KNN on Wine dataset

Pavani Komati | CSE 5243 | 3rd Feb'18

Table of Contents

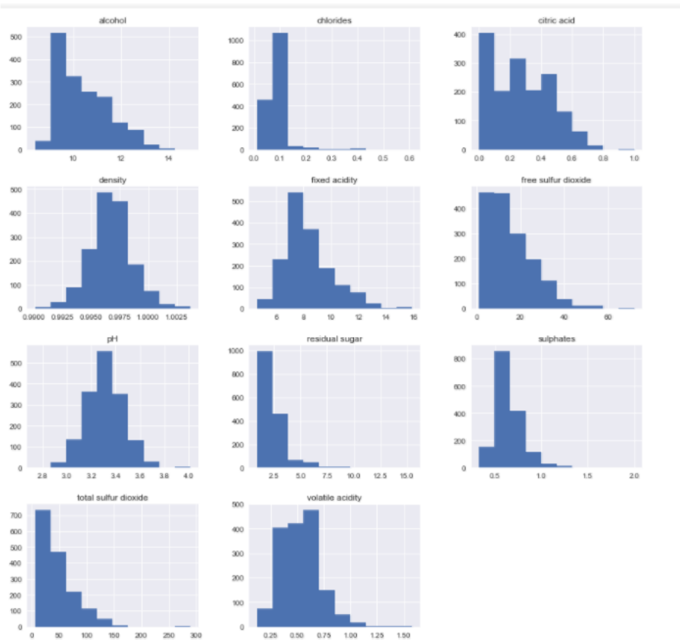
1.INTRODUCTION

This report contains implementation and analysis of KNN algorithm on wine dataset. KNN algorithm predicts the outcome of each record in test data based on its k closest neighbors from the training data. The implementation consists of calculating the proximity measures of test data from the train data, choosing appropriate k value and some data transformations on the given data set.

2.DATA ANALYSIS

The dataset consists of 1599 records with 11 features. These features are spread out with varied distribution(range) as shown below. The aim of the project is to determine the quality of wine with the help of these 11 features.

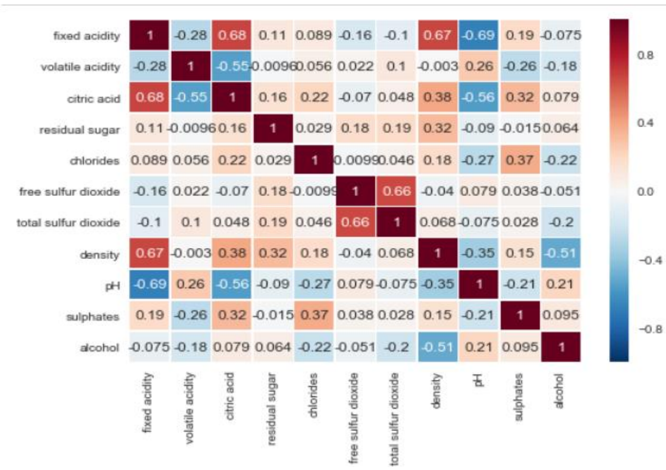
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.
mean	8.319637	0.527821	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.



Distribution of data in features (fig 1)

RELATION BETWEEN FEATURES

The correlation map between features (after normalizing the values) shows that citric acid and fixed acidity features are one of the highly correlated values followed by density -fixed acidity and total sulfur dioxide - free sulfur dioxide.



