Machine Learning Engineer Nanodegree
Capstone Proposal
Jack St. Clair
January 18, 2018

Proposal

Domain Background

The domain in which my proposed project is centered around is the macroeconomy. The inspiration for this comes from two different channels: first, my undergraduate studies were partly in economics and second, I just finished reading "Macroeconomics" by Charles I. Jones, 2nd edition, as part of my preparation for attending the Berkeley MFE. More specifically, I would like to explore how the short run model used to explain the economy could be used to classify positive returning months in the S&P 500 Index.

According to Jones, the short run model consists of actual inflation, expected inflation, the fed funds rate, the strength of the US dollar, among others.[1] While Jones creates a working model using these inputs (among others) to understand the state of the economy, my goal is to use machine learning techniques to generate predictions of the US stock market (using the S&P 500 as the proxy) from the macroeconomic variables stated above. To be precise, my vision is to use machine learning to correctly classify whether or not the S&P 500 will have a positive returning month.

Problem Statement

The problem is classifying winning months and losing months in the S&P 500 by using different features of the macroeconomic environment that are available at the beginning of the month using machine learning techniques such as SVMs, Decision Trees, or Neural Networks, to name a few. This is a binary classification problem. Using data going back more than ten years, I will use features such as expected inflation (average of prior months), actual inflation (most recent inflation number), the fed funds rate, and the strength of the US dollar, among others to try and classify whether the S&P 500 had a winning or losing month. I expect to use accuracy as one of the metrics.

Datasets & Inputs (FRED data are the .csv files on my Github repository)

Overview: the target variable will be whether or not the S&P 500 returned a positive amount or not while the features of each training example will be relevant values of the strength of the USD, the current inflation environment, the Fed Funds Rate, and the TED Spread. More specifically, the change in these features will be used. I expect to use monthly data going back as far as 1975 as well as normalizing the data. There is a total of 383 data points in the sample with 232 positive examples, ~60% of the dataset. Given only the slight imbalance, I think accuracy is still a good choice for metric.

S&P 500 Return (source, Yahoo! Finance): Monthly index levels are gathered through pandas datareader using Yahoo! Finance. From these monthly levels, I calculate the monthly percentage return using the standard percent change formula. Then, if the percent change is positive, it is classified as a '1', otherwise, the month is classified as a '0'. This data is necessary as it's what the machine learning

[1] Jones, Charles I. *Macroeconomics*. 2nd ed., W. W. Norton & Company, Inc., 2011.

algorithm will be attempting to classify. To be clear, I am using the after-cash percent return meaning I subtract out the effective monthly fed funds rate and TED spread at the time.

USD Strength Index (source, FRED website): Monthly levels are downloaded from the FRED online database and read into Python using pandas read_csv. The USD strength is motivated from the text of Jones who explains how the relative level of the USD can impact foreign investment and thus drive economic growth. Link: https://fred.stlouisfed.org/series/TWEXMMTH

Inflation (source, FRED website): Data is acquired similar to the USD Strength Index. This data will be used in a couple different fashions. First, inflation expectations will be calculated as an average of recent months. Second, the difference between actual inflation and expected inflation will be calculated and used as an input into the algorithm. How inflation comes in versus expectations is a factor into the performance of the S&P 500. Link: https://fred.stlouisfed.org/series/CPIAUCSL

Fed Funds Rate (source, FRED website): Acquired like the two features above, the Fed Funds Rate is a vital input in the macroeconomic model developed in the Jones' textbook. Link: https://fred.stlouisfed.org/series/FEDFUNDS

TED Spread (source, FRED website): Also gathered via FRED, the TED Spread will be used as a proxy for the risk premium in the current macroeconomic environment. Most famously during the Great Recession, risk premiums blew out causing the decline in value across almost all asset classes. Link: https://fred.stlouisfed.org/series/TEDRATE#0

Solution Statement

Given the goal of this project will be to correctly classify winning months from losing months in the S&P 500, several machine learning techniques could apply. The machine learning algorithm will output a probability that the S&P 500 will yield positive returns in the given month. For example, a decision tree classifier can learn a decision boundary that separates the data into categories of 'winner' and 'loser'. The model can be evaluated using accuracy as a metric.

Benchmark Model

I will compare the learned model to random chance that that always predicts a winner. This 'random choice' will have an accuracy of ~60%. I will use standard logistic regression as the benchmark model. With no feature normalization and only using the pure Fed Funds rate, TED spread, USD strength index, and CPI, this baseline logistic regression model achieved an accuracy of ~64%. I will work to improve this number throughout the project.

Evaluation Metrics

As far as evaluation of the model goes, accuracy will be the first metric. However, a natural metric that comes to mind is also to consider the outcome of following the predictions in the market and to track the profit & loss from this. In other words, all mistakes are not equal. Predicting a winning month when it actually loses 1 basis point is a lot different than predicting a winner when it actually loses 1,000 basis points. This project is likely a good application for using the Sharpe Ratio (ratio of the excess return to the risk) and I will likely use it as an evaluation metric.

Project Design

Given I am starting out with a framework laid out in Jones' 'Macroeconomics', I do not anticipate my project to initially require much along the lines of feature selection. However, I do anticipate having to think through and test the best way to use the available data. I know the goal is to correctly classify winning months from losing months in the S&P 500 and so the data I feed into the machine learning algorithm will have to have real meaning to explaining the return of the index. For example, simply the change in inflation from the prior month may not be meaningful compared to how inflation came in relative to expectations (measured as a recent average of prior months). These subtle changes in the expression of the data may make big differences in the performance of the algorithm and I intend on testing many of these out in different machine learning algorithms. I will use data up until 2014 to train on and use data from 2015-2017 to test on.

I expect to begin by processing the data so that it is compatible with sklearn's different algorithms (i.e., feature normalization). Specifically, I will try using logistic regression, decision tree classifiers, SVMs, and KNN, among others. Once getting code up and running, I expect to compare across these different techniques and see how they do in terms of accuracy and the Sharpe ratio that follows from trading off the model.

As stated at the top of this section, I am motivated by Jones' 'Macroeconomics' in choosing the features I have described above. There is a chance that these features do not lead to a successful implementation and thus I will search for other features that could improve the performance of the algorithm including lagged values of the S&P 500 or other macroeconomic indicators.