Machine Learning for Portfolio Selection

Introduction:

There are a lot of papers have researched on stock technical analysis by machine learning, and this project's idea comes from them but will have some improvements. In stock markets, there are two common but interesting phenomena: momentum and reversal effects. When the market is under the momentum effect, the past winners will probably continue to outperform the benchmark in the future, and it's opposite for reversal effect. Based on this fact, this project will use several machine learning models to predict these two effects and build a corresponding trading strategy.

Detailed Methods:

1. Description:

This project will research on US market and choose S&P 500 components as the trading universe, and use the history data from 2005 to present. To study the momentum and reversal effects, here we define **observation and holding periods**, which can be in hours, days or any time horizons depending on the trading preference. Because different observation and holding periods will have significantly different impacts for the results, we will back test several pairs of the periods. The stocks in the universe will be ordered by their returns in the observation periods, and the top N stocks will be defined as past winners, and the bottom N stocks will be past losers.

2. Momentum and reversal effect detection:

As we will use supervised machine learning to do prediction, we should pre-defined the market states (bullish and bearish) for S&P500. But there is no generally accepted formal definition of bull and bear markets in the finance literature, after researching on several popular algorithms, this project chooses the Lunde and Timmermann (2004):

In general, this algorithm imposes a minimum on the price change since the last peak or trough. Let λ_{bull} be a scalar defining the threshold of the movement in stock prices that triggers a switch from a bear state to a bull state and similarly, let λ_{bear} be the threshold for shifts from a bull state to a bear state. The algorithm first finds the maximum close price on the time interval [t_0, t], then computes the (inverse of the) relative change:

$$\delta = \frac{P_{max} - P_t}{P_{max}}$$

If $\delta > \lambda_{bear}$, then a new peak is detected at time t_{peak} at which close price attains a maximum on [t_0, t]. The period [$t_0 + 1$, t_{peak}] is labelled as a bull state. A bear state begins from t_{peak}+ 1, which is the new t_0.

Similarly, the logic is used for the trough detection. If $\delta = \frac{P_t - P_{min}}{P_{max}} > \lambda_{bull}$, then a new trough is detected at time t_{trough} at which close price attains a minimum on [t_0, t].

After applying this algorithm, we get the following results as figure 1 shows for bull and bear market states.

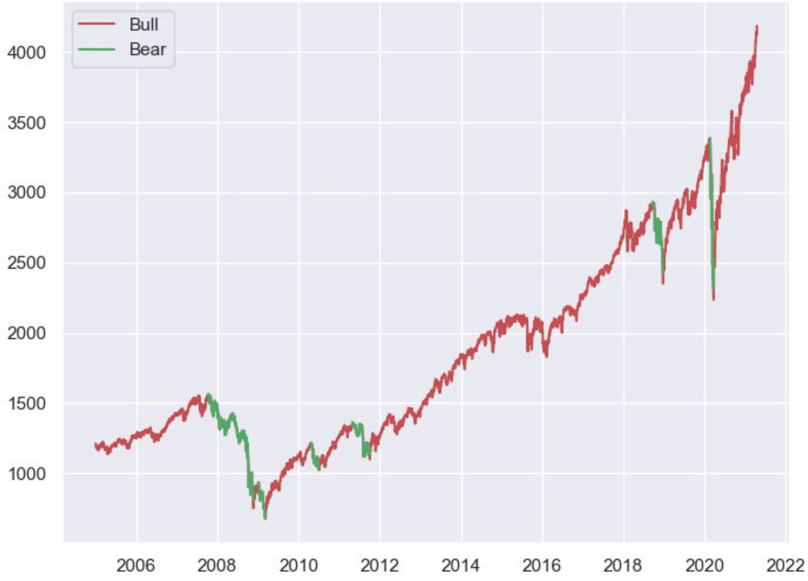

Figure 1. Bull and Bear Market

3. Model Selection:

This momentum and reversal effect detection can be seen as a binary classification model, so we will choose several machine learning models from simple to complex as follows: Naïve Bayes Model, Support Vector Machine, Decision Tree Model, Random Forest, Light Gradient Boosting Machine, and LSTM. This section will be enriched in the future.

4. Feature Engineering

Features play a pretty important role in almost all machine learning problems. Meaningful features will increase the model's performance significantly, while useless features will ruin the model, which is "garbage in, garbage out". Other than the regular market quotes (e.g. prices (open, high, low, close), volume, etc.), we will calculate several kinds of technical indicators including volume indicators (e.g. Money Flow Index, On-Balance Volume, etc.), volatility indicators (e.g. volatility of return, Bollinger Bands, etc.), trend indicators (e.g. Moving Average, Moving Average Convergence Divergence, Average Directional Movement Index, etc.), and momentum indicators (e.g. Relative Strength Index, True Strength Index, Stochastic Oscillator, etc.)

But some technical factors have high correlation as the Figure 2 shows, and too many features will meet the curse of dimensionality, so feature selection is necessary in this part. This project will apply several techniques to reduce the dimension, such as filter-based feature selection methods, Principal component analysis.

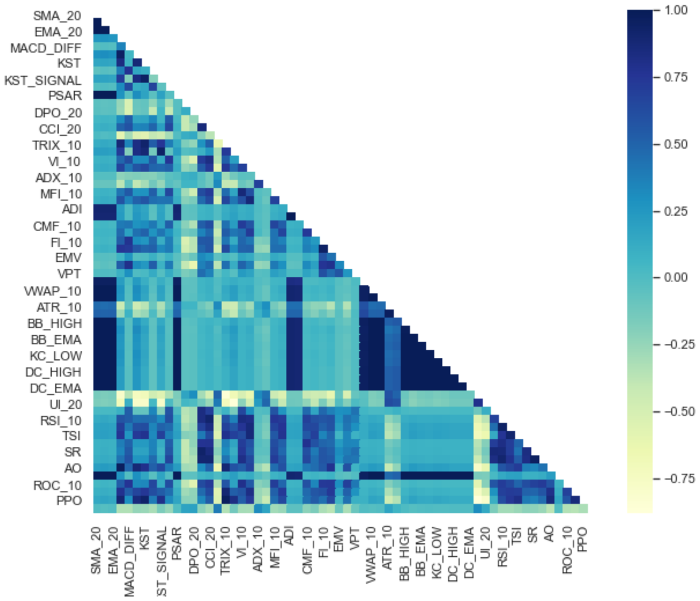

Figure 2. Correlation between features

Will update the follows work in the coming reports.