# Accuracy:

## Pre-processing

All attributes were standardized using the StandardScaler from sklearn. Hyper-parameters were tuned using a grid search. Data was split 60/40 (training / test).

## Models Evaluation

evaluating RandomForest

Accuracy: 0.95163, AUC: 0.98912

[[1103 32]

[ 57 648]]

FScore: 0.93574, Precision: 0.95294, Recall: 0.91915

classification report

precision recall f1-score support

0.0 0.95 0.97 0.96 1135

1.0 0.95 0.92 0.94 705

avg / total 0.95 0.95 0.95 1840

evaluating SVC

Accuracy: 0.80489, AUC: 0.89423

[[918 211]

[148 563]]

FScore: 0.75825, Precision: 0.72739, Recall: 0.79184

classification report

precision recall f1-score support

0.0 0.86 0.81 0.84 1129

1.0 0.73 0.79 0.76 711

avg / total 0.81 0.80 0.81 1840

evaluating DecisionTree

Accuracy: 0.91902, AUC: 0.91930

[[1046 88]

[ 61 645]]

FScore: 0.89646, Precision: 0.87995, Recall: 0.91360

classification report

precision recall f1-score support

0.0 0.94 0.92 0.93 1134

1.0 0.88 0.91 0.90 706

avg / total 0.92 0.92 0.92 1840

evaluating AdaBoost

Accuracy: 0.93098, AUC: 0.97955

[[1066 42]

[ 85 647]]

FScore: 0.91063, Precision: 0.93904, Recall: 0.88388

classification report

precision recall f1-score support

0.0 0.93 0.96 0.94 1108

1.0 0.94 0.88 0.91 732

avg / total 0.93 0.93 0.93 1840

evaluating KNeighbors

Accuracy: 0.79457, AUC: 0.86209

[[936 172]

[206 526]]

FScore: 0.73566, Precision: 0.75358, Recall: 0.71858

classification report

precision recall f1-score support

0.0 0.82 0.84 0.83 1108

1.0 0.75 0.72 0.74 732

avg / total 0.79 0.79 0.79 1840

## Best Model Performance:

RandomForestClassifier

Cost-Aware:

## Set-up

All attributes were standardized using the StandardScaler from sklearn. Hyper-parameters were tuned using a grid search. FScore beta set to 10 to set cost of classifying real emails as spam 10 time expensively than spam

## Models Evaluatiuon

evaluating RandomForest

Accuracy: 0.95163, AUC: 0.98448

[[1078 32]

[ 57 673]]

FScore: 0.92223, Precision: 0.95461, Recall: 0.92192

classification report

precision recall f1-score support

0.0 0.95 0.97 0.96 1110

1.0 0.95 0.92 0.94 730

avg / total 0.95 0.95 0.95 1840

evaluating SVC

Accuracy: 0.82880, AUC: 0.90823

[[937 173]

[142 588]]

FScore: 0.80514, Precision: 0.77267, Recall: 0.80548

classification report

precision recall f1-score support

0.0 0.87 0.84 0.86 1110

1.0 0.77 0.81 0.79 730

avg / total 0.83 0.83 0.83 1840

evaluating DecisionTree

Accuracy: 0.90109, AUC: 0.89939

[[1015 104]

[ 78 643]]

FScore: 0.89150, Precision: 0.86078, Recall: 0.89182

classification report

precision recall f1-score support

0.0 0.93 0.91 0.92 1119

1.0 0.86 0.89 0.88 721

avg / total 0.90 0.90 0.90 1840

evaluating AdaBoost

Accuracy: 0.93261, AUC: 0.97835

[[1052 70]

[ 54 664]]

FScore: 0.92459, Precision: 0.90463, Recall: 0.92479

classification report

precision recall f1-score support

0.0 0.95 0.94 0.94 1122

1.0 0.90 0.92 0.91 718

avg / total 0.93 0.93 0.93 1840

evaluating Multi-NB

Accuracy: 0.75109, AUC: 0.80150

[[920 190]

[268 462]]

FScore: 0.63355, Precision: 0.70859, Recall: 0.63288

classification report

precision recall f1-score support

0.0 0.77 0.83 0.80 1110

1.0 0.71 0.63 0.67 730

avg / total 0.75 0.75 0.75 1840

evaluating KNeighbors

Accuracy: 0.77554, AUC: 0.84720

[[910 196]

[217 517]]

FScore: 0.70456, Precision: 0.72511, Recall: 0.70436

classification report

precision recall f1-score support

0.0 0.81 0.82 0.82 1106

1.0 0.73 0.70 0.71 734

avg / total 0.77 0.78 0.77 1840

## Best Model Performance:

RandomForestClassifier

# Discuss the best models in two different tasks (as well as their performance) in detail, provide some comparisons. Draw some conclusions from the assignment.

In my experiments the RandomForestClassifier performed the best. It had the highest accuracy and the lowest total cost when weighted 10-1 for miss classified real emails (as spam) to spam emails classified as real.

Conclusions:

It is important when evaluating models to think not only about the model’s overall accuracy, but the business case it is trying to solve. For spam email classifications, it would be very bad to not deliver someone their real email (classifying it as spam). As such, we need to take that into account when building and selecting our models.

Work: <https://github.com/randorfer/spamClassifier>

* All ROC curves can be found in the ROC folder