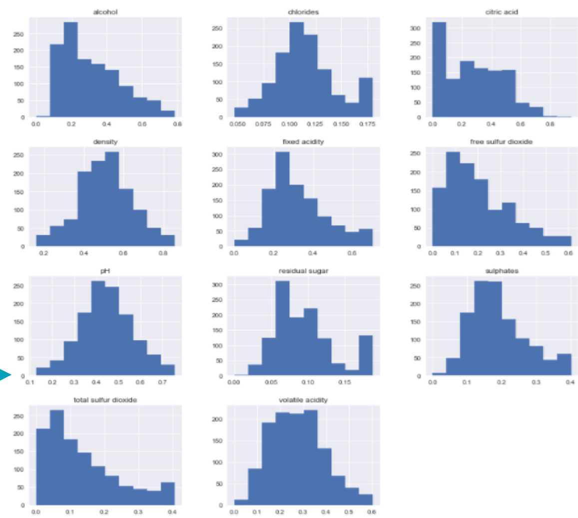
3.DATA TRANSFORMATION

The quality column contains continuous values in the range 1-10. As KNN classifies data as categorical type, the quality column is transformed as Binary value with quality <= 5 as 'Low' and quality >5 as 'High'.

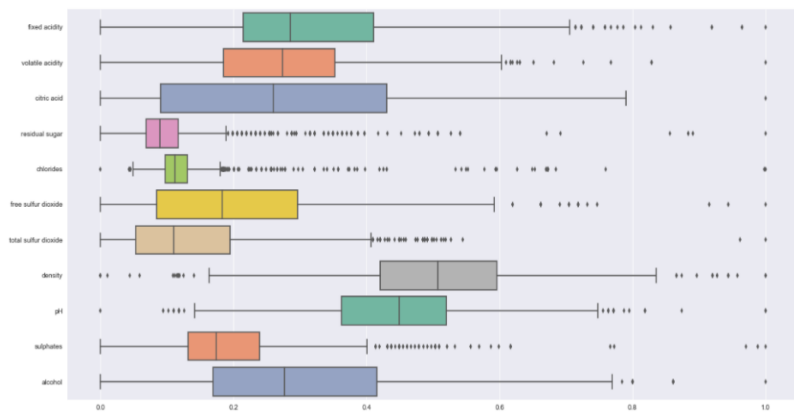
The histogram in fig 1 shows that the data is not distributed evenly across the features.

KNN algorithm requires proximity measure to predict the outcome. As the given data distribution is not uniform across the features, the proximity measures might not be accurate with the current values. Hence values are normalized to range between [0-1] based on min-max values of training data.

After normalizing, the data is distributed evenly when compared to the distribution in fig 1.



Outliers in the data



From the box plot of the features, it can be observed that data consists of too many outliers.

Do we keep the outliers or not??

- Currently, we are not sure if these values are correct or not. Hence, we can test the performance of the model with and without these outliers and determine the correctness of these values.

4.MODEL DEVELOPMENT AND EVALUATION

A. TRAINING AND TEST DATA

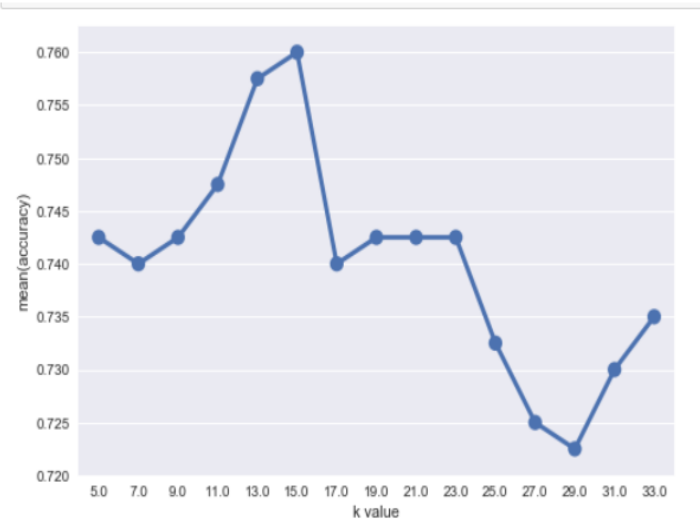
As the data consists mixture only binary outcomes, it is ok to split the data randomly. Hence, the given data is split into 75% of training data and 25% of test data randomly with a seed of 150. The training data consists of 1200 records and test data consists of 399 records.

