Kaggle Results - Wendy Liang

Used “reviewText”, “summary”, and length of text

Notes

- for all classifiers for all classification tasks except AdaBoostClassifier : I only tuned C because in this project specifically, C is the most important to tune in classifier when my TfidfVectorizer already has the following hyperparameters:

TfidfVectorizer(ngram_range= (1,3),lowercase = True, max_df =0.9, min_df=0.01)

Too time-consuming- It’s simply too inefficient to tune further and only optimize accuracy by one percent

- For AdaBoostClassifier, I tuned the learning rate because it is very essential to a ml model when it determines the performance on gradient descent

Classification Tasks	Scores on Kaggle	Best Model
Binary Classification Cutoff 1	0.81005	LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1)
Binary Classification Cutoff 2	0.82506	LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1)
Binary Classification Cutoff 3	0.87244	LinearSVC(random_state=42, class_weight='balanced', C=0.1)
Binary Classification Cutoff 4	0.84083	LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1)
Multiclass Classification	0.60296	OneVsRestClassifier() with LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1)

Clustering	0.5997451777956766 (silhouette)	KMeans(n_clusters=6, algorithm='auto',init='k-means++')

Binary Classification

LogisticRegression

Hyperparameters used: random_state=42, class_weight='balanced', max_iter=500
Hyperparameters tuned: C = [0.01, 0.1, 1, 10]

Binary Classifier for Y1

To determine best Hyperparameters:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 1}
   clf__C  Mean F1 Macro  Ranks
0    0.01        0.747307      4
1    0.10        0.757178      3
2    1.00        0.762236      1
3   10.00        0.759338      2
```

Report on all the validation scores for model with best hyperparameters:

We now proceed with the LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.745223  0.741273  0.755647  0.742007  0.752385
1    0.10  0.754059  0.747920  0.767958  0.755254  0.760697
2    1.00  0.759273  0.757752  0.771879  0.753584  0.768692
3   10.00  0.756759  0.758990  0.765746  0.749749  0.765444
```

Validation scores: 0.759273 0.757752 0.771879 0.753584 0.768692

Report the best model in multiple metrics

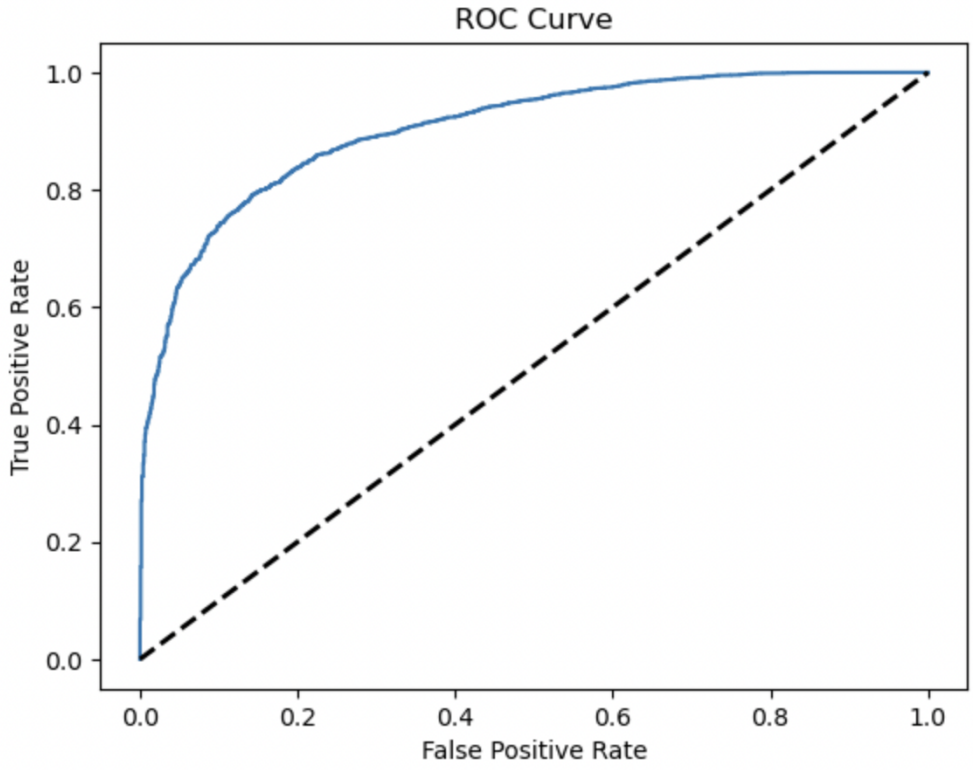
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

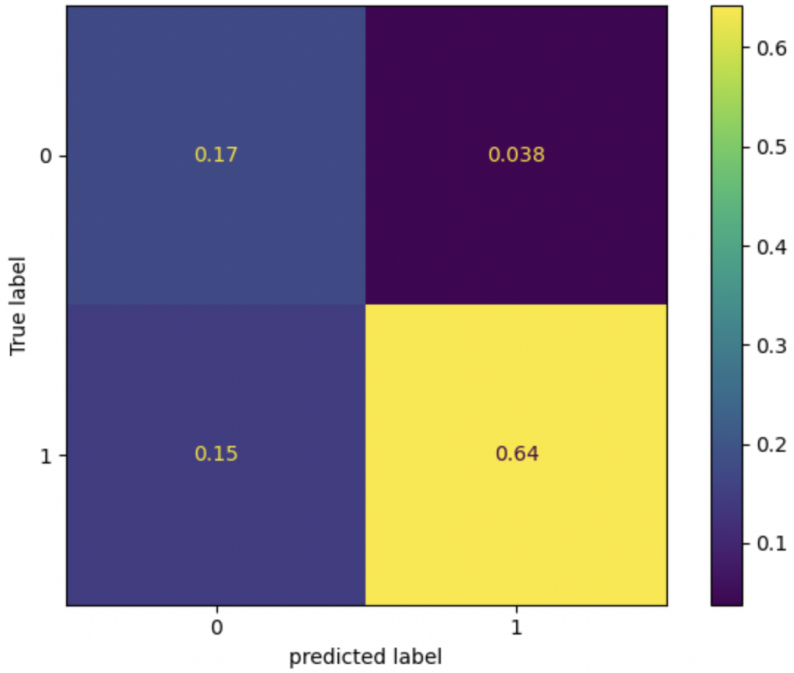
```
Evaluation Report for  LogReg
Accuracy:  0.815861596437136
F1 Macro Score:  0.7639866007673628
AUC:  0.9043778889091426
Classification Report
              precision    recall  f1-score   support

           0       0.54      0.82      0.65      1233
           1       0.94      0.81      0.87      4605

    accuracy                           0.82      5838
   macro avg       0.74      0.82      0.76      5838
weighted avg       0.86      0.82      0.83      5838
```

