Joe Klein

SI 671 HW

Kaggle Challenge Report

My first attempt with a simply svm.SVC setting the kernel parameter to linear and C to 1 (I'm still a little lost as to what this controls) I got .99702. I was surprised that the score was this high because it was just a linear classifier that didn't really consider the various hierarchy of labels. My second submission I used just the plain SVC() with the default settings to see how it compared with the linear kernel- which it received an identical score. I realized then that somehow I had instantiated one of my variables to SVC() instead of the intended linear, C=1 for the first submission so my third submission I made sure to set my classifier to the intended kernel SVC classifier and I received a .99745 (my best score). My fourth attempt I used a RandomForestClassifier with a verbose limit set to 10, estimators at 10, the same amount of n_jobs as the core and a maximum of 20 features- this score in training gave me an extremely high score (.99985) but in the submission got me a lousy .77423. I think I must be processing the training data for this classifier in a non ideal manner. A version of svm_clf=RandomForestClassifier(n_estimators=5, n_jobs=-1) got me .94031. Changing this estimator to a max features of 10 got me 0.967278. Using the max_features of log2 got me a whopping 0.59773! Square root scored a .79832.

Before I would submit anything I used a cross validation process where I split the test into halves and trained a small group of classifiers and compared their results. Some of the classifiers didn't give me high enough scores for me to make a submission including OnevsOneClassifier, LogisticRegression, and a group of SDGClassifiers which gave decent results but not better than the SVC's. I also tried a bunch of times to run a Kneighbors classifier but for some reason I think my preprocessing did something where it just hung up- one point running for 5 hours without a solution. I ended up using a black box encoding for my preprocessing where all of the labels where encoded with numeric values at the same level but I also tried to separate the labels by arrays using pandas where I drop the classes that are normal and try to group classes of intrusions into category families- but I was never able to actually to get a good test.