Report for COMP 472 project

It is first important to note that I was unable to create proper functions that were able to save and load models, therefore each result is based off the fact that the model would need to go through the full training process once ran through main.py.

Model Architecture and Training

The first instance of each model being trained had specific values attached to an attribute that was changed later once variants needed to be tested. The Naïve Bayes was the only model where variants were not made. The main model for the Decision Tree was trained at a maximum depth of 50, while a variant of it was trained at 150. For the MLP, the hidden size variable was first set to 64, then its variant had one of 128. As for the VGG11, the main model started off with filters with 3 and 64 until it reaches a filter value of 512. It also had a size of 4096,4096. The variant starts off with filters of 3 and 32 and also stops increasing until it reaches 512. Its size was also reduced to 1024,512.

The Naïve Bayes model consisted of 10 classes. As for the MLP and VGG11, they both had 10 epochs. They as well both were evaluated on their loss. Each individual epoch for both MLP and VGG11 had their loss value displayed and for VGG11 the total train accuracy once an Epoch is finished is displayed to track the progress of the training.

In order to create the confusion matrices sklearn.metrics is used in order to generate them. As well in VGG1, flattening was used to make the outputs able to be used in a vector to then be able to be used in layers

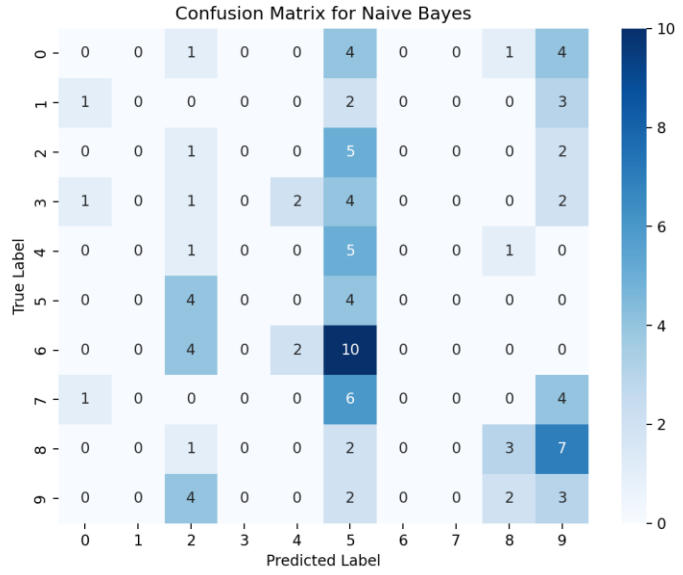
Evaluation

Presentation of metrics

Before beginning the evaluation of each model, it is important to address the fact that there is a likely possibility that models are either poorly optimized or not well made. This can impact the evaluation process of this report as insights and discussions of the results heavily depend on them. I acknowledge this fact therefore there is a certain doubt that should be raised when interpreting these results.

Starting off with the main model for the Naïve Bayes model, through every instance of training it had a constant Training Accuracy of 15%, with a Testing Accuracy of 11%. I have also included the classification report including precision, recall, f1-score and support. The completion time was nearly instant. I have also included the confusion matrix.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	10
1	0.00	0.00	0.00	6
2	0.06	0.12	0.08	8
3	0.00	0.00	0.00	10
4	0.00	0.00	0.00	7
5	0.09	0.50	0.15	8
6	0.00	0.00	0.00	16
7	0.00	0.00	0.00	11
8	0.43	0.23	0.30	13
9	0.12	0.27	0.17	11
accuracy			0.11	100
macro avg	0.07	0.11	0.07	100
weighted avg	0.08	0.11	0.08	100