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fdc599227f0>





Binary Classifier for Y2

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 1}
   clf__C  Mean F1 Macro  Ranks
0    0.01        0.783929      4
1    0.10        0.813168      3
2    1.00        0.824120      1
3   10.00        0.822021      2
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.774852  0.780710  0.791749  0.780252  0.792084
1    0.10  0.807364  0.810242  0.816183  0.815693  0.816358
2    1.00  0.816862  0.821433  0.825078  0.828603  0.828625
3   10.00  0.814594  0.822173  0.818174  0.826716  0.828448
```

Validation scores: 0.816862 0.821433 0.825078 0.828603 0.828625

<u>Report the best model in multiple metrics</u>

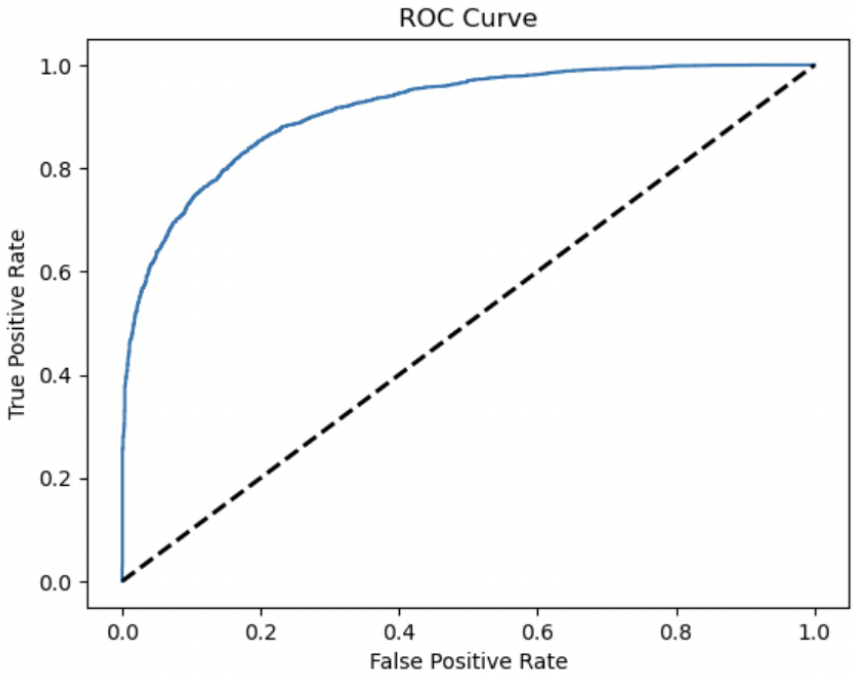
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

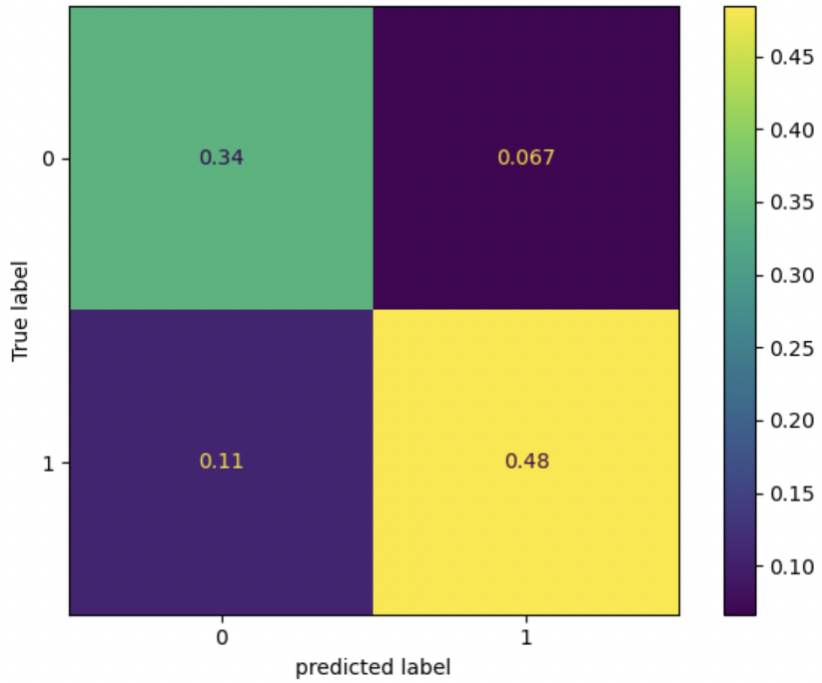
```
Evaluation Report for  LogReg
Accuracy:  0.8254539225762247
F1 Macro Score:  0.8217907879098337
AUC:  0.913220890919179
Classification Report
              precision    recall  f1-score   support

           0       0.76      0.84      0.80      2381
           1       0.88      0.82      0.85      3457

    accuracy                           0.83      5838
   macro avg       0.82      0.83      0.82      5838
weighted avg       0.83      0.83      0.83      5838
```

```
Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f9f3c43de80>
```





Binary Classifier for Y3

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 1}
   clf__C  Mean F1 Macro  Ranks
0    0.01        0.804044      4
1    0.10        0.828948      3
2    1.00        0.835735      1
3   10.00        0.834172      2
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.800865  0.813682  0.805990  0.804819  0.794862
1    0.10  0.823843  0.836848  0.833095  0.830805  0.820150
2    1.00  0.830046  0.841574  0.839287  0.837854  0.829913
3   10.00  0.826062  0.839683  0.839760  0.838082  0.827271
```

Validation scores: 0.830046 0.841574 0.839287 0.837854 0.829913

<u>Report the best model in multiple metrics</u>

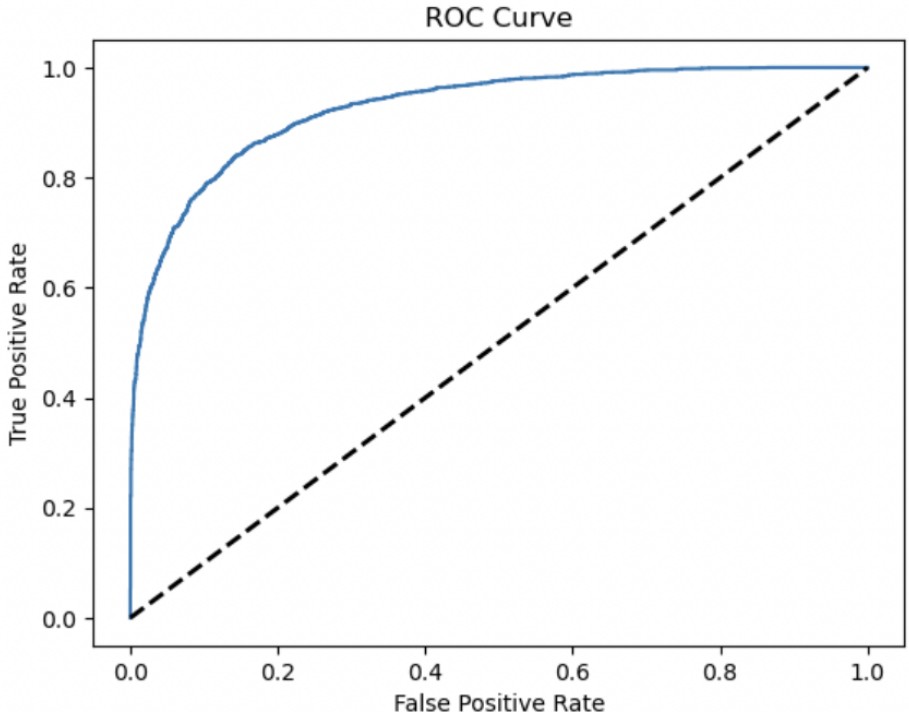
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

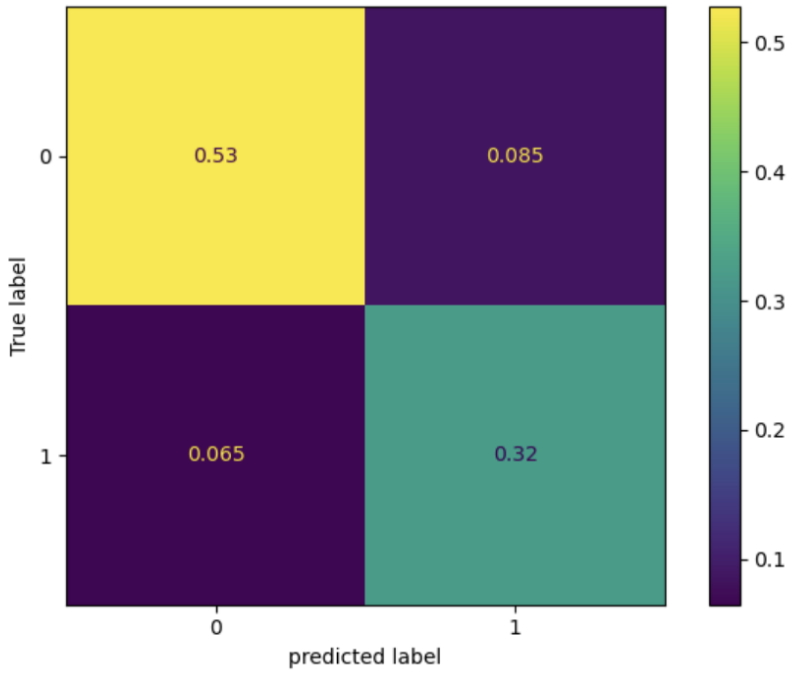
```
Evaluation Report for  LogReg
Accuracy:  0.8504624871531347
F1 Macro Score:  0.8439000858667234
AUC:  0.9275472399749187
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.86      0.88      3576
           1       0.79      0.83      0.81      2262

    accuracy                           0.85      5838
   macro avg       0.84      0.85      0.84      5838
weighted avg       0.85      0.85      0.85      5838
```

```
Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fa0f875f760>
```





