Performance Evaluation of Classifiers on Car Evaluation Dataset

Harshada Phadol

1. Dataset Description:

The dataset used in this study which is a collection of the records on specific attributes on cars donated by Marco Bohanec in 1997 was obtained from the UCI dataset repository (http://archive.ics.uci.edu/ml/datasets/Car+Evaluation).

Dataset Characteristics	**Multivariate**
Attribute Characteristics	Categorical
Associated Tasks	Classification
Number of Instances	1728
Number of Attributes	6
Missing Values	No

Table 1.1 Car Evaluation Dataset

1.1 Class Attributes:

Attributes	**Identified as**
Acceptable	Acc
Good	Good
Unacceptable	Unacc
Very Good	Vgood

Table 1.2 Class Attributes

1.2 Number of instances per class:

Class	**No of records**
Acceptable	384 (22.222%)
Good	69 (3.993 %)
Unacceptable	1210 (70.023%)
Very Good	65 (3.762 %)

Table 1.3 Class Distribution

2. Data Cleaning:

We have converted nominal attributes into numeric attributes. This conversion is essential for data normalization.

Column	Old value	New Numeric value
Buying	vhigh	0
	high	1
	med	2
	Low	3

Table 2.1 Numeric conversion for Buying column

Column	Old value	New Numeric value
Maintenance	vhigh	0
	high	1
	med	2
	low	3

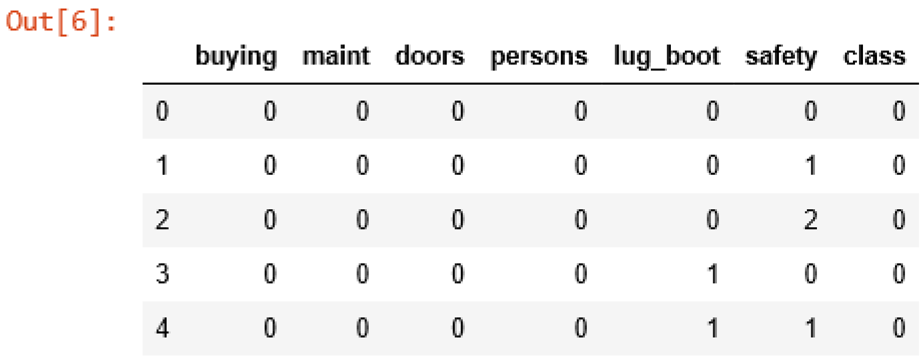
Table 2.2 Numeric conversion for Maintenance column

Column	Old value	New Numeric value
Luggage Boot	small	0
	med	1
	Big	2

Table 2.3 Numeric conversion for Luggage Boot column

Column	Old value	New Numeric value
Safety	Low	0
	med	1
	high	2

Table 2.4 Numeric conversion for Safety column



Out[6]:

	buying	maint	doors	persons	lug_boot	safety	class
0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0
2	0	0	0	0	0	2	0
3	0	0	0	0	1	0	0
4	0	0	0	0	1	1	0

Figure 2.1 Dataset after Numerical Conversion

3. Classification:

   We are using three classification techniques

   1. Decision Tree Classification
   2. Random Forest Classification
   3. Naïve Bayes Classification

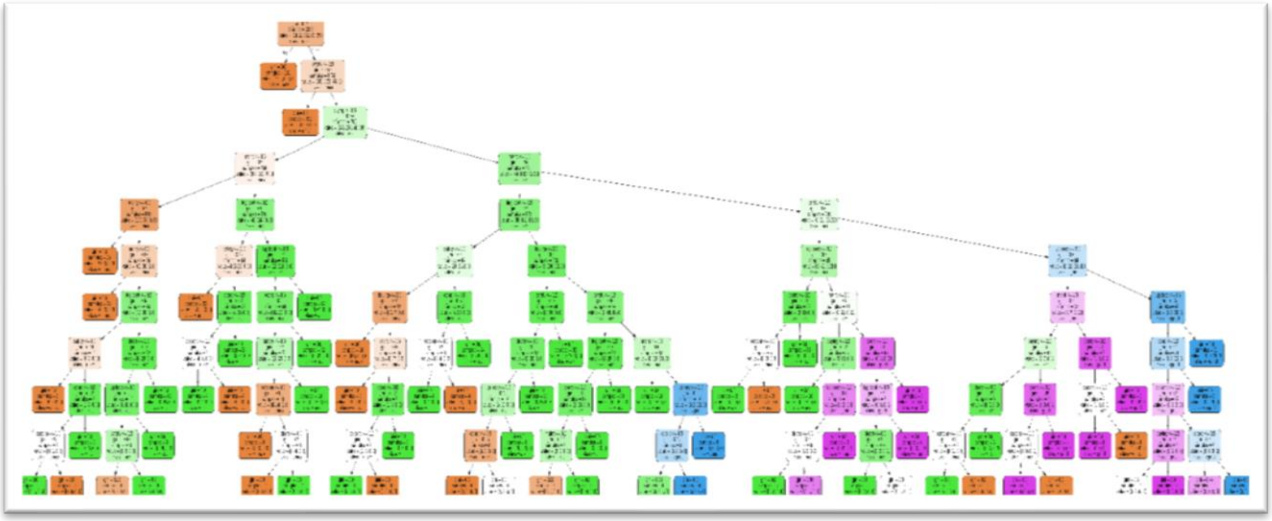
3.1 Decision Tree Classification:

In decision tree classification, it is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values. ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely same the entropy is zero and if the sample is an equally divided it has entropy of one.

As mentioned in below table, we can see the accuracy of Decision tree.

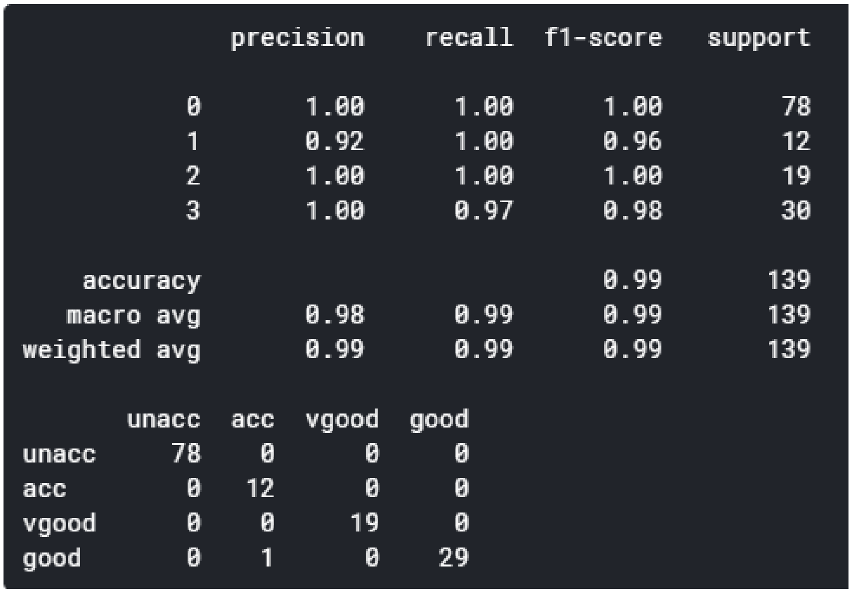
With 92% of training data we can get 99% of accuracy.