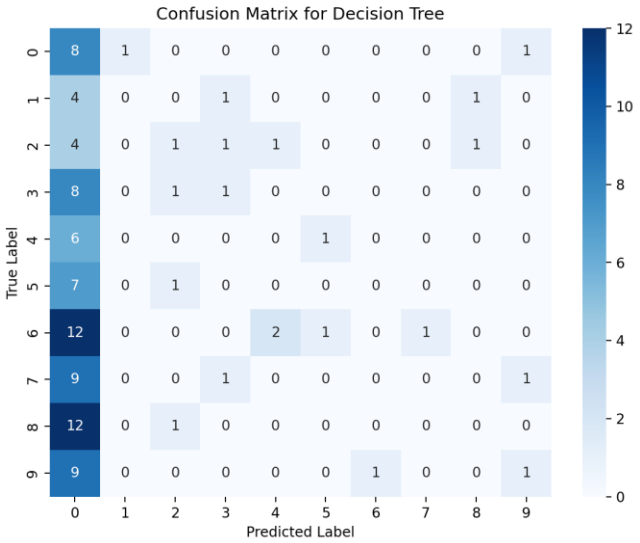
Next, here are the results for the main model of the decision tree, and I have also included the output of the variant as well. In fact, the classification report of the main and variant are the same. Both the main and the variant had the same training accuracy of 39.60% and the same testing accuracy of 11%. The only difference is that the main model had a completion time of 10.76 seconds while the variant had 10.26 seconds. I have included both classification reports as well as the confusion matrices.

```
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.10      0.80      0.18        10
           1       0.00      0.00      0.00         6
           2       0.25      0.12      0.17         8
           3       0.25      0.10      0.14        10
           4       0.00      0.00      0.00         7
           5       0.00      0.00      0.00         8
           6       0.00      0.00      0.00        16
           7       0.00      0.00      0.00        11
           8       0.00      0.00      0.00        13
           9       0.33      0.09      0.14        11

    accuracy                           0.11       100
   macro avg       0.09      0.11      0.06       100
weighted avg       0.09      0.11      0.06       100
```

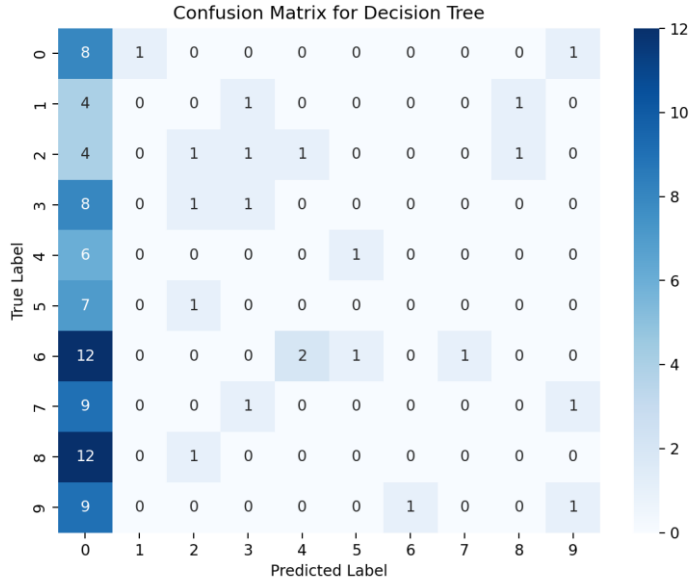
This is for decision tree with depth of 150:

```
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.10      0.80      0.18        10
           1       0.00      0.00      0.00         6
           2       0.25      0.12      0.17         8
           3       0.25      0.10      0.14        10
           4       0.00      0.00      0.00         7
           5       0.00      0.00      0.00         8
           6       0.00      0.00      0.00        16
           7       0.00      0.00      0.00        11
           8       0.00      0.00      0.00        13
           9       0.33      0.09      0.14        11

    accuracy                           0.11       100
   macro avg       0.09      0.11      0.06       100
weighted avg       0.09      0.11      0.06       100
```

Main confusion matrix:



Variant confusion matrix:



Moving on to MLP, the main model had an accuracy of 15.80% and a testing accuracy of 11%. Its completion time was of 0.27 seconds. I will include the output to show the loss of each epoch as well with the classification report.

```
Epoch [1/10], Loss: 2.2668
Epoch [2/10], Loss: 2.2855
Epoch [3/10], Loss: 2.2546
Epoch [4/10], Loss: 2.2902
Epoch [5/10], Loss: 2.3203
Epoch [6/10], Loss: 2.2628
Epoch [7/10], Loss: 2.2147
Epoch [8/10], Loss: 2.2090
Epoch [9/10], Loss: 2.2790
Epoch [10/10], Loss: 2.2634
Evaluating MLP model...
Train Accuracy: 15.80%
Test Accuracy: 11.00%
MLP training completed in 0.27 seconds.
MLP Training Accuracy: 15.80%
MLP Testing Accuracy: 11.00%
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	10
1	0.00	0.00	0.00	6
2	0.00	0.00	0.00	8
3	0.00	0.00	0.00	10
4	0.09	0.57	0.15	7
5	0.00	0.00	0.00	8
6	0.00	0.00	0.00	16
7	0.00	0.00	0.00	11
8	0.00	0.00	0.00	13
9	0.17	0.64	0.27	11
accuracy			0.11	100
macro avg	0.03	0.12	0.04	100
weighted avg	0.03	0.11	0.04	100

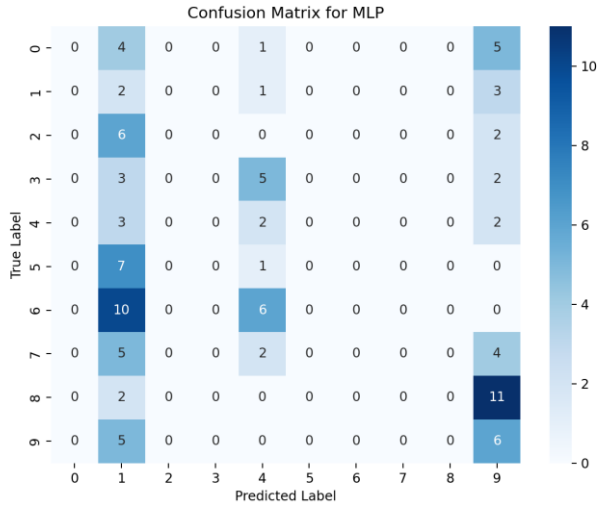
As for the MLP variant model, it had an accuracy of 16.40% with a completion time of 0.20 seconds. It also had a testing accuacy of 11%. I will include the output to show the loss of each epoch along with the classification report.

I have also included both confusion matrices.

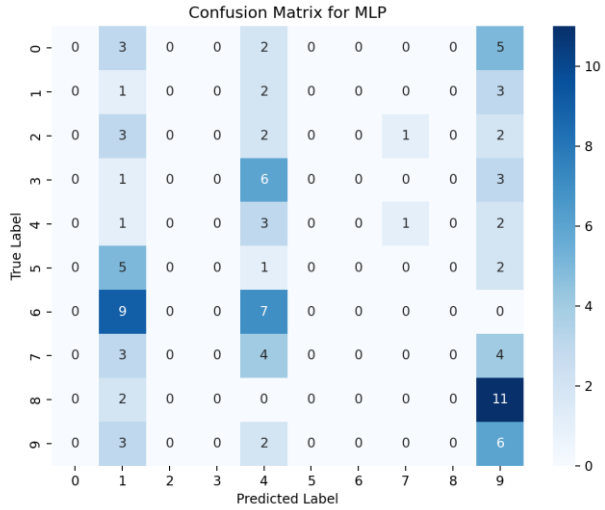
```
Epoch [1/10], Loss: 2.3054
Epoch [2/10], Loss: 2.3095
Epoch [3/10], Loss: 2.2752
Epoch [4/10], Loss: 2.2598
Epoch [5/10], Loss: 2.2562
Epoch [6/10], Loss: 2.2341
Epoch [7/10], Loss: 2.2152
Epoch [8/10], Loss: 2.2082
Epoch [9/10], Loss: 2.3101
Epoch [10/10], Loss: 2.1034
Evaluating MLP model...
Train Accuracy: 16.40%
Test Accuracy: 11.00%
MLP training completed in 0.20 seconds.
MLP Training Accuracy: 16.40%
MLP Testing Accuracy: 11.00%
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	10
1	0.04	0.17	0.07	6
11				
8	0.00	0.00	0.00	13
9	0.15	0.55	0.24	11
accuracy			0.11	100
macro avg	0.13	0.12	0.06	100
weighted avg	0.19	0.11	0.06	100

