SOLFINTECH STock Prediction WEB-APP

Q14518368: PANASHE MANO

Deployed Application: <https://manopanashe-solfintech-artifecttradesenseapp-5tuagu.streamlit.app/>

Table of Contents

[**Introduction** 2](#_Toc124415153)

[1.1 Machine Learning in the stock Market 2](#_Toc124415154)

[1.2 Problem Definition 2](#_Toc124415155)

[1.3 Value Proposition 2](#_Toc124415156)

[1.4 Success Metrics 2](#_Toc124415157)

[2 Data Collection and Describing 3](#_Toc124415158)

[2.1Dataset Variable Types 4](#_Toc124415159)

[2.2 Dataset Preparation for EDA 5](#_Toc124415160)

[2.2.1 Loading Data 5](#_Toc124415161)

[2.1.2 Data variables 5](#_Toc124415162)

[2.1.3 Null Values 6](#_Toc124415163)

[2.1.4 Data Statistics 6](#_Toc124415164)

[3 Data visualisation and exploration 7](#_Toc124415165)

[3.1 Data Correlation 7](#_Toc124415166)

[3.3 Skewness Check 8](#_Toc124415167)

[3.4 Data Visualisation 8](#_Toc124415168)

[3.4.1 Univariate Analysis 8](#_Toc124415169)

[3.4.2 Time Series Analysis 9](#_Toc124415170)

[4 Data Modelling and Visualisation 10](#_Toc124415171)

[4.1 Data Pre-processing 11](#_Toc124415172)

[4.1.1 Data Scaling 11](#_Toc124415173)

[4.1.2 Splitting our data 11](#_Toc124415174)

[4.2 Model Fitting and Construction 12](#_Toc124415175)

[4.3 Model Evaluation 12](#_Toc124415176)

[4.3 Model Visualisation 13](#_Toc124415177)

[5 System Overview 14](#_Toc124415178)

[5.1 User Features 14](#_Toc124415179)

[5.1.2. User Selection 14](#_Toc124415180)

[5.1.3 Data Display 15](#_Toc124415181)

[5.1.4 Registered Users Additional features 16](#_Toc124415182)

[6 Limitations and Challenges 16](#_Toc124415183)

[Conclusion 17](#_Toc124415184)

[References 17](#_Toc124415185)

[**Appendix** 18](#_Toc124415186)

# **Introduction**

## Machine Learning in the stock Market

Over the last few decades there has been an explosive increase in the average person’s interest for stock market (Umer, 2019). In a financially explosive market as the stock market, traders need fast and accurate information in order to make effective decisions on buying or selling stocks. This has made stock prediction applications very valuable in the trader’s world as it gives them a view of future potential profits or losses. Machine Learning techniques have increasingly been utilized in stock prediction as they utilize historical stock data as input to predict new output values. However, market changes and the behaviour of stock prices is uncertain and makes the prediction of future stock prices a very challenging task.

## 1.2 Problem Definition

SOLFINTECH is a leading financial multinational organisation that deals with stocks , shares, saving and investments. The organisation operates an online investment platform which accommodates over 50 million subscribers with over 150b pounds worth of investment. It is a secure organisation that trades on multiple stock exchange platforms . However, as their customer base continues to expand and with new competitors in the industry , the company requires new innovative ways to keep ahead of its competitors.

## 1.3 Value Proposition

To solve this issue this study aims to create a proof-of-concept web application that predicts stock prices. As a result of the advancements in Machine learning over the past few years, we have seen many algorithms that have been deployed for stock price prediction. Like this study similar analysis has been made on ml algorithms for stock prediction such as (Malti Bansal, 2022). The study conducted a comparison analysis on various ml models to see which one provided the highest level of accuracy. It concluded that amongst the K-Nearest Neighbours, Linear Regression, Support Vector Regression, Decision Tree Regression, and Long Short-Term Memory algorithms, the Long Short-Term Memory algorithm provided the best results during stock price prediction.

