**World Quant University**

**Professor: Tiberiu Stoica**

**Alpha Design**

Erich Ratzat - [eratzat@hotmail.com](mailto:eratzat@hotmail.com)

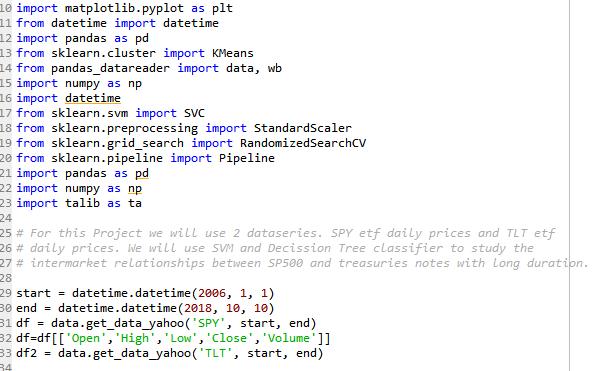
Nikolas Lippmann Pareschi - [nikolaslippmann@gmail.com](mailto:nikolaslippmann@gmail.com)

**Mini Project 2**

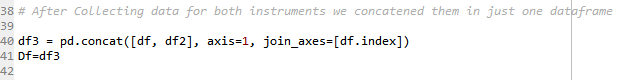
### Overview

In this project we will use intermarket analysis and 2 machine learning algorithms: Support Vector Machine and Tree Classifiers to predict if we should be buying the SP500 (Spy etf) in the next day of if we should be flat or selling it. The 2 markets that we use the data for our analysis are the SP500 (SPY) and Treasuries Bunds with long duration (TLT etf). We have separated the 2 algorithms in 2 python files.

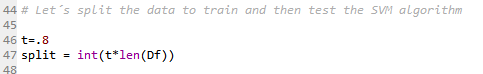
The first step is to download the data from SPY and TLT to 2 dataframes.



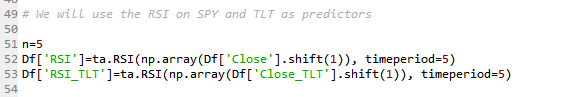
After downloading and renaming the columns so they do not have the same name we concatenate them into one single data frame:



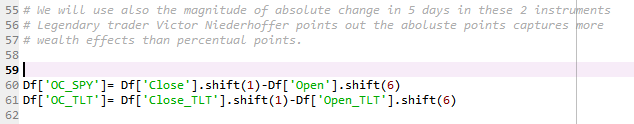
We then split the data, using 80% of our dataset to train the algorithm and 20% as an out sample.



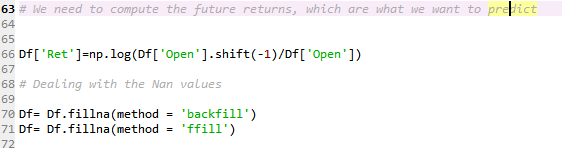
Let´s use RSI for overextended areas. One should have installed Talib, for reference check the readme file:



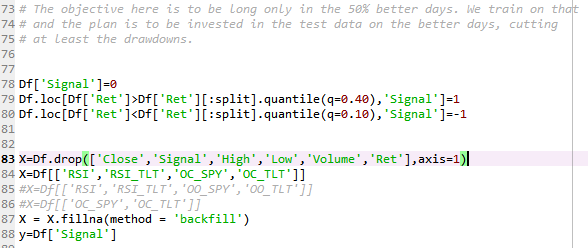
We will use also the magnitude of absolute change in 5 days in these 2 instruments. Legendary trader Victor Niederhoffer points out the abolute points captures morewealth effects than percentual points. Salaries are more or less stables, market returns are subject to bubble dynamics all the time.



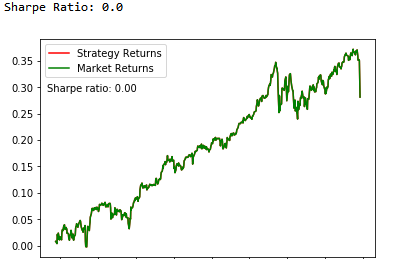
We need to compute futures returns, which is what we want to predict in the test data using the trained algorithm. Also we need to deal with Nan values:



We now generate the signal data in which we will train the algorithm. We used some different values.



