Machine Learning Engineer Nanodegree

Capstone Proposal

Shreyas Matade

December 21st, 2018

Proposal

Deep Learning Stock Price Predictor

Project intuition

Like most of the tech engineers, I was impressed and surprised by the bitcoin-cryptocurrency boom. Like any other human, I was drawn to FOMO and started investing in crypto (and I can say proudly, i did not lost nor gained). This is where I started my curiosity toward trading bots.

After taking Machine learning nanodegree course, I learned a lot about stats, data analysis and various machine learning models, methodologies.

I really wanted to apply my newly learned skills to work. However, cryptocurrency market is highly volatile and as of now it is becoming unreliable. Hence, just to get the taste of analyzing time series data I thought of exploring “comparatively” more reliable Stock Market.

Domain Background

Stocks Market and Trading

Let’s see what is stock market, it’s an equity market or share market is the aggregation of buyers and sellers of stocks (also called shares), which represent ownership claims on businesses [Wiki]. Anyone can make profits in stock market by two ways, Trading and Investing.
Stock trading is about buying and selling stocks for short-term profit, with a focus on share prices. Investing is about buying stocks for long-term gains.
There are different types of indices available in US for trading. I have chosen to use S&P500 which is
American stock market index based on the market capitalization of 500 large companies having common stock listed on the NYSE or NASDAQ.
Stock markets can be highly profitable if traded smartly. Reward that trader wants to achieve here is simple, to maximize the profits and avoid risks. Considering volatile and rapid nature of Exchange Traded Funds(ETF), it make more sense to have computer programs to do trading work for you, if nothing this will definitely avoid the influence of emotions.

Algorithms in Trading - Computational Trading

When algorithmic trading strategies were first introduced, they were wildly profitable and swiftly gained market share. In May 2017, capital market research firm Tabb Group said that high-frequency trading

(HFT) accounted for 52% of average daily trading volume. But as competition has increased, profits have declined. In this increasingly difficult environment, traders need a new tool to give them a competitive advantage and increase profits.
Algorithms like simple moving average, Exponential moving average are fair to predict stock prices for shorter period in future but these techniques has limitations to predict prices at far time in future. These limitations can be avoided by momentum based algorithms. But more appropriate solution to this problem would be Machine Learning.

Supervised Learning

Machine learning techniques are much more capable of learning hidden patterns in the historical data and come up with better and profitable strategy. We can use Supervised Machine learning algorithms linear regressions, neural networks, support vector machines, and naive Bayes, to name a few.
However, time series prediction problems are a difficult type of predictive modeling problem.

Deep Learning

Time series nature of stocks adds complexity of a sequence dependence among the input variables. Such a complexity can be handled by applying Deep Neural networks. A powerful type of neural network designed to handle sequence dependence is called recurrent neural networks. The Long Short-Term Memory network or LSTM network is a type of recurrent neural network used in deep learning because very large architectures can be successfully trained.

Project outline

In this project, I am planning to start from simple supervised learning methods that I learned over this machine learning coursework to the deep learning RNN models. I am looking forward to see expected performance improvements as I move from simple to complex models. While researching, I found that a lot of academic work has been done applying RNNs for time series data, and I feel prepossessing data and come up with significant features would be challenging since it requires expert domain knowledge which is not easily available for obvious reasons.

Problem Statement

Given a stock and time period in future, apply Deep learning model, RNN, to predict the close value.
The model shall be trained on historical time series data of a given stock and the model shall be backtested to compare its accuracy over the span of time period.
This model then shall be applied different stocks to see its performance.

Datasets and Inputs

I will be using fix-yahoo-finance python package to get the stock data.
Below are the variables of the time series data of a single stock.
Solution shall be able to get the historical data given a Ticker symbol of

the desired stock.

I will be using data from Apple(AAPL), Microsoft(MSFT), Amazon(AMZN) and IBM(IBM) from their respective start dates till current date.

So, the dataset varies from ticker to ticker depending on when the company entered into the stock market. Below is the shape of data by ticker

1. AAPL (9589, 6)
2. AMZN (5438, 6)
3. IBM (12356, 6)
4. MSFT (8264, 6)

Example: For IBM

```
from pandas_datareader import data as pdr

import fix_yahoo_finance as yf
yf.pdr_override()

# download dataframe
data = pdr.get_data_yahoo("IBM", start="1970-01-01")
```

This data will then be split in training, validation and testing data considering timeseries nature

1. Open
2. High

3. Low
4. Close
5. Adj Close
6. Volume

Derived Features:

Windowed Rolling Mean - This value is ((High + Low)/ 2).
Days Since Market is Open - It is the days since market is open. This might affect the prices if market was closed for multiple days.
Since, optimizing features, finding out significant features is a open problem, I am constantly looking for better features which can be vital predicting target. I will be looking for significant features throughout this project as I deepen my understanding of stock market dynamics.
*Bollinger Bands**

Solution Statement

Solution would be programmed in python using Multiple Jupyter notebooks. The optimized Recurrent Neural Net with LSTM shall be able to predict the close value of the given stock for given time period. The expected solution shall involve

1. Come up with significant features that would be significant in predicting closing value, target variable.
2. Prepossessing time series data
3. Training RNN model with time series stock data.
4. Optimizing model by tuning hyperparameters.

5. Predict the stock close value.

Solution would be compared against the benchmark model and it is expected to perform better than the benchmark model.

Benchmark Model

I will be keeping simple regression model as my benchmark model.

Evaluation Metrics

I will use MSE (mean squared error) as my evaluation matrix to compare predicted and actual values of the target stock at market close. MSE would be used to evaluate the performance of the benchmark model and final deep learning model. It is expected to see final deep learning model , RNN with LSTM , gives us better (lesser) MSE.

Project Design

I will be using Python Jupyter notebook to program the solution. I will be using numpy, pandas and scikit-learn python libraries for initial EDA and developing benchmark (regression) model. For Deep Learning models, I will be using Keras implementation of Tensorflow. For Visualization matplotlib and seaborn will be used.

Data Collection

Get the Data using fix-yahoo-finance. Function to return data for specified Tikcer symbol.

Exploratory Data Analysis

1. Get the overall Statistics of the data.
2. Using pandas find the correlation between features and target variables.
3. Look for missing values and discrepancy in the data.

   Data Preprocessing

4. Impute the data into Dataframe using pandas.
5. Append derived features to the dataframe.
6. Normalize the data.
7. Sort Data and Split the data 70/15/15 or 60/20/20 percent in training, validation and testing. This split will be according to time.

   Considering the timeseries nature of data, I will make sure that model training performed initial 70% of data, then validation on next 15% of data, then the testing on latest 15% of data in order to “look ahead” issue.

Training and Validating Benchmark Model

1. Using Scikit-learn develop regression model.
2. Test it, log results - MSE.
3. Use GridSearchcv to get better model. Optimize model with hyperparameters.

   Designing RNN with LSTM

4. Design and develop basic RNN with LSTM.
5. Tune the hyperparameters to get optimized model.
6. Compare the results with benchmark model.

I will be trying to visualize most of the results using seaborn and matplotlib.

Even though I am laying out project pipeline to develop benchmark and RNN models, I am keen to see how MSE varies for supervised learners like SVM , naive bayes etc.

Links

1. What is Stock Market
2. Stocks Tutorials
3. Stock Trading vs Stock Investment
4. Different Market Index and Why I choose S&P 500
   5 . https://sigmoidal.io/machine-learning-for-trading/
5. https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
6. https://cloud.google.com/solutions/machine-learning-with-financial-time-series-data
7. Understanding LSTM