

Implementing a Neural Net for Stock Forecasting

Helen Mehreteab and Anusha Murali

December 9, 2019

1 Description

Predicting the price of a company stock is one of the most difficult tasks, even for experienced traders. The future value of a stock is dependent on numerous factors such as the company's financial health, the historical stock prices, analyst reports, and the general market volatility. In this project, we use the historical stock prices of a company to predict its future stock prices. Using three popular machine learning algorithms, namely linear regression, Prophet and Long Short Term Memory (LSTM) algorithm, and elementary data manipulations on widely available CSV data for publicly traded companies, we demonstrate that one could predict stock prices with a high degree of accuracy.

Our program, called `cs50stock` was written in Python, and can be run from any platform that has Python version 3.7.5 or higher.

2 Getting Started

The following instructions outline how to get the project up and running on a local machine for both development and testing purposes.

2.1 Prerequisites

Our project was developed using Python 3.7.5 on Mac OSX. The source code is entirely contained in a single Python file called `cs50stock.py`. The following Python libraries are necessary to run `cs50stock`:

2.1.1 Mandatory Python Packages

Our project has absolute dependency on the following Python packages.

1. `pandas`
2. `numpy`

3. `matplotlib`
4. `sklearn`
5. `keras`
6. `fbprophet`
7. `fastai` It is important to have version 0.7.0 of `fastai` installed as the most recent version does not support the `structured` module. It took as more than three hours to figure out that the latest version of `fastai` has many modules renamed from the earlier versions.
8. `torch` It is important to have version 0.1.2 of `torch` installed on the system as `fastai` is dependent on it.

2.1.2 Supplementary Python Packages

The following packages are not mandatory to successfully run our program. The absence of the following packages will result in warning messages, but will not prevent one from running the code to successful completion.

1. `sys`
2. `warnings`
3. `platform`
4. `subprocess`
5. `time`

2.1.3 Data Files

The program expects one or more CSV files in the directory where `cs50stock.py` is found. These files contain the historical stock data in CSV format. We have made the following files available for testing purposes: `IBM.csv`, `MSFT.csv`, `ORCL.csv`, `FB.csv`, `AMZN.csv`, `GOOG.csv`, `NFLX.csv`, and `AAPL.csv`. More CSV files can be downloaded from `finance.yahoo.com`.

3 Running `cs50stock`

Please note that our program requires Python version 3.0. The program can be started by running

```
$ python3 cs50stock.py
```

from the terminal on a Mac.

The program can also be run from IDLE, which is one of the popular Python editors. However, we have noticed that the Python shell spawned from IDLE may not always properly format tabular outputs generated by the `PrettyTable` Python package. Therefore, we strongly recommend that you run our program from a terminal window on a Mac.

When the program is launched, the following screen first appears allowing the user to select from a set of eight sample stock symbols.

```
*****
*                               CS50 Stock Forecast                               *
*****
Enter a stock symbol (MSFT, IBM, AAPL, ORCL, FB, NFLX, AMZN, GOOG):
```

Selecting one of the above symbols takes the user to the following menu:

```
*****
*                               CS50 Stock Forecast: Main Menu                               *
*****
1: Prediction using a single model
2: Prediction comparison between Linear Regression, FB Prophet and LSTM: Graph
3: Prediction comparison between Linear Regression, FB Prophet and LSTM: Accuracy Table
4: Exit
Please enter a choice: █
```

As seen from the above menu, the program has three major features:

1. Stock forecasting using a specific machine learning model (i.e: Linear regression, FB Prophet, and LSTM).
2. Graphical comparison of the three machine learning models (i.e: Linear regression, FB Prophet, and LSTM).
3. Error rates of the three machine learning models (i.e: Linear regression, FB Prophet, and LSTM).

Selecting Option #1 from the Main menu presents the user with the following set of options:

```
*****
*      CS50 Stock Forecast: Prediction Using a Single Model      *
*****

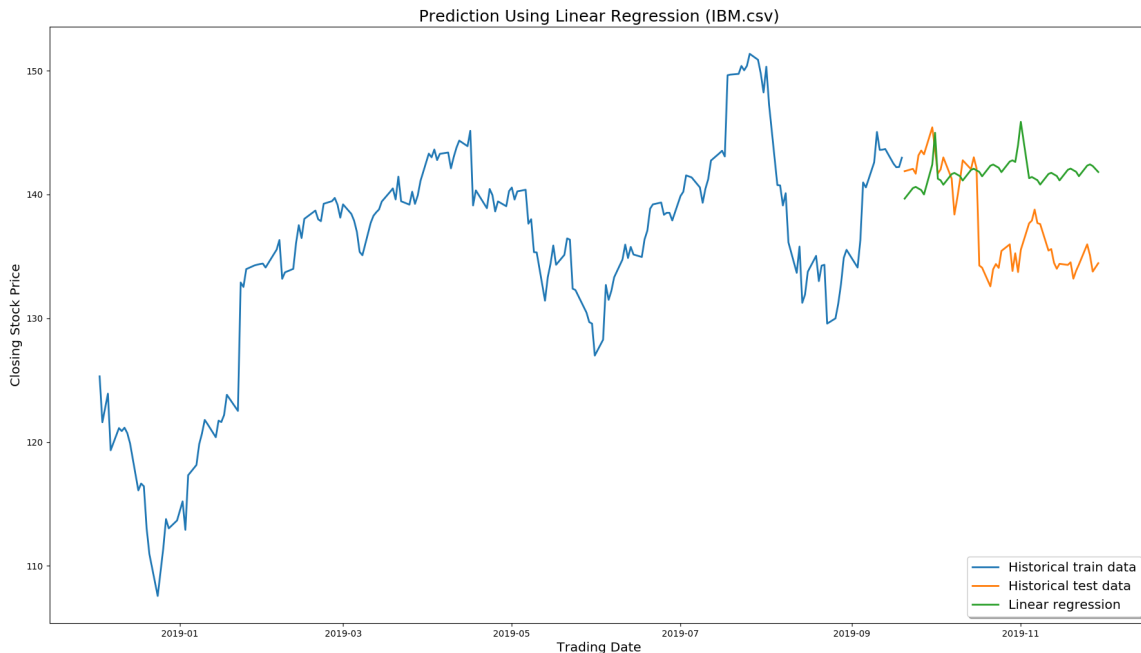
1: Linear Regression
2: Facebook Prophet
3: LSTM
4: Return to Main Menu

Please enter your choice: █
```