B. PROXIMITY MEASURE AND CALCULATION

In this implementation Euclidean distance is chosen as the proximity measure. KNN algorithm calculates the distances of each test record to all the training records and classifies the data based on the k closest neighbors' class. The current algorithm computes distance of every training record from the given test record and saves the closest 35 neighbor's quality as a list for every test record. An approximate of sqrt(number of train records) can be considered as ideal value to predict the outcome. Hence, from these 35 records, performance can be determined with various k values ranging from 1-35.

C. DETERMINING K- VALUE

It is very important to choose appropriate k value to determine the correct output. This value shouldn't be too small or too large. It must be chosen such that the model should be able to predict the class labels with high accuracy. Hence, the optimal k-value is determined by using different k-values (odd numbers in this case, as it is a binary classifier) and take choose the k which classifies maximum number of records correctly.



Accuracy for every k value is calculated as (number of test records correctly classified)/(total test number of records).

From the graph it is observed that for **k=15** maximum number of test records are correctly classified (76%). Hence, the current model is built with a k value of 15.

	k value	accuracy
0	5.0	0.7425
1	7.0	0.7400
2	9.0	0.7425
3	11.0	0.7475
4	13.0	0.7575
5	15.0	0.7600
6	17.0	0.7400
7	19.0	0.7425
8	21.0	0.7425
9	23.0	0.7425
10	25.0	0.7325
11	27.0	0.7250
12	29.0	0.7225
13	31.0	0.7300
14	33.0	0.7350

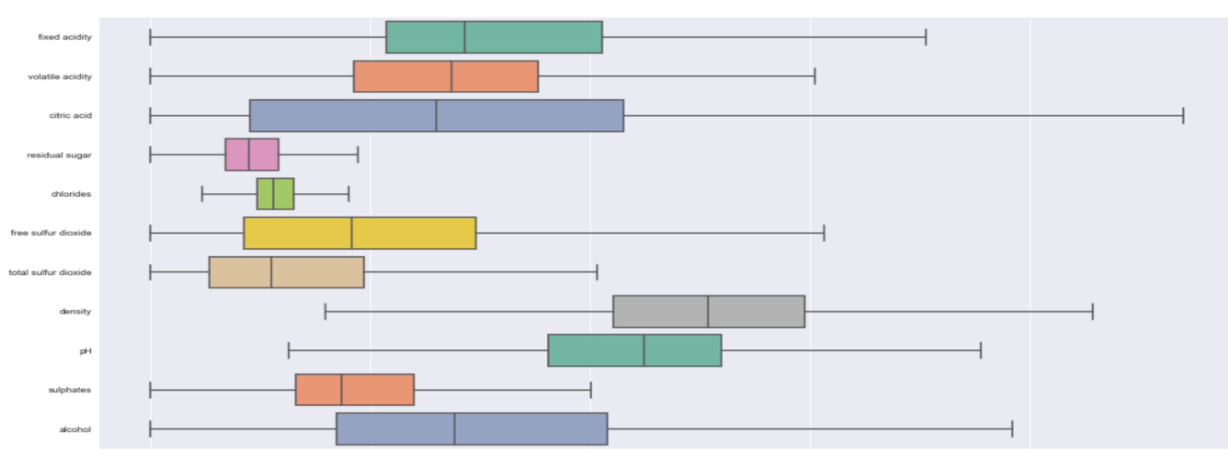
D.REMOVAL OF FEATURES??

From the correlation plot it is observed that (citric acid -fixed acidity), (density -fixed acidity) and (total sulfur dioxide - free sulfur dioxide) are strongly correlated. This indicates, only one feature in each pair can be useful. But this assumption might not be true in all the cases. This can be verified by dropping 'density', 'free sulfur dioxide' and 'citric acid' and determine the model performance. After this feature removal it is observed that the performance of the model has not improved (75% from 76%). Hence, removal of features is not recommended in this data set.

E. REMOVAL OF OUTLIERS??

As it is discussed earlier, the outliers present in the data need not necessarily indicate incorrect values. This can be verified by changing the outlier values and predicting the model performance.

The maximum and minimum values of each feature is calculated using the interquartile range values and outliers greater than maximum are replaced with the maximum value and outliers less than the minimum value are replaced by the minimum value.