Main confusion matrix:



Variant confusion matrix:



Finally, for the VGG11, the main model had a training accuracy of 53% with a testing accuracy of 29%. The completion time was roughly 20 minutes. I will include both the output and the classification report.

```
Training VGG11 model...
Epoch [1/10], Loss: 2.3194, Train Accuracy: 13.00%
Epoch [2/10], Loss: 2.2685, Train Accuracy: 21.60%
Epoch [3/10], Loss: 2.2347, Train Accuracy: 21.40%
Epoch [4/10], Loss: 1.9977, Train Accuracy: 28.40%
Epoch [5/10], Loss: 1.8468, Train Accuracy: 31.60%
Epoch [6/10], Loss: 2.1701, Train Accuracy: 34.00%
Epoch [7/10], Loss: 1.7253, Train Accuracy: 39.80%
Epoch [8/10], Loss: 1.6296, Train Accuracy: 45.60%
Epoch [9/10], Loss: 1.4472, Train Accuracy: 45.80%
Epoch [10/10], Loss: 1.3573, Train Accuracy: 53.00%
Test Accuracy: 29.00%
VGG11 Training Accuracy: 53.00%
VGG11 Testing Accuracy: 29.00%
```

```
VGG11 Classification Report:
              precision    recall  f1-score   support

           0       0.25      0.40      0.31        10
           1       0.29      0.67      0.40         6
           2       0.00      0.00      0.00         8
           3       0.31      0.40      0.35        10
           4       0.11      0.14      0.12         7
           5       0.20      0.12      0.15         8
           6       0.17      0.06      0.09        16
           7       0.36      0.45      0.40        11
           8       0.50      0.62      0.55        13
           9       0.17      0.09      0.12        11

    accuracy                           0.29       100
   macro avg       0.23      0.30      0.25       100
weighted avg       0.25      0.29      0.25       100
```

Next is the results for variant. It had a training accuracy of 54% with a testing accuracy of 29%. The completion time of this was roughly 5 minutes. I will include both the output and the classification report.

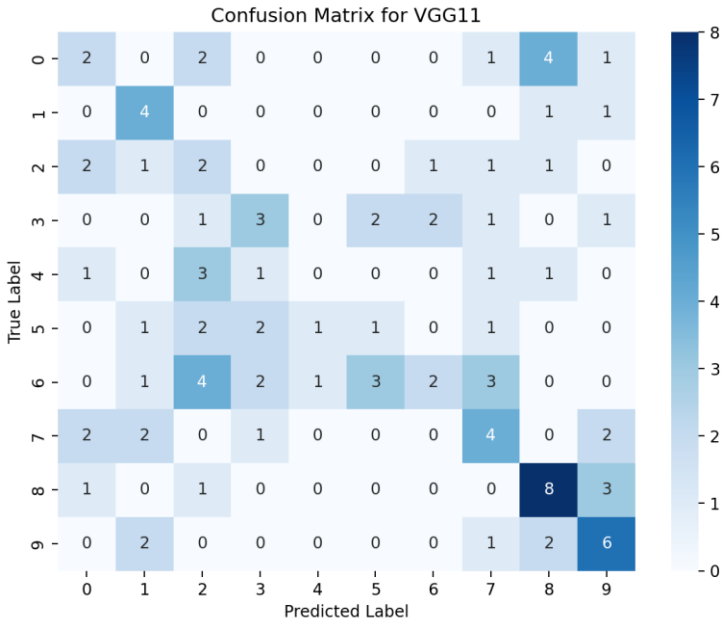
I have also included both confusion matrices.

```
Training VGG11 model...
Epoch [1/10], Loss: 2.3180, Train Accuracy: 15.80%
Epoch [2/10], Loss: 2.4024, Train Accuracy: 17.40%
Epoch [3/10], Loss: 2.1903, Train Accuracy: 22.60%
Epoch [4/10], Loss: 1.9337, Train Accuracy: 28.80%
Epoch [5/10], Loss: 1.7904, Train Accuracy: 37.40%
Epoch [6/10], Loss: 1.6522, Train Accuracy: 41.80%
Epoch [7/10], Loss: 1.6086, Train Accuracy: 44.00%
Epoch [8/10], Loss: 1.4943, Train Accuracy: 46.40%
Epoch [9/10], Loss: 1.5002, Train Accuracy: 47.60%
Epoch [10/10], Loss: 1.3494, Train Accuracy: 54.00%
Test Accuracy: 29.00%
VGG11 Training Accuracy: 54.00%
VGG11 Testing Accuracy: 29.00%
```

