WINE QUALITY PREDICTION

Using DecisionTree and Neural Network.

Manish Meshram & Supratik Chanda

INTRODUCTION

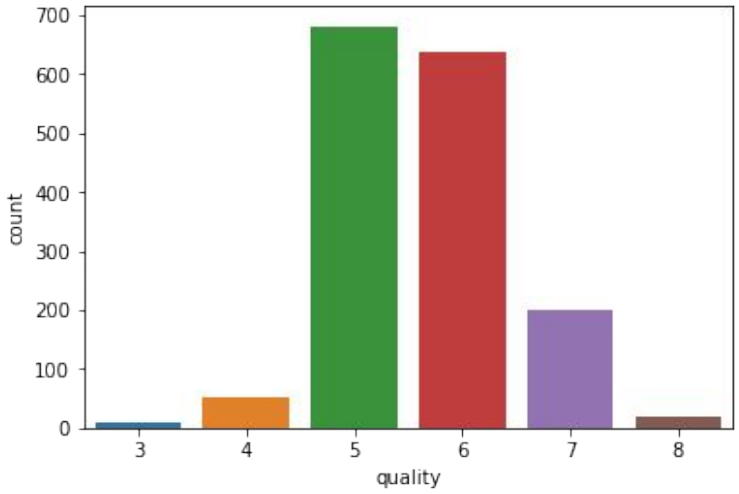
- Red wine dataset
- Prediction of quality of wine using Decision Trees and Neural Networks

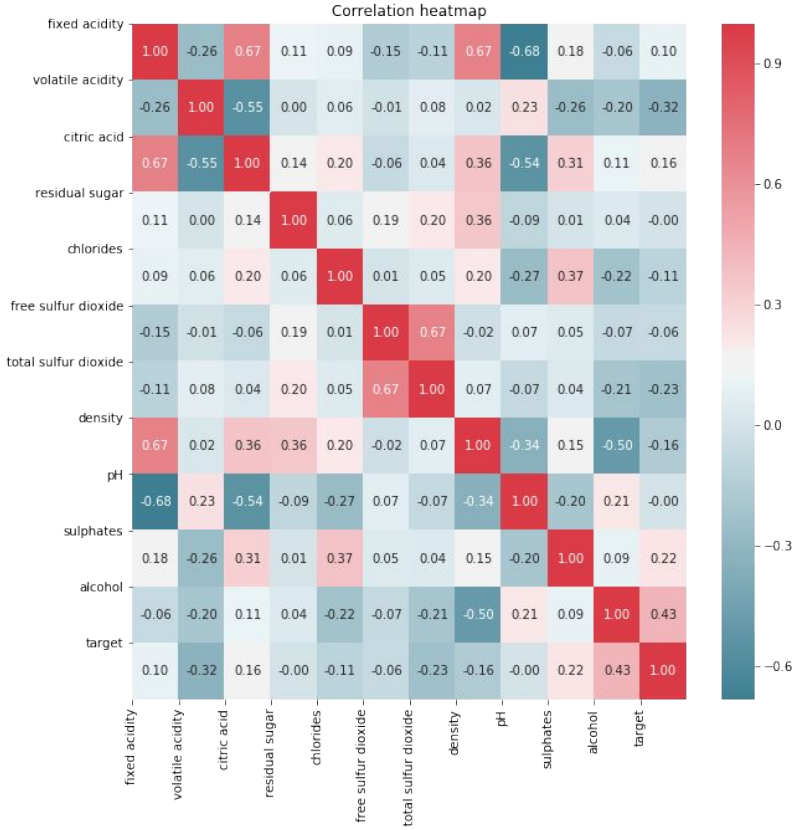
DATASET

- Independent variables:
  - fixed acidity
  - volatile acidity
  - citric acid
  - residual sugar
  - chlorides
  - free sulfur dioxide
  - total sulfur dioxide
  - density
  - pH
  - sulphates
  - alcohol
- Dependent variable is quality score from 1 to 10 (10 being the highest)
- 1599 instance of data