Some interesting results. If we try to eliminate only the 10% terrible days SVM algorithm believes it is optimal to be always bought:



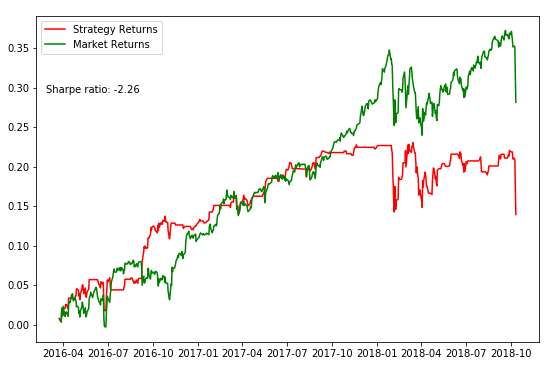
Here we are bought on 50% quintile, sold in the 10% down data and flat otherwise. Again the SVM maximized the prediction being bought always:



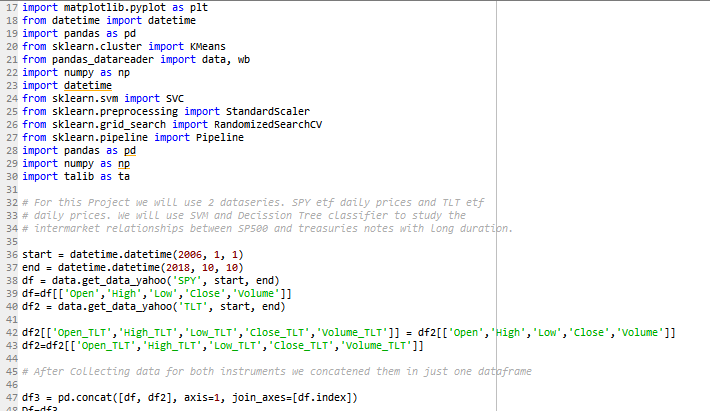
We now tweaked a bit the train quintiles. Now we got something new, but it is suboptimal to buy and hold:



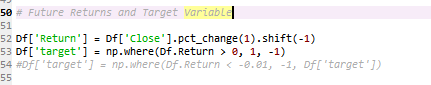
Results:



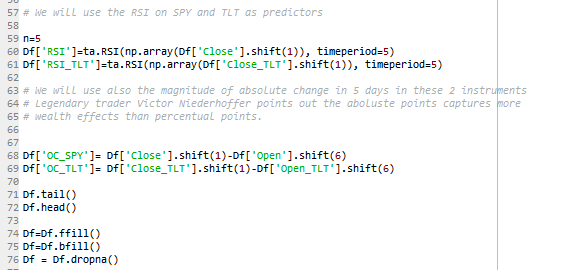
Now we will analyze using a decision tree classifier. We only want to predict if the next day will be an up or a down day. The beginning of the code is similar to the SVC algorithm:



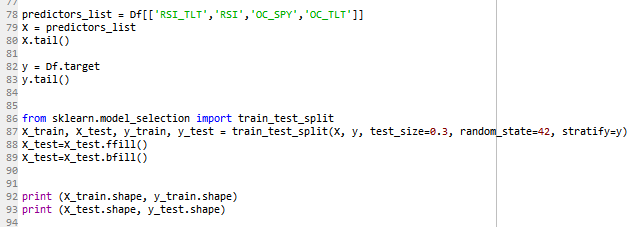
We have defined the future returns and the target classification (up day x down day) in the beginning:



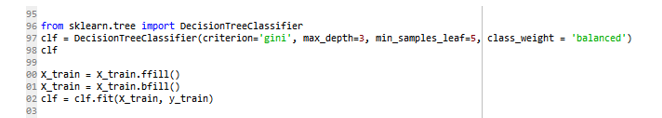
The classifiers and handling with Nans:



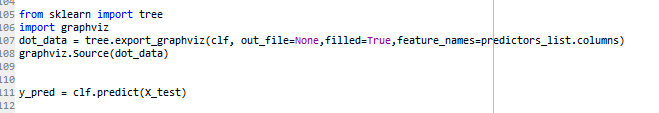
Predictors list and organizing the sets of data:



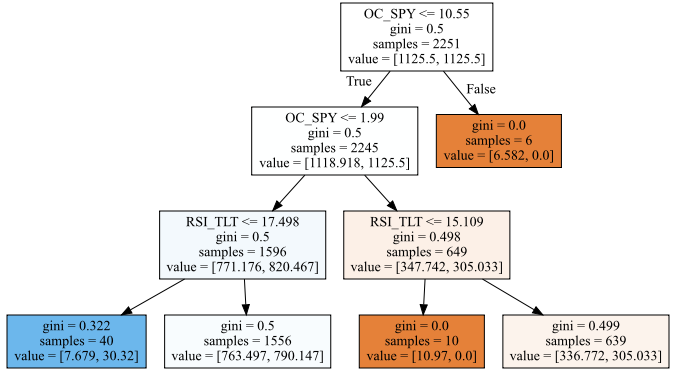
Training the model and using it in the out sample (30% of data).



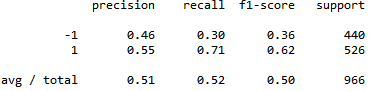
Printing the tree and verifying the statistics of the predictions:



These are the classifications that we made. This is a tree which uses 4 classifiers, Open to Close considering 5 days on SPY and TLT and RSI(5) on SPY and TLT:

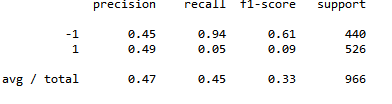


This tree is good for predicting up days, but not down days. Any F1-Score above 0.50 is a good score.

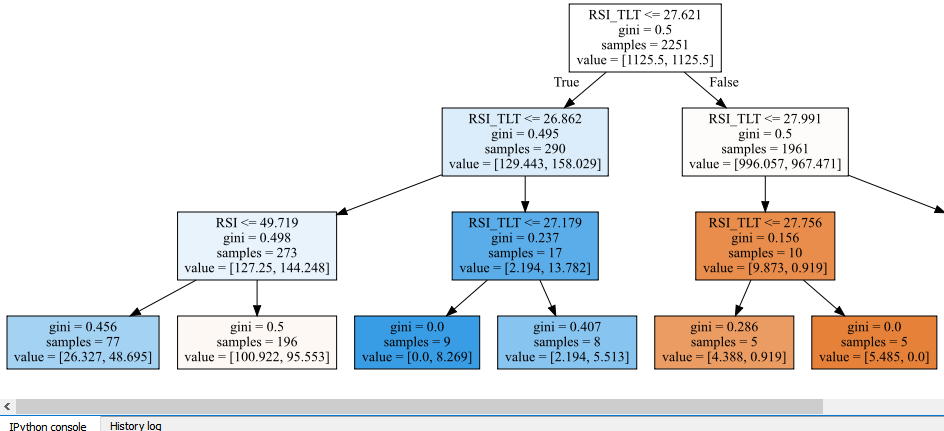
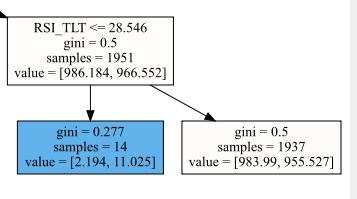


We also test Victor´s hypothesis, that absolute values usually have more predictive power than percentage values. So we compared the RSI(5) predictors for SPY and TLT to the OPEN to CLOSE variation considering 5 days.

Using only RSI(5) SPY and RSI(5)TLT for prediction of the day.



It seems that RSI has some power in predicting the negative days.

The Absolute Values have basically the same predictive power but for the positive days, it is inconclusive:

