FitsnBits

Yung Han Jeong
Flatiron School DSI
nyc-mhtn-ds-091420

Table of Contents

Introduction

Investing in the stock market has gotten much more accessible recently due to the spread of free market brokerage tools such as Robinhood and TDAmeritrade. According to a market research performed by BMO Global Asset Management (2018) showed stark contrast in financial goals in investment. Many professionally established generations, baby boomers and gen-x, mostly invest for retirement, but younger generations, millennials, mostly invest for short term goals. Additionally, JPMorgan Chase found that over 60% of baby boomers are working with an advisor for their portfolio, but over 70% of the millennials are investing on their own according to Yahoo Finance. (2020)

Business Problem

Stock market prediction is crucial information to any investors to maximize their profit. For the young, often independent, investors their predictions solely depend on their news and intuition. In addition, the small and short tendency of these accounts make them unlikely to receive large detail or assistance from professional advisors. A low cost data driven prediction is required to assist these accounts.

Solution

This project will provide stock price prediction by utilizing neural networks for regression analysis. The prediction model can be made interactive by allowing the user to curate their own data set based on their need. For instance, a model can be made to predict the stock price dependent on other stock pricing allowing the user to gauge investment options between companies.

Data

A stock quote is a time series generally consisting of open, high, low, and close prices with total volume traded in the given interval. The data available projects are both historic and live stock prices up to the limit set by the API. For the scope of this project the pre-market and post-market movements will not be utilized.

Collection

The data will be collected through twelvedata API (twelvedata.com). The API can automatically format the data into a pandas dataframe, json, or sql. Pandas dataframe will be utilized to leverage the datetime data format to analyze the API data.

The twelvedata API provides:

- 800 calls per day
- Batch call up to 150 companies/symbols in a given market
- Time series up to 5000 rows
- Formatted output: json, csv, pandas dataframe, etc

Formatting

It is expected that all API returns are clean and formatted data. Few logic will be required to build a database if required data is larger than a single API call. A chained scheduled call or a data mining may be required for creating a large training data set. A scripted can be built to mine for specified data. The returned data can be formatted in many ways including pandas dataframe. The mining can leverage this format to utilize pandas datetime indexing for sorting mined data.

Storage

The anticipated data size is not large, but should be organized similar to API call to streamline the data flow. The initial plan is to create csv files of each data and store the date time index along with the data. If more organization is required a database can be deployed to manage the data. These data will most likely not be posted on github.

Model

Creating and predicting stock prices using neural network regression can produce acceptable results within a very short amount of time. The user of this model will be provided resources to create curated data sets from the stock market and produce a custom model depending on their needs quickly to leverage their investments.

Technique

A regression model can be made with a neural network by removing the activation layer at the terminal layer. To increase model performance the input value will be scaled with scalers available with sklearn. These scalers will be saved to transform the prediction to provide real data to the user.

The neural network could be built using few different layer types. Initial testing with linear layers returned acceptable ranges of values. Utilization of RNN or LSTM should result in model performance increase. This project will explore a few network and layer variations to find optimal neural network design.

Evaluation

The model will be evaluated using root mean squared error formula (RMSE). The goal is to minimize the variation between the prediction and true value.

Deployment

The deployment of this project will consist of a few modules and a web application deployment project should include all modules. Each module represents the core functionality of this project and will serve the basis of all user interaction.

Data Module

- Collection/query data as required
- Combine multiple query and provide a single data set as user curates
- Scale data as necessary
  - Save the scaler as needed
- Create a train test split in sequence

Model Module

- Regression with neural network built on pytorch
- Optimize by SGD/Adam with RMSE

Schedule Module

- Create data query and train schedule
- Return model performance

Tools

- Data
    - twelvedata TDClient
    - Pandas
        - Datetime Index
    - Numpy
        - Random Selection
        - Array vectorization
    - matplotlib/seaborn
        - Visualization
- Modeling
    - Pytorch
        - Neural Network
        - Tensor data type
    - Numpy
        - Error Calculation
        - Random Selection
    - Sklearn
        - Data Scaling
- Deployment
    - Flask
        - Web deployment
    - Steamlit
        - Lightweight deployment