EXPLORATORY DATA ANALYSIS

Frequency count plot

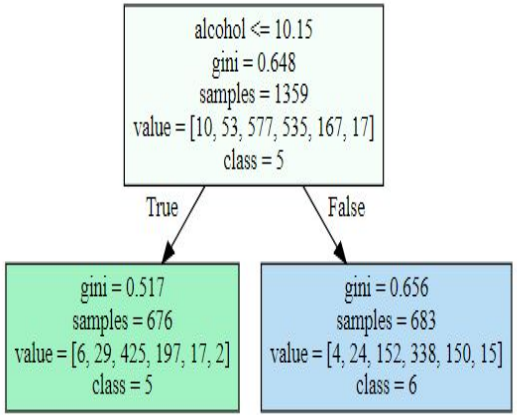




DECISION TREES
MAX_DEPTH AND MAX_LEAF_NODES ANALYSIS

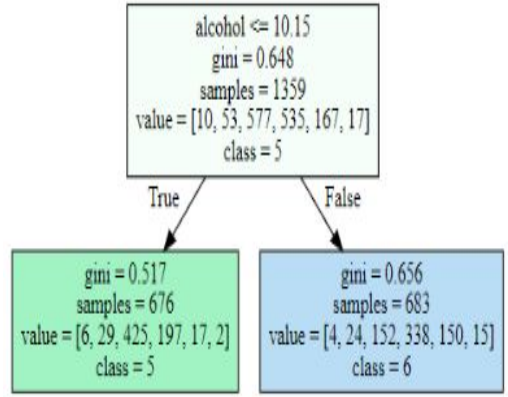
```
dTree_Max_Leaf_Nodes_1= DecisionTreeClassifier(criterion='gini',max_depth=5,max_leaf_nodes=2,min_samples_leaf=10,random_stat
dTree_Max_Leaf_Nodes_1.fit(X,Y)
graph = Source(tree.export_graphviz(dTree_Max_Leaf_Nodes_1,feature_names=list(FeatureDTFrame.Features[0:10]),
                                    class_names=['3','4','5','6','7','8'],filled=True))
tempGraph = graph.pipe(format='svg')
tempSVG=SVG(tempGraph)
print('Max_depth is 5 and max_leaf_nodes is 2 while min_samples_leaf is 10')
display(tempSVG)
```

```
Max_depth is 5 and max_leaf_nodes is 2 while min_samples_leaf is 10
```



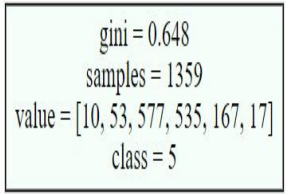

```
dTree_Max_Depth_1= DecisionTreeClassifier(criterion='gini',max_depth=1,min_samples_leaf=10,random_state=4)
dTree_Max_Depth_1.fit(X,Y)
graph = Source(tree.export_graphviz(dTree_Max_Depth_1,feature_names=list(FeatureDTFrame.Features[0:10]),
                                    class_names=['3','4','5','6','7','8'],filled=True))
tempGraph = graph.pipe(format='svg')
tempSVG=SVG(tempGraph)
print('Max_depth is 1  while min_samples_leaf is 10')
display(tempSVG)
```

```
Max_depth is 1  while min_samples_leaf is 10
```

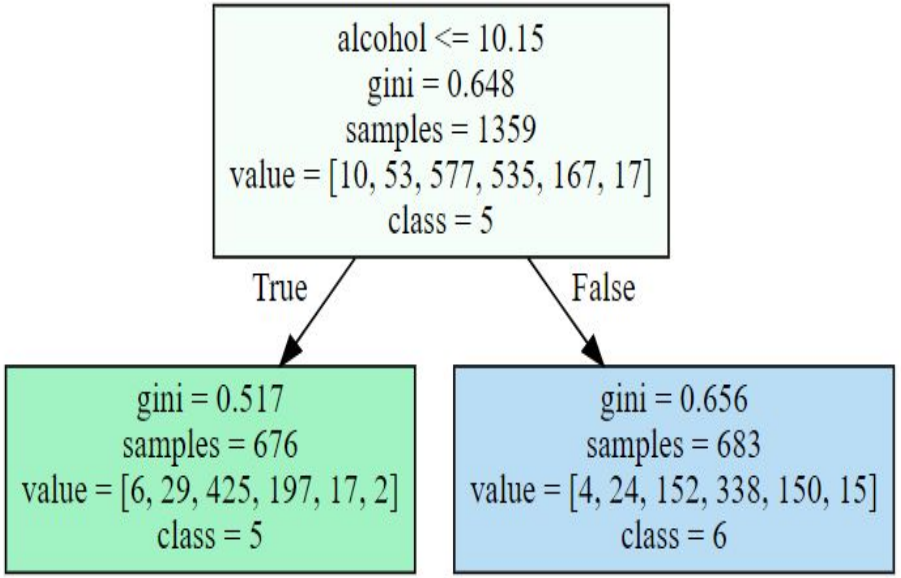


DECISION TREES
MINIMUM_LEAF_SAMPLES

Max_depth is 3 and max_leaf_nodes is 10 while min_samples_leaf is 1000



Max_depth is 3 and max_leaf_nodes is 10 while min_samples_leaf is 500



DECISION TREES
F1_Score before and after Predictor Variable Modification

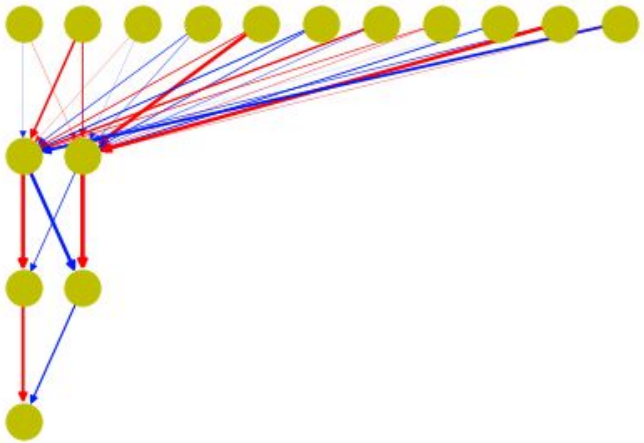
The accuracy score is 0.6029411764705882