Binary Classifier for Y4

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 0.1}
   clf__C  Mean F1 Macro  Ranks
0    0.01        0.799284      3
1    0.10        0.807767      1
2    1.00        0.803466      2
3   10.00        0.792766      4
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.797888  0.786422  0.813234  0.794065  0.804810
1    0.10  0.805599  0.799864  0.812591  0.812894  0.807886
2    1.00  0.803565  0.795610  0.811967  0.804540  0.801649
3   10.00  0.791602  0.784885  0.797779  0.793198  0.796368
```

Validation scores: 0.805599 0.799864 0.812591 0.812894 0.807886

<u>Report the best model in multiple metrics</u>

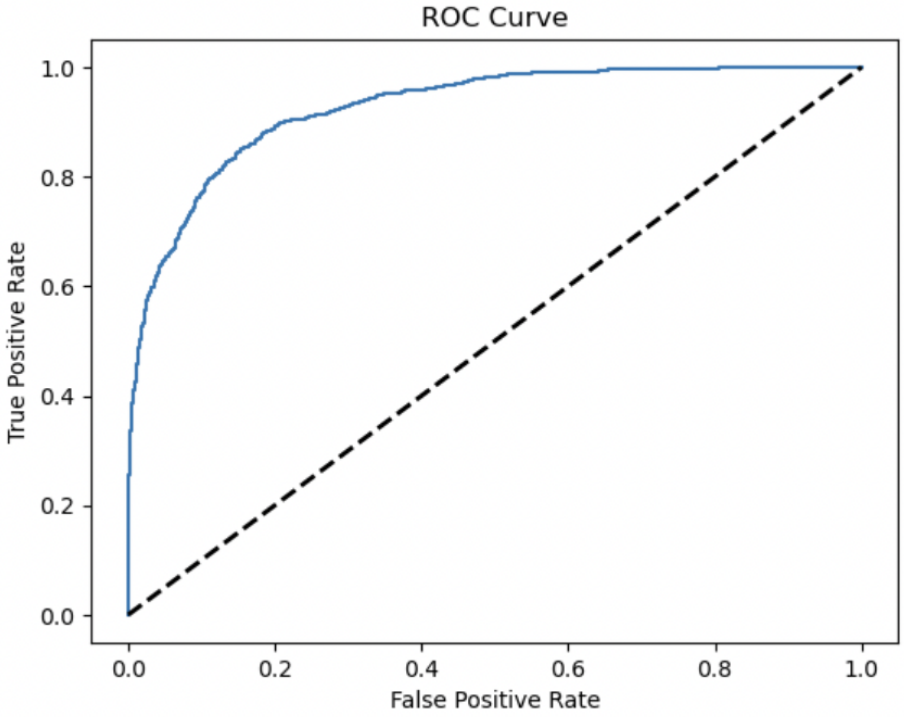
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

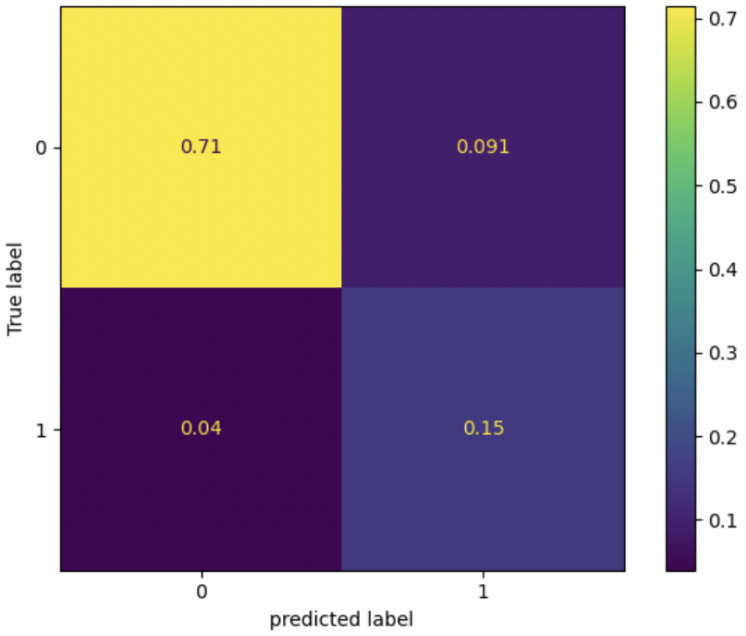
```
Evaluation Report for  LogReg
Accuracy:  0.8689619732785201
F1 Macro Score:  0.8091087222482691
AUC:  0.9265583832714442
Classification Report
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      4705
           1       0.63      0.80      0.70      1133

    accuracy                           0.87      5838
   macro avg       0.79      0.84      0.81      5838
weighted avg       0.89      0.87      0.87      5838
```

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f7a19a78130>





Support Vector Classifier

Hyperparameters used: random_state=42, class_weight='balanced'
Hyperparameters tuned: C = [0.01, 0.1, 1, 10]

Binary Classifier for Y1

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 0.1}
   clf__C  Mean F1 Macro  Ranks
0    0.01       0.757747      2
1    0.10       0.761610      1
2    1.00       0.756718      3
3   10.00       0.754719      4
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the LinearSVC(random_state=42, class_weight='balanced', C=0.1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.754748  0.748252  0.769807  0.757382  0.758546
1    0.10  0.760097  0.757719  0.771442  0.753647  0.765142
2    1.00  0.752220  0.758131  0.762267  0.746264  0.764710
3   10.00  0.751525  0.756814  0.758167  0.743685  0.763403
```

Validation scores: 0.760097 0.757719 0.771442 0.753647 0.765142

Report the best model in multiple metrics

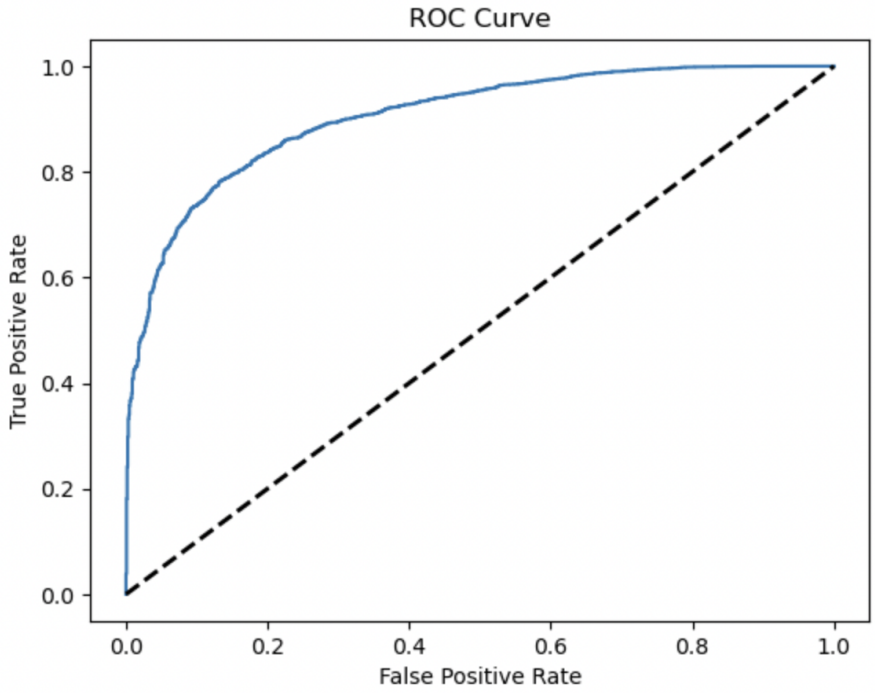
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

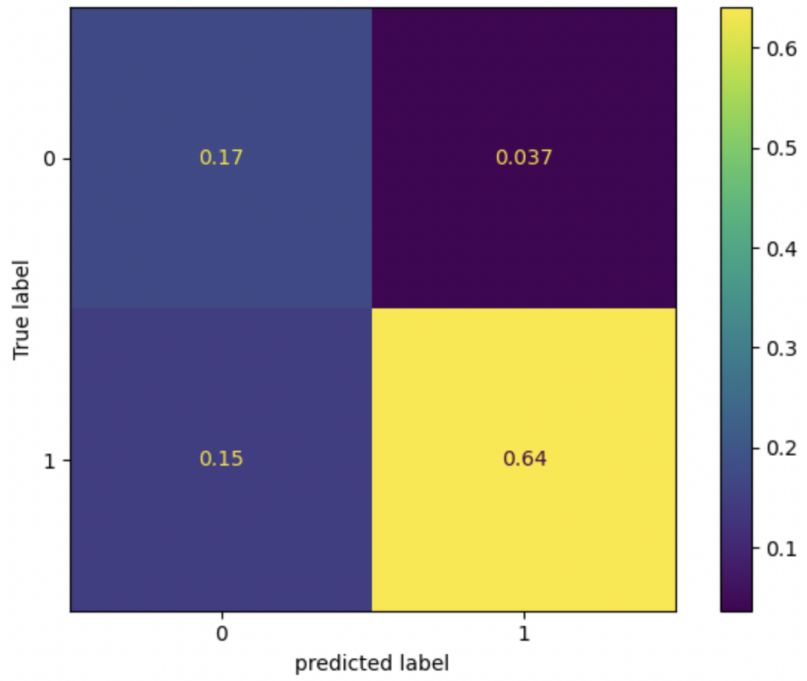
```
Evaluation Report for  SVC
Accuracy:  0.8139773895169579
F1 Macro Score:  0.7623035463630583
AUC:  0.9044596083279837
Classification Report
              precision    recall  f1-score   support

           0       0.54      0.82      0.65      1233
           1       0.94      0.81      0.87      4605

    accuracy                           0.81      5838
   macro avg       0.74      0.82      0.76      5838
weighted avg       0.86      0.81      0.83      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fbd383ee550>
```





