Project for Naireen Hussain: Classification of Transit signals

The goal of this homework is to build up a frame work of running classificuation algorithms on stellar light curves, aiming to identify those stars with transit singals. In this week, you should have a script that be able to read in the light curve data, apply a few of the standard classification algorithms, conduct cross validation, and assess the result using the confusion matrix and ROC curves.

Software requirements:

scikit-learn package

For documentation of the package, go to online documentations.

Classification algorithms requirement:

Logistic regression

Support Vecotor Machines (SVC)

Random Forest

Gradient Boosting Classifier

Final Goal of the homework:

You are expected to run all the above classifers on the simulated datasets with the balanced components, using the light curve representation of the data. You are expected to present the results from the classification in a few different formats, such as confusion matrix, ROC curves, and cross validation scores. You are expected to try to discuss what you learned from the results. *Fine tuning of the individual algorithms are encouraged, but not required.*

Break down of the tasks:

(1) Load and Visuallize the data

The time series will be provided to you in the shared git repository.

The first step is to examine the data. Use matplotlib to make figures of the time series of the transit, each of the false positive catagories (8 figures in total, you are encouraged to use subplot in matplotlib, or overplot some of the lines, to reduce the total number of figures), at the minimum and maximum noise level. Describe what you observe.

(2) build a minimum working script for one classfier

(2a) Assemble a balanced dataset.

Assemble the dataset to classify with all 1000 transit light curves we have, draw randomly 330 examples from each of the false positive catagories. Shuffle the total 1990 light curves.

(2b) Classify

Use Logestic regression algorithm to classify, assuming only two classes, transit, and false positives, and report the confusion matrix and ROC curve.

(3) add in principle component decomposition (PCA) and cross validation

(3a) Run PCA.

Use PCA to reduce the number of features (for the time series) to 20 components.

(3b) Crossvalidation with Support vector machines.

Use SVM and 5 fold cross validatation using f1 as the CV score method.
The syntax will be:
clf = svm.SVC()
scores = cross_validation.cross_val_score(clf,X,Y,cv=5,scoring='f1')
Only the crossvalidation score is required to be reported.

(4) learn to use the other algorithms

(4a) Repeat (3) with all the other algorithms, and compare the cross validation result from their default settings.

(4b) For the best algorithm from your choice, report the confusion matrix, and ROC curve.