We utilized 2 datasets for our training and classification:

1) Original hand sign language MNIST dataset
2) American Sign Language dataset

About the datasets:

1)Original hand sign language MNIST dataset

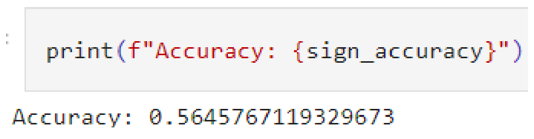
- The dataset can be found at https://www.kaggle.com/datasets/datamunge/sign-language-mnist
- This dataset contains images of letters A-Z, but except J and Z which involve rotation so they were excluded in the original dataset itself. So in total we have 24 classes here.
- This dataset has images of size 28X28 pixels
- We used 19218 training, 8237 validation, 7172 testing images

2)American Sign Language dataset

- The dataset can be found at https://www.kaggle.com/datasets/grassknoted/asl-alphabet/data
- This dataset contains images of letters from A-Z as well as del, space, and nothing. So in total we have 29 classes here.
- This dataset has images of size 100X100 pixels
- We had a big dataset file (can be found in github) but for faster and easier fine-tuning .of the CNN network we utilized a subset of it for time being which with more error analysis, fine-tuning knowledge and transfer learning can be potentially applied to the bigger dataset as well had we have had better GPU power for processing.

SparkML models used and results: -

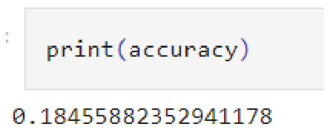
1) **Random forest classifier on original MNIST dataset:** Achieved an accuracy of 56.45%



```
print(f"Accuracy: {sign_accuracy}")
```

```
Accuracy: 0.5645767119329673
```

2) **Multi Layer Perceptron (MLP) model on original MNIST dataset:** Achieved an accuracy of 18.45%



```
print(accuracy)
```