Binary Classifier for Y2

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 0.1}
   clf__C  Mean F1 Macro  Ranks
0    0.01        0.813051      4
1    0.10        0.824401      1
2    1.00        0.822091      2
3   10.00        0.821303      3
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the LinearSVC(random_state=42, class_weight='balanced', C=0.1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.806838  0.809584  0.815878  0.816300  0.816651
1    0.10  0.818153  0.821433  0.824523  0.828833  0.829063
2    1.00  0.813768  0.824108  0.817321  0.826277  0.828979
3   10.00  0.812085  0.822750  0.817260  0.825720  0.828702
```

Validation scores: 0.818153 0.821433 0.824523 0.828833 0.829063

Report the best model in multiple metrics

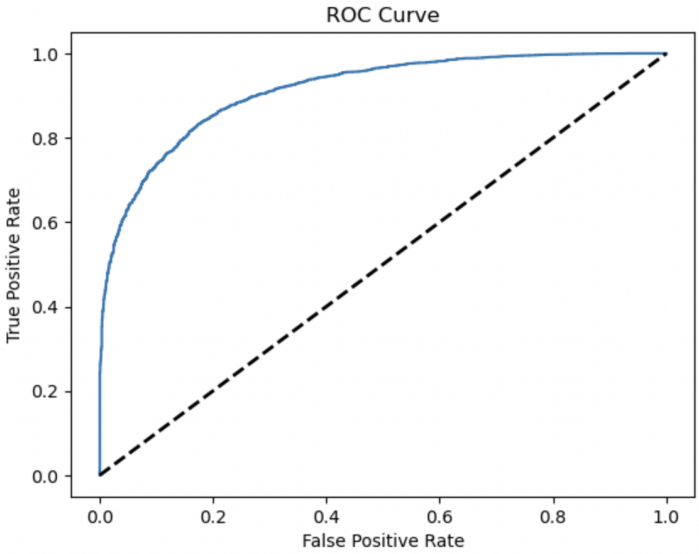
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

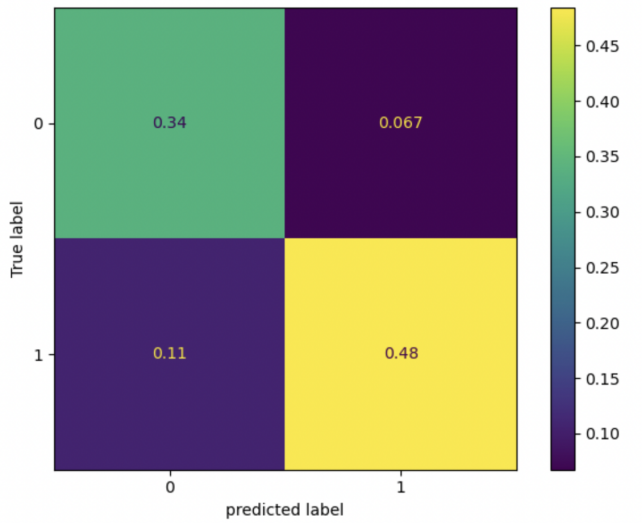
```
Evaluation Report for  SVC
Accuracy:  0.8247687564234327
F1 Macro Score:  0.8210732756106411
AUC:  0.9125107807360775
Classification Report
              precision    recall  f1-score   support

           0       0.76      0.83      0.80      2381
           1       0.88      0.82      0.85      3457

    accuracy                           0.82      5838
   macro avg       0.82      0.83      0.82      5838
weighted avg       0.83      0.82      0.83      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f9f9f584dc0>
```





Binary Classifier for Y3

To determine best Hyperparameters:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__C': 0.1}
   clf__C  Mean F1 Macro  Ranks
0    0.01        0.829806      4
1    0.10        0.835924      1
2    1.00        0.835112      2
3   10.00        0.833546      3
```

Report on all the validation scores for model with best hyperparameters:

We now proceed with the LinearSVC(random_state=42, class_weight='balanced', C=0.1) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__C     Test1     Test2     Test3     Test4     Test5
0    0.01  0.826137  0.838086  0.836279  0.829252  0.819277
1    0.10  0.830046  0.841033  0.840330  0.837882  0.830332
2    1.00  0.825108  0.840936  0.839343  0.839350  0.830824
3   10.00  0.826823  0.839711  0.835593  0.835221  0.830380
```

Validation scores: 0.830046 0.841033 0.840330 0.837882 0.830332

Report the best model in multiple metrics

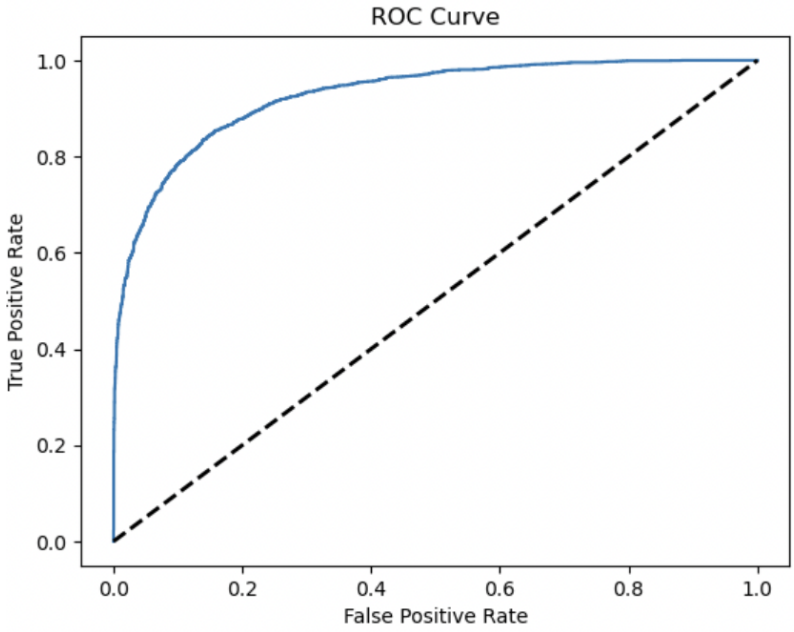
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

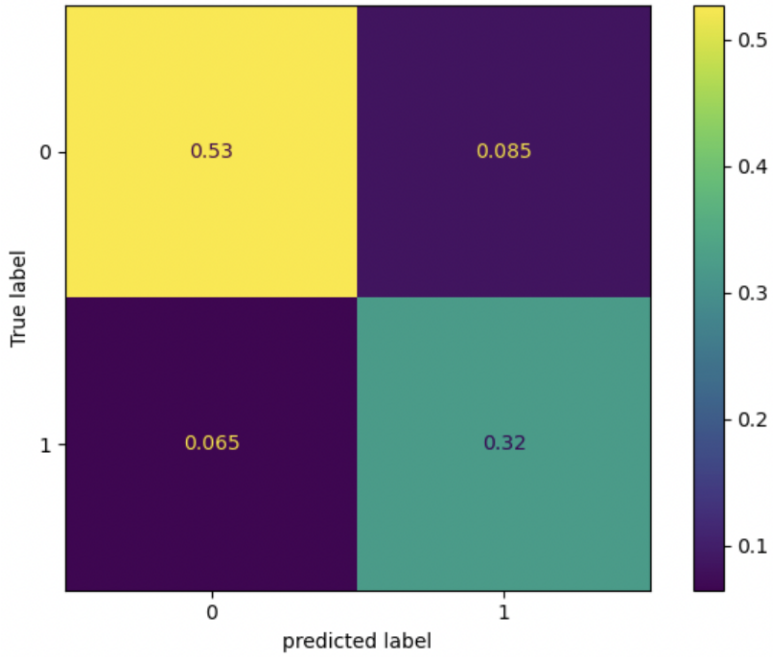
```
Evaluation Report for  SVC
Accuracy:  0.8496060294621446
F1 Macro Score:  0.8430405264380183
AUC:  0.9271609086611401
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.86      0.88      3576
           1       0.79      0.83      0.81      2262