Another study (Wei Chen, 2022) also conducted a similar comparison analysis on machine learning methods (random forest (RF), support vector regression (SVR), long short-term memory networks (LSTM) ) . However, this study concluded that the Random Forest model provided the best efficient prediction and highest accuracy results. Therefore in-order to ensure that we are using the best model for the application extensive research will be conducted on the Random Forest Model and the Long Short-Term Memory Model.

## 1.4 Success Metrics

The success of our application will be evaluated by achieving meaningful prediction of stock prices that can help the company’s customers make informed trading and investment choices in the future. It will also be evaluated on the application’s usability and user friendliness as the application should make stock trading much easier for the company’s customers.

# 2 Data Collection and Describing

Machine learning models learn patterns and trends through historical numerical data, therefore, to obtain data for our model we collected a list of the standard and poorest companies on the stock market. As seen in Figure 1 the list contains the company’s information such as industry, sector, location etc.



A screenshot of a computer

Description automatically generated with medium confidence

Figure : Companies Information

The application will allow the users to select the company they would like to view the Stock prediction for as seen in Figure 2.

Background pattern

Description automatically generated

Figure : User Company Selection

The stock information for the companies on the list will then be retrieved by Yahoo finance’s API open-source tool called yfinance. The yfinance api then uses the name of the selected company from our list to download stock data from the preferred year range, we collected stock data from 2016 till present day for our application which was then saved to a CSV file.

Table

Description automatically generated

Figure : Downloaded Stock Data

## 2.1Dataset Variable Types

|  |  |  |  |
| --- | --- | --- | --- |
| Field Variable | Field Description | Type of variable | Data Type |
| Date | Date of stock information | Continuous | Time |
| Open | The price at which the financial security opens in the market | Independent | Numerical |
| High | Highest Trading price for that day | Independent | Numerical |
| Low | Lowest Trading Price for that day | Independent | Numerical |
| Close | Last price at the end of trading session | Independent | Numerical |
| Adj Close | The price of stock after paying off the dividends | Independent | Numerical |
| Volume | The number of shares traded in a stock | Independent | Numerical |

Figure : Dataset features

## 2.2 Dataset Preparation for EDA

After retrieving the stock data for the selected company, our next step was exploring the data quality using various exploration analysis and methods.

### 2.2.1 Loading Data

To perform analysis and prepare our dataset for the ml algorithm , we used jupyter notebook to visualise and explore key features within the dataset. To ensure that the data had been downloaded correctly into the csv file we used the pandas functions .shape() and .head().

Graphical user interface, text

Description automatically generated

Figure : Data description

### 2.1.2 Data variables

Machine learning models learn patterns from numerical data therefore to ensure our data variables with the exclusion of the date column, were numerical data types we used the pandas .dtypes() function which returns a series with the datatype of each column.

Text

Description automatically generated

Figure : Data Variables

### 2.1.3 Null Values

Our next step was checking for any null values in our dataset and as seen in Figure 7 there are no null values in our data.

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Null Values check

### 2.1.4 Data Statistics

To further understand our data, we retrieved basic statistical data for our dataset using the pandas .describe() function which calculates statistical data like percentile, mean and std for the numerical values of our data as seen in Figure 8.

Graphical user interface, text, application

Description automatically generated

Figure : Data description

# 3 Data visualisation and exploration

Data exploration allows us to identify any patterns , trends or outliers within our data. It gives us insights on the key characteristics of our dataset which will aid in selecting he appropriate model for our application.

## 3.1 Data Correlation

To understand the relationship of the variables within our dataset, we performed a correlation analysis which shows the relationship between the variables in terms of positive and negative correlations which we then visualised using a heatmap graph from the seaborn library.

