Machine Learning Engineer Nanodegree

Capstone Proposal

CHENG HAN WU January 7, 2018

Stock Price Indicator Proposal

Domain Background

In investment and trading, the buy and sell decision have been made by human. But there are lots of factors that influence human and cause loss. So how about machine? there are wealth of information is available in the form of stock prices and company performance, so it is suitable for machine learning.

Here is a paper that using machine learning to predict the stock price, and there are many other papers doing research on this field. So, apply machine learning to predict stock price is feasible.

Investment firms, hedge funds and even individuals have been using financial models to better understand market behavior and models can provide some predicts for trader reference.

We would use **TWII(Taiwan Stock Exchange)** and **TSMC(Taiwan Semiconductor Manufacturing Company)** in this project, TSMC is the target stock we want to predict in this project, the detail about data would be described in **Datasets and Inputs** section below.

Problem Statement

Our problem is that trading decisions made by human is not always reasonable, such as me. So, I want try to use machine learning to train a model

which can give the predicted stock price value in some days range, and see it can help trader to make better decision or not.

The problem is a regression problem, because the output value which is the price in this project is continuous value. We take features like **Open price**, **Highest price**, **Lowest Price** etc. input and predict the adjusted close price value in one day later for example.

Datasets and Inputs

I decided to get data from Yahoo finance, it provide historical data that can be download as csv, I choose **TWII(Taiwan Stock Exchange)** which is similar to **NASDAQ**, we can know whether the market was opened or closed by **TWII**. And **TSMC(Taiwan Semiconductor Manufacturing Company)** which is the weighted stock in Taiwan stock market, so it has high liquidity and it's not easily affected by single major institutional investors.

The historical data fetch from Yahoo finance has seven columns, **Date, Open, High, Low, Close, Adjusted Close and Volume**, the difference between **Adjusted Close** and **Close** are described here, we would use **Adjusted Close** as our outcome variables which is the value we want to predict. **Open, High, Low, Volume** would be used for train in this project, these features are continuous. **Date** is a kind of categorial data, and we won't directly use it to train, we would use **Date** to sorting the data.

The data date range I choose is from 2003/01/01 to 2017/12/29, the data has 3750 rows in total. And There are some null values because stock market is close in these days, and I would preprocess the data before training.

We would use TimeSeriesSplit function in scikit-learn to split the data into training set and testing set, it could split the whole dataset into several packs and in each packs, the indices of testing set would be higher than training set. By doing this can prevent look ahead bias, which means the model would not use future data to train itself.

Solution Statement

Because the output is linear, so I would like to choose some supervised regression algorithms, such as **Random Forest Regressor, SVM, RVM** and a model that bagging them together. I would try them separately and choose the one which performance the best.

The model take input which include the original data(from Yahoo finance) and some calculated indicators, like Moving Average and Bollinger Bands, the way to get them would be described in **Project Design** part. The model would return the forecast stock **Adjust Close** value in the chosen day range (7 days for example).

Benchmark Model

I would choose simple **Linear Regression** and use default arguments as the benchmark model. Use the same data and features as the solution model above. By doing so, could compare whether the solution model we picked up and adjusted is better or worse than the simple model, the evaluation metric would be described below.

Evaluation Metrics

I supposed to use **RMSE** as the evaluation metric, **RMSE** (root-mean-square error) is a popular evaluation metric used in regression problem. **RMSE** is the square root of the average of squared errors, the formula is:

$$\sqrt{\frac{\sum_{i=1}^{n}(Predicted - Actual)^2}{N}}$$

And the lower **RMSE** value represent the predicted value has less error, which means the value predicted by the model is accuracy, so when we apply this evaluation metric to both our solution model and benchmark model, we choose the one has lower **RMSE** value and say that one is better.

Project Design

In my opinion, this project can be separated into three parts. Data, Model and Showcase.

Data part

The data fetch from Yahoo Finance has seven columns, **Date, Open, High, Low, Close, Adjusted Close and Volume**, as mentioned above, we choose Adjust Close as our target value, so we would not use Close. We want to forecast it by input **Open, High, Low, Volume, Moving Average and Bollinger Bands**, there are some different time series in **Moving Average** like 5days, 20days etc., I supposed to use 5days and 10days because I think that there are too many factors to affect the stock price, so the technical analysis is hard to have accuracy result in predicting the price in far future. And below are formulas of Moving Average and Bollinger Bands.

$$\frac{\sum_{i=1}^{n} pi}{n}$$

n is the time series, we choose 5days and 10days in the project, so **n** is 5 and 10. **p** is the close price, we use **Adjust Close** in this project.

$$nMA \pm 2 * (n\sigma)$$

n is the same as the time series of **Moving Average**. For example, if we choose 5MA, the **Bollinger Bands** are **5MA +- 2 * the Standard Deviation of close price in 5 days**.

There are some null values in the data, so we should preprocess them before using them to train the model, by using **fillna** function in pandas, we can easily replace null values to the price of previous trading day and throw the null value out in training, because the price in the days that the market close would not change, so we can simply not use it.

Model part

In the project, we want to build a model that can forecast the Adjusted Close price in the chosen days after, like predict the Adjusted Close price 7 days later. And we first take these features to train, **Open, High, Low, Volume, 5MA, 10MA, Bollinger Bands in 5MA**, and by using the feature importance function, we can discover which feature is important and then modify our training feature.

The data need to be normalize before train, there are two different normalize types in these features. The price part, such as **Open, High, Low, 5MA, 10MA, Bollinger Bands**, these features would be divided by the Adjusted Close price in the first day we choose. The **Volume** value is usually about tens of billion, so we could divide them by one hundred billion, so the Volume value after normalize would be in range of 0 to 1.

Date	Open	High	Low	Close	Adj Close	Volume
2003/1/1	null	null	null	null	null	null
2003/1/2	33.1663	33.4746	32.7035	32.7804	19.28072	29925256697
2003/1/3	33.9373	34.7084	33.6291	34.5546	20.32427	52579680536
2003/1/6	34.5546	35.2488	34.2456	34.5546	20.32427	49129151006
2003/1/7	35.2488	36.1743	35.0174	35.0174	20.59648	61698511243
2003/1/8	35.0943	36.5595	35.0174	35.8653	21.09519	67404812442
2003/1/9	35.4802	35.7884	35.0943	35.4802	20.86869	48948024610
2003/1/10	36.4057	36.4826	36.1743	36.4057	21.41305	57046931096
2003/1/13	37.0223	38.951	37.0223	38.951	22.91014	1.37E+11

And we try these algorithms, **Random Forest Regressor, SVM, RVM** and Ensemble Learner combine them together, after adjust and test them, we compare the performance of these algorithms and pick up the best one.

Showcase part

I suppose to use Jupyter notebook to show the research and the complete model, because Jupyter notebook is good to show the result and description.