Train Test Split				**Decision Tree**			
Training	Testing	Training %	Testing %	Correct	Incorrect	Correct %	Incorrect %
1589	139	92	8	138	1	99	1
1261	208	73	12	204	4	98	2
760	208	44	12	180	28	87	13
501	208	29	12	169	39	81	19



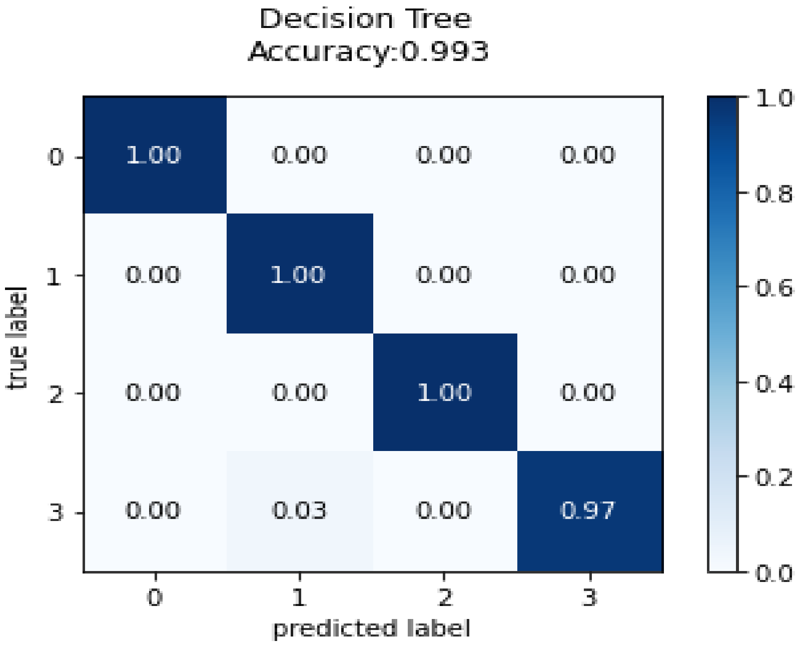
3.1.1 Detailed Classification Report: Decision Tree

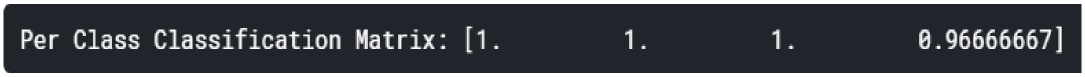
Train data: 1589 (92%)
Test data: 139 (8%)



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        78
           1       0.92      1.00      0.96        12
           2       1.00      1.00      1.00        19
           3       1.00      0.97      0.98        30

    accuracy                           0.99       139
   macro avg       0.98      0.99      0.99       139
weighted avg       0.99      0.99      0.99       139

       unacc  acc  vgood  good
unacc     78    0      0     0
acc        0   12      0     0
vgood      0    0     19     0
good       0    1      0    29
```

Figure 3.1.1 Classification Report



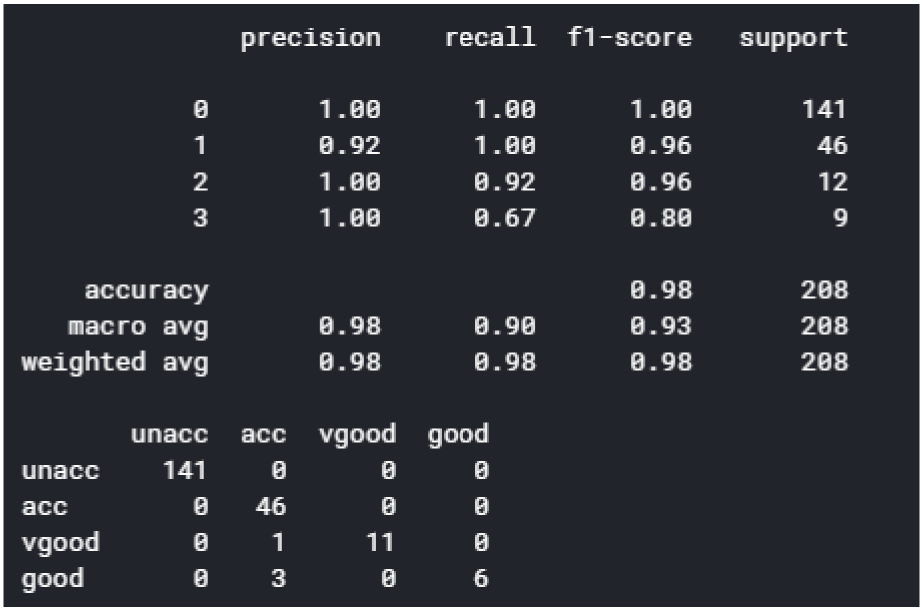


```
Per Class Classification Matrix: [1.         1.         1.         0.96666667]
```

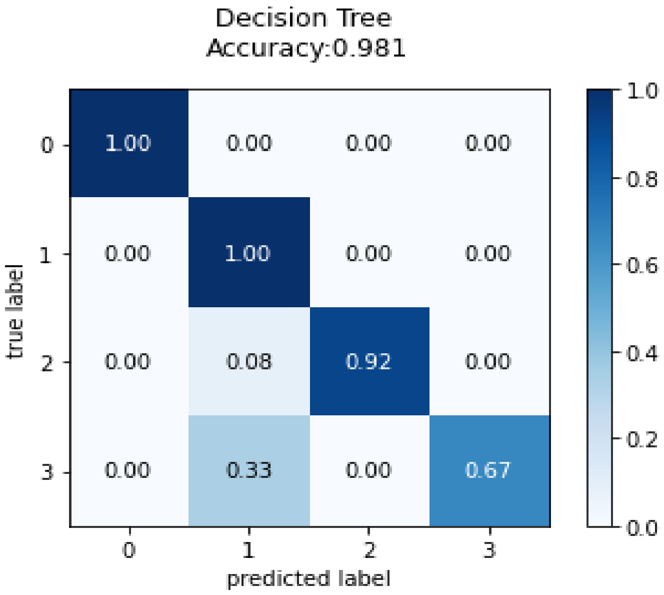
3.1.2 Detailed Classification Report: Decision Tree

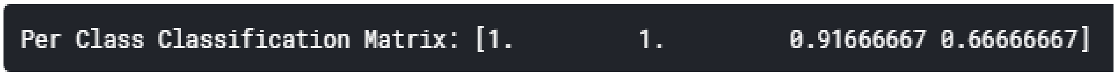
Train data: 1261(73%)

Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       141
           1       0.92      1.00      0.96        46
           2       1.00      0.92      0.96        12
           3       1.00      0.67      0.80         9

    accuracy                           0.98       208
   macro avg       0.98      0.90      0.93       208
weighted avg       0.98      0.98      0.98       208

       unacc  acc  vgood  good
unacc    141    0      0     0
acc        0   46      0     0
vgood      0    1     11     0
good       0    3      0     6
```



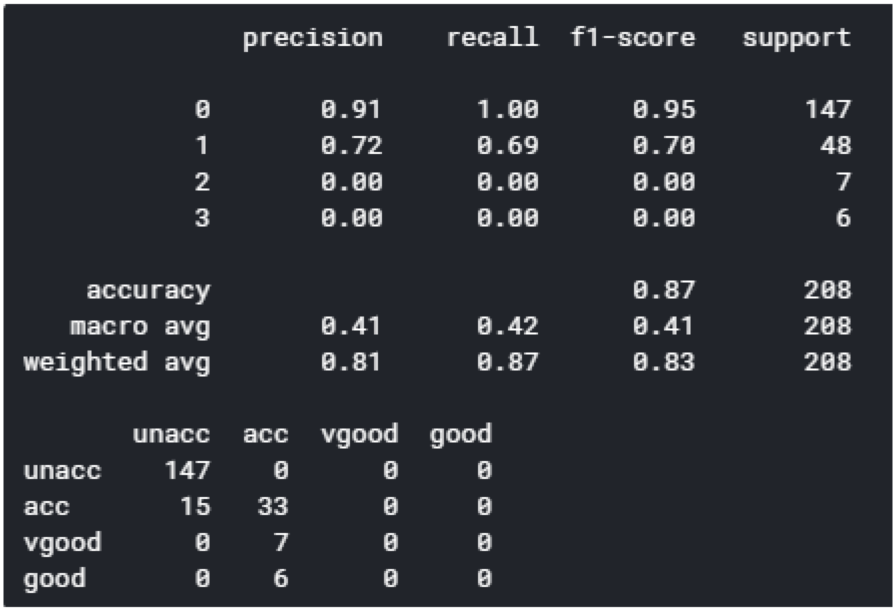


```
Per Class Classification Matrix: [1.         1.         0.91666667 0.66666667]
```

