Mon- 13Mar2023

Report # 3

Algorithm Testing and Evaluation:

This report provides an overview of the machine learning testing and evaluation techniques that were applied within our movie revenue classification project. In the below sections, methods used for splitting the data into training and test sets are outlined. In addition, 4- different machine learning algorithms are applied against multiple splitting methods. Confusion matrixes were generated to visualize the performance of each machine learning algorithm.

The modified dataset that was prepared in Report #2, had the following logistic regression, Gaussian Naïve Bayes, Support Vector Machine and Decision Tree machine learning algorithms fitted to it.

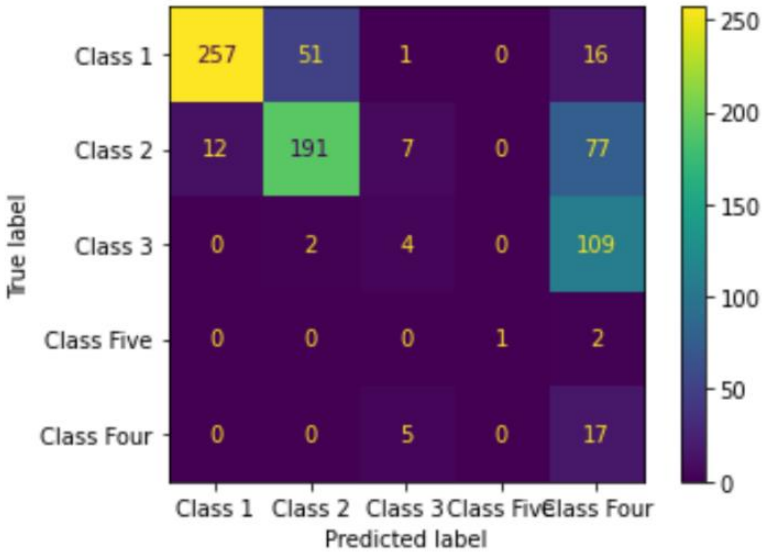
The modified data was partitioned into test and validation sets, utilizing the **Classes** feature as the y variable in our model. The remaining features made up our basis data by which the model would base its predictions off.

After fitting each model to the modified dataset, gross revenue **Class predictions** were generated, and the models were tested utilizing a 10-fold cross validation technique.

Using **Gaussian Naïve Bayes**, the model performed very poorly, predicting multiclass categorization at approximately a 30% rate. The below array displays each of accuracy score of each of the 10 tests that were taken.

```
array([0.30263158, 0.31578947, 0.28 , 0.25333333, 0.24 , 0.33333333, 0.28
, 0.29333333, 0.28 , 0.32 ])
```

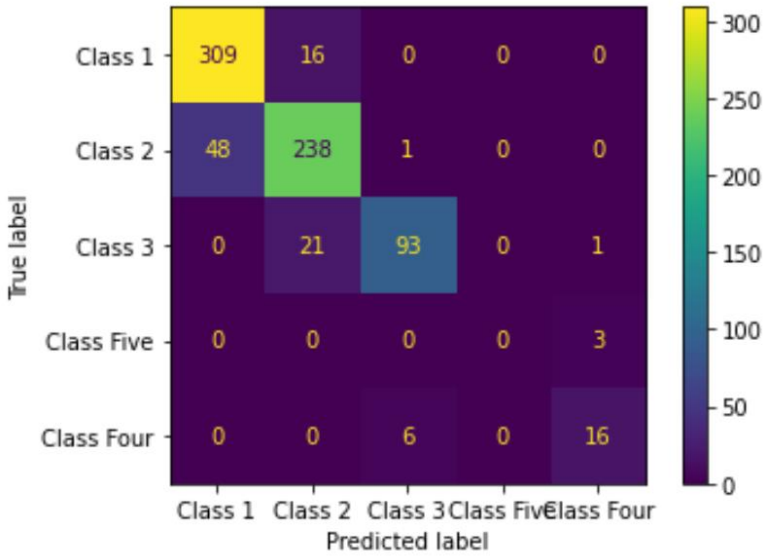
In the confusion matrix below, the predictions generated by **Gaussian Naïve Bayes** are evaluated against the class variable.



Using **Support Vector Machine**, the model performed significantly better than Gaussian Naïve Bayes. SVM performed it' multiclass categorization at approximately an 80% rate in its first test while incrementally decreasing in accuracy for each subsequent test that was performed. The first two tests in the cross- validation step resulted in 78% and 82%, respectively, while the final test resulted in an accuracy score of 69%. The below array displays the accuracy scores of each of the 10 tests that were taken.

```
array([0.78947368, 0.82894737, 0.72 , 0.69333333, 0.70666667, 0.62666667,
0.61333333, 0.82666667, 0.70666667, 0.69333333])
```

In the confusion matrix below, the predictions generated by **Support Vector Machine** are evaluated against the class variable.

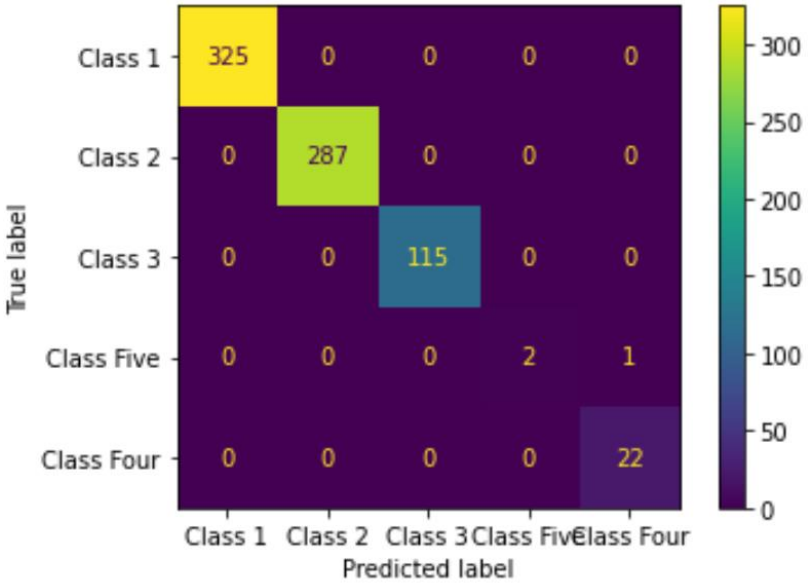



```
Accuracy: GAUSSIAN NAIVE BAYES: 0.625
Accuracy: SVM: 0.8723404255319149
Accuracy: DECISION TREE: 0.9986702127659575
```

Using **Decision Tree**, the model performed better than both Support Vector Machine and Naïve Bayes. The accuracy scores for the 10 tests ranged from 98-100%. The below array displays the accuracy scores of each of the 10 tests that were taken.

```
array([1. , 0.98684211, 1. , 1. , 1. , 1. , 1. , 1. , 1. , 1. ])
```

In the confusion matrix below, the predictions generated by **Decision Tree** are evaluated against the class variable.



This machine learning testing and evaluation report presents an overview of the steps taken to begin the process of drawing potential insights from the dataset through measuring the accuracy of our model's predictions. Gaussian Naïve Bayes performed the poorest, Decision Tree performed very well, and Support Vector Machine was performed in the middle.

To learn more about the project, the full commented source code and output results are referenceable via the following GitHub repository link:

https://github.com/adivalentino/820.git