```
main_train(X_train =X_train_scaled , X_val=X_val_scaled, X_test=X_test_scaled, y_train=y_train, y_val=y_val, y_test=y_test, model_method ="knn",
        directory = "/content/drive/Shareddrives/ECEN 758 project/saved_models_and_data/Normal_Scaled_Data_Models/",
        data_changes_note="", attempt_set=3, param1 = 4, param2 = 2)
#this would then run my definition with the number of neighbors set to 4 and p set to 2, because those parameters that the paper used
```

```
1 KNN n= 4 p = 2 Test Classification Report:
           precision    recall  f1-score   support

        0       0.74      0.88      0.80      1000
        1       0.98      0.97      0.98      1000
        2       0.74      0.82      0.78      1000
        3       0.90      0.85      0.87      1000
        4       0.78      0.75      0.76      1000
        5       0.99      0.88      0.93      1000
        6       0.64      0.56      0.59      1000
        7       0.89      0.97      0.92      1000
        8       0.98      0.92      0.95      1000
        9       0.92      0.95      0.93      1000

 accuracy                           0.85     10000
 macro avg       0.86      0.85      0.85     10000
weighted avg     0.86      0.85      0.85     10000

1 KNN n= 4 p = 2 Test Confusion Matrix:
[[875    0  13  18    6    0  81    1    6    0]
 [  7 969    4  12    5    0    3    0    0    0]
 [ 25    1 818    9  77    0  69    0    1    0]
 [ 47   12  13 855   29    0  43    0    1    0]
 [  4    0 123  33 747    0  91    0    2    0]
 [  1    0    0    2    0 875    5   63    2   52]
 [207    2 120  22   89    0 557    0    3    0]
 [  0    0    0    0    0    5    0 968    0   27]
 [  9    0   10    4   10    3   25   12 924    3]
 [  1    0    0    0    0    4    0   49    0 946]]
```

```
1 KNN n= 4 p = 2 Validation Classification Report:
           precision    recall  f1-score   support

        0       0.74      0.88      0.80       597
        1       0.99      0.98      0.98       608
        2       0.74      0.82      0.78       611
        3       0.87      0.86      0.86       587
        4       0.78      0.76      0.77       627
        5       0.99      0.87      0.92       621
        6       0.66      0.53      0.59       619
        7       0.89      0.97      0.93       550
        8       0.97      0.92      0.95       590
        9       0.91      0.95      0.93       590

 accuracy                           0.85      6000
 macro avg       0.85      0.85      0.85      6000
weighted avg     0.85      0.85      0.85      6000

1 KNN n= 4 p = 2 Validation Confusion Matrix:
[[524    0  11  15    1    0  41    0    5    0]
 [  1 593    1  13    0    0    0    0    0    0]
 [ 15    0 503    3  48    0  42    0    0    0]
 [ 33    4    7 505   24    0  13    0    1    0]
 [  2    0  70  28 477    0  47    0    3    0]
 [  1    0    0    0    0 539    9   35    1   36]
 [128    0  82  18   58    0 329    0    4    0]
 [  0    0    0    0    0    0    1 536    0   13]
 [  3    1    8    1    7    2   13    6 545    4]
 [  0    0    0    0    0    4    2   24    0 560]]
```

1 KNN n= 4 p = 2 Validation Confusion Matrix

1 KNN n= 4 p = 2 Test Confusion Matrix Confusion Matrix