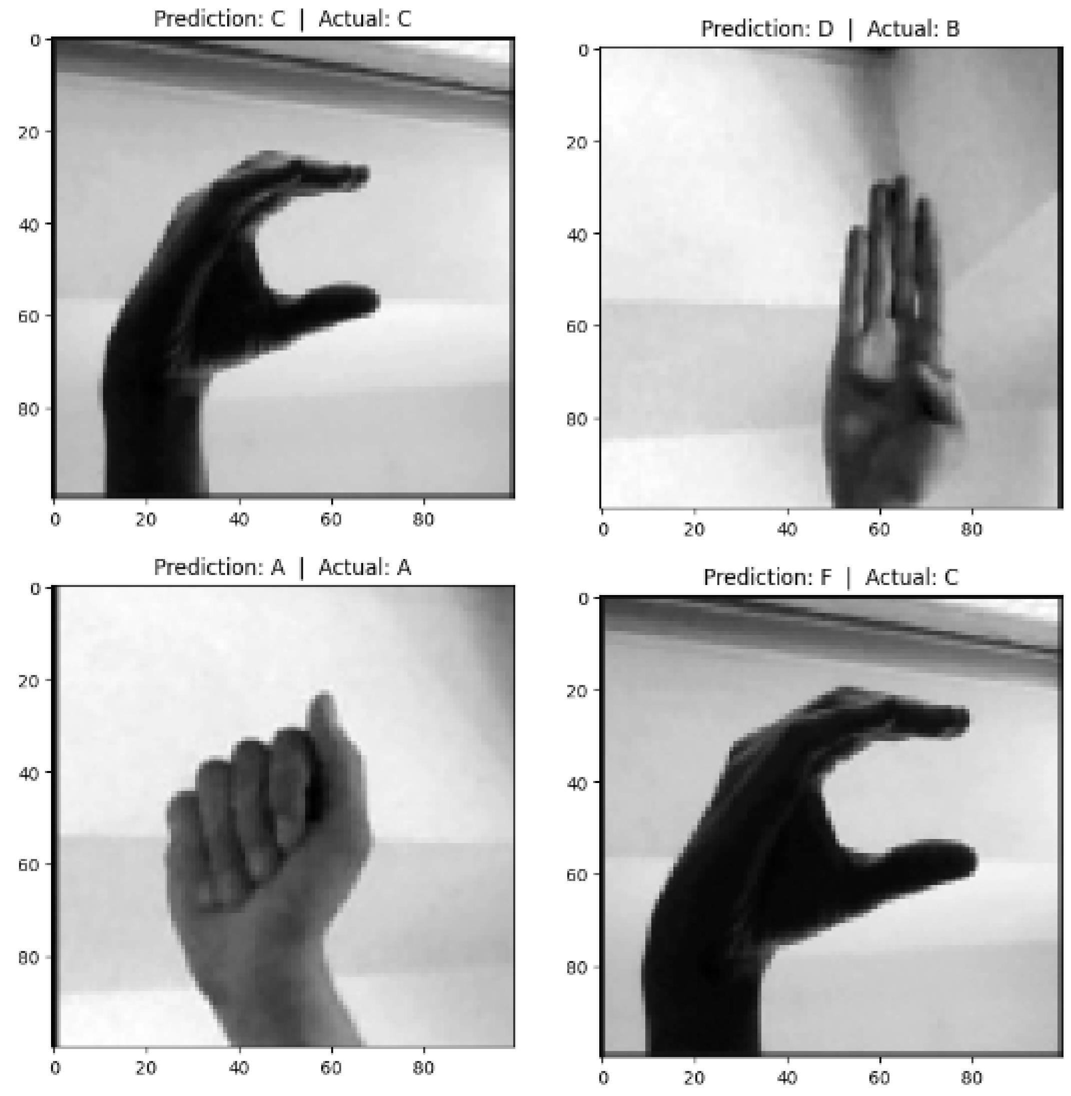
```
0.18455882352941178
```

Since we did not get good accuracy from the SparkML models we started to look into deep learning models involving deep neural networks which were not available in SparkML. We found that CNN are a good solution in many image classification problems, so we utilized CNN model architecture and performance metrics from tensorflow.keras and sklearn.

Tensorflow.keras , sklearn models and results: -

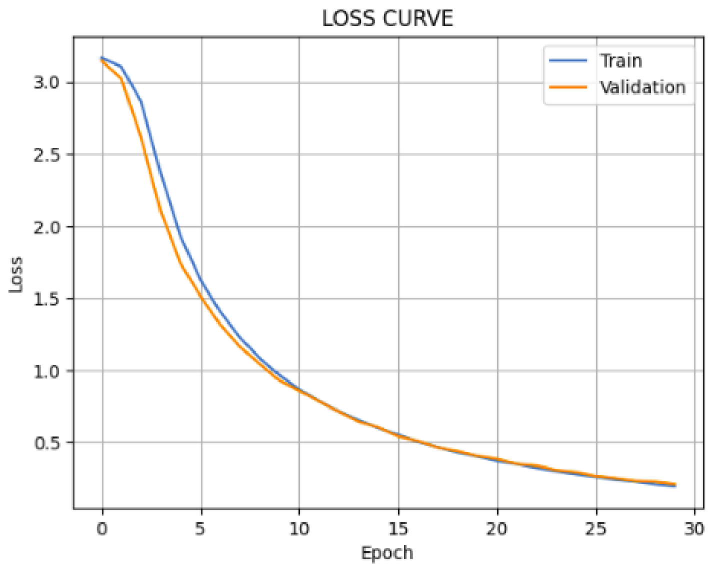
1) **CNN on original MNIST dataset:** Achieved an F1 score of 79%

