# NACME Project 1

Compare the performance of several classifiers using both accuracy and F1-Score on the “spam” dataset. All results shown in this assignment are based on a 70/30 train/test split with a random\_state of 42.

## Data

The “spam.csv” datafile contains a collection of emails that have been classified as “spam” or “ham” (not spam). Load the “spam.csv” datafile from canvas. Displaying the top 5 rows (via the head method of the pandas library) should produce the following.

A picture containing table

Description automatically generated

The label (target) associated with each message is in the first column and the actually email message is located in the second column (the other columns are not used). Separate the data into a training and test set, where training is composed of 70% and testing is 30% of the data. You can use the train\_test\_split function in scikit-learn.

### **Step 1: SVM Classifier**

Train and test a Support Vector Machine **linear** classifier to operate on the spam dataset. Display both the accuracy and F1-Score. Explain the discrepancy between the accuracy and the F1-Score (why is the accuracy high but not the F1-Score)

Accuracy: 0.965311004784689

F1-Score: 0.8527918781725887

### Train and test a Support Vector machine classifier with a RBF kernel. Sweep through gamma values from 1 to 10 to optimize the hyperparameter. Display the accuracy and F1-score associated with the best performing gamma.

Accuracy: 0.9192583732057417

Best F1-Score is 0.8571428571428572 at gamma 1

### **Step 2: Logistic Regression Classifier**

Train and test a logistic regression classifier on the spam dataset. Display both accuracy and F1-Score.

Accuracy: 0.9401913875598086

F1-Score: 0.7076023391812866

### **Step 3: Decision Tree Classifier**

Train and test and decision tree classifier on the spam dataset. Display both accuracy and F1-Score

Accuracy: 0.9497607655502392

F1-Score: 0.8099547511312217

### Train and test a random forest classifier on the spam dataset. Sweep through the number of estimators parameter to identify the most appropriate number of estimators to use to maximize F1-Score while limiting execution time. Display a graph of F1 performance and the associated F1-score at the most appropriate number of estimators. *Note: your results may vary but should be consistent with what’s shown.*

Chart, line chart

Description automatically generated

Best F1-Score 0.881012658227848 occurred with 410 trees

### Step 4: Show Decision Path Through Tree

Show the decision path the classifier takes through the trained decision tree. Use the “decision\_path” method associated with the Decision Tree object. You can refer to the Scikit learn documentation under the “Decision Path” section of the page: <https://scikit-learn.org/stable/auto_examples/tree/plot_unveil_tree_structure.html>

Select at least three observations from the test dataset where the target label is ‘spam’ and show the decision path. Make sure to confirm that the decision tree prediction matches the target label.

Observation 1:

node#0 trust > 0.11061882600188255 [0.20896969]

node#228 trust <= 0.320525199174881 [0.20896969]

node#229 speechless <= 0.10843262076377869 [0.]

node#230 usb <= 0.2118046134710312 [0.]

node#231 murder <= 0.09743401408195496 [0.]

node#232 downloaded <= 0.2603650391101837 [0.]

node#233 telling <= 0.17131425440311432 [0.]

node#234 sugar <= 0.22542085498571396 [0.]

node#235 platt <= 0.20889881253242493 [0.]

node#236 frndshp <= 0.09118067473173141 [0.]

node#237 quick <= 0.26631879806518555 [0.]

node#238 due <= 0.13619211316108704 [0.]

node#239 nookii <= 0.07844974845647812 [0.]

node#240 orange <= 0.3800553232431412 [0.]

node#241 na <= 0.14464452862739563 [0.]

node#242 secure <= 0.26715490221977234 [0.]

node#243 shijas <= 0.13335944712162018 [0.]

node#244 shhhhh <= 0.15749488770961761 [0.]

node#245 nething <= 0.1199624240398407 [0.]

node#246 traffic <= 0.19488725066184998 [0.]

node#247 cantdo <= 0.2886410057544708 [0.]

node#248 yan <= 0.16461408138275146 [0.]

node#249 erutupalam <= 0.20466133952140808 [0.]

node#250 watever <= 0.20079296827316284 [0.]

node#251 min <= 0.4731675088405609 [0.]

node#252 hidden <= 0.22162039577960968 [0.]

node#253 avoiding <= 0.2746322453022003 [0.]

node#254 gave <= 0.1607898324728012 [0.]

node#255 suite <= 0.4078381061553955 [0.]

node#256 mtnl <= 0.49081951379776 [0.]

Observation is spam and predicted spam

Observation 2:

node#0 trust <= 0.11061882600188255 [0.]

node#1 surrender > 0.07418157905340195 [0.2013789]

node#211 cstore <= 0.13116006553173065 [0.]

node#212 clock <= 0.1528775542974472 [0.]

node#213 surrender <= 0.4033806025981903 [0.2013789]

node#214 rply <= 0.24827760457992554 [0.]

node#215 yep <= 0.12227099388837814 [0.]

node#216 dental <= 0.1793350726366043 [0.]

Observation is spam and predicted spam

Observation 3:

node#0 trust <= 0.11061882600188255 [0.]

node#1 surrender <= 0.07418157905340195 [0.]

node#2 pear <= 0.16091206669807434 [0.]

node#3 attended > 0.1476808860898018 [0.28077471]

node#157 hint > 0.07069701701402664 [0.27407409]

node#187 gene <= 0.13633722066879272 [0.]

Observation is spam and predicted spam

Based on your observations, what general statements can you make about how the decision tree classifier works.