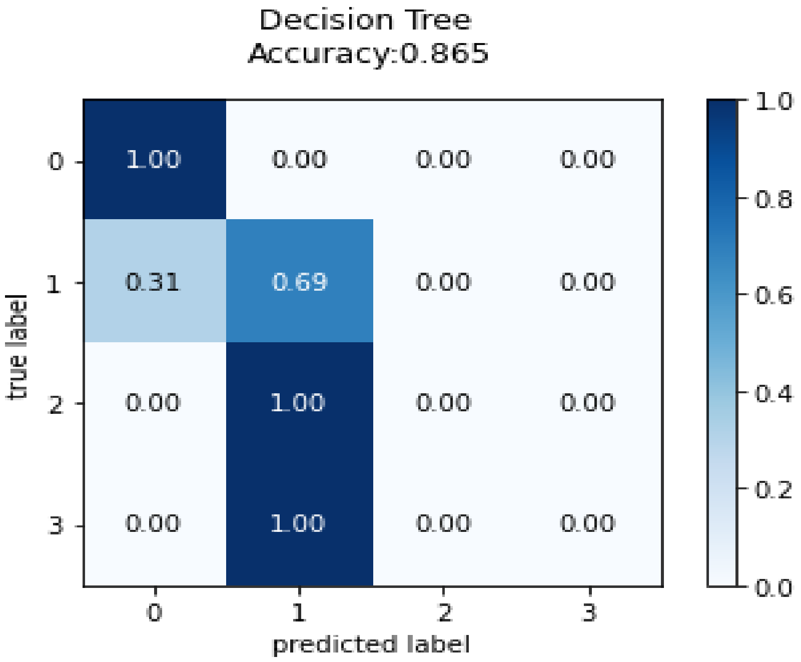
3.1.3 Detailed Classification Report: Decision Tree

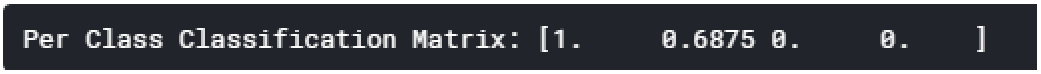
Train data: 760(44%)

Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.91      1.00      0.95       147
           1       0.72      0.69      0.70        48
           2       0.00      0.00      0.00         7
           3       0.00      0.00      0.00         6

    accuracy                           0.87       208
   macro avg       0.41      0.42      0.41       208
weighted avg       0.81      0.87      0.83       208

       unacc  acc  vgood  good
unacc    147    0      0     0
acc       15   33      0     0
vgood      0    7      0     0
good       0    6      0     0
```

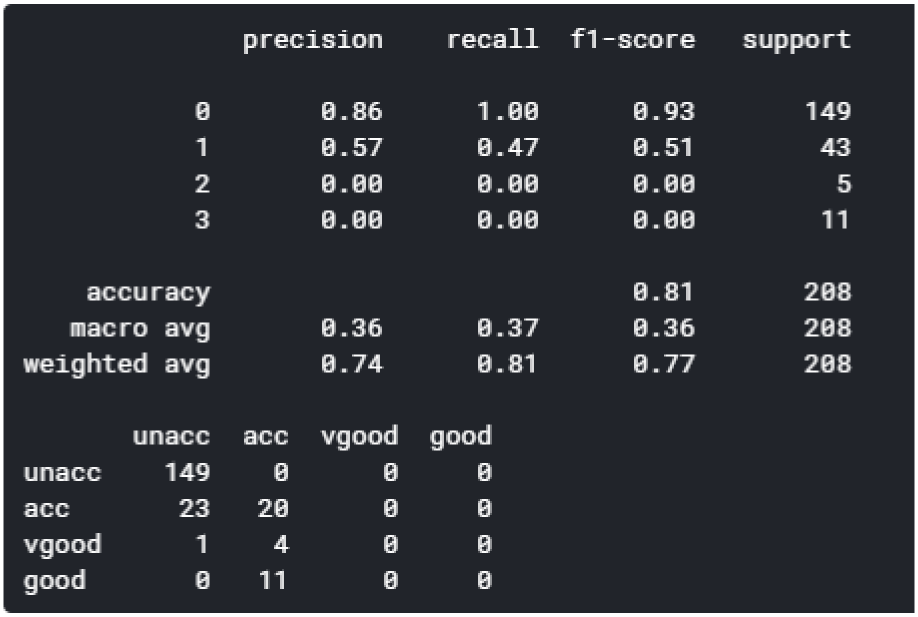




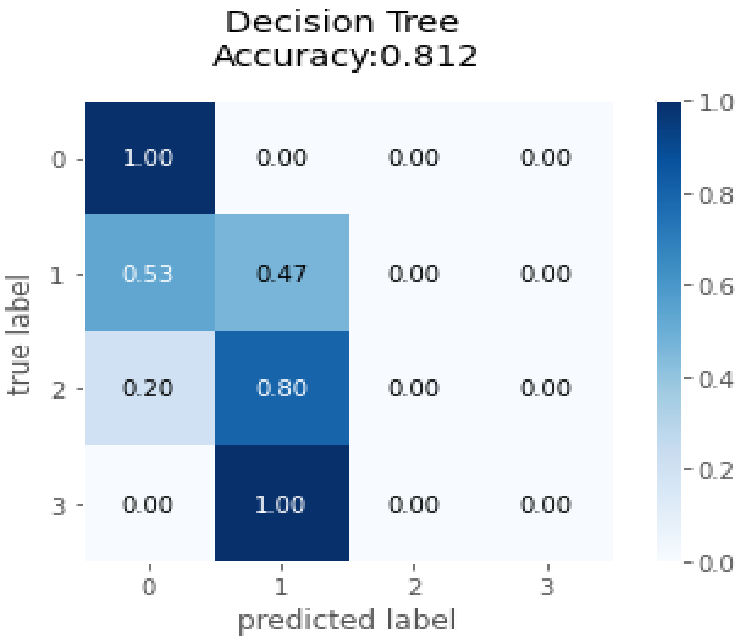
```
Per Class Classification Matrix: [1.     0.6875 0.     0.    ]
```

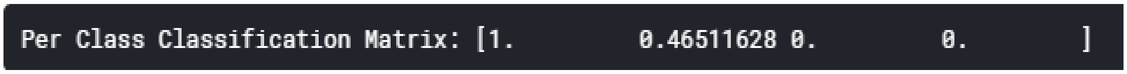
3.1.4 Detailed Classification Report: Decision Tree

Train data: 501(29%)
Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.86      1.00      0.93       149
           1       0.57      0.47      0.51        43
           2       0.00      0.00      0.00         5
           3       0.00      0.00      0.00        11

    accuracy                           0.81       208
   macro avg       0.36      0.37      0.36       208
weighted avg       0.74      0.81      0.77       208

       unacc  acc  vgood  good
unacc    149    0      0     0
acc       23   20      0     0
vgood      1    4      0     0
good       0   11      0     0
```





```
Per Class Classification Matrix: [1.         0.46511628 0.         0.        ]
```