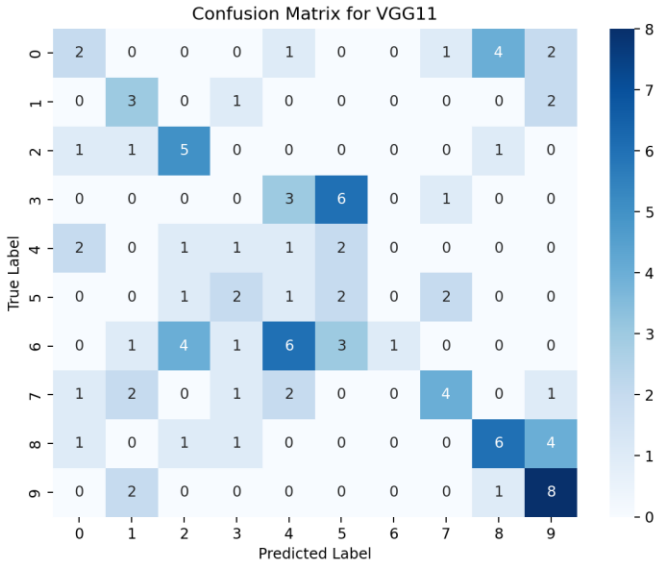
VGG11 Classification Report:

	precision	recall	f1-score	support
0	0.42	0.50	0.45	10
1	0.31	0.67	0.42	6
2	0.30	0.38	0.33	8
3	0.00	0.00	0.00	10
4	0.33	0.29	0.31	7
5	0.05	0.12	0.07	8
6	0.67	0.25	0.36	16
7	0.67	0.18	0.29	11
8	0.50	0.31	0.38	13
9	0.31	0.36	0.33	11
accuracy			0.29	100
macro avg	0.35	0.31	0.29	100
weighted avg	0.39	0.29	0.30	100

Main confusion matrix:



Variant confusion matrix:



Discussion of results

Between each model and their variant, the changes made to certain attributes, with the exception of VGG11 had no changes between them. For the decision tree, the max depth changes have not seem to make any difference on the results. For MLP, while giving different results, the changes are not significant enough for them to be noteworthy.

When it comes to my findings, changing the depth of the models did not make any significant changes to any of the models.

The only significant change is present when looking at both models of VGG11. When adjusting the filters and the size, the variant (with the smaller size and lower values for the filters) took significantly less time for it complete than the main model (with a bigger size and higher values for the filters. The variant took around 5 minutes while the main model took around 20 minutes, which means the variant was 4 times faster to complete.

When it comes to comparing accuracies (knowing the fact that some models are likely poorly organized and made), VGG11 was the most accurate followed by Decision Tree and then Naïve Bayes/MLP are interchangeable (as there is no distinguishing gap that helps dictate which is more accurate than the other). If I had to assume why VGG11 performed best it would be because it is likely the one that is most well designed out of the 3 since the other 3 models all have relatively low accuracies.

As for which of the models was the most confused, MLP seemed to be the most confused as they it is hard to see the diagonal of the matrix. There are darker colored values however, they are more present vertically, which doesn't give any hints into seeing the diagonal. When comparing the VGG11 with the MLP, the diagonal is more obvious as to where it is.

Conclusion

Overall, based of the results of my program training the models, the "best" model, which I evaluate as the most accurate model and least confused model had the highest accuracy and had the most obvious diagonal with darker colors on the confusion matrix. The "worst" model would be the MLP as it had one of the lowest accuracies while also being the most confused when evaluating the confusion matrix.