Text

Description automatically generated

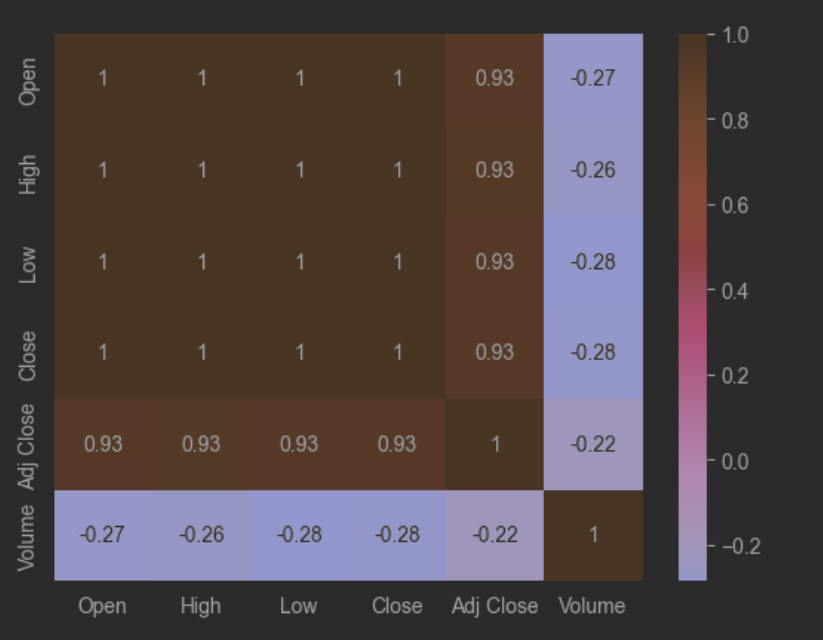
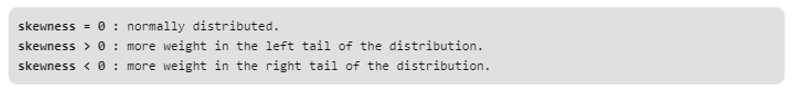


Figure : Correlation Analysis

## 3.3 Skewness Check

Figure 9 shows us that there is strong relationship between the columns Open, Close , High and Low in our dataset. Our next Step was to understand the distribution within our variables. To achieve this we began by calculating the skewness of our data distribution. Skewness divides the distribution into 3 section which indicate the weight of our data distribution.



Text

Description automatically generated

Figure : Skewness Check

The skewness score show us that the distribution in our data set is mostly Right Skewed with more weight on the left tail of the distribution.

## 3.4 Data Visualisation

Visualising the patterns and trends within our dataset is also very crucial step in data pre-processing. Therefore, to visualise the significant patterns and trends within our data we applied various visualization analysis and techniques using various libraries.

### 3.4.1 Univariate Analysis

Univariate analysis is the most basic form of statistical data analysis. It is conducted through several descriptive ways such as Frequency Distribution Tables, Histograms, Pie Charts and Bar Charts. It explores one variable at a time, and we can represent a variable’s distribution using mean, median or mode. To visualise the distribution of data we used matplotlib’s .hist() function which displays individual histograms for each column in our dataset.

Chart, histogram

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Data distribution

As we can see in Figure 11 the univariate analysis supports the results calculated by the skewness check.

### 3.4.2 Time Series Analysis

Once we explored the distribution of our data, the next step was understanding the how the variables perform over time. Time series analysis is very useful in the stock market as it can help understand the cause of trends and systematic pattern over time. We visualised our analysis with matplotlib’s line graphs. The rest of time series analysis for the rest of our data can be found in our **Appendix**.

Chart, line chart

Description automatically generated

Figure 12: Open Column Time series Analysis

# 4 Data Modelling and Visualisation

After Exploring the trends and patterns in our dataset , Our next step was conducting a comparative analysis of the two candidate models for our application. The analysis will assist us in making an informed decision for which model to use for our application and which model is the most appropriate for our data set.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Definition | Pro’s | Con’s |
| Random Forest Model | The random forest algorithm is made up of a collection of decision trees. It utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. | * Reduced risk of overfitting: Decision trees run the risk of overfitting as they tend to tightly fit all the samples within training data * Easy to determine feature importance: Random Forest makes it easy to evaluate variable importance, or contribution, to the model. | * Time-consuming process: Since random forest algorithms can handle large data sets, they can provide more accurate predictions, but can be slow to process data as they are computing data for each individual decision tree. * Requires more resources: Since random forests process larger data sets, they’ll require more resources to store that data |
| Long Short-Term Memory Model | It is used in the field of Deep Learning. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems | * Built to handle challenging tasks * Is the most powerful RNN for forecasting as it the LSTM cell adds long-term memory which allows even more parameters to be learned by the model * The model predicts future stock prices based on sequential data which provides greater accuracy. | * They require a lot of resources and time to get trained and become ready for real-world applications * LSTMs are prone to overfitting and it is difficult to apply the dropout algorithm to curb this issue |

Overall, the LSTM model is more powerful compared to the random forest , both models require a lot of resource and time however for time series forecasting the Long Short Term Memory Neural Network model (LSTM) is superior. Therefore, we decided to use the LSTM model for our application.