    accuracy                           0.85      5838
   macro avg       0.84      0.85      0.84      5838
weighted avg       0.85      0.85      0.85      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fa0f875fdf0>
```





Binary Classifier for Y4

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

Best Parameters: {'clf__C': 0.01}

	clf__C	Mean F1 Macro	Ranks
0	0.01	0.807895	1
1	0.10	0.804209	2
2	1.00	0.792330	3
3	10.00	0.786690	4

Report on all the validation scores for model with best hyperparameters:

We now proceed with the LinearSVC(random_state=42, class_weight='balanced', C=0.1) because above gridsearch shows the best result

Scores for Best Hyper:

	clf__C	Test1	Test2	Test3	Test4	Test5
0	0.01	0.805993	0.800567	0.812552	0.812446	0.807920
1	0.10	0.804879	0.796021	0.812213	0.805416	0.802515
2	1.00	0.794517	0.782756	0.797724	0.791795	0.794858
3	10.00	0.786539	0.776592	0.790270	0.784538	0.795511

Validation scores: 0.805993 0.800567 0.812552 0.812446 0.807920

Report the best model in multiple metrics

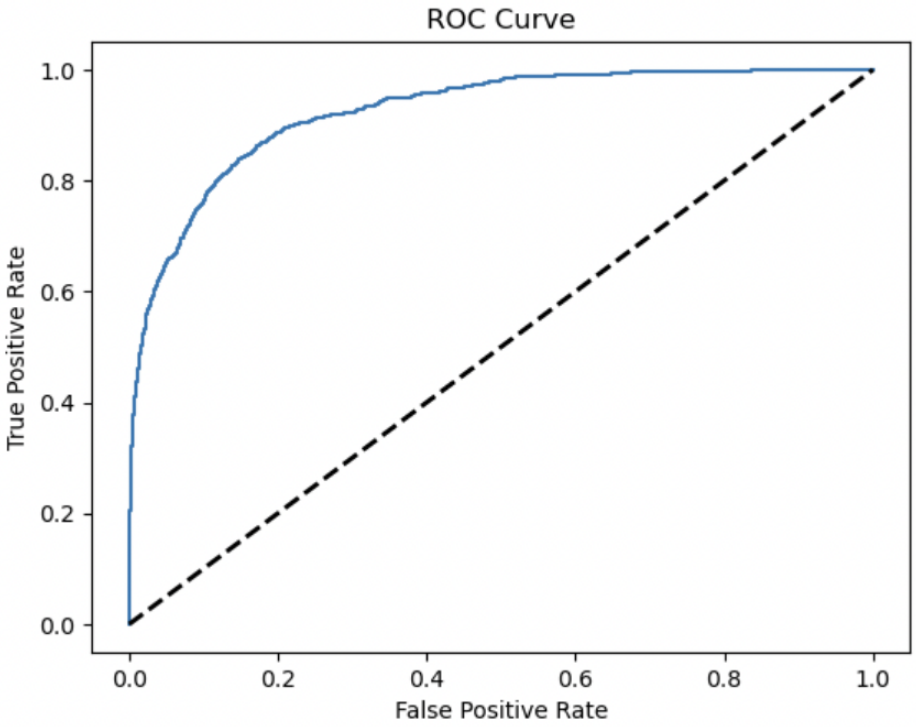
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

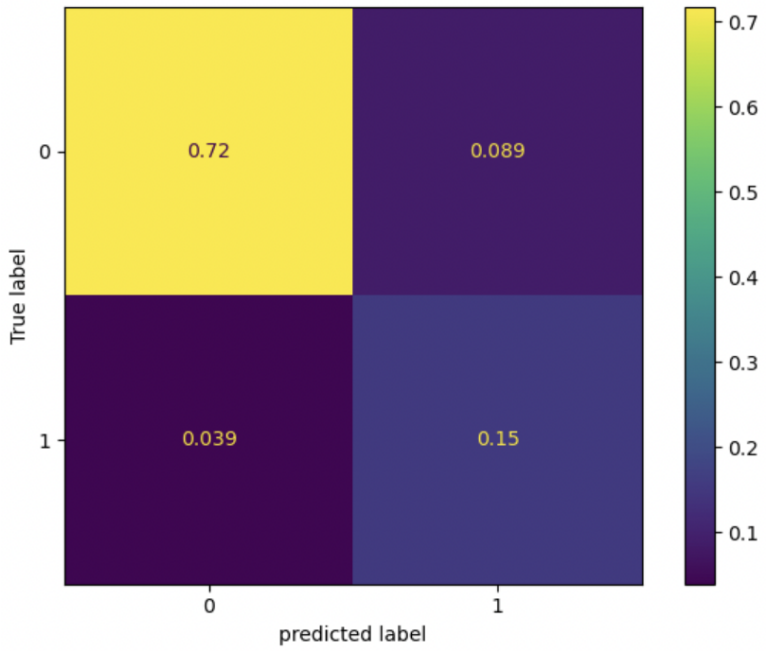
```
Evaluation Report for  SVC
Accuracy:  0.8717026378896883
F1 Macro Score:  0.8123655631367024
AUC:  0.9249008350583829
Classification Report
              precision    recall  f1-score   support

           0       0.95      0.89      0.92      4705
           1       0.64      0.80      0.71      1133

    accuracy                           0.87      5838
   macro avg       0.79      0.84      0.81      5838
weighted avg       0.89      0.87      0.88      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f79ce7de9a0>
```





AdaBoostClassifier

Hyperparameters used: random_state=42
Hyperparameters tuned: learning_rate= [0.8, 0.9, 1.0]
Binary Classifier for Y1

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__learning_rate': 1.0}
   clf__learning_rate  Mean F1 Macro  Ranks
0                 0.8       0.722807      3
1                 0.9       0.728648      2
2                 1.0       0.734429      1
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the AdaBoostClassifier(random_state=42, learning_rate= 1.0) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__learning_rate     Test1     Test2     Test3     Test4     Test5
0                 0.8  0.727488  0.736636  0.721985  0.705517  0.722411
1                 0.9  0.738022  0.735399  0.726970  0.714306  0.728542
2                 1.0  0.739178  0.743068  0.736123  0.720380  0.733396
-------------
```

Validation scores: 0.739178 0.743068 0.736123 0.720380 0.733396

Report the best model in multiple metrics

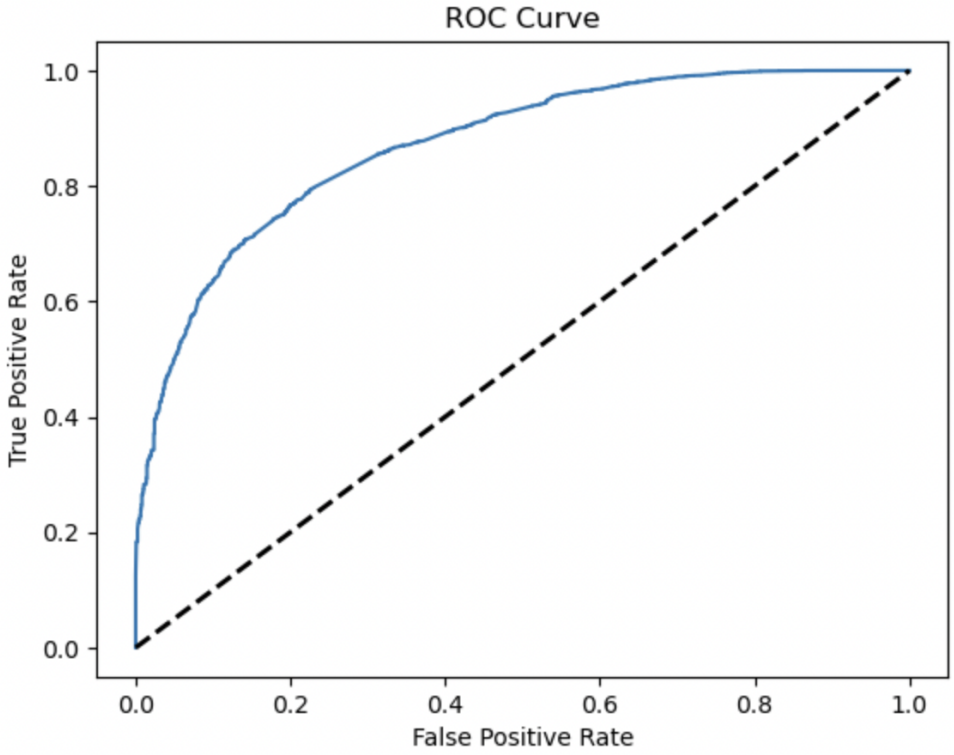
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

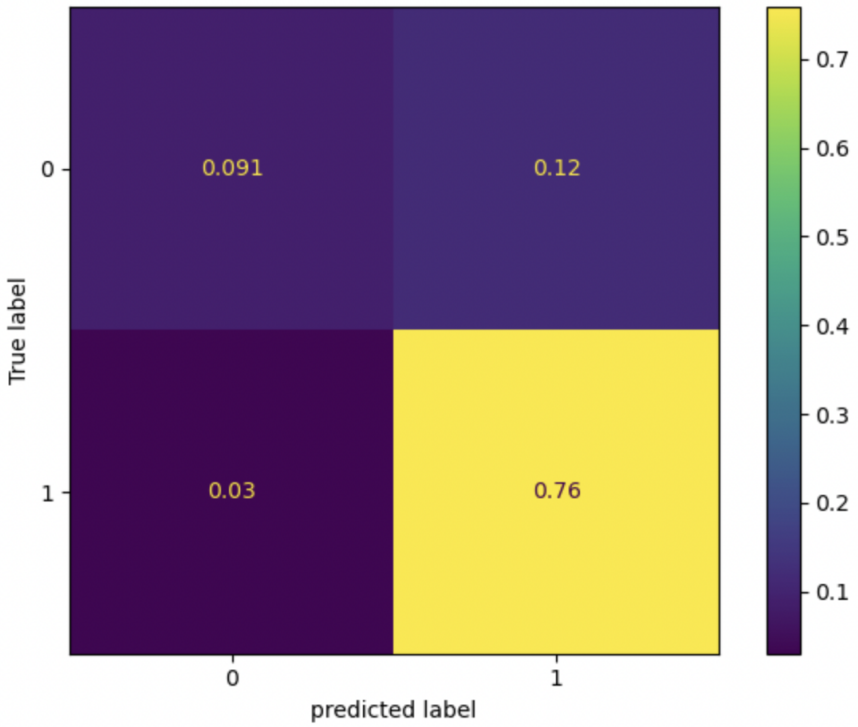
```
Evaluation Report for  Ada
Accuracy:  0.8502911956149366
F1 Macro Score:  0.730080068017293
AUC:  0.8700431404561317
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.43      0.55      1233
           1       0.86      0.96      0.91      4605

    accuracy                           0.85      5838
   macro avg       0.81      0.70      0.73      5838
weighted avg       0.84      0.85      0.83      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fdc7a72a430>
```





