Machine Learning Engineer Nanodegree
CAPSTONE PROJECT REPORT

Title: Using Supervised learning to find whether the quality of wine is good or bad.

June 29th,2018

Sonia Agarwal

I.Definition

Project Overview

Wine is an alcoholic beverage produced through the partial or total fermentation of grapes. Other fruits and plants like berries, apples, cherries, dandelions, and palm can also be fermented. It is one of the most consumed drink by the people. The Wine we are going to study here is White wine .It can be made with either white or red grapes. It is one of the most consumed drink by the people. Wine sector contributes a lot of economy to most of the countries like UK, Portugal etc. Producing a good quality of wine is not an easy task. According to the technical point of view, a good quality wine consists of factors meeting specific criteria and set according to considerations of scientific, chemical and technical factors, which can always be verified analytically. The quality of wine is mainly affected by the ingredients used for making it. So, we use these ingredients to predict the quality of the wine. The major motivation for exploring this project is to develop the skill and understanding how to work out on real time datasets and how Machine-Learning is potentially making the lives of people easier.

Reference1: https://en.wikipedia.org/wiki/Wine_chemistry

Reference2: https://winefolly.com/tutorial/how-is-white-wine-made/

Reference3:https://en.wikipedia.org/wiki/Wine

Problem Statement

By using the dataset, the task is to determine of the quality of wine. Since the task is to figure out whether the quality of the wine is good or bad based on the chemical composition in manufacturing the wine. And hence we have to tackle with a binary classification problem that has two possible outcomes (‘0’ for bad

quality and ‘1’ for good quality). By using machine learning techniques we can determine quality of wine. Several steps are involved in the project like Data exploration, Data processing and finally testing various algorithms and techniques

Description of features

The dataset has 4898 instances and 11 attributes and a target variable. The 11 attributes are as follows:

- Fixed acidity -It is the level of fixed acidity
- Volatile acidity -it is the level of volatile acidity
- Citric acid -amount of citric acid
- Residual sugar -amount of residual sugar
- Chlorides -amount of chlorides
- Free sulfur dioxide -amount of free sulphur dioxide
- Total sulfur dioxide -amount of total sulphur dioxide
- Density -density of the wine
- pH –pH of the wine
- Alcohol-alcohol content in wine
- sulphates- amount of sulphates in the wine

output variable is(based on sensory data)

- Quality (score between 0 and 10)

By considering all the above features, we can predict the quality of the wine to be good or bad

Evaluation Metrics

In this project I used the evaluation metric of fbeta_score. It measures the effectiveness of retrieval with respect to a user who attaches beta times as much importance to recall as precision. I didn’t choose accuracy score as my metric because my data is unbalanced.

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}.$$

II. Analysis

Data Exploration

In this section, I calculated number of Attributes and Instances in the data theme of the project is to determine the wine quality. Then I described the data to get some insights.

```
display(data.head(n=5))
```