1.2 Random Forest Classification:

Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

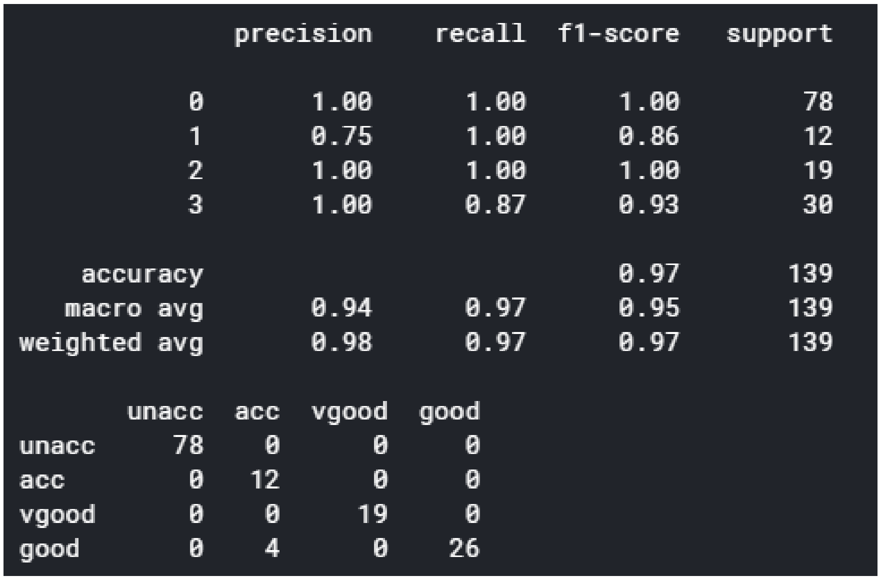
As mentioned in below table, we can see the accuracy of Random Forest. With 92% of training data we can get 97% of accuracy.

Train Test Split				**Random Forest**			
Training	Testing	Training %	Testing %	Correct %	Incorrect %	Correct	Incorrect
1589	139	92	8	97	3	135	4
1261	208	73	12	97	3	201	7
760	208	44	12	84	16	174	34
501	208	29	12	78	22	162	46

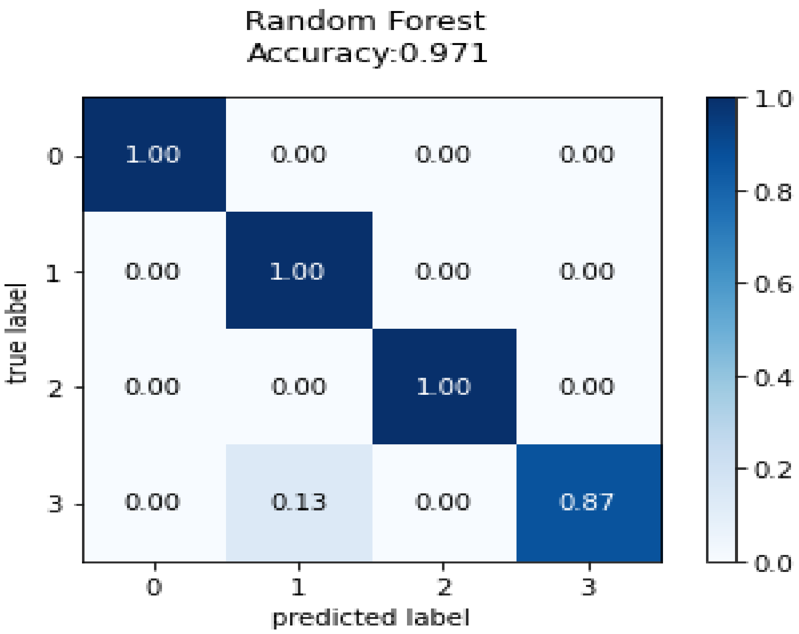
1.2.1 Detailed Classification Report – Random Forest

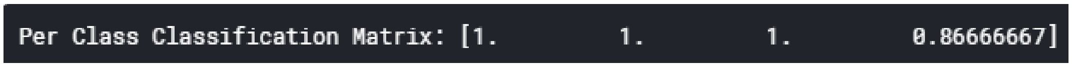
Train data: 1589 (92%)

Test data: 139 (8%)



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        78
           1       0.75      1.00      0.86        12
           2       1.00      1.00      1.00        19
           3       1.00      0.87      0.93        30

    accuracy                           0.97       139
   macro avg       0.94      0.97      0.95       139
weighted avg       0.98      0.97      0.97       139

       unacc  acc  vgood  good
unacc     78    0      0     0
acc        0   12      0     0
vgood      0    0     19     0
good       0    4      0    26
```



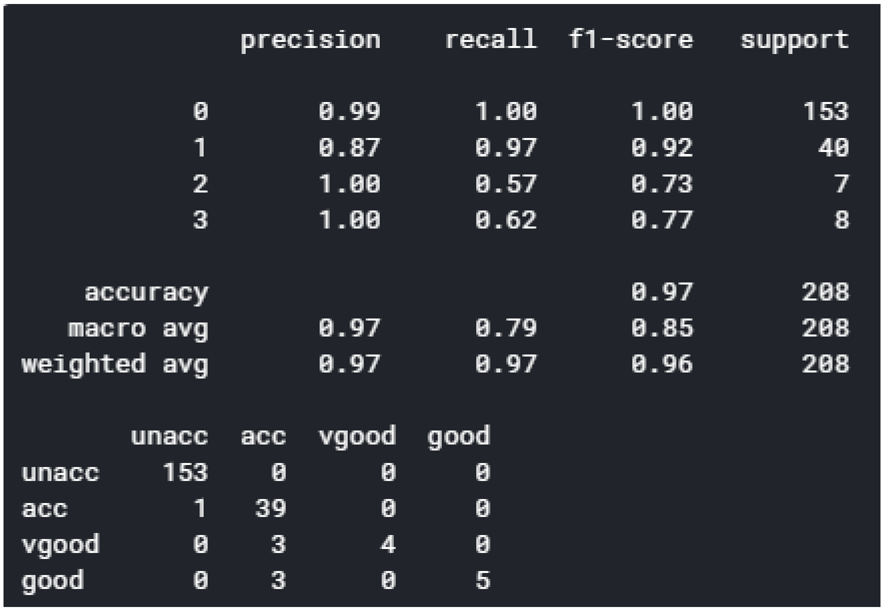


```
Per Class Classification Matrix: [1.         1.         1.         0.86666667]
```

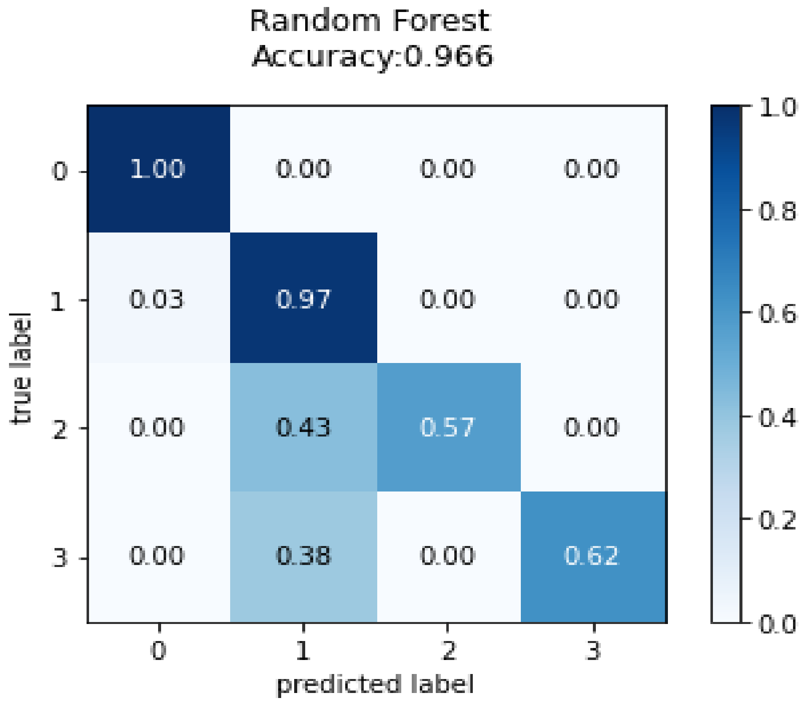
3.2.2 Detailed Classification Report: Random Forest

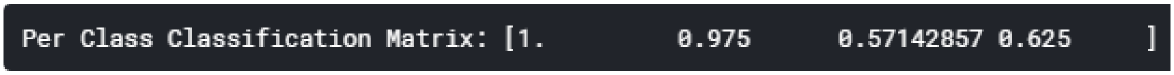
Train data: 1261(73%)

Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       153
           1       0.87      0.97      0.92        40
           2       1.00      0.57      0.73         7
           3       1.00      0.62      0.77         8

    accuracy                           0.97       208
   macro avg       0.97      0.79      0.85       208
weighted avg       0.97      0.97      0.96       208

       unacc  acc  vgood  good
unacc    153    0      0     0
acc        1   39      0     0
vgood      0    3      4     0
good       0    3      0     5
```