## 4.1 Data Pre-processing

The accuracy of our model is significantly dependent on the data. Therefore, preparing our data for the model is a crucial step. It ensures that our data is in the right format and has the appropriate scale with meaningful features to solve our problem.

### 4.1.1 Data Scaling

Our dataset contains numerical data therefore it is in the correct format, the next step was ensuring that the data was in the appropriate scale. We began by creating a copy of the dataset with our independent and dependent variables which were then scaled with the sklearn MinMaxScaler.

Text

Description automatically generated

Figure : Scaling our data

### 4.1.2 Splitting our data

To avoid overfitting or model , we split the data into train and test data.

Text

Description automatically generated

Figure : Splitting the data

Once the data was split we then created the sequence input layers for the test and train data , which will be the input our data used to train and train our model.



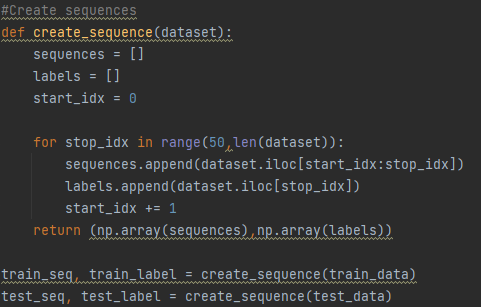


Figure : Sequences

## 4.2 Model Fitting and Construction

Once our data was split into train and test data, the next step was creating our model. We imported the Sequential Model from the karas library and other necessary packages.

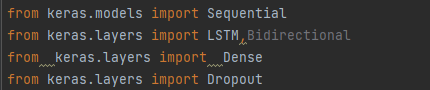


Figure : Importing necessary libraries

Once we had imported our packages , the next step was setting the necessary parameters for our model. Each LSTM model requires

* The number of units which are the dimensions of the inner cells in LSTM
* The training sequence
* A dropout layer which helps prevent overfitting
* The number of layers for our model which is set by Dense
* The evaluation metrics which our model will use to calculate it’s accuracy
* The optimizer which will reduce the overall loss and improve the accuracy of our model .

Figure 18 Shows us the summary of these parameters.

Text

Description automatically generated

Figure : Our Model

Text

Description automatically generated with medium confidence

Figure : Model Summary

## 4.3 Model Evaluation

After training our model with our train data , we then used the test data for testing the prediction of our model. LSTM models evaluate their absolute error and accuracy score on each epoch. Our highest validated accuracy score was 55% and a validated loss of 13%.

A picture containing text

Description automatically generated

Figure 19: Model Evaluation

## 4.3 Model Visualisation

To better understand our predictions , we used various visualisation techniques to see the predicted data and the actual values . However, as we had scaled our data to train our model, we first needed to inverse the transformation of our data to their original values. Once the data was reverted back to its original value we then merged the predicted data to our original dataset.

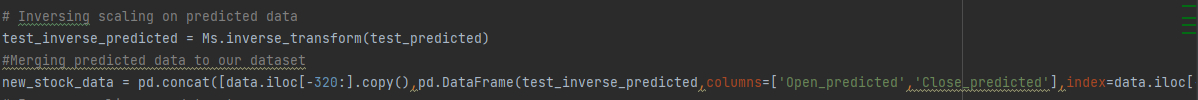


Figure 20: Merging datasets

Once our datasets were merged we used matplotlib’s line graph to visualise the predicted values against the actual values. As we can see in Figure 21, the graph shows us that our model prediction is very close to the actual values.

Chart, histogram

Description automatically generated

Figure : Predicted Values

# 5 System Overview

Once the prediction model was built and tested we then proceeded to develop the user interface for the customers to interact and view the predictions made by the model. The application was developed using Stream lit which is an open-source app framework in python language. To meet the requirements of the company for the application the following features were implemented.

## 5.1 User Features

### 5.1.2. User Selection

The first requirement of the application was for the system to anticipate stock prices on a Daily/weekly/Monthly Basis. This Requirement was met by the development of our model ,however, to allow users to specify the choice of stock for prediction and the period of time for viewing the stock’s historical data we created a navigation Menu to enable users to make these choices.

Graphical user interface, application, Teams

Description automatically generated

Figure 22: Navigation Menu

### 5.1.3 Data Display

The selection made by the user will then be handled by the system server which will return the historical data of the company they have selected and will be displayed by a table and lineplot that shows a time series analysis of the closing price over their prefered period .

Chart

Description automatically generated

Figure 23: Raw stock data

The application will then display the forecasted price for the stock that the user will have selected and display the forecast of the upcoming week on a line plot graph.

Text

Description automatically generated with low confidenceChart, line chart, scatter chart

Description automatically generated

Figure 24: Forecasted Closing price for next day

### 5.1.4 Registered Users Additional features

Another requirement of the system was to allow registered users to be able to view further prediction or analysis to help them make more informed decisions, therefore, to meet this requirement registered users will be able to view the forecasted Opening price for the week ahead.

Graphical user interface, application

Description automatically generated

Chart

Description automatically generated

Figure 25: forecasted Open price for week ahead

# 6 Limitations and Challenges

LSTM models are among the best performing models for Time series forecasting; however, they take longer to train . This impacts the load time for our application which will cause some users to lose interest in the application and losing the company a large number of potential customers. To partially remedy this issue there is a load time notification on the page letting users know we are calculating their predictions, therefore users may remain interested in the application as they anticipate what the result will be . Another limitation with our model is the accuracy level of prediction , the highest accuracy score our model achieved was 55%, this may have been due to over fitting or the number of units in our model therefor to remedy this issue, we added more dropouts to our model and increased the number of epochs which raised our accuracy score to 69%.

# Conclusion

Overall Machine learning models have proved they are capable of carrying out prediction of the complex and constantly changing stock market prices. LSTM models have also proved to be capable of handling complex stock market data as its ability to process information, forget it, then remember it again puts it ahead of Random Forest Models. However, as the data is very complex, this also increases the complexity of the machine learning model which can result in high computational times and slow predictions.

The stock market is a fast-paced environment , therefore slow ml models in the context of stock prediction can be viewed as unreliable to a certain extent. The development of this application has enabled me to gain understanding of the challenges and learning problems in the context of data models. Challenges we encountered in the data pre-processing stage around scaling the data for the model required extensive amount of research before a solution was found. Overall, the application will be of high value to the average person interested in the stock market as its user-friendly features makes it easy to understand stock market prices. However, as the model run time is quite slow users may lose interest in the application, therefore, more time efficient models will be required for the application to be able to attract and retain users.

# References

Barone, A. (2022). Opening Price: Definition, Example, Trading Strategies. *Investopedia*.

Hayes, A. (2021). What Is Closing Price? Definition, How It's Used, and Example. *Investopedia*.

Malti Bansal, A. G. (2022). *Stock Market Prediction with High Accuracy using Machine Learning Techniques,.*

Nath, T. (2021). How Big Data Has Changed Finance. *Investopedia*, 1.

Umer, M. (2019). *Stock Market Prediction Using Machine Learning(ML)Algorithms.*

Wei Chen, H. Z. (2022). *A novel two-stage method for well-diversified portfolio construction based on stock return prediction using machine learning,.*

# **Appendix**

Chart

Description automatically generated

Figure : High Column Time series Analysis

Chart, line chart

Description automatically generated

Figure :Low Column Time series Analysis

Chart

Description automatically generated

Figure : Close Column Time series Analysis

Chart, line chart

Description automatically generated

Figure :Adj Close Column Time series Analysis

Graphical user interface, chart

Description automatically generated

Figure : Volume Column Time series Analysis