Logo

Description automatically generated

4/30/2021

Shivi Bhatt  
Dharmik Harkhani

DATA SCIENCE ENGINEERING METHODS AND TOOLS

Stock PRICE DATA Analysis & PREDICTION

Contents

[Topic: 3](#_Toc69671701)

[Course: 3](#_Toc69671702)

[Professor: 3](#_Toc69671703)

[Group Members: 3](#_Toc69671704)

[Abstract: 3](#_Toc69671705)

[Introduction: 4](#_Toc69671706)

[Background: 4](#_Toc69671707)

[Motivation: 4](#_Toc69671708)

[Goal: 4](#_Toc69671709)

[Dataset Description: 5](#_Toc69671710)

[Software and Library Used In this Project: 5](#_Toc69671711)

[Feature Selection: 6](#_Toc69671712)

[Data Wrangling: 6](#_Toc69671713)

[Exploratory Data Analysis: 6](#_Toc69671714)

[Model Used: 6](#_Toc69671715)

[Model Descriptions: 6](#_Toc69671716)

[Linear Regression: 6](#_Toc69671717)

[RegressorGRU (Gated Recurrent Unit): 7](#_Toc69671718)

[Decision Tree Regression: 7](#_Toc69671719)

[Recurrent Neural Networks: 7](#_Toc69671720)

[Long Short-Term Networks (LSTM): 8](#_Toc69671721)

[Visualization: 8](#_Toc69671722)

[Dataset Details: 9](#_Toc69671723)

[Dataset Source: 10](#_Toc69671724)

[Data Analysis: 11](#_Toc69671725)

[Missing values Treatment: 11](#_Toc69671726)

[Exploratory Data Analysis: 13](#_Toc69671727)

[Visualizations: 14](#_Toc69671728)

[Predict the Stock Price: 17](#_Toc69671729)

[Linear Regression: 18](#_Toc69671730)

[Regressor GRU(Gated Recurrent Unit): 19](#_Toc69671731)

[Decision Tree Regressor: 23](#_Toc69671732)

[LSTM ( Long Short Term Memory) a Recurrent Neural Network 28](#_Toc69671733)

[Conclusion: 32](#_Toc69671734)

[References: 32](#_Toc69671735)

# Topic:

* Stock Price Data Analysis and Prediction

# Course:

* Data Science Engineering Methods and Tools

# Professor:

* Dr. Liu Handan

# Group Members:

* Dharmik Harkhani (1029591)
* Shivi Bhatt (1027605)

# Abstract:

This report consists of the final findings of the Supervised Machine Learning Algorithms used for the prediction of the stock market prices from the historical stock market data set from year 1970 to 2018 in the Unites States of America.

The dataset used in the project has been taken from the Kaggle. The data is then cleaned as per the business requirement and then various models were utilized in Jupyter Notebook to see which model is the good fit for accurate predictions.

After studying the dataset and understanding the data models, it became evidently clear that the problem statement to predict stock prices fall under the time-series and regression. The study focuses on the various algorithms by using regression such as Linear regression, decision tree regression, Regressor Gated Recurrent Unit and time series such as Recurrent Neural Network (LSTM= Long Short-Term Memory method).

The major finding is that the machine learning approach should be suitable for this problem due to many aspects.

# 

# Introduction:

Background:Stock price data analysis and prediction aims to predict the close price for the one of the top companies from the data set we have taken example of NYL& AAPL, the dataset is available for 1970 to 2018, for predicting the score we filtered the data to 2017 year. This system as ‘Close’ and ‘Adjusted Close Price’ as the target variable and its predictions are based on variety of intrinsic and contextual attributes such as ticker (company name), date, adj\_close, open, high and volume.

Motivation:Stock price prediction is a classic and important problem. With a successful model for stock prediction, we can gain insight about market behavior over time, spotting trends that would otherwise not have been noticed. With the increasingly computational power of the computer, machine learning will be an efficient method to solve this problem.

The motivated idea is that, if we know all information about today’s stock trading (of all specific traders), the price is predictable. Thus, if we can obtain an information from previous trends, we can expect to improve the current prediction lot.

Thus, our motivation is to design a system model incorporating historical data and price predictions to make a stronger model that will benefit everyone.

Goal:In the current emerging competitive market, predicting the stock prices as well as the company's financial status in advance will provide more benefits for the investors in order to invest confidently. Stock prediction can be done by using the current and previous data available on the market.

We wish to present this system model to predict the stock price trends for a company, by providing deep analysis on the close price in simpler visualizations formats for quick and easy understanding. Our goal is to get an outcome   
- demonstrate NLY company stock price prediction as an example to satisfy the customer  
- Visualize various stock parameters such close and open price, high and volume  
- Predict Adjusting closing price using LSTM

- Predict stock price for AAPL using LSTM

# Dataset Description:

This dataset has the daily stock prices for a selection of several thousand stock tickers from NYSE and NASDAQ. This is historical stocks data from 1970 to 2018, this dataset has 20.97 million records (2 GB size), so model is applied using below two files  
- historical stocks  
- historical stock prices

# Software and Library Used In this Project:

The data set is obtained from the Kaggle, reference will be available at the end of the report, we have used Jupyter Notebook and often times Google Colab , below are the libraries which is utilized in this project

* Numpy
* Pandas
* Matplotlib
* Seaborn
* Metrics
* Mean Squared Error
* Sklearn
* ScikitLearn Tree
* Train\_test\_split
* Standard Scalar
* Sklearn.ensemble
* sklearn.preprocessing
* MinMaxScaler
* linear\_model
* Linear Regression
* Keras
* Sequential
* Dense
* Dropout
* GRU
* Bidirectional
* LSTM
* keras.optimizers – SGD
* math
* sklearn.tree
* DecisionTreeRegressor
* Tensorflow

# Feature Selection:

The data originally has less columns so there was not so much noise in the database, however, there were NaN values which needs to be cleaned and removed, the data model is built on data frame with 6 columns.

# Data Wrangling:

Data provided was raw and has many missing values as the data is from the year 1970 to 2018. Therefore, many rows had NaN values which would not give accuracy when built a model, so we have removed those NAN values from our dataset.

# Exploratory Data Analysis:

After data wrangling and sorting the data, we have shown top 10 companies, top 10 industries from the data using matlplot lib, we have also shown what all sectors are there for different companies.

# Model Used:

Target variable we have used is ‘close’ and adj\_close, both are the stock price of the ticker(company)  
Below are the different types of Models utilized for the project are

Linear Regression

* RegressorGRU(Gated Recurrent Unit)
* Decision Tree Regression
* Recurrent Neural Network (LSTM = Long Short-Term Memory method)

# Model Descriptions:

## Linear Regression:

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.  
Linear Regression fits a linear model with coefficients w = (w1, …, wp) to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation.

RegressorGRU (Gated Recurrent Unit):In simple words, the GRU unit does not have to use a memory unit to control the flow of information like the LSTM unit. It can directly make use of all hidden states without any control. GRUs have fewer parameters and thus may train a bit faster or need less data to generalize. But, with large data, the LSTMs with higher expressiveness may lead to better results.

They are almost similar to LSTMs except that they have two gates: reset gate and update gate. Reset gate determines how to combine new input to previous memory and update gate determines how much of the previous state to keep. Update gate in GRU is what input gate and forget gate were in LSTM. We don't have the second nonlinearity in GRU before calculating the output, neither they have the output gate.

## Decision Tree Regression:

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.

A 1D regression with decision tree. The [decision trees](https://scikit-learn.org/stable/modules/tree.html#tree) is used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve

## Recurrent Neural Networks:

In a recurrent neural network, we store the output activations from one or more of the layers of the network. Often these are hidden later activations. Then, the next time we feed an input example to the network; we include the previously stored outputs as additional inputs. You can think of the additional inputs as being concatenated to the end of the “normal” inputs to the previous layer. For example, if a hidden layer had 10 regular input nodes and 128 hidden nodes in the layer, then it would actually have 138 total inputs (assuming you are feeding the layer’s outputs into itself à la Elman) rather than into another layer). Of course, the very first time you try to compute the output of the network you’ll need to fill in those extra 128 inputs with 0s or something.

A picture containing text, clock, sign

Description automatically generated

Now, even though RNNs are quite powerful, they suffer from Vanishing gradient problem which hinders them from using long term information, like they are good for storing memory 3-4 instances of past iterations, but larger number of instances don't provide good results, so we don't just use regular RNNs. Instead, we use a better variation of RNNs: **Long Short-Term Networks (LSTM).**

Long Short-Term Networks (LSTM):Long short-term memory (LSTM) units (or blocks) are a building unit for layers of a recurrent neural network (RNN). A RNN composed of LSTM units is often called an LSTM network. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Each of the three gates can be thought of as a "conventional" artificial neuron, as in a multi-layer (or feedforward) neural network: that is, they compute an activation (using an activation function) of a weighted sum. Intuitively, they can be thought as regulators of the flow of values that goes through the connections of the LSTM; hence the denotation "gate". There are connections between these gates and the cell.

The expression long short-term refers to the fact that LSTM is a model for the short-term memory which can last for a long period of time. An LSTM is well-suited to classify, process and predict time series given time lags of unknown size and duration between important events. LSTMs were developed to deal with the exploding and vanishing gradient problem when training traditional RNNs.

Diagram

Description automatically generated

# Visualization:

The visualizations in the project we have done in exploratory data analysis as well as while predicting the values by comparing it from actual values. We have utilized matplotlib, plotly graph for visualizing data using bar graph, zigzag graphs, line graph, pie charts etc.

# Dataset Details:

Above dataset contains two files:

* historical\_stocks: it has list of all the stocks and data of all the companies.
* historical\_stocks\_prices: It has the list of prices of all the stocks from 1970 to 2018.

The data set has 8 columns from historical stocks prices file while 6 columns from historical stocks, the dataset size is 2 GB  
 the data set has the following data types:  
String, date, decimal and others

List of Attributes from the file historical\_stock

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Attributes | Description | Nullable |
| 1 | ticker | This is the name of the company share | No |
| 2 | exchange | Indicates the type of exchange made | No |
| 3 | name | refers the company's name | No |
| 4 | Sector | refers to the actual sector where the given company operates | Yes |
| 5 | Industry | specifies the type of services that can be provided | Yes |

List of Attributes from the file historical\_stock\_price

|  |  |  |  |
| --- | --- | --- | --- |
| S.no | Attributes | Description | Nullable |
| 1 | ticker | This is the name of the company share | No |
| 2 | open | describe the open price of that share in a specific day | No |
| 3 | close | describe the final share price in the end of a day | No |
| 4 | adj-close | it´s a tricky column, describes the adjusted price of a share, thats normally different from the close price | No |
| 5 | low | the lowest value paid for that share | No |
| 6 | high | the highest value paid for that share | No |
| 7 | volume | # of shares purchased in that day | No |
| 8 | date | represents the date (year-month-day) | No |

# Dataset Source:

We have used below dataset from Kaggle:

Daily Historical Stock Prices (1970 - 2018) Historical stock prices for several thousand unique stock tickers (20.97 million records – 2GB data)

<https://www.kaggle.com/ehallmar/daily-historical-stock-prices-1970-2018>

# Data Analysis:

Since, we have a huge data set of 20 million, we extracted 1.29 million of data and did exploratory data analysis so that we can predict the model accurately. When analyzed we have found that there were null values in one of the files and we removed those NaN as well.

## Missing values Treatment:

**Table

Description automatically generated**

**After Removal:**

**Graphical user interface, text, application

Description automatically generated**

## Exploratory Data Analysis:

Visualizing NASDQ and NYSE stocks count  
Graphical user interface, application

Description automatically generated

## Visualizations:

Finance Sectors

Graphical user interface

Description automatically generated

Total Sectors:  
Graphical user interface, text, application, email

Description automatically generated

Total Industry count:

Table

Description automatically generated

Sectors sort by ticker exchange  
Chart

Description automatically generated

Top 10 Industry Visualization using bar graph and Pie Chart:  
Graphical user interface, application

Description automatically generated

Top 10 industry sorted by ticker - exchange

Chart

Description automatically generated

Distribution of NLY company stock data:  
Graphical user interface, application

Description automatically generated

Normalizing and Splitting the data:  
Graphical user interface, application

Description automatically generated

# Predict the Stock Price:

Training and Testing data:  
Graphical user interface

Description automatically generated with low confidence

Feature Scaling – Standardization:  


Linear Regression:  
Graphical user interface, text, application, email

Description automatically generated

Text, letter

Description automatically generated

Visualized the actual and predicted score:  
Chart, bar chart

Description automatically generated

Metric:  
Graphical user interface, text, application, email

Description automatically generated

### Regressor GRU(Gated Recurrent Unit):

Importing various libs:  
Graphical user interface, text, application, email

Description automatically generated

Training Data:  
Graphical user interface, text, application, email

Description automatically generated

Defining Layers and Training Data:  
Graphical user interface, text, application, email

Description automatically generated

Result of NLY training:  
A picture containing text, document, screenshot

Description automatically generated

Plotting the Result of loss  
Chart, line chart

Description automatically generated with medium confidence

Training, Testing method and Predictions:  
Graphical user interface, text, application

Description automatically generated

Visualizing Actual and Predicted value:  
Graphical user interface

Description automatically generated with medium confidence

Test Method 2:  
Graphical user interface, text, application, email

Description automatically generated

Comparison between two Methods: Visualized through graph  
Graphical user interface, application

Description automatically generated

Decision Tree Regressor:Exploratory Data Analysis for Decision Tree Regressor: **Table

Description automatically generated**

Visualize opening prices for all Tickers:  
Chart

Description automatically generated with medium confidence

Visualize high price for tickers:  
Chart

Description automatically generated

Visualize low price for tickers:  
Chart

Description automatically generated

Visualize close price for tickers:  
A picture containing graphical user interface

Description automatically generated

Prediction for 100 futureDays:  
Graphical user interface, text, application, email

Description automatically generated

Train, Test, fit and Compare both models:  
Graphical user interface, text, application

Description automatically generated

Predict Future:  
Table

Description automatically generated with medium confidence

Prediction values:  
A computer screen capture

Description automatically generated with low confidence

Visualized:

A picture containing graphical user interface

Description automatically generated

LSTM ( Long Short Term Memory) a Recurrent Neural Network  
Predict Stock Price for AAPL

Load Data and Train data set based on target value: adj\_close:  
Graphical user interface, text, application, email

Description automatically generated

Using Tensor flow create layers:  
Graphical user interface, text, application, email

Description automatically generated

Fit Model and visualize the Model Loss:  
Graphical user interface, table

Description automatically generated

Train, Test, Model

Graphical user interface, text, application, email

Description automatically generated

Predicting Adj Close value and Visualize them:  
Graphical user interface, text, application

Description automatically generated

Calculate Metric:  
Graphical user interface, application

Description automatically generated

# Conclusion:

With Accuracy Score of all the used algorithms, RegressorGRU (Gated Recurrent Unit) produced significant score of accuracy for predicting the Target variable (close)

With Accuracy Score of all the used algorithms, LSTM produced significant score of accuracy for predicting the Target variable (adj close)

# References:

1. <https://www.kaggle.com/thebrownviking20/intro-to-recurrent-neural-networks-lstm-gru>
2. <http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm>
3. <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html>
4. <https://en.wikipedia.org/wiki/Gated_recurrent_unit>
5. <https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html>
6. <https://www.kaggle.com/guilhermeviveiros/predicting-stock-prices>
7. <https://www.kaggle.com/rishidamarla/stock-market-prediction-using-decision-tree>
8. <https://www.kaggle.com/piseykorn/kernel27b53324dd>
9. <https://www.kaggle.com/maimom/predict-adjusted-close-using-lstm>
10. https://www.saedsayad.com/decision\_tree\_reg.htm#:~:text=Decision%20tree%20builds%20regression%20or,decision%20nodes%20and%20leaf%20nodes.
11. https://www.researchgate.net/publication/313408030\_A\_Review\_on\_Prediction\_of\_Stock\_Market\_using\_Various\_Methods\_in\_the\_Field\_of\_Data\_Mining#:~:text=Objectives%3A%20In%20the%20current%20emerging,data%20available%20on%20the%20market.