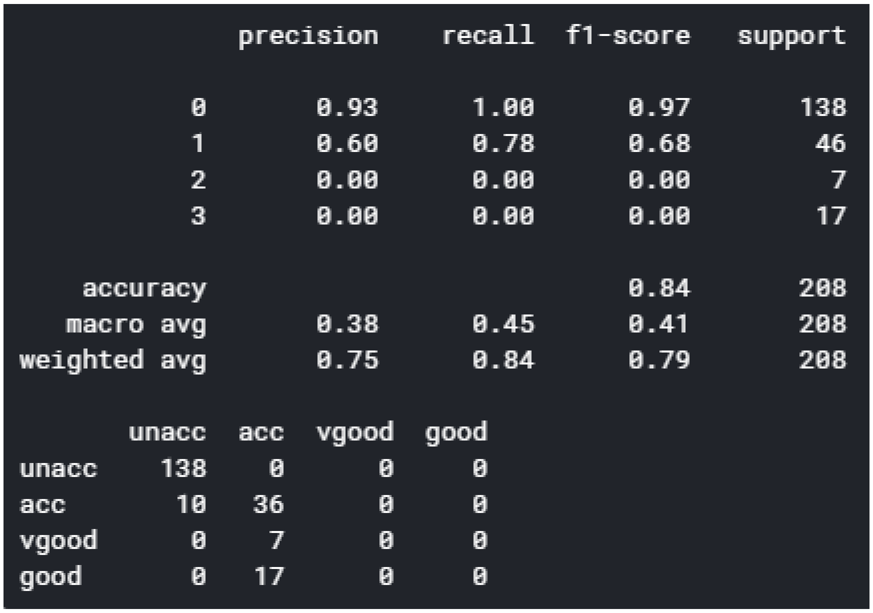




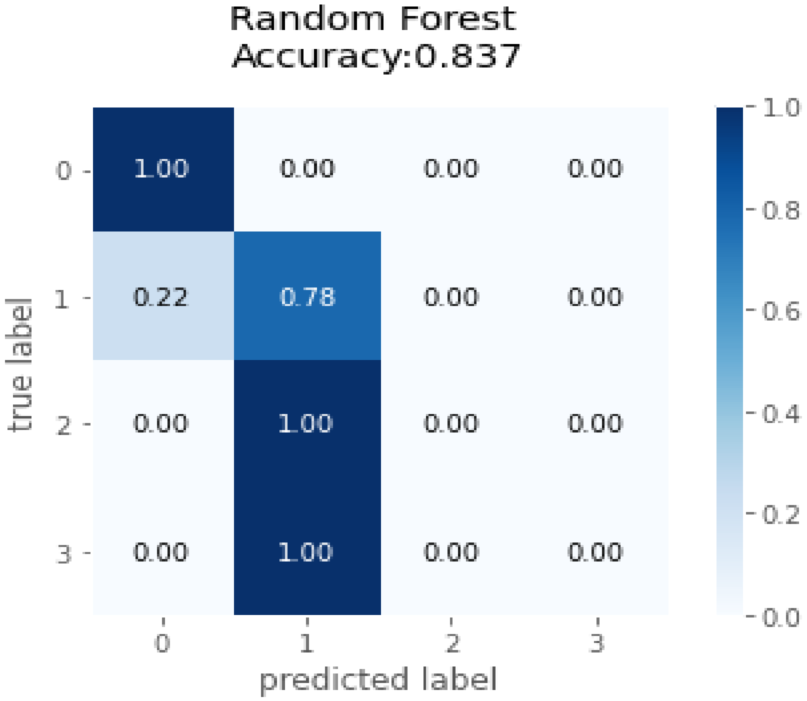
```
Per Class Classification Matrix: [1.         0.975      0.57142857 0.625     ]
```

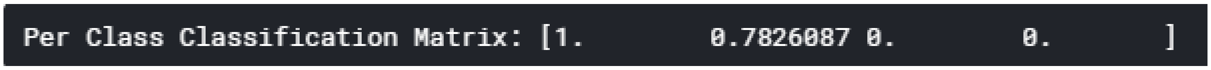
3.2.3 Detailed Classification Report: Random Forest

Train data: 760(44%)
Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.93      1.00      0.97       138
           1       0.60      0.78      0.68        46
           2       0.00      0.00      0.00         7
           3       0.00      0.00      0.00        17

    accuracy                           0.84       208
   macro avg       0.38      0.45      0.41       208
weighted avg       0.75      0.84      0.79       208

       unacc  acc  vgood  good
unacc    138    0      0     0
acc       10   36      0     0
vgood      0    7      0     0
good       0   17      0     0
```

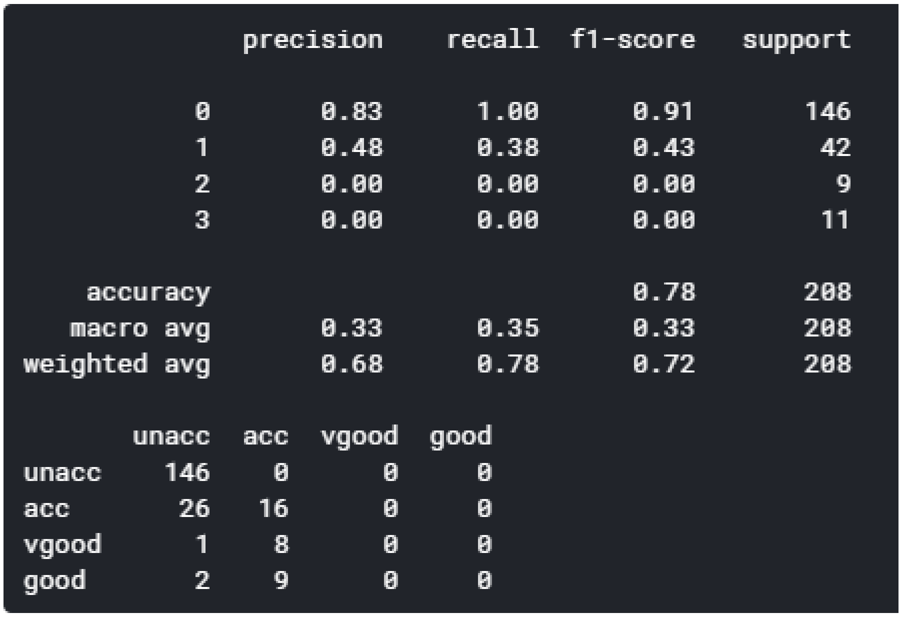




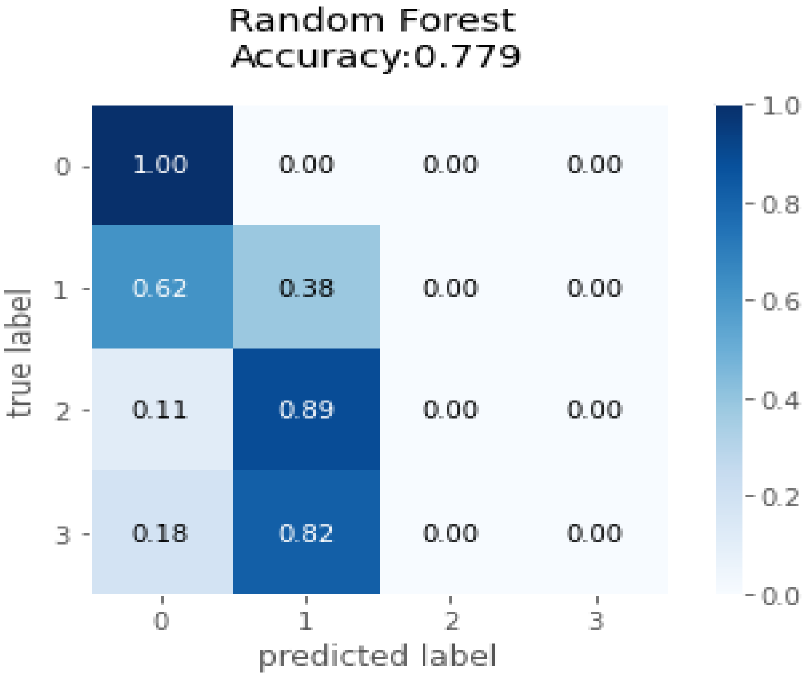
```
Per Class Classification Matrix: [1.         0.7826087 0.         0.        ]
```

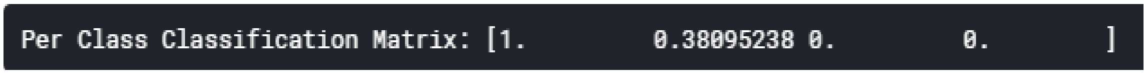
3.2.4 Detailed Classification Report: Random Forest

Train data: 501(29%)
Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.83      1.00      0.91       146
           1       0.48      0.38      0.43        42
           2       0.00      0.00      0.00         9
           3       0.00      0.00      0.00        11

    accuracy                           0.78       208
   macro avg       0.33      0.35      0.33       208
weighted avg       0.68      0.78      0.72       208

       unacc  acc  vgood  good
unacc    146    0      0     0
acc       26   16      0     0
vgood      1    8      0     0
good       2    9      0     0
```





```
Per Class Classification Matrix: [1.         0.38095238 0.         0.        ]
```

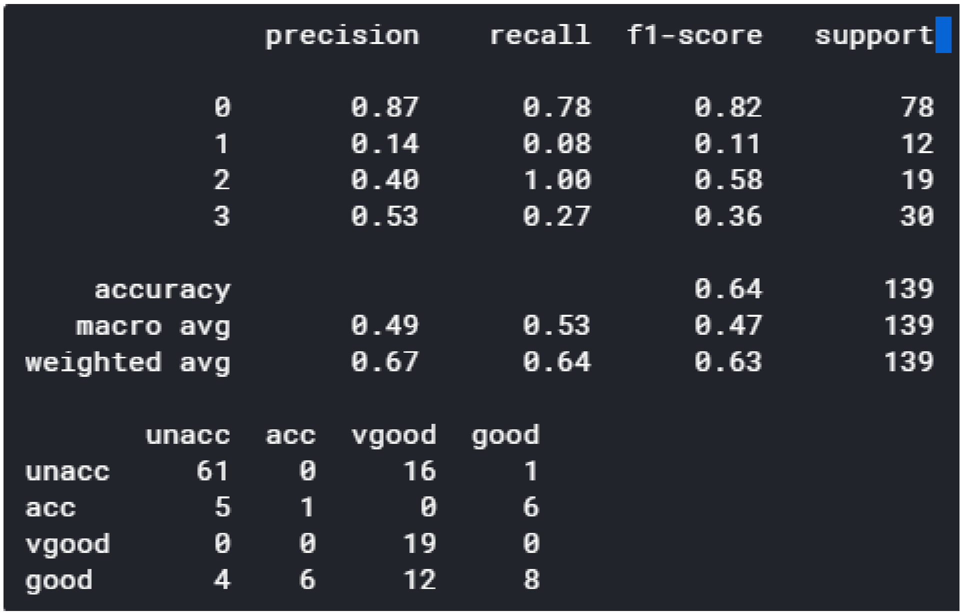
3.3 Naïve Bayes Classification:

Naïve Bayes classification is based on Bayes' Theorem with an assumption of individuality among predictors. Naive Bayes classifier considers that the presence of a feature in a class is unrelated to the presence of any other feature. As mentioned in below table, we can see the accuracy of Naïve Bayes. With 92% of training data we can get 64% of accuracy.

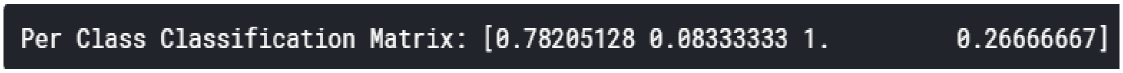
Train Test Split				**Naïve Bayes**			
Training	Testing	Training %	Testing %	Correct	Incorrect	Correct %	Incorrect %
1589	139	92	8	89	50	64	36
1261	208	73	12	155	53	75	25
760	208	44	12	162	46	78	22
501	208	29	12	136	72	65	35

3.3.1 Detailed Classification Report – Naïve Bayes

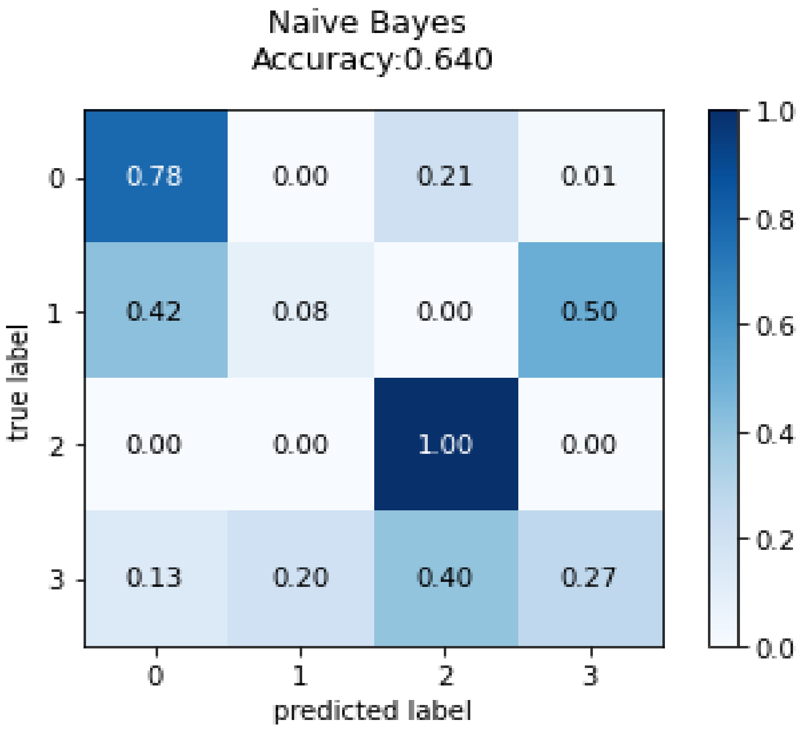
Train data: 1589 (92%)
Test data: 139 (8%)



```
              precision    recall  f1-score   support

           0       0.87      0.78      0.82        78
           1       0.14      0.08      0.11        12
           2       0.40      1.00      0.58        19
           3       0.53      0.27      0.36        30

    accuracy                           0.64       139
   macro avg       0.49      0.53      0.47       139
weighted avg       0.67      0.64      0.63       139

       unacc  acc  vgood  good
unacc     61    0     16     1
acc        5    1      0     6
vgood      0    0     19     0
good       4    6     12     8
```

