11/2/2020

Stock Price Predictor

Statement of Work (SoW)

Kartik Sojitra
100723768

Contents

Executive Summery

Investment firms, hedge funds, and even individuals have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process.

Can we predict stock prices with machine learning? Investors make educated guesses by analyzing data. They will read the news, study the company history, industry trends, and other lots of data points that go into making a prediction. The prevailing theories are that stock prices are random and unpredictable but that raises the question of why top firms like Morgan Stanley and Citigroup hire quantitative analysts to build predictive models. We have this idea of a trading floor being filled with adrenaline infuse men with loose ties running around yelling something into a phone but these days they're more likely to see rows of machine learning experts quietly sitting in front of computer screens. About 70% of all orders on Wall Street are now placed by software, we are now living in the age of the algorithm.

This project seeks to utilize Deep Learning models, Long-Short Term Memory (LSTM) Neural Network algorithm, to predict stock prices. For data with timeframes recurrent neural networks (RNNs) come in handy but recent research have shown that LSTM, networks are the most popular and useful variants of RNNs. I will use Keras to build the LSTM to predict stock prices using historical closing price and trading volume and visualize both the predicted price values over time and the optimal parameters for the model.

Problem Statement

The challenge of this project is to accurately predict the future closing value of a given stock across a given period in the future. In the past few years, we have seen lots of academic papers published using neural nets to predict stock prices with varying degrees of success but until recently the ability to build these models has been restricted to academics. Now with libraries like tensor flow, anyone can build powerful predictive models trained on massive datasets. For this project, I will use a Long Short-Term Memory network – usually just called "LSTMs" to predict the closing price of the S&P 500 using a dataset of past prices.

Goals:

1. Explore stock prices.

2. Implement basic model using linear regression

3. Implement LSTM using Keras library.

4. Compare the results and submit the report

Datasets and Inputs

I will be using the daily prices of the S&P 500 from January 2000 to June 2017; this is a series of data points indexed in time order or a time series. My goal will be to predict the closing price for any given date after training. All the necessary data for the project will come from Google Finance.

Solutions Statement

For this project, according to my research, the best possible solution is to utilize a LSTM Neural Net model capable of learning from time-series data. This project will be programmed in a Jupyter Notebook (iPython) for ease of reproducibility. Using a Keras implementation of the Tensor Flow library, the solution will utilize a LSTM Neural Net model and will be supported by a Pandas DataFrame library for convenient time series data schema. The measures of performance will be based on the predicted stock ticker price in comparison to both the actual price and the benchmark model's predicted price.

Benchmark Model

For this project, I will use a Linear Regression model as its primary benchmark. As one of my goals is to understand the relative performance and implementation differences of machine learning versus deep learning models. This Linear Regressor will be based on the examples presented in Udacity's Machine Learning for Trading course and will be used for error rate comparison MSE and RMSE utilizing the same dataset as the deep learning models.

Evaluation Metrics

For this project, I will measure performance using the mean squared difference between predicted and actual values of the target stock at the adjusted close price and the delta between the 4 performance of the benchmark model (Linear Regression) and our primary model (Deep Learning).

Project Design

This project will be implemented through the Keras/Tensor Flow library using LSTM Neural Networks. Development workflow will follow the below sequence:

1. Set Up Infrastructure
   - iPython Notebook
   - Incorporate required Libraries (Keras, Tensor flow, Pandas, Matplotlib, Sklearn, Numpy)
   - Git project organization
2. Prepare Dataset
   - Incorporate data of S&P 500 companies
   - Process the requested data into Pandas
   - Dataframe Develop function for normalizing data

- Dataset will be used with 80/20 split on training and test data across all models

3. Develop Benchmark Model
    - Set up basic Linear Regression model with Scikit-Learn
    - Calibrate parameters
4. Develop Basic LSTM Model
    - Set up basic LSTM model with Keras utilizing parameters from Benchmark Mode
5. Improve LSTM Model
    - Develop, document, and compare results using additional labels for the LSMT model
6. Document and Visualize Results
    - Plot Actual, Benchmark Predicted Values, and LSTM Predicted Values per time series
    - Analyze and describe results for report