Machine Learning Engineer Nanodegree

Capstone Proposal

Liangliang Yang

Domain Background

Stock prediction is always a hot topic. Also, it is extremely hard to try and predict the stock market. There are some many factors that could affect the price: the earnings of the company, the trends of the overall economy, the political climate and so on.

Traditionally, people built many stock indicators, and use several statistical models (especially with time series models) to predict whether a stock will go up or down in the future. In recent years, as machine learning and deep learning become more and more popular, people start to use various new techniques to perform stock market prediction [1].

Problem Statement

In this project, I will be working on the “Stock Predictor” problem. There are mainly two goals in this project. First I will try to build a machine learning/deep learning model that can predict the future stock “Adj Close” price, based on historical data. Training and evaluation will be run to test the model performance.

In addition to the stock prediction model, the second main piece of this project is that, I will build an interactive web application that can enable users to pick different stock symbols and time ranges for model running.

Datasets and Inputs

There are multiple ways that we can get stock prices. For the purpose of this project, I will use yahoo finance (https://finance.yahoo.com/). There is a python package named finance [2], which can enable us to pull stock prices easily.

The stock data from yahoo finance is fairly simple, as we can see below. There are mainly 7 columns including the Date. In my project, I plan to use “Adj Close” as our target (the prediction price).

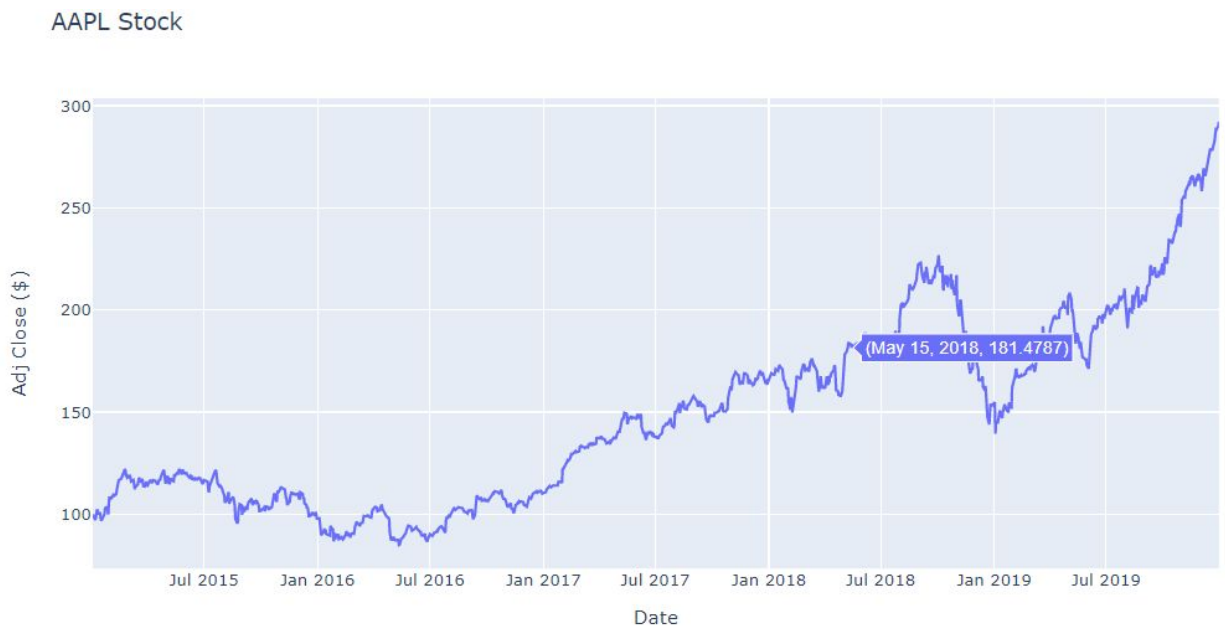
	Date	Open	High	Low	Close	Adj Close	Volume
0	2015-01-02	111.389999	111.440002	107.349998	109.330002	99.945885	53204600
1	2015-01-05	108.290001	108.650002	105.410004	106.250000	97.130241	64285500
2	2015-01-06	106.540001	107.430000	104.629997	106.260002	97.139420	65797100
3	2015-01-07	107.199997	108.199997	106.699997	107.750000	98.501518	40105900
4	2015-01-08	109.230003	112.150002	108.699997	111.889999	102.286186	59364500

Also since most machine learning/deep learning techniques work best with normalized data, I will use MinMaxScaler to normalize the data.

Solution Statement

As we mentioned in the problem statement, there are mainly two parts of our project. The first part is to implement a machine learning / deep learning model, which can predict the stock closed price well. There are several possible methods for the solution (Linear Regression, KNN, random forest, xgboost, etc). Among those techniques, a deep learning method named LSTM (Long Short Term Memory) [3, 4] seems very promising. LSTM is very powerful in sequence prediction, since it is able to store the past information. I will mainly focus on the model by LSTM, and hope its performance is acceptable.

Once I find an appropriate model, the next main task is to build an interactive web application. For this part, I plan to use Dash and Plotly [5]. It is a Python framework for building analytic web applications. Personally I have used it for one work project, and it works pretty well. I also used it a lot in my daily analysis work inside jupyter notebook, as it provides an interactive way to check the data point values as below.



Benchmark Model

In the project, I plan to use a traditional prediction method, such as the moving average model as my benchmark model. For the purpose of comparison, I will train and predict the stock price for the same time range (for example, train on the first 90% of data and predict on the next 10%).

Evaluation Metrics

I will primarily use both Root Mean Squared Error (RMSE) [6] and Mean Absolute Percentage Error (MAPE) to evaluate the model.

Here,

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(y_{t-real} - y_{t-pred})^2}{T}}$$

$$MAPE = \frac{1}{T} \times \left(\sum_{t=1}^{T} \frac{|y_{t-real} - y_{t-pred}|}{|y_{t-real}|}\right) \times 100$$

The Mean Absolute Percentage Error will be used to calculate the percentage difference. If we see the MAPE and RMSE of our machine learning / deep learning model are less than the moving average model, and the MAPE is within 5% for 7 days forecasting, this is a good sign that our model is good.

Project Design

As I can see now, the project design mainly contains the following steps:

1. Import datasets: setup conda environment and install proper packages, especially the packages to download data from yahoo

2. Start with one stock (eg, AAPL stock), use jupyter notebook to perform some initial data analysis, and see how can we use Dash plotly to plot the stock price

3. Implement machine learning model(s), especially try the LSTM model (or other models that can beat the moving average model). Explore this step with a jupyter notebook first.

4. Write a python package, convert the notebook code to the package

5. Build a dash web application with python, dash and plotly. Implement functions that enable users to choose a stock symbol, and choose time range for model run.

Reference:

[1] https://www.udacity.com/course/machine-learning-for-trading--ud501
[2] https://pypi.org/project/yfinance/
[3] https://colah.github.io/posts/2015-08-Understanding-LSTMs/
[4] https://en.wikipedia.org/wiki/Long_short-term_memory
[5] https://github.com/plotly/dash
[6] https://en.wikipedia.org/wiki/Root-mean-square_deviation