Binary Classifier for Y2

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__learning_rate': 1.0}
   clf__learning_rate  Mean F1 Macro  Ranks
0                 0.8       0.779029      3
1                 0.9       0.780465      2
2                 1.0       0.780774      1
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the AdaBoostClassifier(random_state=42, learning_rate= 1.0) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__learning_rate     Test1     Test2     Test3     Test4     Test5
0                 0.8  0.776527  0.773887  0.781277  0.776245  0.787208
1                 0.9  0.776624  0.775031  0.782327  0.781337  0.787004
2                 1.0  0.772125  0.778005  0.787330  0.780944  0.785464
```

Validation scores: 0.772125 0.778005 0.787330 0.780944 0.785464

<u>Report the best model in multiple metrics</u>

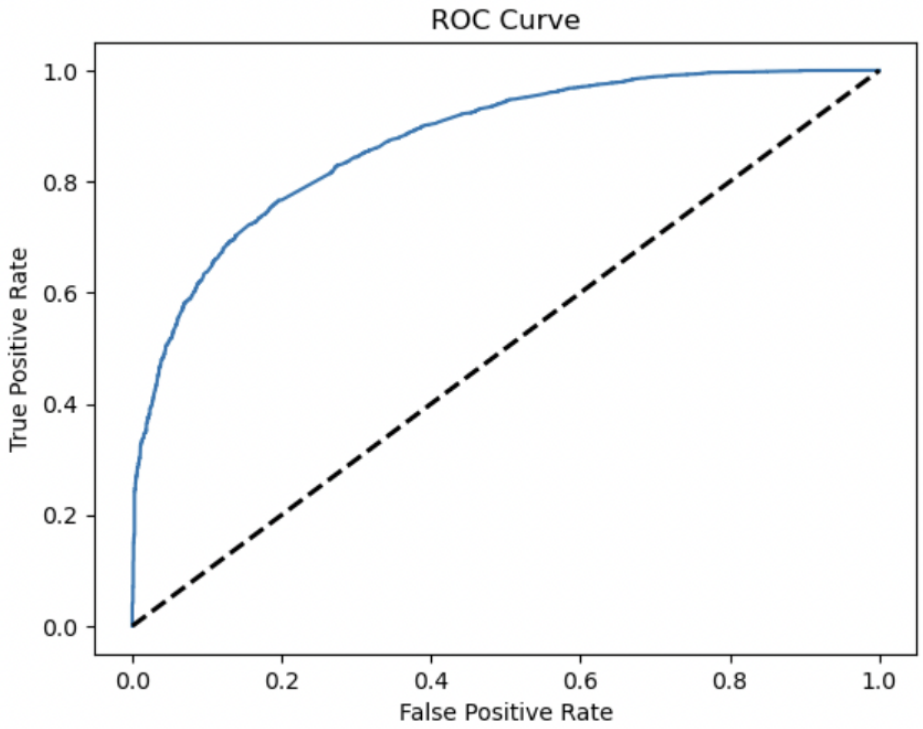
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

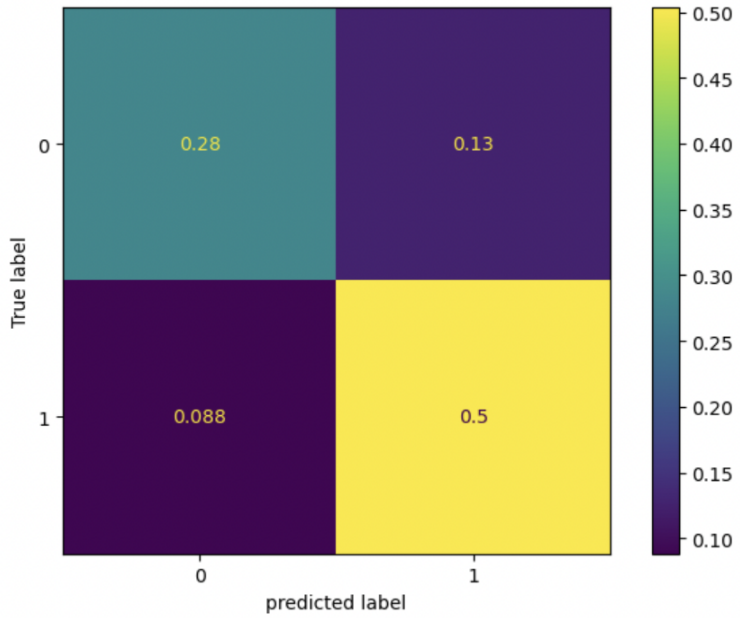
```
Evaluation Report for  Ada
Accuracy:  0.7858855772524838
F1 Macro Score:  0.774822783447742
AUC:  0.8725700898189153
Classification Report
              precision    recall  f1-score   support

           0       0.76      0.69      0.72      2381
           1       0.80      0.85      0.82      3457

    accuracy                           0.79      5838
   macro avg       0.78      0.77      0.77      5838
weighted avg       0.78      0.79      0.78      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f9f49835ac0>
```





Binary Classifier for Y3

To determine best Hyperparameters:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__learning_rate': 0.8}
   clf__learning_rate  Mean F1 Macro  Ranks
0                 0.8       0.791509      1
1                 0.9       0.790675      3
2                 1.0       0.791117      2
```

Report on all the validation scores for model with best hyperparameters:

We now proceed with the AdaBoostClassifier(random_state=42, learning_rate= 1.0) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__learning_rate     Test1     Test2     Test3     Test4     Test5
0                 0.8  0.801599  0.788364  0.793065  0.794046  0.780471
1                 0.9  0.800812  0.787291  0.794878  0.793157  0.777239
2                 1.0  0.796166  0.789321  0.790970  0.796800  0.782331
```

Validation scores: 0.801599 0.788364 0.793065 0.794046 0.780471

Report the best model in multiple metrics

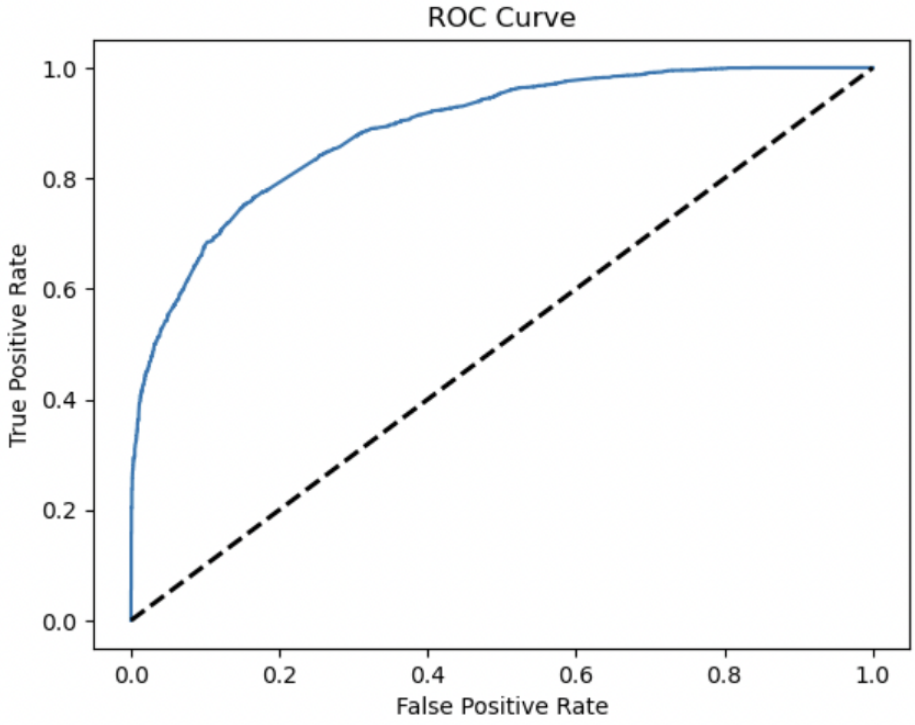
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

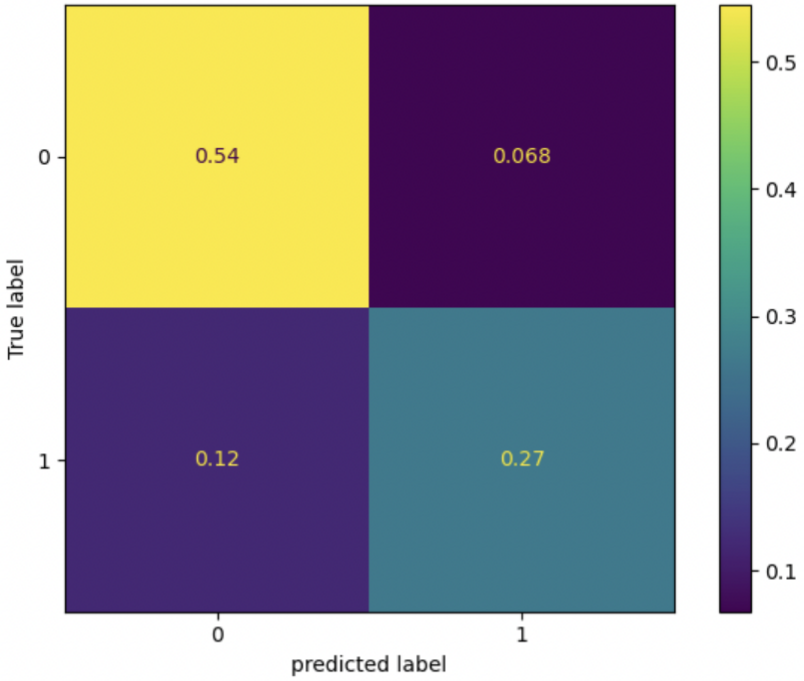
```
Evaluation Report for  Ada
Accuracy:  0.8115793079821857
F1 Macro Score:  0.7957758957677927
AUC:  0.8881727851656687
Classification Report
              precision    recall  f1-score   support