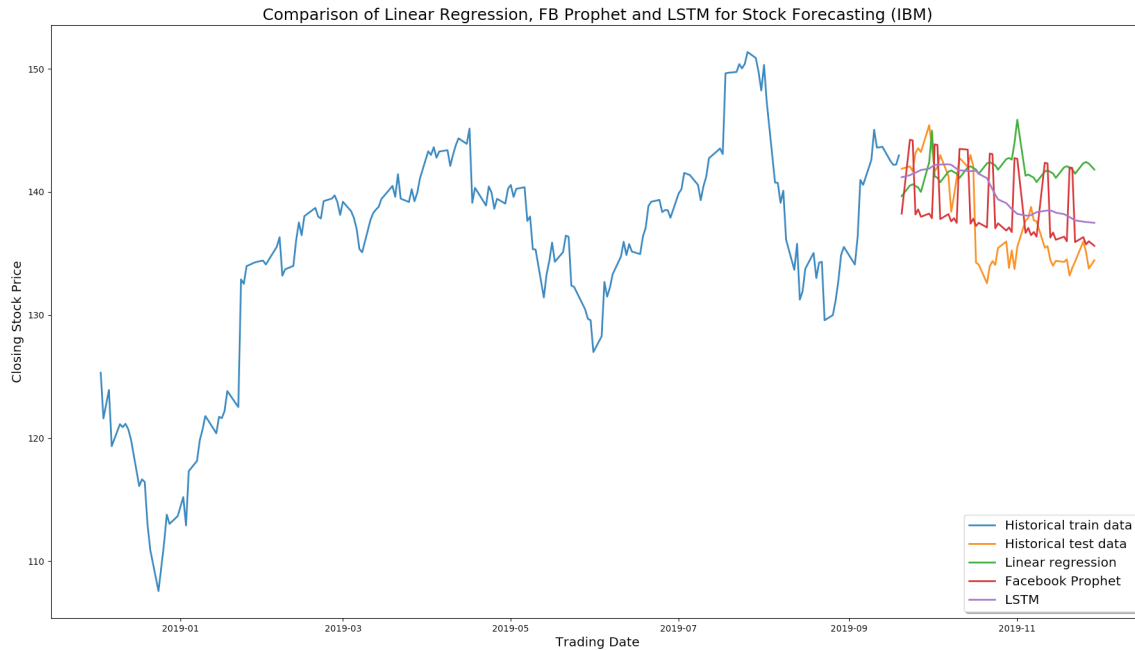
As can be seen from the above menu, the user can now choose one of the following Machine Learning algorithms for stock forecasting:

1. Linear Regression,
2. Facebook Prophet or
3. LSTM

For example, selecting `IBM.csv` and `Linear Regression` generates the following plot. The green line on the right corresponds to the test data (which is nearly 20% of the historical data), and closely tracks the actual test data presented in orange. Therefore, the user can see that the `Linear Regression` model is fairly accurate enough to predict the stock values.



Selecting Option #2 from the Main menu generates a combined plot from all three algorithms, which allows the user to compare their performances for various data inputs. For example, for `IBM.csv`, we obtain the following plot:



Finally, selecting Option #3 from the Main menu generates the following tabular output, which shows the error rate from the three algorithms:

```

*****
*      CS50 Stock Forecast: Prediction Accuracy o Linear Regression, Prophet and LSTM      *
*****
+-----+-----+
| Algorithm Used | Error Rate % |
+-----+-----+
| Linear Regression | 2.93 |
| Facebook Prophet | 0.73 |
| LSTM | 2.52 |
+-----+-----+

```

4 Authors

- Helen Mehreteab
- Anusha Murali

5 Acknowledgments

1. Gad, Ahmed., From $y-x$ to building a complete artificial neural network
2. Koehrsen, Will., Stock Analysis in Python

3. Numerous ML videos from [3blue1brown.com](https://www.3blue1brown.com)