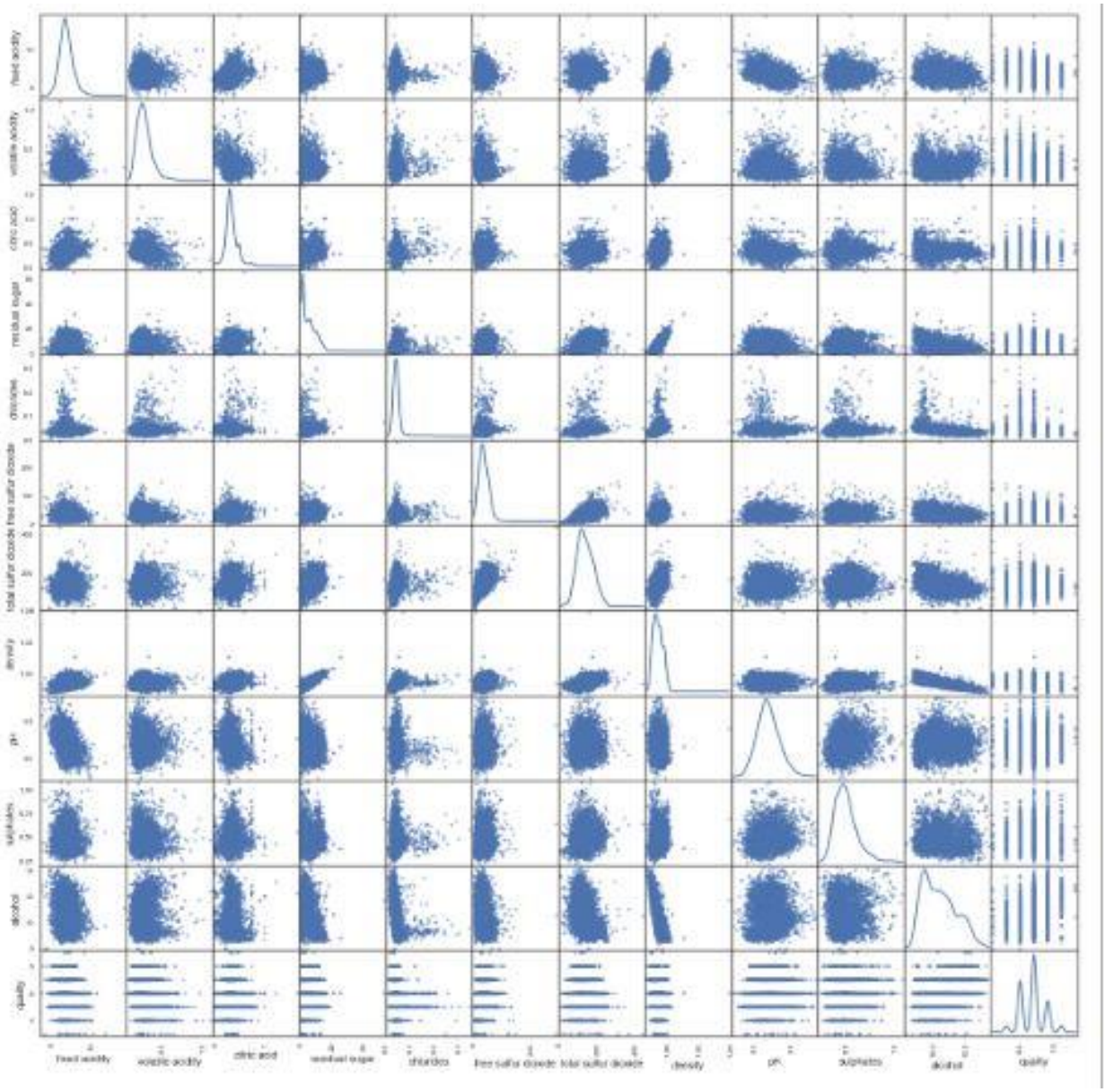
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

```
data.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267	5
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621	0
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000	3
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000	5
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000	6
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000	6
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000	9

Exploratory Visualization

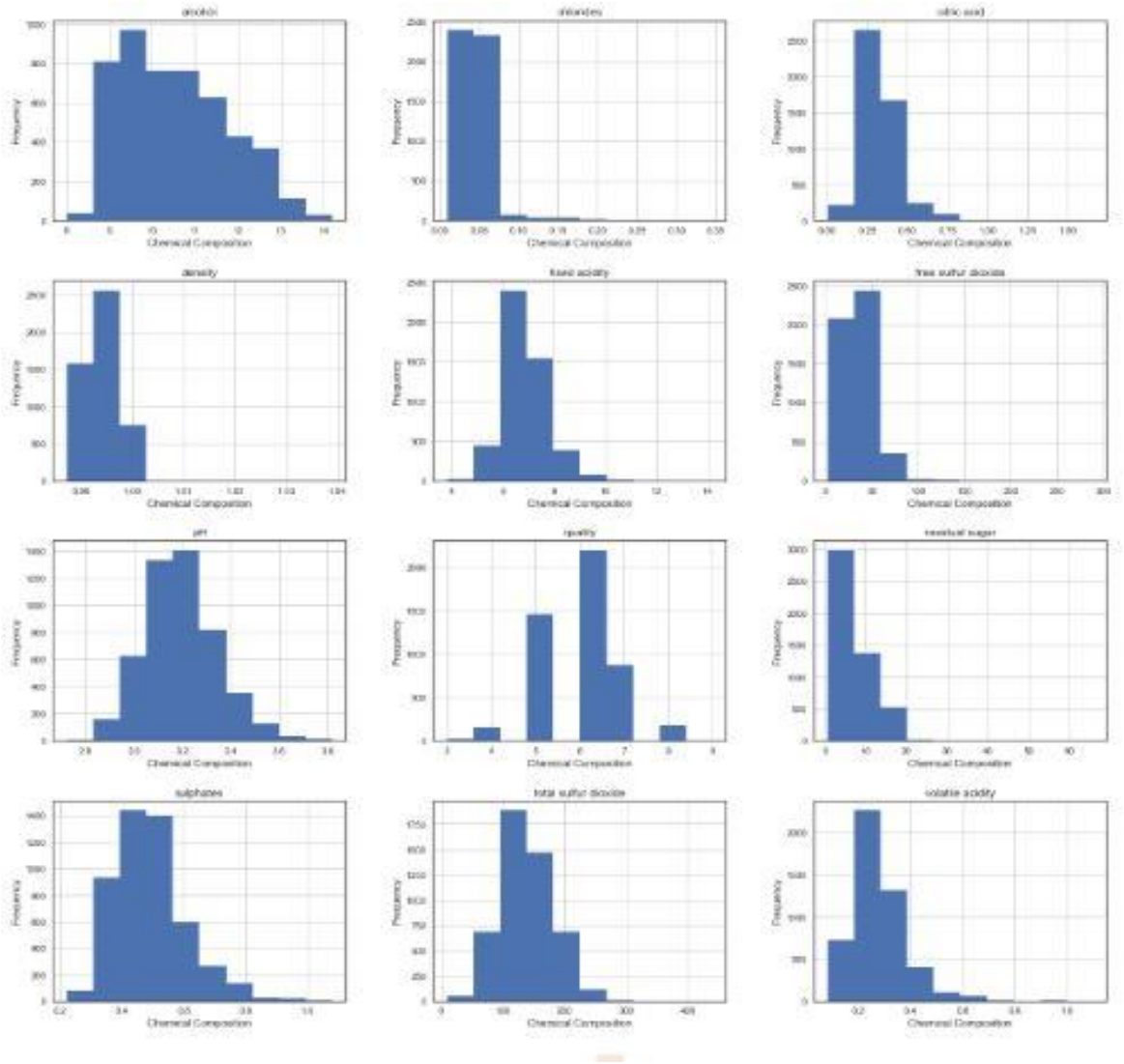
I plotted a scatter matrix to know the relationship (correlation) between the features.



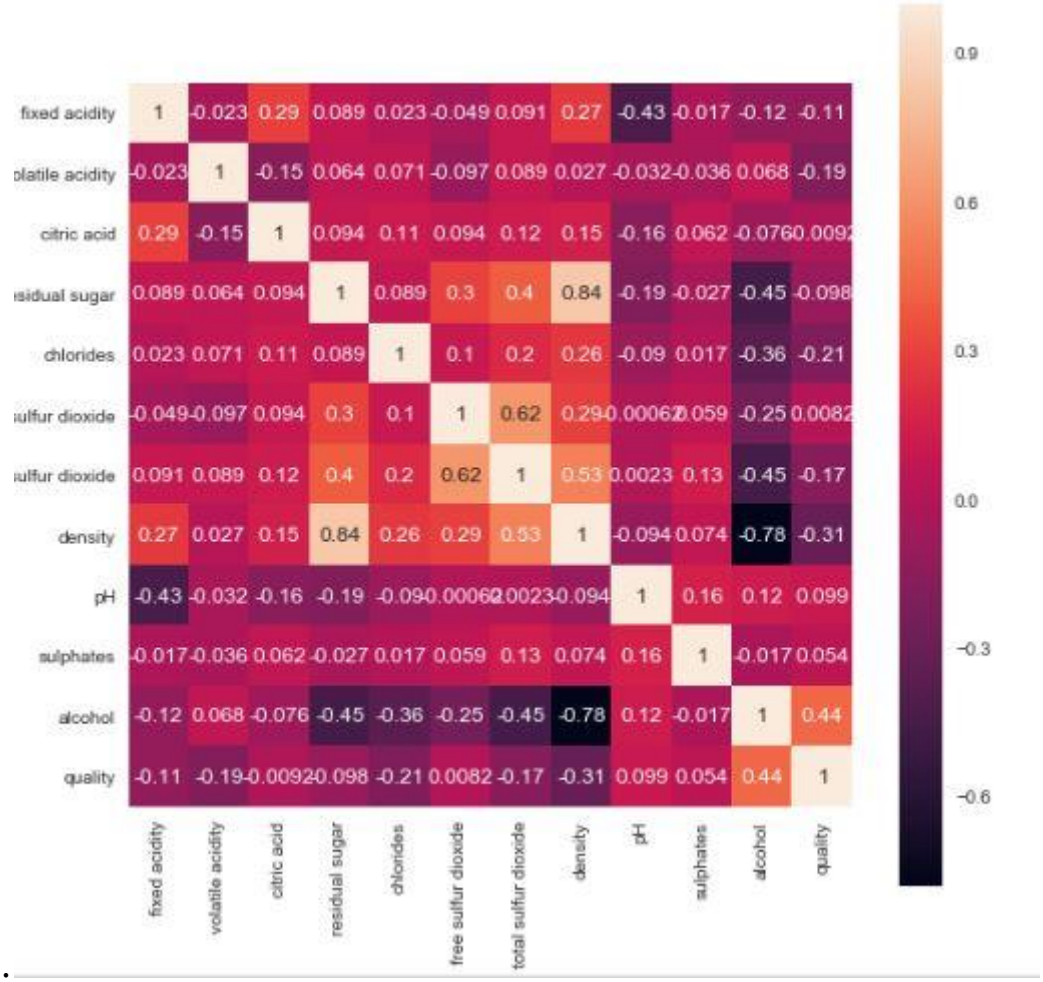
From the above visualizations , it is clear that most of the features are skewly – distributed.

Data Visualization:

Histograms of features:



From heatmap below, we can infer that there is both positive and negative correlation in between the features .Most of the features are not highly correlated . Consider some of the examples.



.

pH - Fixed Acidity : -0.43

It is reasonable because pH depends on acidity . It is negative correlation which means as acidity increases pH decreases.

Residual sugar - Density : 0.84

Residual sugar and Density are positively correlated . If there is more residual sugar , then density of wine also more.

Free sulfur di-oxide - Total sulfur di-oxide : 0.62

If the content of one feature increases ,then other feature will also increase . Therefore , there exists postive correlation.

Residual sugar - Total sulfur di-oxide : 0.4

