Machine Learning Engineer Nanodegree

Capstone Proposal

Joe Importico

January 28, 2019

Proposal

Domain Background

Forecasting asset prices is a practice that has been adopted by many and in recent years has been modernized by a group of investors known as Quants. Many investors of this ilk create complex models with a core focus of understanding the behavior of an assets' price movement and how those prices will evolve over time. For any researcher hoping to achieve success in that endeavor, they are first faced with determining what data should be used for their research, as well as how they will go about acquiring it. Financial data is a scarce asset for many, especially for those not working in academia or at a large institutional asset manager, and it shines a light on the disadvantage retail traders face when beginning to tackle an already difficult problem in forecasting asset prices.

Luckily, there are a few platforms and publicly available data sources that have been made available to the broader public in an effort to distribute the opportunity to take part in these types of research studies. For my capstone project, I will be using a number of these sources, such as Yahoo Finance to query historical asset prices and Quantopian for back-testing my signal in a simulated trading environment.

Problem Statement

In this project I will be constructing a classification model that uses several price-based attributes to predict returns for a cross-sectional universe of assets. There are a few, but important, considerations I want to point out with respect to the adjustments I will be making throughout this analysis. The first is with respect to stationarity. Many statistical techniques have a deep-rooted assumption that the data being used is stationary, which, unfortunately, financial data is not. It is for that reason that I will be switching my target variable from closing price to an n-day return horizon. This adjustment should provide the benefit of working with a more stationary time-series; however, returns can, especially in more volatile markets, have a fair amount of noise associated with them, so once my target has been transformed into a return it will then be binarized relative to the median n-day return horizon of the universe. Assets with a return horizon greater than the median return will receive a value of one, otherwise zero. I will address the other adjustments, such as the temporal dependencies that need to be accounted for, being applied later in this proposal.

Datasets and Inputs

The majority of the data being used for this project was obtained using the pandas-datareader[1]. Using that public API, I queried a series of open, high, low, close, and volume (OHLCV) information for a large number of publicly traded securities. In addition to the OHLCV data pulled for the assets in my study, I am also incorporating information from the Fama/French Data Library[2], as well as historical OHLCV data for a variety of sector ETF's. Both the Fama & French and sector ETF data will be used to construct time-series exposures to common fundamental variables (size, value, and momentum) and sector classifications. This is a common approach used to approximate exposures when the underlying information is unavailable, and a great method for us to incorporate financial variables into our research that are too expensive or difficult to otherwise acquire. To estimate these exposures, we will use OLS regression and a fixed rolling window of asset returns.

The target variable will be an n-day forward return horizon. Given the nature of the feature set (OHLCV data) I'll be working with, I'd like to evaluate the efficacy of my features on a few return horizons, some near-term and some farther out. The forward return horizon will be shifted back to align with the n-day horizon that has been selected and will be binarized relative to the median asset's return for the period. Asset's with a value greater than the median value will receive a one else a zero.

The feature space will consist of a variety of technical indicators and time-series exposures estimated via OLS regression. Some of the technical indicators will be derived using the TA-Lib[3] and others will be derived using other libraries from the Python ecosystem, such as Pandas[4] and Numpy[5]. I've included a series of links to more thorough descriptions of each feature listed below.

Features:

- Relative Strength Index (RSI)
- Moving Average Signal Convergence Divergence (MACD)
- Long-Term Momentum: Price change over several horizons
- Short-Term Reversal: Price change over several near-term horizons
- Percent above and below the 52-week high and low price
- Percent above and below the 30 and 200 days moving average price
- Volatility: Standard deviation of returns over several horizons
- Money Flow Index
- Directional Movement Index (DX)
- Stochastic Oscillator
- Average True Range (ATR)
- Time-Series Exposures:
    - Sectors:
        - Beta to Communication Services (XLC)
        - Beta to Consumer Discretionary (XLY)
        - Beta to Consumer Staples (XLP)
        - Beta to Energy (XLE)
        - Beta to Financials (XLF)
        - Beta to Health Care (XLV)
        - Beta to Industrials (XLI)

        - Beta to Materials (XLB)
        - Beta to Technology (XLK)
        - Beta to Utilities (XLU)
    - Styles:
        - Beta to Fama & French SMB (Small Minus Big) portfolios
        - Beta to Fama & French HML (High Minus Low) portfolios

Solution Statement

Based on the structure of this problem, with a binarized target variable, we will use a series of classification algorithms to evaluate our ability to reliably predict our target. Our candidate models will consist of the following methods:

- Logistic Regression
- Random Forest
- AdaBoost
- Feed Forward Neural Network

Benchmark model

Since each stock has an equal probability of earning a return that is greater or less than the median value for the period, the appropriate naïve model for this analysis will be to predict that 50% of the companies will fall into each category.

Evaluation Metrics

A common method for evaluating a set of candidate models is to measure the accuracy of the model. This is an effective metric for many research projects, however it is not well suited to evaluate the promise of our models given how our target variable has been structured. With 50% of the observations falling into each class, we'll want to pay closer attention to the number of accurate predictions we make, and for that reason we'll be using the F1 score which provides a balance of evaluating both precision and recall. We are concerned with both components of the F1 score, so we'll leave our value of beta equal to .5.

$$\text{precision} = \frac{tp}{tp + fp},$$

$$\text{recall} = \frac{tp}{tp + fn},$$

$$F_\beta = (1 + \beta^2)\frac{\text{precision} \times \text{recall}}{\beta^2 \text{precision} + \text{recall}}.$$

Project Design

1) Data Collection & Aggregation

The first step in preparing the data described in the earlier section of this proposal has to deal with data aggregation. I will leverage several resources to collect, clean, and align the raw data for the correct time periods.

2) Feature Engineering & Exploratory Data Analysis (EDA)

Once the requisite data has been collected and loaded into our environment, it will need to be transformed into the variables described in the Datasets & Inputs section before being used as inputs into our candidate models. The feature engineering process is especially important here as we will be working with many rolling window transformations, so it is imperative that we accurately track the information and properly surface it so it is reflected on an 'as was' basis when the information would have been known by the public. Ultimately, we will be taking the OHLCV data and creating a series of technical indicators and time-series exposures to exogenous attributes. In addition to engineering the feature space, we will also be creating our target variable and shifting the returns appropriately.

Following the feature engineering step, we will proceed to our EDA where we will track the statistical properties of our data. Some of the areas we will explore include:

- Generating descriptive statistics
- Checking the availability of our features
- Visualizing the distributions of our features & interrelationships with the other features

3) Model Construction, Evaluation, and Tuning

In this segment of the project, we will be cross-validating our data while preserving the temporal nature of the dataset using TimSeriesSplit[7] from scikit-learn. After the data has been properly divided into training and test sets, we will begin to train our classifiers and selecting the best models based on their F1 scores. Given how dense the hyperparameter search space is, we'll be using RandomizedSearchCV[8] over GridSearchCV. An excellent description for why we are favoring this approach can be found here[9].

4) Create Predictions & Implement the Model

In the final segment of this project, a final model will be selected and used to make predictions out of sample (OOS). Those OOS predictions will then be used in a simulated trading environment to test the strength of those signals.

References

1 https://pandas-datareader.readthedocs.io/en/latest/
2 Fama/French Data Library
3 TA-Lib
4 https://pandas.pydata.org/
5 http://www.numpy.org/
6 https://scikit-learn.org/stable/modules/model_evaluation.html#precision-recall-f-measure-metrics
7 https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html
8 https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
9 http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf