Predicting Popular Social Media Stock Prices

Jared Pippin, Humayun Khan, Lili Balazs

San Diego State University

***Abstract*—Machine learning is a subset of artificial intelligence that is made of algorithms that can be improved through the experience without it being explicitly programmed. It allows us to create algorithms with an immense amount of data as the input that the algorithm can analyze and make recommendations, predictions, and decisions based on the data that was provided. Thus it is widely used to solve many real-world problems in areas such as finance, thanks to its ability to easily identify patterns. Therefore, machine learning is a key part of several financial services and applications such as calculating credit scores, approving loans, managing assets, and making predictions of future stock prices. In this paper, our goal is to discuss the importance of machine learning when it comes to predicting stock prices. In our research, we experiment with popular social media stocks to try to predict future stock prices with the help of the Long short-term memory (LSTM) model. Furthermore, we analyze our results, debate challenges, and aim to provide solutions to predicting stock prices.**

I. Introduction

Machine learning is a sub-area of artificial intelligence. It is composed of computer algorithms that can be improved through experience and without it being explicitly programmed. In order to make recommendations, predictions or decisions we need a mathematical model that relies on sample data, otherwise known as "training data". These algorithms are universally used in a variety of applications where it is hard to develop conventional algorithms to solve the problems we face. Machine learning has a relation to computational statistics that aims to make predictions with the use of computers. In addition, in many business related areas, machine learning might be called predictive analytics.

Machine learning is used to make predictions or classifications based on the data that was given as an input which can be labeled or unlabeled. Based on the provided data, the machine learning algorithm will look for patterns in the data and make recommendations, decisions or predictions.

With the continuous growth of the financial market many inverters aim to find approaches that improve their return on investment by trying to avoid certain risks when making an investment decision. However, the stock market is known to be very dynamic and nonlinear and full of uncertainty. It is also challenging to predict future stock prices since many factors can influence the outcome such as global economic conditions, unexpected events or a company's financial performance. Still we have huge amounts of data to work with that helps us to find patterns and to understand trends which leads us to prediction making. Nevertheless machine learning is crucial in finance and business. For that reason, our goal in this paper is to predict popular social media stock prices.

In spite of how many different factors can influence stock prices it is possible to analyze them as a sequence of discrete-time data. Meaning we observe the data that was taken at successive points in time, such as daily. With time series forecasting we can achieve a high accuracy of predictions when it comes to stock prices. Therefore we applied the Long short-term memory (LSTM) model on the social media stocks dataset that contains information about social media stocks between 2012 and 2022. We chose the LSTM model to do our experiments with because it has the ability to capture historical trend patterns that allows us to predict future stock prices with high accuracy.

During our research we studied the social media stocks dataset. Then we trained and split the dataset while we also developed our LSTM model. Our next step was to create a testing dataset which then allowed us to finally make our predictions. Once we had predictions, we were able to visualize, analyze them and conclude our experiment.

II. Task Description

Machine learning is an excellent approach to making predictions about the future values of a company's stocks. On the other hand, it is not an easy task due to many different factors influencing the outcome of the stock prices. In this paper our goal is to predict social media stock prices. In order to predict them we need to execute the following tasks.

The first step is to select a platform where we wanted to do our project. Since most machine learning projects need heavy computing powers (GPUs) or powerful servers we decided to use Google Colab to make our project since it is a cloud based platform that does not require the need of a GPU. Other cloud based platforms would also be sufficient to use such as Kaggle Kernel or Amazon Cloud.

Once we decide on a platform our next step is to import the libraries that we need to perform our project. During our project we used popular machine learning libraries such as keras, and scikit-learn. We used google.colab to import our dataset. In order to study, manipulate and visualize our data we used the math, pandas, numpy and matplotlib libraries.

Our next step is to load and study the training dataset. In order to load the dataset we use the files.upload() function from the google.colab library. The dataset has eight columns which are the Date, Symbol, Adj Close (Price), Close (Price) , High (Price), Low (Price), Open (Price), and Volume. The

```
import math
from pandas.core.frame import DataFrame
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense,LSTM
from google.colab import files
from datascience import *
import pandas_datareader as web
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use('bmh')
```

Fig. 1. Libraries used in the project.

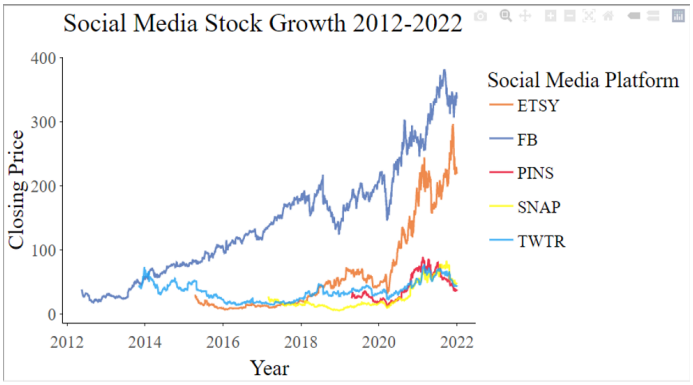

Fig. 2. Social Media Stock Growth 2012-2022.

dataset contains 8398 rows which each indicates a different social media stock on a different day. Once we have the dataset we study and visualize the chosen social media stocks such as Facebook, Twitter and Snapchat to understand them better.

The next task is to normalize and scale the dataset. Then we can create the training dataset so we can split the data into x_train and y_train sets. Once we have the new sets we can convert them into numpy arrays and reshape them.

Now that we have the data that we can work with we can build our model. We decided to use the Long short-term memory (LSTM) model to make our predictions. Once we have our model we can proceed to compile and train the LSTM model.

After we train our model, the next task is to create the testing dataset and convert it into a numpy array and reshape it just as we did it with the training data.

Now we can get our Long short-term memory (LSTM) model to predict the price values. One we have our predictions it is important to Root Mean Square Error (RMSE) to see how concentrated the predicted values are around the line of best fit. The lower the RMSE score the better our model was able to fit the dataset.

Finally can visualize the predicted values and see how accurate our predictions were.

III. Major Challenges and Solutions

LSTM is a special version of RNN(Recurrent neural network) which helps solve the problem of short term memory that Recurrent neural networks have. If we go back to a traditional neural network it has a major shortcoming of not being able to use previous instances to guide future ones. It is like reading a book but forgetting what the previous pages said, while trying to read the current page. Recurrent Neural Networks are built to solve the problem of not being able to remember previous instances by incorporating a loop. The problem with RNNs is that they can only remember things from a short while ago. The problem of long term dependencies is solved using Long Short Term Memory networks (LSTM). LSTM networks are made to handle long term dependencies by default, which is why we decided to use them for our predictions.

We initially used simple linear regression as a baseline method to predict the closing prices. The root mean square deviation was fairly low, which meant there was a decent fit of the data. The accuracy of the results from simple linear regression was not the best.

Since we are dealing with a time series we decided to use a method that can deal with long term dependencies, which is why we went with LSTM. Using LSTM we were able to get much more accurate closing prices than our baseline method of simple linear regression.

IV. Experiments

Dataset Description

The dataset that we have chosen to work with is from Kaggle. The dataset contains information about popular social media stocks such as Meta (Facebook), Twitter, Snapchat, Etsy and Pinterest. The included data is the trading volume, open price, low price, high price and closing price of each trading day. The Volume indicates the total amount of stocks that were traded. The Open column reveals the price that the stock started with when the stock exchange opened. The Low column and the High column show the lowest and highest prices the stock reached during the day. The Close column shows the price of an individual stock the the time when the stock exchange closed for the day while the Closing adjusted column refers to the closing price that also considers other factors like dividends and new stock offerings.

The daily stock prices were conducted from Yahoo Finance from 2012 - 2022. In our experiment we decided to focus on the closing price of Facebook, Twitter and Snapchat. We use the closing price to make our predictions due to the fact that investors use this to determine changes in stock prices over time.

Evaluation Metrics

To evaluate our model, we calculated both the Mean Absolute Percentage Error and the Root Mean Squared Error. First, we utilized Mean Absolute Percentage Error (MAPE),

social media stocks 2012-2022.csv (966.92 kB)

Detail | Compact | Column

Date	Symbol	Adj Close	Close	High	Low	Open	Volume
2012-05-18	FB	38.229999542236 33	38.229999542236 33	45.0	38.0	42.849999237060 55	573576400.0
2012-05-21	FB	34.029998779296 875	34.029998779296 875	36.659999847412 11	33.0	36.529998779296 875	168192700.0
2012-05-22	FB	31.0	31.0	33.590000152587 89	30.940000534057 617	32.610000610351 56	101786600.0
2012-05-23	FB	32.0	32.0	32.5	31.360000610351 562	31.370000839233 4	73600000.0
2012-05-24	FB	33.029998779296 875	33.029998779296 875	33.209999084472 656	31.770000457763 672	32.950000762939 45	50237200.0
2012-05-25	FB	31.909999847412 11	31.909999847412 11	32.950000762939 45	31.110000610351 562	32.900001525878 906	37149800.0
2012-05-29	FB	28.840000152587 89	28.840000152587 89	31.690000534057 617	28.649999618530 273	31.479999542236 328	78863400.0
2012-05-30	FB	28.190000534057 617	28.190000534057 617	29.549999237060 547	27.860000610351 562	28.700000762939 453	57267900.0

Fig. 3. Social Media Stocks Dataset.

```
#Get the mean absolute percentage error(MAPE)
#----facebook----
mape_f= np.mean(np.abs((y_test_f-predictions_f) / y_test_f)) * 100
print("Facebook MAPE:",mape_f)
#----twitter----
mape_t= np.mean(np.abs((y_test_t-predictions_t) / y_test_t)) * 100
print("Twitter MAPE:",mape_t)
#----snapchat----
mape_s= np.mean(np.abs((y_test_s-predictions_s) / y_test_s)) * 100
print("Snapchat MAPE:",mape_s)
```

```
Facebook MAPE: 3.175974494571898
Twitter MAPE: 15.391523914264921
Snapchat MAPE: 11.827289981069246
```

Fig. 4. Mean Absolute Percentage Error.

```
#Get the root mean squared error(RMSE)
#----facebook----
rmse_f=np.sqrt( np.mean((predictions_f - y_test_f)**2))
print("Facebook RMSE:",rmse_f)
#----twitter----
rmse_t=np.sqrt( np.mean((predictions_t - y_test_t)**2))
print("Twitter RMSE:",rmse_t)
#----snapchat----
rmse_s=np.sqrt( np.mean((predictions_s - y_test_s)**2))
print("Snapchat RMSE:",rmse_s)
```

```
Facebook RMSE: 11.982386439330508
Twitter RMSE: 8.999685130142751
Snapchat RMSE: 8.210158507873079
```

Fig. 5. Root Mean Squared Error.

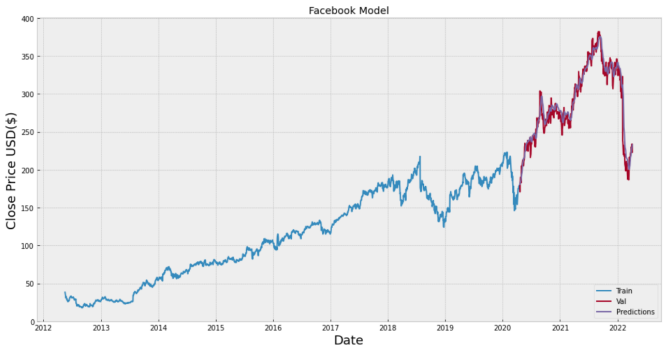

Fig. 6. Facebook Model.

which is the ratio of the average absolute difference between the predicted and actual values divided by the actual value. It is often used to measure the accuracy of a prediction in time series problems. A model is more accurate and works better if the calculated MAPE is a lower value.

As seen above, the MAPE values are all fairly low since they are spread between 3-15%, which shows that our models were able to predict values close to the actual value. The model worked very well for Facebook, as its MAPE value was only 3.15% compared to Twitter and Snapchat's 15.395% and 11.835% respectively.

Facebook's MAPE value may have been lower than the others because the average of its stock values are generally higher. Facebook shares are more expensive than those of Twitter and Snapchat, which might lead to an inflated MAPE value. With something as volatile and unpredictable as the stock market, these relatively low MAPE values seem to indicate the model is doing a good job.

The next evaluation metric we used was Root Mean Squared Error (RMSE), which is the square root of the average of the error squares. It is popular for measuring the quality of a forecasting model, which makes it useful for our project. Like MAPE, a lower value of RMSE indicates higher performance. It is also always positive, which makes it easy to understand and judge the values. A value of 0 would mean that the data was perfectly predicted by the model, so it is best to have a RMSE close to 0.

As seen above, the RMSE values were also fairly low, spread between 8-12%. The model returned a RMSE result that was best for Snapchat and Twitter, while Facebook was slightly worse. This is the opposite of what occurred when the stocks were evaluated with MAPE, which could once again be the result of Facebook dealing with larger price numbers. However, the MAPE numbers were still fairly low, which should indicate that our model did a good job of predicting prices.

Major Results

Using our LSTM model, we were able to create a program that was able to predict stock prices after being trained using a portion of the dataset. The model was able to provide fairly accurate stock predictions that were close to the actual values of these social media stock prices. While the model was unable to predict the exact stock prices, it was able to remain fairly accurate due to its ability to notice and follow trends that the real stock took. When viewing the graphs of the actual closing prices and the predicted closing prices, the model did a good job of following the trends and staying close to the actual values. While our predicted values were sometimes higher or lower than the actual value, it was often able to recover and bring the predictions back towards the actual values.

As seen in the images above, the Facebook model was trained using the data from 2012 to 2020 in order to predict into 2021. As you can see, it was able to properly predict the movement of the stock. It was not always accurate to the exact

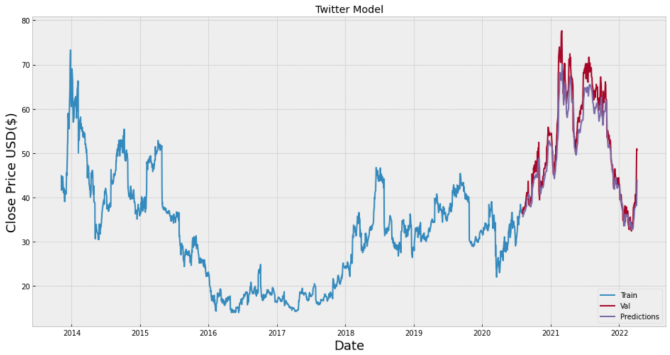

Fig. 7. Twitter Model.

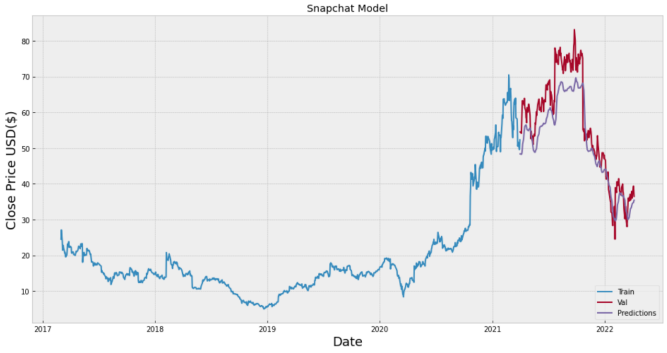

Fig. 8. Snapchat Model.

price, but the model was able to follow the trends of the real stock price, which was the goal of the model. This can be seen particularly well in both Twitter and Snapchat's graphs, where the predictions will be under the actual values, but still able to follow the direction that the price is moving. The model was unable to follow every small change in trend, but it did a good job of recovering and continuing to stay near the actual values.