If the content of one feature increases ,then other feature will also increase . Therefore , there exists postive correlation.

Alcohol - Quality : 0.44

If the content of one feature increases ,then other feature will also increase . Therefore , there exists postive correlation.

Alcohol - Density : -0.78

If the content of one feature increases ,then other feature will decrease . Therefore , there exists negative correlation.

Algorithms and Techniques

1. Logistic Regression : I have choosen Logistic Regression as it is very simple classification algorithm. It is more robust and to apply this algorithm , it doesn't require linear relationships exists in between predictors and target variables. The problem with this algorithm is it is more prone to over-fitting.

2. Decision Trees: This model is chosen as it converts our classification task into tree like structure which is simple to understand and interpret. Prediction is quite fast, easy visualization. Disadvantage is can take a lot of memory (with more features, decision tree is deeper and larger), overfitting happens very easily. Since, this model can handle both numerical and categorical data , able to handle lot of data easily and easy to visualize.

3. Adaboost: Adaboost is one of the ensemble model. An efficient algorithm which boosts the performance (predictive power) of a model by combining a set of weak learners into a single strong learner. Simple models can be combined to build a complex model, which is computationally fast and quite robust to overfitting. Ensemble methods are considered as high quality classifiers.This model produce more accurate predictions.

Data Preprocessing

Finding Missing Values

```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

There are no missing values present in the data

Outliers

```
for feature in data.keys():

    Q1 = np.percentile(data[feature], q=25)
    Q3 = np.percentile(data[feature], q=75)
    IQR = Q3 - Q1
    step = 1.5 * IQR
    print("Outliers for  '{}':".format(feature))
    display( data[~((data[feature] >= Q1 - step) & (data[feature] <= Q3 + step))] )
```

There are many outliers present in the data. But removing all these outilers may lead to loss of information. I am not going to remove them as manufacturers add extra chemicals(composition) to improve the taste and quality.

Converting quality variable to suit binary classification task

Splitting data:

```
X_train , X_test , y_train , y_test = train_test_split( X , y , test_size = 0.2 , random_state =0 )
print(X_train.shape , y_train.shape)
print(X_test.shape , y_test.shape)
print("Training set has {} samples.".format(X_train.shape[0]))
print("Testing set has {} samples.".format(X_test.shape[0]))
```

```
((3918, 11), (3918,))
((980, 11), (980,))
Training set has 3918 samples.
Testing set has 980 samples.
```

Standardising data

```
# Standardizing the data


scaler = MinMaxScaler().fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
print X_train_scaled.mean(axis=0)
print X_test_scaled.mean(axis=0)
```

```
[ 0.29412696  0.19489485  0.20062178  0.08804495  0.12995591  0.11579185
  0.29950795  0.13311767  0.4148186   0.3132976   0.40502904]
[ 0.29214089  0.19219188  0.20411237  0.09194551  0.13448689  0.11711228
  0.30267058  0.13432817  0.41603364  0.31568581  0.40751756]
```

Benchmark model

I will use Logistic regression is used as benchmark model. Fbeta_score of benchmark model is used as reference and other model will be judge to perform better if their fbeta score will be greater than Logistic regression model.

Implementation

Performance of Benchmark model

```
def performance(clf, X_train_scaled, y_train, X_test_scaled, y_test):

    print 'Classifier:',clf.__class__.__name__
    clf.fit(X_train , y_train)
    y_preds = clf.predict(X_test)
    print 'F-score is',fbeta_score( y_test , y_preds , beta = 0.5)
```

```
clf1 = LogisticRegression(random_state = 10)
performance(clf1 , X_train_scaled, y_train, X_test_scaled, y_test)
```

```
Classifier: LogisticRegression
F-score is 0.803256445047
```

Testing other Algorithms

```
clf2 = DecisionTreeClassifier(random_state = 0)
performance(clf2 , X_train_scaled, y_train, X_test_scaled, y_test)
clf3 = AdaBoostClassifier(random_state = 30)
performance(clf3 , X_train_scaled, y_train, X_test_scaled, y_test)
```

Classifier: DecisionTreeClassifier
F-score is 0.838176812481
Classifier: AdaBoostClassifier
F-score is 0.808425846855

F - score score for AdaBoost classifier & Decision Tree classifier are almost similar. I am going to choose Ada Boost Classifier as my final model because it has more hyper parameters than Decision Trees which may increase F - score.

Hyper Parameter Tuning using Grid Search

The final model AdaBoost Classifier with hyper parameter tuning using Grid Search achieved an F - score of 82%.

```
clf = AdaBoostClassifier()

n_params = { 'n_estimators' : [10 , 50 , 75 ] , 'learning_rate' : [0.1 , 0.2, 0.3 , 0.5] }
scorer = make_scorer( fbeta_score , beta=0.5 )
grid = GridSearchCV(clf , param_grid=n_params , scoring=scorer )
best_model  = grid.fit( X_train , y_train ).best_estimator_
```

```
performance(best_model , X_train_scaled, y_train, X_test_scaled, y_test)
y_preds = best_model.predict(X_test)
```

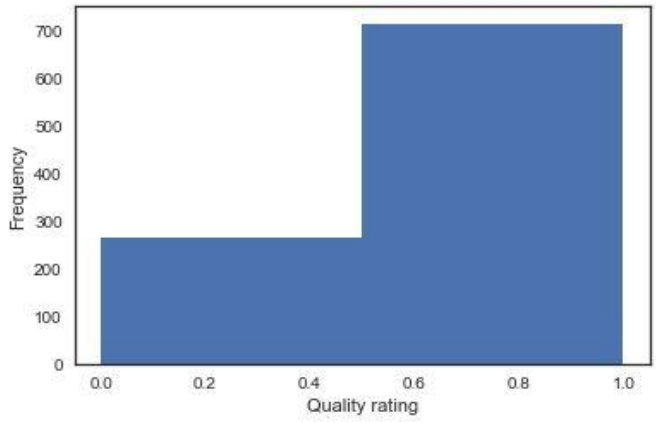
Justification:

Metric	Benchmark model	Un-optimized model	Optimized model
F-score	0.8032	0.8084	0.8203

we can clearly say that our final model works pretty well than our Benchmark model . As our final model got F-score of 82% the model is quite significant to solve the problem.

Free-Form Visualization:

data contains more examples of good quality wine than bad quality wine.



Reflection

First , I read the data using Pandas DataFrame.Then I defined my problem statement as to predict the quality of wine. I found the no. of instances and attributes present in the data at the Data exploration step. I have learn using heatmap and used it to check whether the features are correlated or not. I tried to find missing values but no missing values are found. Then I tried to find out Outliers in the data.I defined a Benchmark model which is Logistic Regressor. Fbeta_score of benchmark model is used as reference and other model will be judge to perform better if their fbeta score will be greater than Logistic regression model. After ,I haven chosen Decision Tree and Ada Boost Classifier and tried to improve the performance of Ada Boost Classifier with Grid Search technique.I liked this project.

Improvements:

Using CNN architecture,we can achieve better performance.More balanced data would increase performance. The algorithmms like XGBoost, gradient descent may increase performance. Features like Clever Penalisation of Trees, Proportional shrinking of leaf nodes made XGBoost popular.

References

http://benalexkeen.com/feature-scaling-with-scikit-learn/

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.fbeta_score.html

https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.histogram.html

https://stackoverflow.com/questions/43972304/how-to-use-different-axis-scales-in-pandas-dataframe-plot-hist

https://stackoverflow.com/questions/25440008/python-pandas-flatten-a-dataframe-to-a-list

https://seaborn.pydata.org/generated/seaborn.heatmap.html

https://recast.ai/blog/machine-learning-algorithms/2/
https://hackernoon.com/boosting-algorithms-adaboost-gradient-boosting-and-xgboost-f74991cad38c