Machine Learning Engineer Nanodegree

Capstone Proposal

Pengfei Gao
2017-11-27

Machine learning to predict stock price trend

Domain background

Predicting the stock price from chaotic market data has always been attractive for both investors and researchers, technical analysis and fundamental analysis are the two main schools of thought when it comes to analyzing the financial market. Technical analysis looks at the price movement of a security and uses this data to predict future price movements while fundamental analysis instead looks at economic and financial factors that influence a business. E.g. technical analysis is trying to interpret the stock indicator like 'sample moving average' which is the average stock price over a period, 'moving average convergence divergence' which reveal changes in strength, direction, momentum, and duration of a trend in a stock price in order to predict the price trend, fundamental analysis trying to analysis the company financial report like balance sheet, incoming statement, cashflow statement in order to calculate company's value and to see whether the company's stock price is overvalued or underestimated. Machine learning technology has already applied to stock prediction *(https://www.researchgate.net/publication/259240183_A_Machine_Learning_Model_for_Stock_Market_Prediction).*

Problem Statement

This project is focusing on the technical analysis, I will use the stock indicator such as SMA, MACD, Bollinger Bands... to predict the stock price trend in next few days, i.e. the output for this project has 2 classes(+1, and -1), I will classify the output as +1 if next 10 day's average adjusted close price is greater than today's adjusted close price and -1 if next 10 day's average adjusted close price is less than today's adjusted close price. It is very hard to predict the stock price trend by just one indicator nowadays, but I believer

by combining some the indicators and use machine learning methods(e.g. SVM, Adaboosting) to create a model, we may get a better prediction than just one or few indicators.

Datasets and Inputs

Yahoo finance(*https://finance.yahoo.com/*) provide the stock information(i.e. open, high, low, close, adjusted close, volume) for any given days, and pandas_datareader.data provide the interface to accessing these data, e.g. if we want to get Apple's stock information from 2010-1-1 to 2016-12-31, we just need to call pandas_datareader.data.DataReader('AAPL', datetime.date(2010,1,1),datetime.date(2016,12,31)), the function will return a pandas dataframe which contains the columns of date, open, high, low, close, adjusted close, volume. Then I will use these raw data to compute the indicators(e.g. 6 days SMA(sample moving average) = df['Adj Close'].rolling(window = 6).mean()), and treat all these indicators as features. The label is the average price in next few days compare with today's price, +1 if greater than current price and -1 if less than current price, e.g. if the given date is 2010-1-1, and the price is 100, we want to predict 6 days price trend, the average price for price from 2010-1-2 to 2010-1-7 is 110, we say the label of 2010-1-1 for the given stock is +1 assuming all these days are trading days.
Since stocks are heavily influenced by the market, i.e. if market moves up, the stock price is more likely to move up, and vice versa, I will also include the market index(e.g. NASDAQ, SP&500)
I plan to use following 29 features for 5 stocks ['MMM', 'MSFT', 'AAPL', 'XOM', 'BAC']:
['Adj Close', 'Volume', 'SMA3', 'EMA6', 'EMA12', 'EMA26',
'MACD12_26_9', 'MACD_signal12_26_9', 'SMA14', 'Upper_band14', 'Lower band14', 'RSI6EMA', 'RSI12EMA',
'Momentum1', 'Momentum3', 'RateOfChange3', 'RateOfChange12', 'CCI12', 'CCI20', 'WILLR14', 'ATR14',
'TripleEMA6', 'OBV', 'MFI14', 'PDI14', 'NDI14', 'ADX14', 'PDI20', 'NDI20', 'ADX20']
These Indicators are computed against different period, e.g. SMA3 is the 3days simple moving average, the formula for these indicators can be find at *http://stockcharts.com/*.
I will include 2 market index NASDAQ and SP&500, so the total features for the stocks are 29*3 = 87.
I plan to use the data from 2010-1-1 to 2015-12-31 as training data and data from 2016-1-1 to 2016-12-31 as testing data.

Solution statement

As I described from Datasets and Inputs section, we already have features X and labels y, I plan to use SVM algorithm to train the model from earlier period of time in the dataset, i.e. from 2010-1-1 to 2015-12-31, and then test the model from later period of time in the dataset, i.e. from 2016-1-1 to 2016-12-31. Since some stock features are extremely larger than others e.g. volumes, I will use minMaxScaler in sklearn to normalize all the features, I will also use the gridsearch and use fbeta_score to score the gridsearch in order to find the best estimator for SVM.

Benchmark Model

I plan to use some simple trading strategy as the benchmark model, e.g. if the stock price is above the simple moving average price given the period, we predict stock price will move up, and vice versa.

Evaluation Metrics

I plan to use accuracy and f-1 score to compare the performance for benchmark model since the classes for stock data will not be balanced (stock values increase more than they decrease overall) and F1 metric will not be affected by class imbalances.

Project Design

The first step of my project is data collection, I will collect raw information from these 5 stocks ['MMM', 'MSFT', 'AAPL', 'XOM', 'BAC'] and these 2 index [S&P500, NASDAQ] from yahoo finance.
After pull all raw information for all SP&500 stocks as well as market index from yahoo finance, I will create features and labels for each stock and market index. Detail for create features are in **Datasets and Inputs** section.
After the calculation for all these indicators, the data frame will have following columns(X)[Date, Open, High, Low, Closed, Adjusted Closed, SMA6, EMA6, MACD, ...], I also plan to add the market index indicators as features, so columns(X) will also include[SMA6_SP500, EMA6_SP500,...,SMA6_NASDAQ,...]
The label y is the up or down of the price given next few days, e.g. for predicting next day's price trend:
y(t) = 1 if price(t+1)>price(t), otherwise -1
For predicting the next 3-day average price trend,
y(t) =1 if SMA(t+3)>SMA(t); otherwise -1
After the feature collection, I will use minMaxScaler in sklearn to normalize all the features, and use SVM to create the model I plan to use the gridsearch and fbeta_score to score the gridsearch in order to find the best estimator. Last but not least, I will compare the score for SVM model and benchmark model.