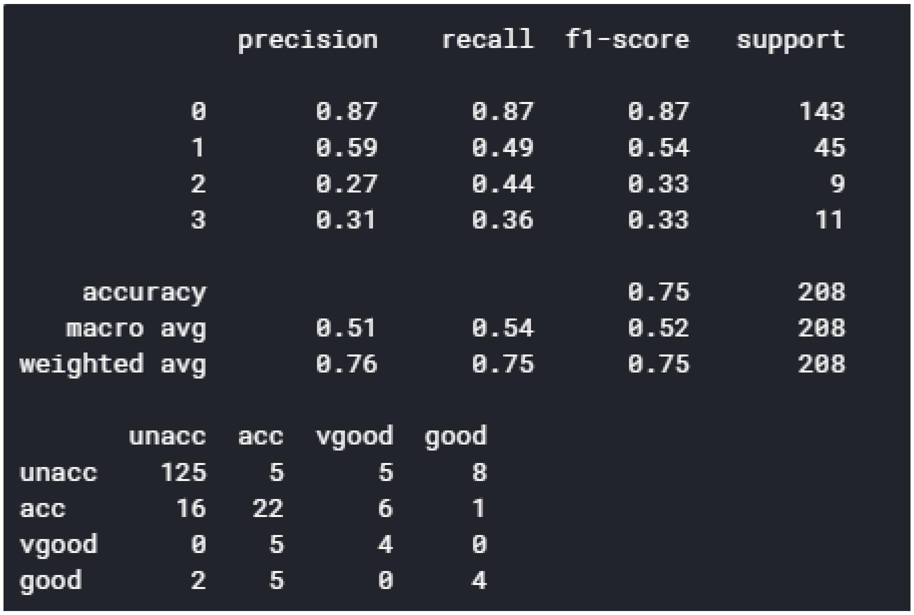


```
Per Class Classification Matrix: [0.78205128 0.08333333 1.         0.26666667]
```

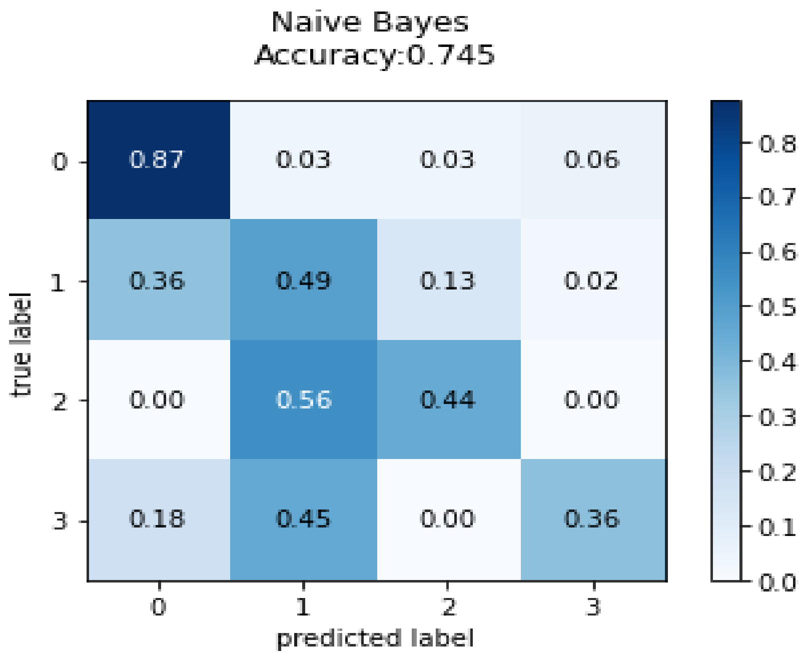


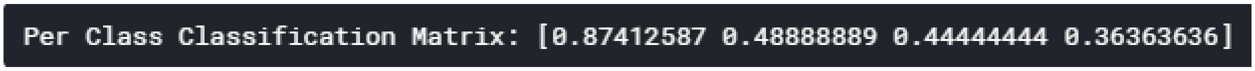
3.3.2 Detailed Classification Report: Naïve Bayes

Train data: 1261(73%)
Test data: 200 (12%)



```
              precision    recall  f1-score   support

           0       0.87      0.87      0.87       143
           1       0.59      0.49      0.54        45
           2       0.27      0.44      0.33         9
           3       0.31      0.36      0.33        11

    accuracy                           0.75       208
   macro avg       0.51      0.54      0.52       208
weighted avg       0.76      0.75      0.75       208

       unacc  acc  vgood  good
unacc    125    5      5     8
acc       16   22      6     1
vgood      0    5      4     0
good       2    5      0     4
```

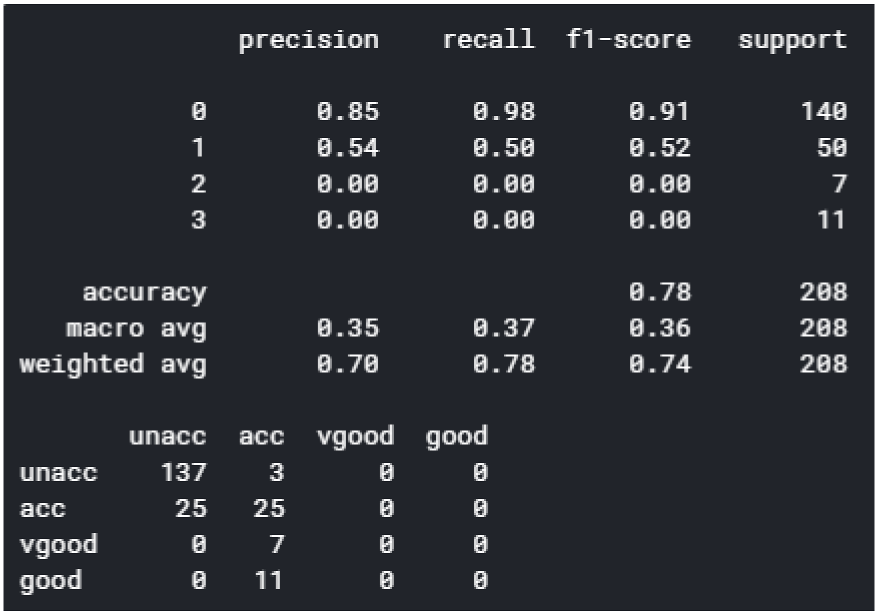




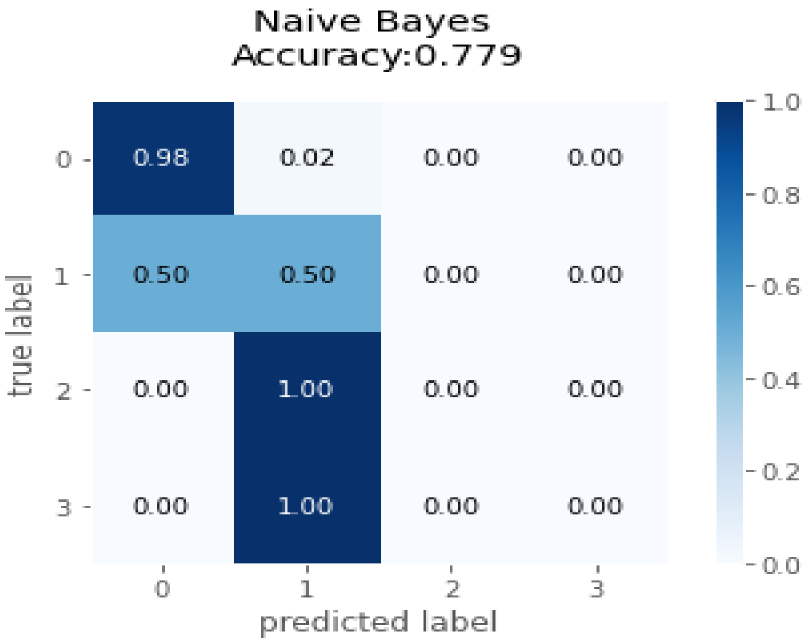
```
Per Class Classification Matrix: [0.87412587 0.48888889 0.44444444 0.36363636]
```

