```
f1-score is 0.791
```

Here I got some complications that Grid searchcv did not perform well. But after I change the cv size by taking the k-folds cross validation. I appled the Gridsearchcv for svm and at first I applied it without preprocess the data. This leads to taking so much time for running even it is the small dataset.Then I preprocess the data and applied it again. And it works well this time. The gradient descent does not gave best accuracy scores then I went to the ensemble method to boost my model and for getting better results.

Refinement

To find the optimized parameter I used Grid Search method where we provide list of parameters in dictionary and model runs and score the model on different parameters combination and optimized the model. In the above section we calculated the f1_score for all the models, there we can see that stochastic and svc got low scores .Then I used the gridsearch for parameter tuning. I took the best parameters and I used the kernel ‘rbf’ with a cross validation size by taking the k-fold cross by taking the n_splits. I used ensemble learning by building the multiple models which are used above and feed the voting classifier based different prediction criteria. Then these two models gives better scores after tuning the parameters. I used normal splitting of data and in grid search also I did k-fold splits of train data.

Results

Model Evaluation and Validation

I used Random forest classifier as my benchmark model it initially gave me better results and performing the normal classifiers without tuning any parameter got some low scoring results. Then I used some performance increasing models for increasing the f1_score.Then those algorithms performed well when compared to models with parameter tuning. The f1_scores never crossed the score made by random forest classifier. It made as a good model for the problem. Taking simple validation of the data gives better results for this model. Then I took the cross validation k-folds and I trained the with that data and I gave the results for k-cross validation in notebook. The small changes by k-fold may effect the model in a resonable way.

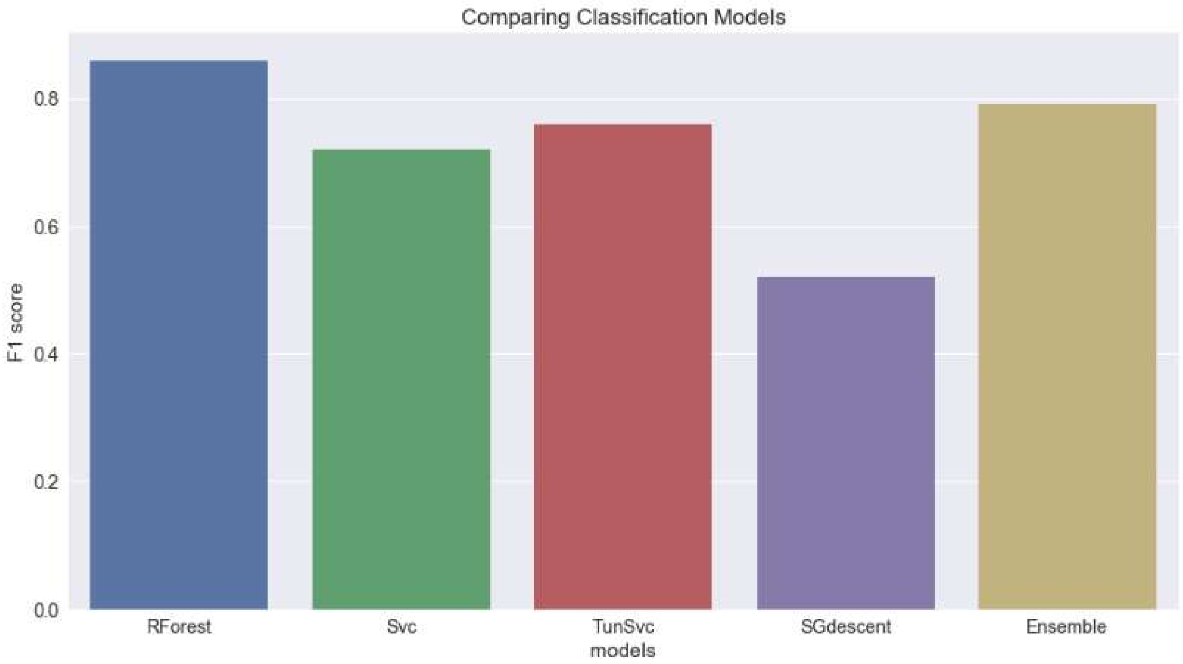
Justification

The RandomForest classifer used as Benchmark model.F1_score of benchmarkmodel is act like a threshold measure for the other models. . With choice of different classifier in hand such as Stochastic Gradient Boosting, SVM, Ensemble learning. I decided to go with Random Forestclassifier because this method increase the accuracy score of the model. And Ensembling methods also a good model for getting high accuracy. Performance of ensemble learning will not be slow and it uses weak learner to identify the features and label it with result this was also best choice. The optimized model scores are more compared to the unoptimized models except RandomForestsclassifier.It attains a F1_score of 0.87which is high compared to the other models. Ensembling also get some high score of 0.79.

Conclusion

Free Form Visualization

I made the different models by training them and testing them against the test data. Then I calculated the scores for every model with the f1_score metric and I got some better results. Then I plotted a barchart for all the models against to their scores and based on the chart we can decide that which algorithm is better among them. The barchart is shown below.



Reflection

At first we have to read the csv file into the dataframe with the help of pandas library and after that I will explore the data.I will check whether the columns have any null values or missing values. If they are I will preprocess the data. After preprocessing and I will describe the data. And made some visualizations on the features to make conclusions from it.As the quality values are mostly normal wines so,I will make this as binary classification as good or bad.For this will transform the quality values like equal or more than 7 quality will be treated as good otherwise bad. Then I will split my data into 80% training and 20% testing with some random state and by taking the stratify for the target variable. After I will perform RandomForestClassifier and find the F1score of that model. Then different models are trained and tested using this data. For increasing the accuracy I will tune the parameters by using the Gridsearchcv method. In that I used the k-fold cross validation method. The different algorithms include ensemble methods like RandomForestClassifier, Stochastic Gradient Descent Classifier , ensembling method voting classifier and SVM. Here I found intresting to train the ensemble model and using the randomforest classifier make better results. And I found somewhat difficult on tuning the parameters to increase the score of the model.And it takes more time to tune the parameters.

Improvement

We can improve the model scores by taking the k-fold validation in Grid search.Then the train and test data perform well on the model.Another one can be implemented as the data is imbalanced then we can do upsampling the minority class or downsampling the majority class and combine the to form a new dataframe there the ratio's are equal for the two classes. Future improvement can be made if more data can be collected on both low-quality and high-quality wine.

References

- http://www.realsimple.com/holidays-entertaining/entertaining/food-drink/alcohol-content-wine
- http://www.winegeeks.com/articles/85/high_alcohol_is_a_wine_fault_not_a_badge_of_honor/
- http://en.wikipedia.org/wiki/Wine_fault
- http://en.wikipedia.org/wiki/Sulfite
- http://www.calwineries.com/learn/wine-chemistry/wine-acids/citric-acid