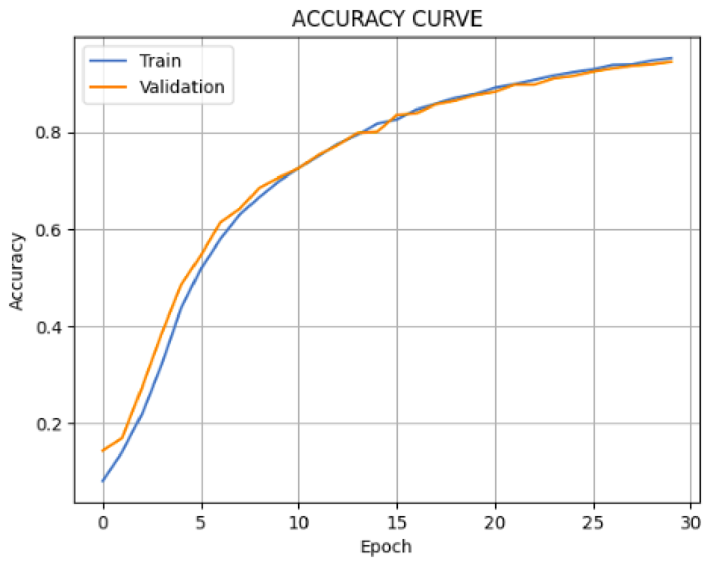
Some of the predictions from the model look like below:


Prediction: C | Actual: C
Prediction: D | Actual: B
Prediction: A | Actual: A
Prediction: F | Actual: C

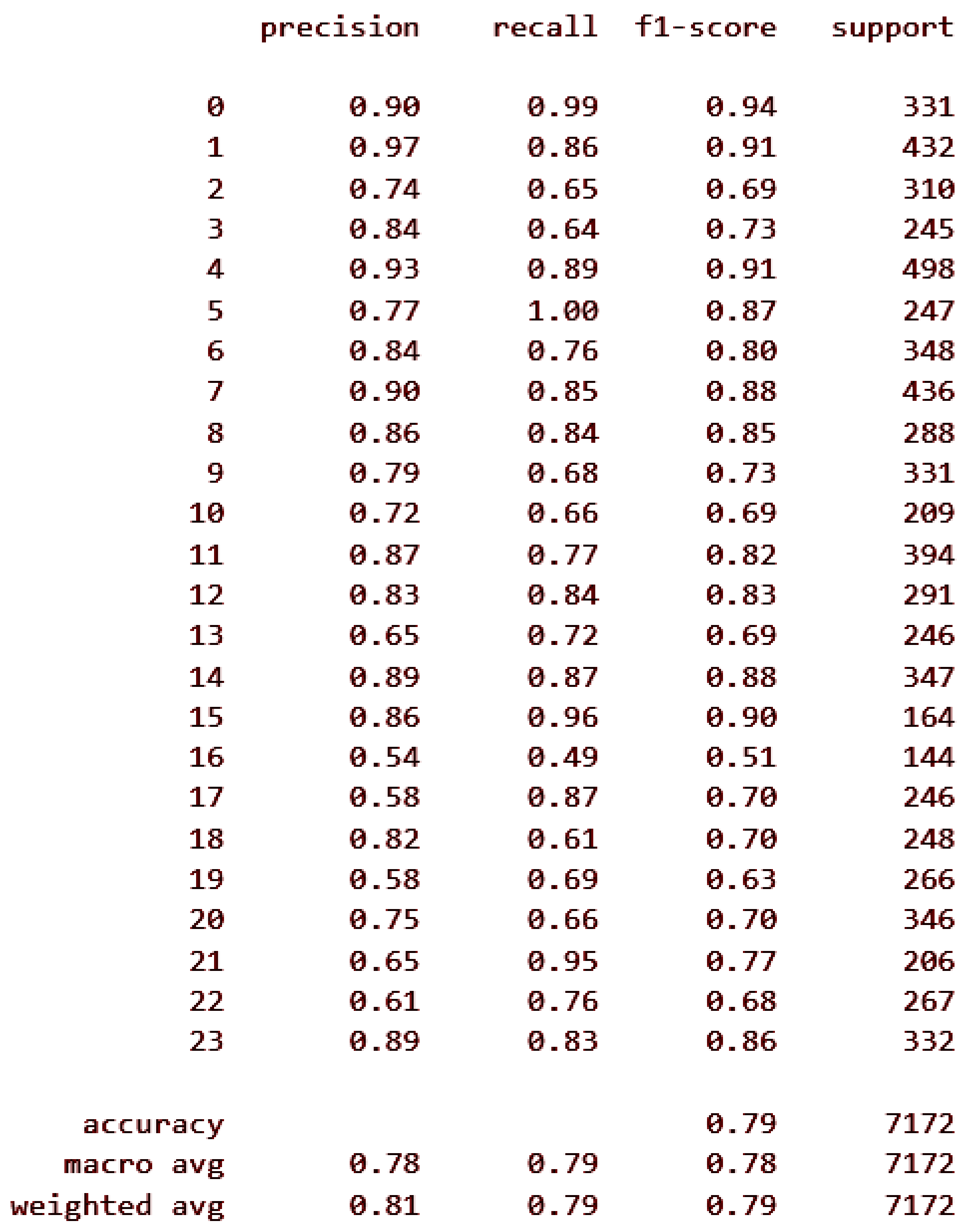
(left: correct prediction samples)
(right: wrong prediction samples)