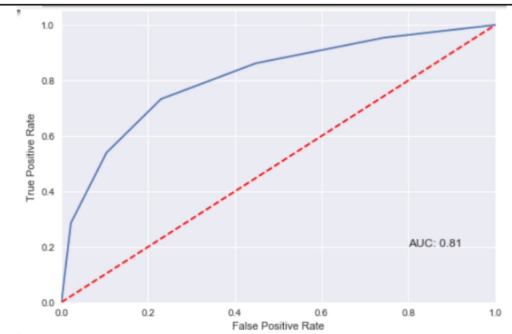
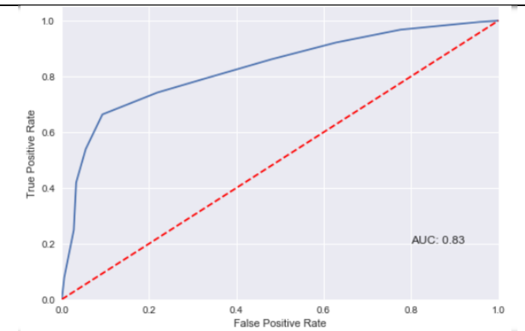
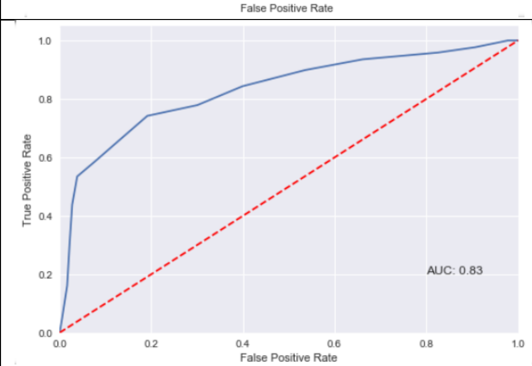
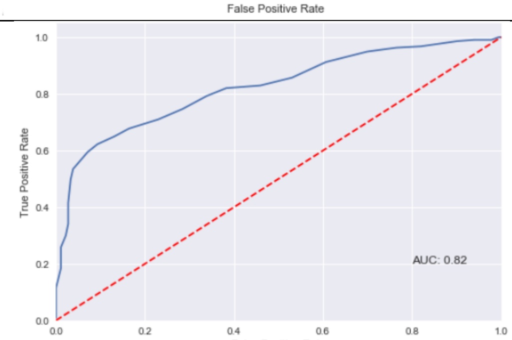


Features after removal of outliers

After the removal of outliers, the accuracy of the model has improved from 76 to 77.25%. Since, the performance hasn't changed drastically, only some of these outliers could be incorrect values.