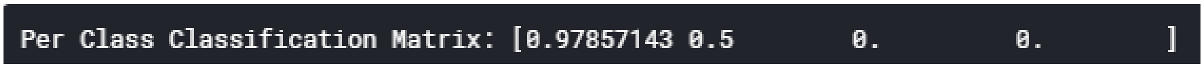
3.3.3 Detailed Classification Report: Naïve Bayes

Train data: 760(44%)
Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       140
           1       0.54      0.50      0.52        50
           2       0.00      0.00      0.00         7
           3       0.00      0.00      0.00        11

    accuracy                           0.78       208
   macro avg       0.35      0.37      0.36       208
weighted avg       0.70      0.78      0.74       208

       unacc  acc  vgood  good
unacc    137    3      0     0
acc       25   25      0     0
vgood      0    7      0     0
good       0   11      0     0
```

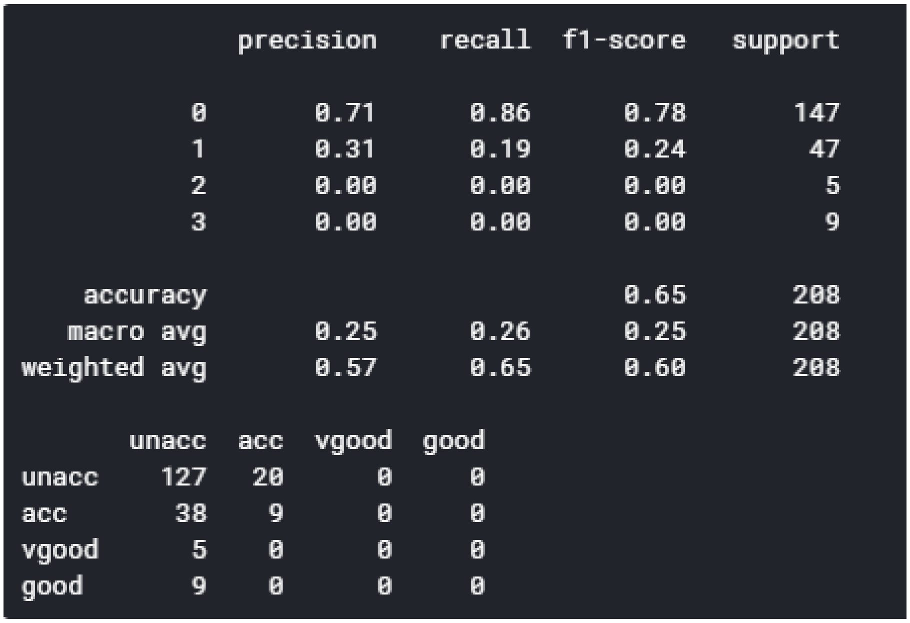




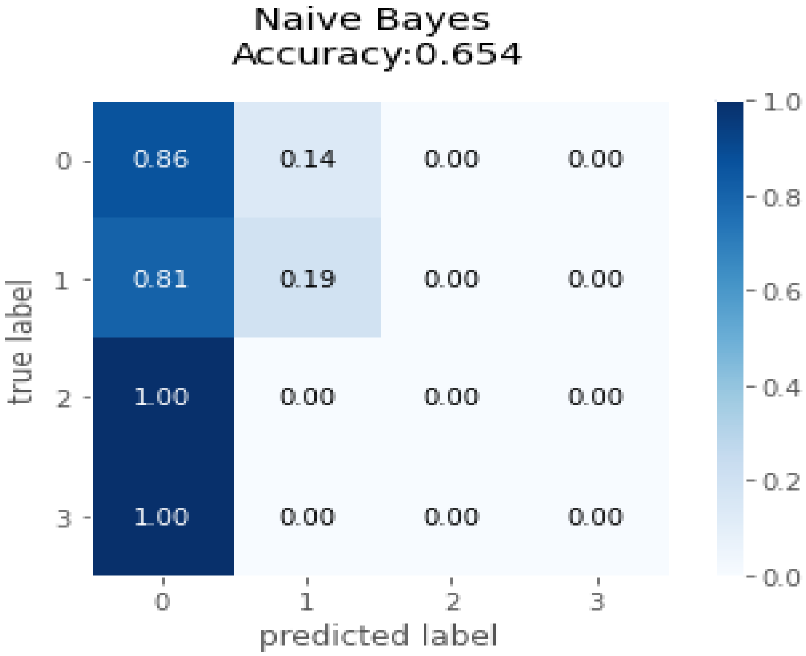
```
Per Class Classification Matrix: [0.97857143 0.5        0.         0.        ]
```

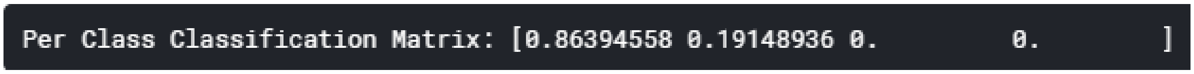
3.3.4 Detailed Classification Report: Naïve Bayes

Train data: 501(29%)
Test data: 208 (12%)



```
              precision    recall  f1-score   support

           0       0.71      0.86      0.78       147
           1       0.31      0.19      0.24        47
           2       0.00      0.00      0.00         5
           3       0.00      0.00      0.00         9

    accuracy                           0.65       208
   macro avg       0.25      0.26      0.25       208
weighted avg       0.57      0.65      0.60       208

       unacc  acc  vgood  good
unacc    127   20      0     0
acc       38    9      0     0
vgood      5    0      0     0
good       9    0      0     0
```





```
Per Class Classification Matrix: [0.86394558 0.19148936 0.         0.        ]
```

4. Comparison of Classifiers:

Train Test Split		**Decision Tree**		**Random Forest**		**Naïve Bayes**	
Training %	Testing %	Correct %	Incorrect %	Correct %	Incorrect %	Correct %	Incorrect %
92	8	99	1	97	3	64	36
73	12	98	2	97	3	75	25
44	12	87	13	84	16	78	22
29	12	81	19	78	22	65	35

- The classified dataset has a class with four attribute values (i.e acc, unacc, good, vgood), thus; having a model with the highest accuracy to be 99% (92:8) with Decision Tree classification.
- The classified dataset has a class with four attribute values (i.e acc, unacc, good, vgood), thus; having a model with the lowest accuracy to be 64% (92:8) with Naïve Bayes classification.
- The classified dataset result from the comparison between the three classifiers shows (Decision Tree, Random Forest, Naïve Bayes) that Decision Tree and Random Forest have the close accuracy across the two (73:12) settings.
- A general observation on the dataset with regards to accuracy is the train test split dimensions. This means, more the training dataset; the higher the accuracy of the model.
- In Naïve Bayes classification, we can observe that accuracy increases when we decrease training dataset and after a point it starts decreasing.

Source Code Link:

1. Decision Tree: https://www.kaggle.com/harshadaaphadol/decision-tree-classification-in-python
2. Random Forest: https://www.kaggle.com/harshadaaphadol/random-forest-classifier
3. Naïve Bayes: https://www.kaggle.com/harshadaaphadol/naive-bayes-classification-in-python