Analysis

We were able to create a model that predicted trends and remained fairly accurate to the actual values of the stock prices. Using an LSTM model, we found that using only one epoch produced the strongest results with the lowest values of MAPE and RMSE. When using MAPE, the results were close to zero for all of the stocks, but were best for Facebook due to its higher stock price point. This shows that the model was working really well for the prediction of these stock prices. For RMSE, the values were also close to zero for all of the stocks and more similar, which suggested that our model continued to work equally well for the three different stocks.

The predicted values were always able to stay within about 10-20 units of the actual values and trends were almost always followed by these predictions. There were instances where there was a larger difference between the predicted values and actual values, but these were often corrected over time to continue to produce strong results.

The LSTM model works really well for time series forecasting of overall trends and patterns in data, but should not be used for finding exact values within these series.

V. Conclusion and Future works

The stock market is something that has a large number of variables that impact how people choose to buy and sell shares, which makes it important to be able to try to look ahead to the future when your money is involved. There are many factors that make forecasting stock prices an extremely difficult task. Utilizing machine learning to forecast stock prices can help many people, as it is something that is such a large part of our daily lives and can have a huge impact on what we do with our money. Rather than focus on attempting to predict the exact values of a closing stock price on a given day, it is more beneficial to focus on the general trends and directions of the prices. While this still might not make up for the unpredictability of the outside factors, it could help give people an idea of what the future might hold for the stocks they are interested in purchasing shares of.

We were able to create a model that utilized an LSTM to predict the trends of a stock's price over a period of time after training it with data from over a couple of years. While this situation proved to create accurate results, it might lead to less accurate predictions after moving farther away from the dates of the training data.

In the future, we might try to implement a way to predict further into the future by periodically retraining the model to produce better results. However, within this project we simply wanted to focus on creating a model that was able to predict stock prices within some level of accuracy for a smaller amount of time. When attempting to go further into the future, we found that the trends were not always picked up or it focused too much on one single trend and was unable to recover.

In the future, we might also try to implement time series decomposition to allow our model to think about additional components. Since there are so many factors involved in forecasting the stock market, it might be beneficial to implement components such as level, trend, seasonality, and noise. It could help produce better predictions and prevent some of our problems with trends not being noticed by the model. However, we felt that our LSTM model was producing the proper results that we were aiming for within the span of this project.

Overall, we felt that our LSTM model worked really well for forecasting stock price trends within a short period of time. Using machine learning as a tool for time series forecasting is really useful for the stock market, even if the prices are not always incredibly accurate. The ability for a model to predict stock price trends showcases the versatility of machine learning and the wealth of possibilities for utilizing technology to analyze and make predictions from data.

REFERENCES

[1] https://www.simplilearn.com/tutorials/machine-learning-tutorial/stock-price-prediction-using-machine-learning: :text=Stock%20Price%20Prediction%20using%20machine%20learning%20helps%20you%20discover%20the,is%20to%20gain%20significant%20profits.

[2] https://www.hindawi.com/journals/complexity/2021/5360828/

[3] https://neptune.ai/blog/predicting-stock-prices-using-machine-learning

[4] https://corporatefinanceinstitute.com/resources/knowledge/other/machine-learning-in-finance/

[5] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[6] https://www.youtube.com/watch?v=LfnrRPFhkuYab_channel=codebasics

[7] https://www.youtube.com/watch?v=QIUxPv5PJOYab_channel=ComputerScience

[8] https://drive.google.com/file/d/1qofr9pW9_v7pTf4CmcaXGTghhjhO6-pY/view?usp=sharing

[9] https://colab.research.google.com/github/knightow/mltraining/blob/master/Stock_Price_Prediction_Using_Python_%26_Machine_Learning.ipynbscrollTo=hl24K-e79ajt