<u>Training and Validation curves:</u>
Our loss curves and accuracy curves for training and validation are as below:





Testing Statistics:



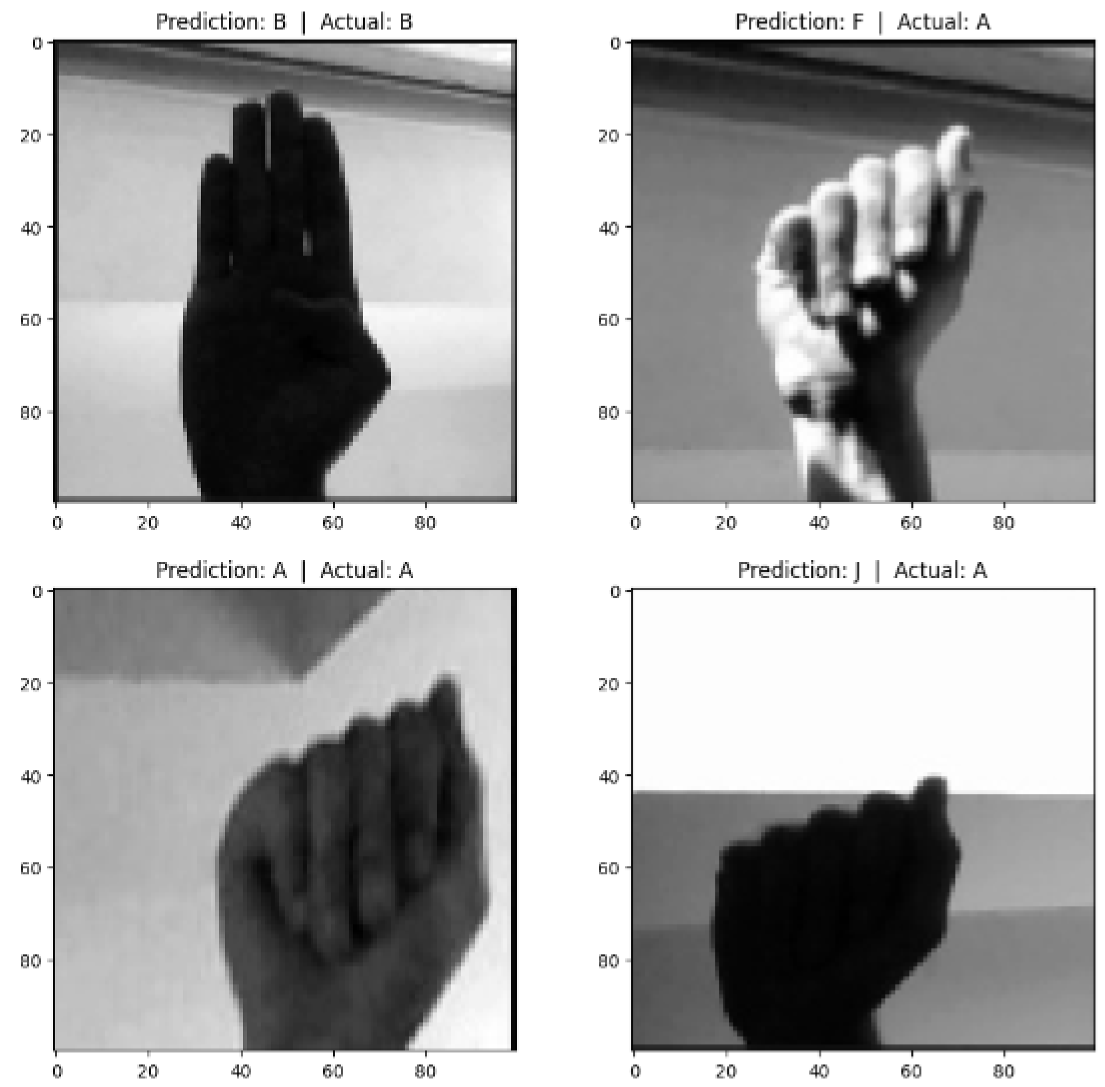
```
              precision    recall  f1-score   support

           0       0.90      0.99      0.94       331
           1       0.97      0.86      0.91       432
           2       0.74      0.65      0.69       310
           3       0.84      0.64      0.73       245
           4       0.93      0.89      0.91       498
           5       0.77      1.00      0.87       247
           6       0.84      0.76      0.80       348
           7       0.90      0.85      0.88       436
           8       0.86      0.84      0.85       288
           9       0.79      0.68      0.73       331
          10       0.72      0.66      0.69       209
          11       0.87      0.77      0.82       394
          12       0.83      0.84      0.83       291
          13       0.65      0.72      0.69       246
          14       0.89      0.87      0.88       347
          15       0.86      0.96      0.90       164
          16       0.54      0.49      0.51       144
          17       0.58      0.87      0.70       246
          18       0.82      0.61      0.70       248
          19       0.58      0.69      0.63       266
          20       0.75      0.66      0.70       346
          21       0.65      0.95      0.77       206
          22       0.61      0.76      0.68       267
          23       0.89      0.83      0.86       332

    accuracy                           0.79      7172
   macro avg       0.78      0.79      0.78      7172
weighted avg       0.81      0.79      0.79      7172
```

Overall, we achieved a 79% F1 score on test dataset. In the above classification report results, we can see which class was how well classified. The class best classified for our model was 0 which was letter A with an F1 score of 0.94 and the worst classified class which needs improvements using error analysis or data augmentation is class 16 which is letter R with an F1 score of only 0.51.

Additionally, from precision values we can find "out of all the instances predicted as positive, how many were actually positive?" and from the recall values we can find "out of all the actual positive instances, how many did the model correctly identify as positive?" for each of our 29 classes.

2) **CNN on our ASL dataset:**

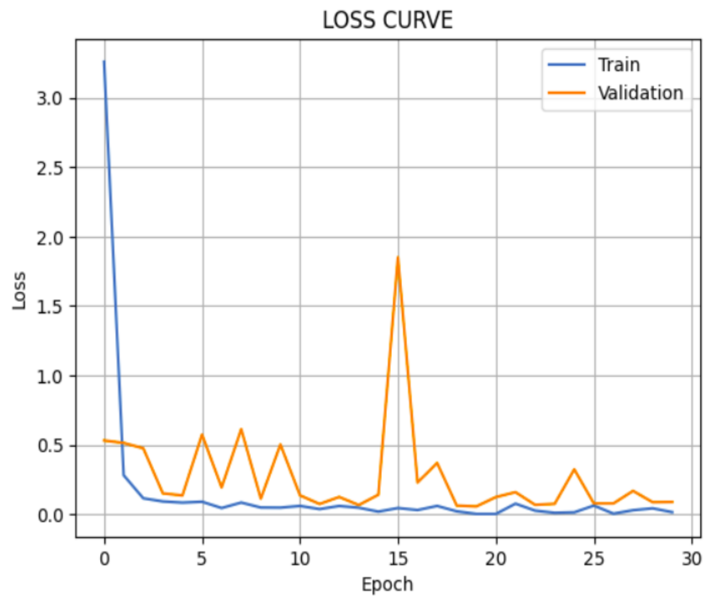
Some of the predictions from the model look like below:

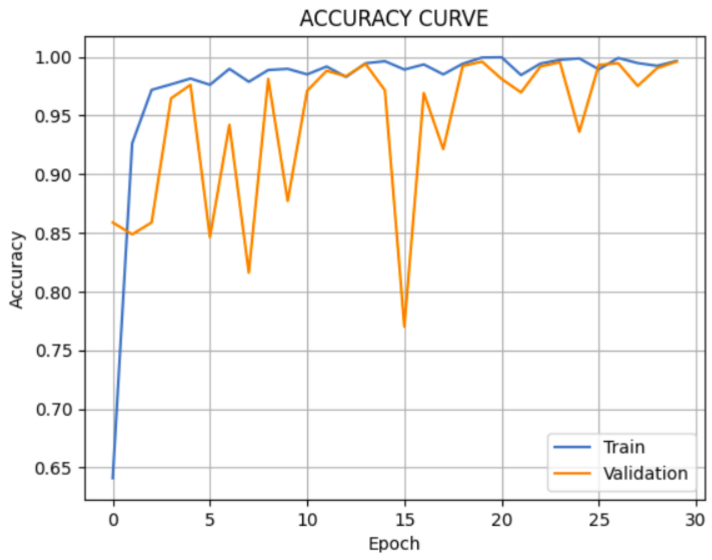

Prediction: B | Actual: B
Prediction: F | Actual: A
Prediction: A | Actual: A
Prediction: J | Actual: A

(left: correct prediction samples)
(right: wrong prediction samples)

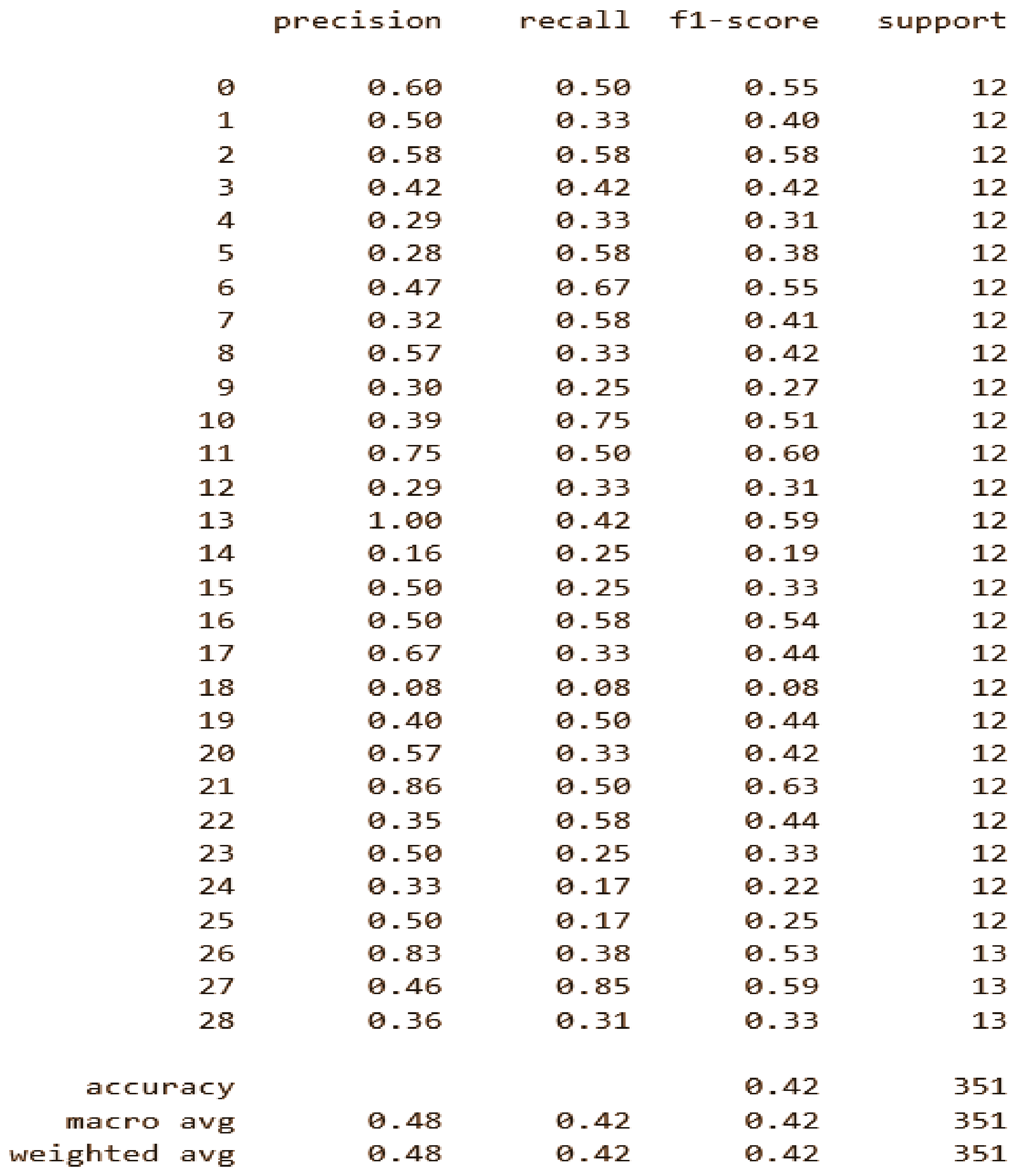
Training and Validation curves:

Our loss curves and accuracy curves for training and validation are as below:





Testing Statistics:



```
              precision    recall  f1-score   support

           0       0.60      0.50      0.55        12
           1       0.50      0.33      0.40        12
           2       0.58      0.58      0.58        12
           3       0.42      0.42      0.42        12
           4       0.29      0.33      0.31        12
           5       0.28      0.58      0.38        12
           6       0.47      0.67      0.55        12
           7       0.32      0.58      0.41        12
           8       0.57      0.33      0.42        12
           9       0.30      0.25      0.27        12
          10       0.39      0.75      0.51        12
          11       0.75      0.50      0.60        12
          12       0.29      0.33      0.31        12
          13       1.00      0.42      0.59        12
          14       0.16      0.25      0.19        12
          15       0.50      0.25      0.33        12
          16       0.50      0.58      0.54        12
          17       0.67      0.33      0.44        12
          18       0.08      0.08      0.08        12
          19       0.40      0.50      0.44        12
          20       0.57      0.33      0.42        12
          21       0.86      0.50      0.63        12
          22       0.35      0.58      0.44        12
          23       0.50      0.25      0.33        12
          24       0.33      0.17      0.22        12
          25       0.50      0.17      0.25        12
          26       0.83      0.38      0.53        13
          27       0.46      0.85      0.59        13
          28       0.36      0.31      0.33        13

    accuracy                           0.42       351
   macro avg       0.48      0.42      0.42       351
weighted avg       0.48      0.42      0.42       351
```

Overall, we achieved a 42% F1 score on test dataset. In the above classification report results, we can see which class was how well classified. The class best classified for our model was 20 which was letter V with an F1 score of 0.63 and the worst classified class which needs improvements using error analysis or data augmentation is class 18 which is letter S with an F1 score of only 0.08.

Additionally, from precision values we can find "out of all the instances predicted as positive, how many were actually positive?" and from the recall values we can find "out of all the actual positive instances, how many did the model correctly identify as positive?" for each of our 29 classes.

We can still improve our model architecture and better fine-tune, data augment, use error analysis, or transfer learning for even more accuracy which can be taken up in future scope of this project as well since we have established a strong pipeline to solve the overall problem definition of our project.