F. MODEL EVALUATION FOR VARIOUS VALUES OF K

Analysis of the model is done by using various values of k: 5, 11, 15 and 29

<table>
<tr><th>K</th><th>Confusion Matrix</th><th>Analysis</th><th>ROC curve</th></tr>
<tr><td>5</td><td><table><tr><th></th><th colspan="3">Predicted class</th></tr><tr><td rowspan="3">Actual class</td><td></td><td>High</td><td>Low</td></tr><tr><td>High</td><td>159</td><td>42</td></tr><tr><td>Low</td><td>42</td><td>58</td></tr></table></td><td>----------------------------
True Positive: 159
False Positive: 42
True Negative: 141
False Negative: 58
Precision: 0.791044776119
Recall: 0.732718894009
F-Measure: 0.760765550239
Classification rate: 0.75
Error rate: 0.25
----------------------------</td><td></td></tr>
<tr><td>11</td><td><table><tr><th></th><th colspan="3">Predicted class</th></tr><tr><td rowspan="3">Actual class</td><td></td><td>High</td><td>Low</td></tr><tr><td>High</td><td>161</td><td>40</td></tr><tr><td>Low</td><td>56</td><td>143</td></tr></table></td><td>----------------------------
True Positive: 161
False Positive: 40
True Negative: 143
False Negative: 56
Precision: 0.800995024876
Recall: 0.741935483871
F-Measure: 0.77033492823
Classification rate: 0.76
Error rate: 0.24
----------------------------</td><td></td></tr>
<tr><td>15</td><td><table><tr><th></th><th colspan="3">Predicted class</th></tr><tr><td rowspan="3">Actual class</td><td></td><td>High</td><td>Low</td></tr><tr><td>High</td><td>161</td><td>35</td></tr><tr><td>Low</td><td>56</td><td>148</td></tr></table></td><td>----------------------------
True Positive: 161
False Positive: 35
True Negative: 148
False Negative: 56
Precision: 0.821428571429
Recall: 0.741935483871
F-Measure: 0.779661016949
Classification rate: 0.7725
Error rate: 0.2275
----------------------------</td><td></td></tr>
<tr><td>29</td><td><table><tr><th></th><th colspan="3">Predicted class</th></tr><tr><td rowspan="3">Actual class</td><td></td><td>High</td><td>Low</td></tr><tr><td>High</td><td>154</td><td>42</td></tr><tr><td>Low</td><td>141</td><td>63</td></tr></table></td><td>----------------------------
True Positive: 154
False Positive: 42
True Negative: 141
False Negative: 63
Precision: 0.785714285714
Recall: 0.709677419355
F-Measure: 0.745762711864
Classification rate: 0.7375
Error rate: 0.2625
----------------------------</td><td></td></tr>
</table>

From this data, it is observed that very low values of k or high values of k have lower precision than the optimal value of k. With k =15, the model has achieved highest accuracy of 77.25% and a precision of 82.1%. Also, for k =15 ROC curve has the maximum AUC value of 0.83.

For k =11 it is observed that the True Positive rate is increasing only for a shorter length indicating the lower accuracy of the model compared to other k values. Similarly for k =15, the True positive rate of the curve is increasing at a higher rate (closer to the y axis) indicating higher accuracy compared to the other k values.

G. FINAL RESULTS

	Actual Class	Predicted Class	Posterier Probability
0	Low	Low	0.333333
1	Low	Low	0.400000
2	Low	Low	0.400000
3	Low	Low	0.000000
4	Low	Low	0.466667
5	High	Low	0.400000
6	Low	Low	0.333333
7	Low	Low	0.333333
8	Low	Low	0.266667
9	High	Low	0.333333
10	Low	Low	0.200000
11	Low	Low	0.400000
12	Low	Low	0.200000
13	Low	High	0.800000
14	Low	Low	0.066667
15	Low	Low	0.466667
16	Low	Low	0.133333

The final results of the test data with k =15 are calculated with the Posterior Probability as P(High/x).

If the Posterior Probability is >0.5 the record is classified as 'High' and if the Probability is <=0.5 record is classified as 'Low'

H. OFF THE SHELF COMPARISION

KNeighborsClassifier from sklearn.neighbors is used for off the shelf comparision of the model. The model is built with the normalized data which is used for the current classifier.

Like the current model, KNeighborsClassifier is tested with different values of k and it is observed that it gives highest accuracy with k =15.

SURPRISING RESULTS!!

KNeighborsClassifier	Current Classifier
True Positive: 161 False Positive: 35 True Negative: 148 False Negative: 56 Precision: 0.821428571429 Recall: 0.741935483871 F-Measure: 0.779661016949 Classification rate: 0.7725 Error rate: 0.2275	---------------------------- True Positive: 161 False Positive: 35 True Negative: 148 False Negative: 56 Precision: 0.821428571429 Recall: 0.741935483871 F-Measure: 0.779661016949 Classification rate: 0.7725 Error rate: 0.2275 ----------------------------

The current model accuracy matches exactly with that of KNeighboursClassifier with same value of k.

The only difference in the dataset is that, current model performs classification on the data after removal of outliers. Whereas, the KNeighborsClassifier uses only the standardized data.

One drawback in the current model is, the time taken to perform the computation. This might be due to the fact that, the rows are not vectorized for distance calculation. ()