	precision	recall	f1-score	support
3	0.00	0.00	0.00	2
4	0.00	0.00	0.00	11
5	0.61	0.87	0.72	116
6	0.59	0.44	0.51	107
7	0.57	0.48	0.52	33
8	0.00	0.00	0.00	3
micro avg	0.60	0.60	0.60	272
macro avg	0.30	0.30	0.29	272
weighted avg	0.56	0.60	0.57	272

The accuracy score is 0.796875

	precision	recall	f1-score	support
0	0.78	0.79	0.78	149
1	0.82	0.80	0.81	171
micro avg	0.80	0.80	0.80	320
macro avg	0.80	0.80	0.80	320
weighted avg	0.80	0.80	0.80	320

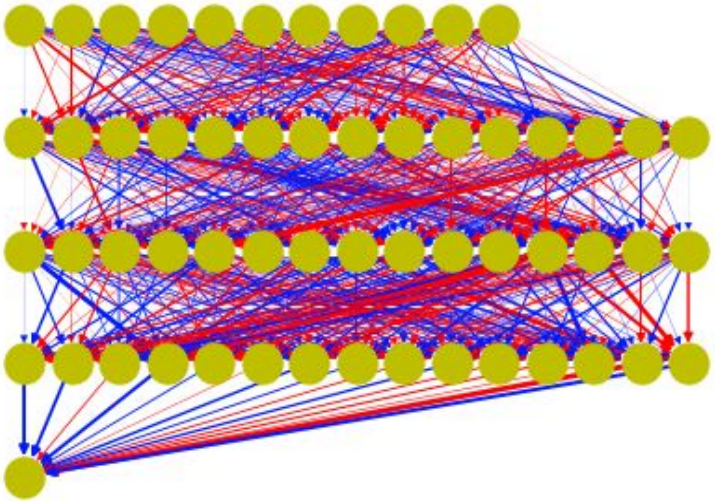
NEURAL NETWORK 1



Architecture: 11 x 2 x 2 x 1
Solver: Adam
Activation function: ReLU
Maximum iterations: 500

	precision	recall	f1-score	support
0.0	0.67	0.73	0.70	178
1.0	0.77	0.71	0.74	222
avg / total	0.72	0.72	0.72	400

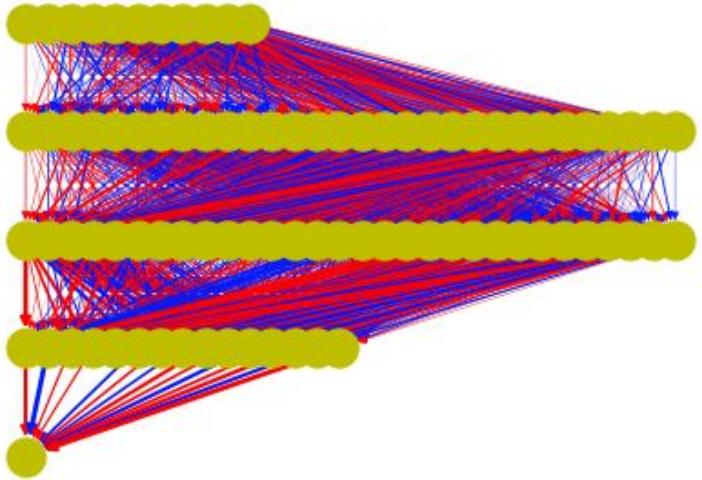
NEURAL NETWORK 2



Architecture: 11 x 15 x 15 x 15 x 1
Solver: Adam
Activation function: ReLU
Maximum iterations: 10000

	precision	recall	f1-score	support
0.0	0.75	0.75	0.75	178
1.0	0.80	0.80	0.80	222
avg / total	0.78	0.78	0.78	400

NEURAL NETWORK 3



Architecture: 11 x 30 x 30 x 15 x 1
Solver: Adam
Activation function: ReLU
Maximum iterations: 10000

	precision	recall	f1-score	support
0.0	0.78	0.79	0.79	178
1.0	0.83	0.82	0.83	222
avg / total	0.81	0.81	0.81	400