           0       0.82      0.89      0.85      3576
           1       0.80      0.69      0.74      2262

    accuracy                           0.81      5838
   macro avg       0.81      0.79      0.80      5838
weighted avg       0.81      0.81      0.81      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fa04b8de8e0>
```





Binary Classifier for Y4

<u>To determine best Hyperparameters</u>:

5-fold cross-validation score (Mean F1 Macro)

```
Best Parameters:  {'clf__learning_rate': 1.0}
   clf__learning_rate  Mean F1 Macro  Ranks
0                 0.8       0.772835      3
1                 0.9       0.777100      2
2                 1.0       0.778448      1
```

<u>Report on all the validation scores for model with best hyperparameters:</u>

We now proceed with the AdaBoostClassifier(random_state=42, learning_rate= 1.0) because above gridsearch shows the best result

```
Scores for Best Hyper:
   clf__learning_rate     Test1     Test2     Test3     Test4     Test5
0                 0.8  0.778201  0.755888  0.782294  0.783454  0.764337
1                 0.9  0.777675  0.768405  0.789841  0.774135  0.775444
2                 1.0  0.777083  0.766338  0.787583  0.778498  0.782737
```

Validation scores: 0.777083 0.766338 0.787583 0.778498 0.782737

<u>Report the best model in multiple metrics</u>

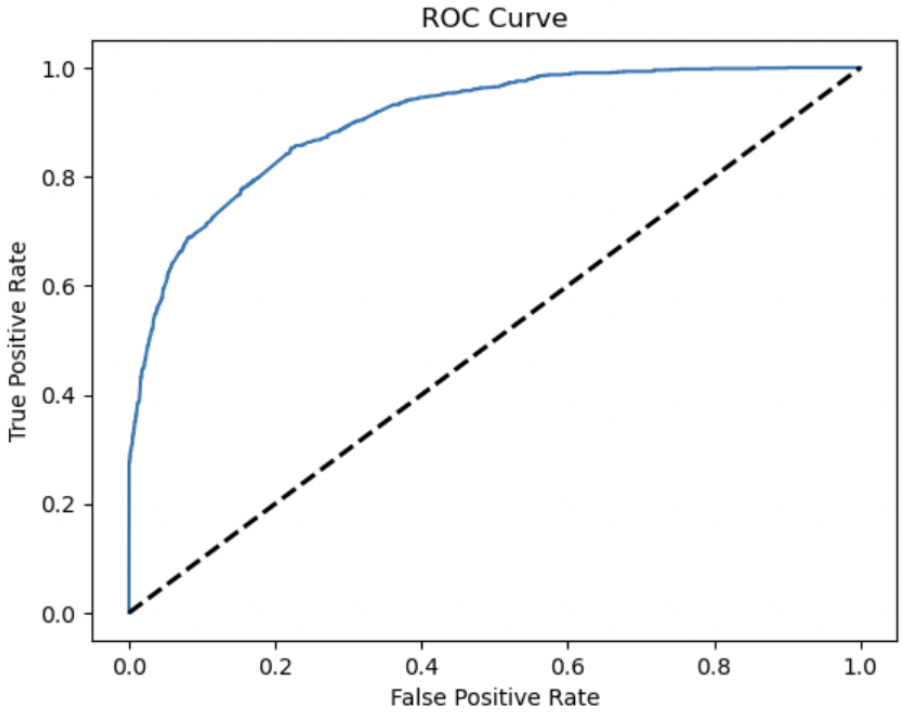
(e.g. confusion matrix, ROC, AUC, macro F1 score, and accuracy)

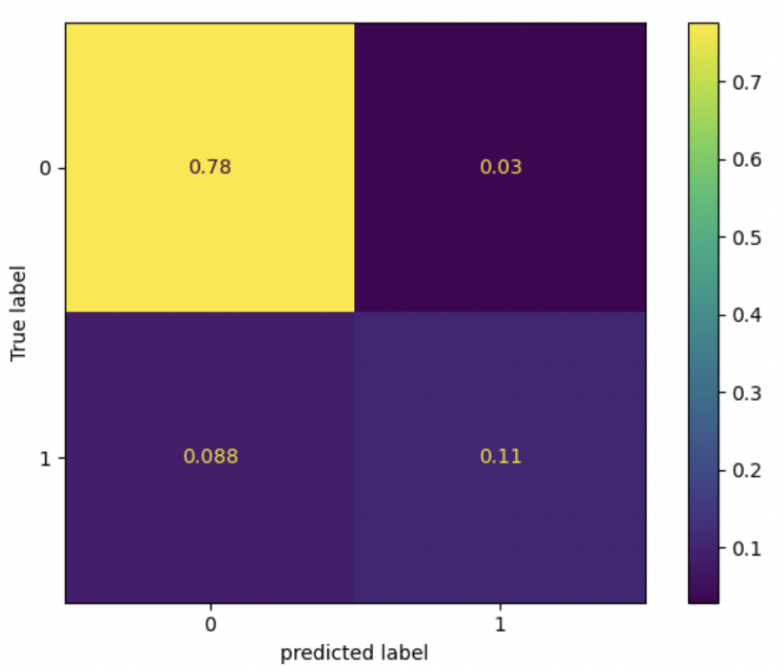
```
Evaluation Report for  Ada
Accuracy:  0.8826652963343611
F1 Macro Score:  0.7873078465264505
AUC:  0.9039891085050645
Classification Report
              precision    recall  f1-score   support

           0       0.90      0.96      0.93      4705
           1       0.78      0.55      0.64      1133

    accuracy                           0.88      5838
   macro avg       0.84      0.76      0.79      5838
weighted avg       0.88      0.88      0.87      5838

Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f7964a481f0>
```



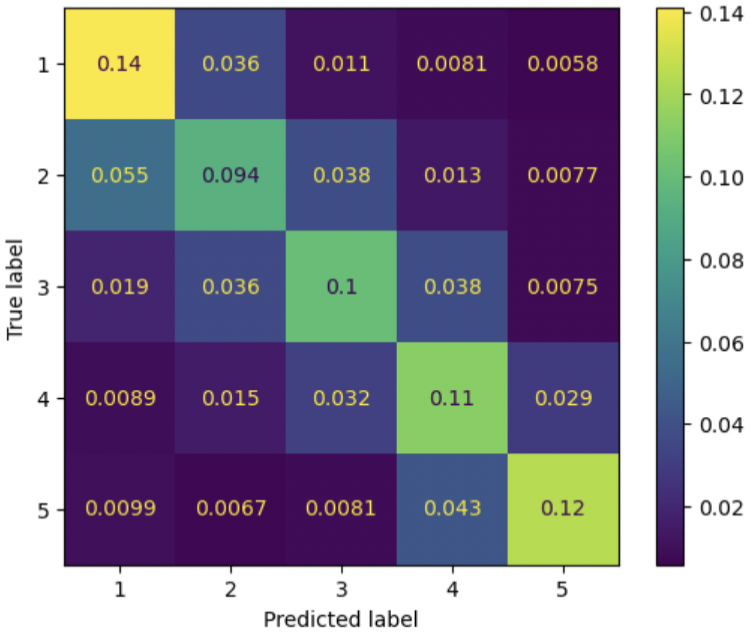


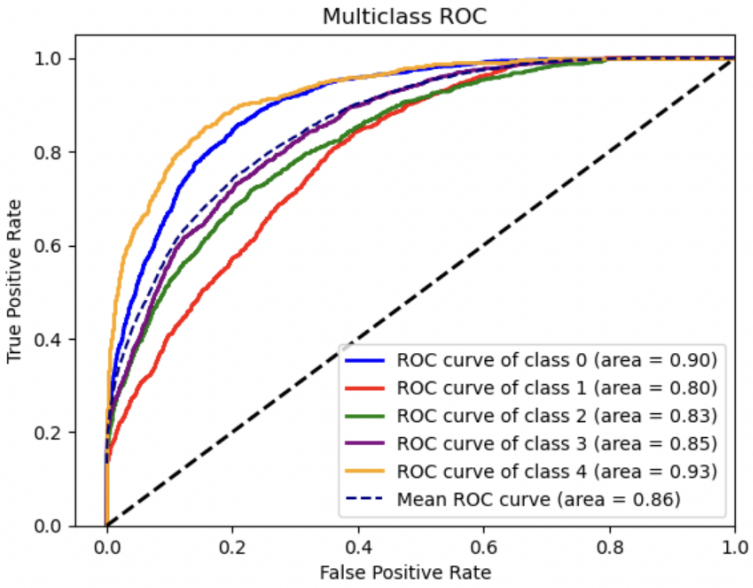
Multiclass classification: One vs. Rest

Logistic regression

Show 6 curves in one plot: 5 curves from each category and the average curve

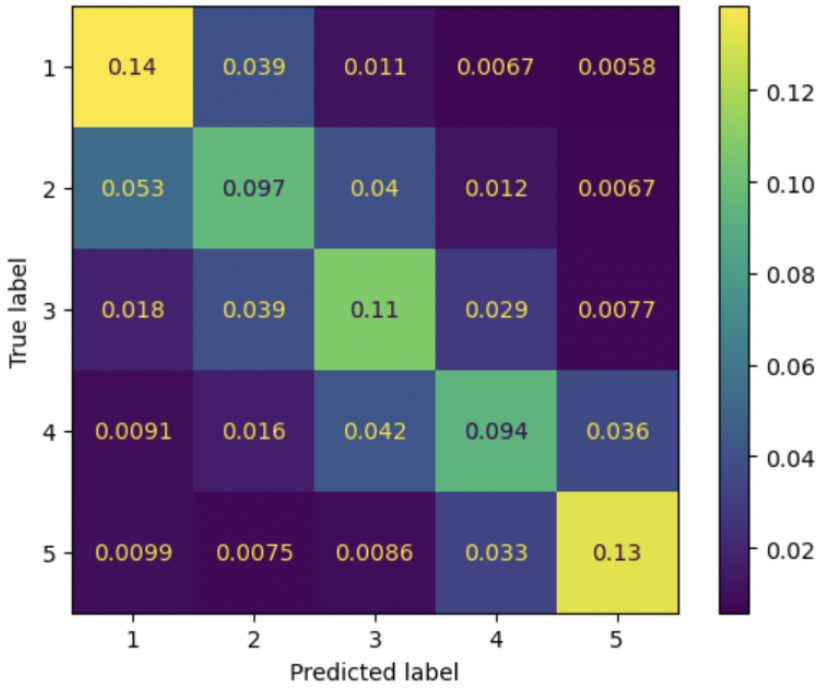
```
Accuracy:  0.5721137375813635
F1 Macro Score:  0.572307874885549
Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fef337942e0>
AUC:  0.7332648387668288
```

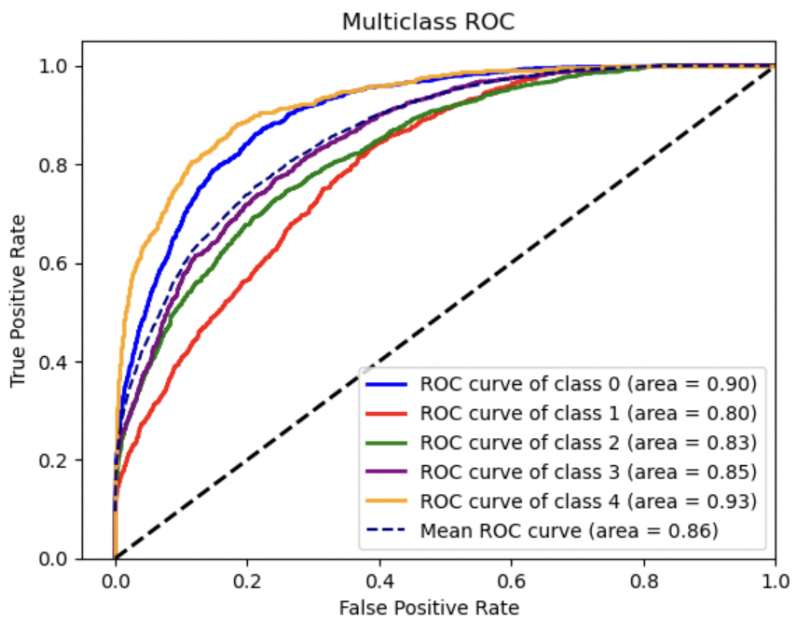




SVC regression

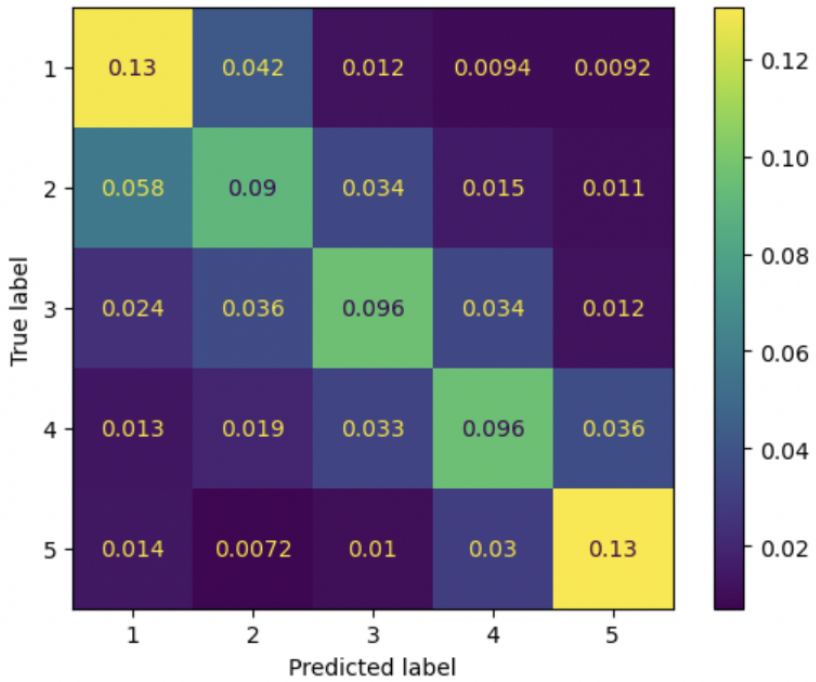
```
Accuracy:  0.5695443645083933
F1 Macro Score:  0.5693046892904505
Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7f97aa376580>
AUC:  0.7316108875957799
```

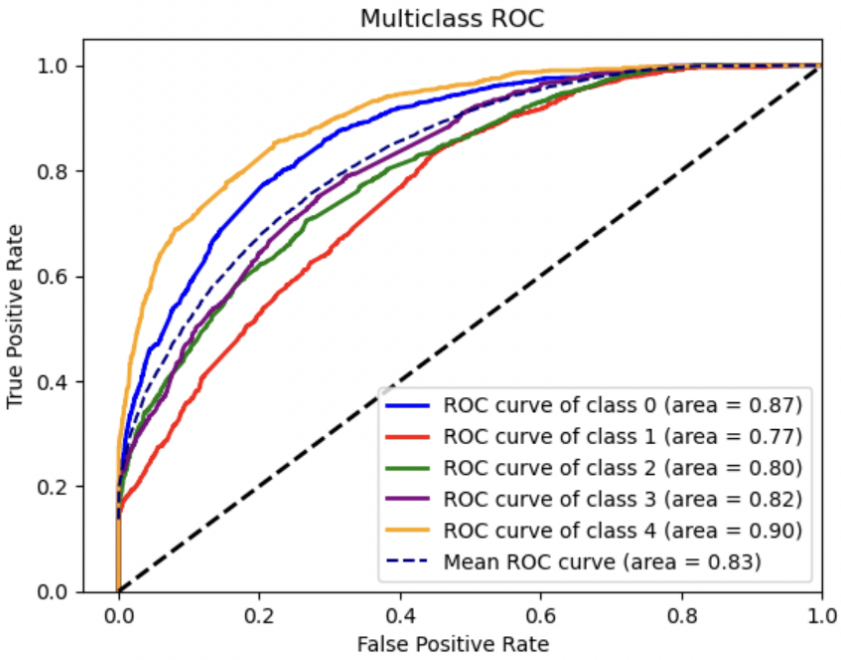




Ada regression

```
Accuracy:  0.5416238437821171
F1 Macro Score:  0.5406889073348216
Confusion Matrix
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay object at 0x7fa911ac48b0>
AUC:  0.71435979080731
```





OneVsRestClassifier() with LogisticRegression(random_state=42, class_weight='balanced', max_iter=500, C=1) works best