A Mini Project Report

On

"Stock Price Prediction"

Of

TE(Computer Engineering)
(Academic Year :2023-24)

SUBMITTED BY

Sarthak Niranjan Kulkarni
Roll. No (T211075)

Guided By

Prof. Yogita Pore

Department of Computer Engineering Zeal Education Society's

Zeal College Of Engineering and Research, Narhe, Pune 411041

<u>CERTIFICATE</u>

This is to certify that, Sarthak Niranjan Kulkarni (T211075) of class T.E COMP; have successfully completed their mini project work on "**Stock Price Prediction**" at **ZEAL College of Engineering and Research, Pune** the partial fulfillment of the Graduate Degree course in T.E at the department of **<u>Computer Engineering</u>**, in the academic Year 2023- 2024 Semester –VI as prescribed by the **Savitribai Phule Pune University.**

(Prof. Yogita Pore)

Guide

Computer Engineering Dept

(Prof. A. V. Mote)

Head

Computer Engineering Dept

Place: Pune

Date:

Acknowledgements

We take this opportunity to thank our project guide **Prof. Yogita Pore** and Head of the department **Prof.A. V. Mote** for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of Computer Engineering the Department for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

Sarthak Niranjan Kulkarni
(T211075)

ABSTRACT

This project aims to predict the closing price of a stock using a TensorFlow model. The historical price data is retrieved from Yahoo Finance using the vfinance package in Python. The data is preprocessed by scaling and splitting into training and test sets. A sequential model with multiple layers is trained on the training data, and the performance is evaluated on the test set using mean squared error (MSE) and root mean squared error (RMSE) metrics. The model is then used to predict the closing prices for a given period in the future. The results indicate that the model performs well in predicting stock prices and could be used as a reliable tool for forecasting.

INDEX:

INTRODUCTION

Stock price prediction is an essential area of research in financial analysis, which involves forecasting the future prices of stocks based on historical data. With the rapid growth of the stock market, it is imperative to have accurate and reliable methods of predicting stock prices. Accurate predictions of stock prices can help investors make informed decisions on buying and selling stocks, resulting in significant profits.

In recent years, advances in machine learning and artificial intelligence have led to the development of sophisticated algorithms capable of predicting stock prices with high accuracy. One such approach is the use of neural networks, a class of machine learning algorithms that can learn complex patterns in data.

In this project, we aim to predict the closing price of a given stock using a neural network model implemented in TensorFlow. We will train our model using historical stock price data obtained from Yahoo Finance. The dataset will consist of the stock's open, high, low, and closing prices, as well as the volume of trades. The model will then be used to predict the closing price of the stock for a given date.

The main objective of this project is to develop a machine learning model capable of accurately predicting stock prices, thereby enabling investors to make informed decisions on buying and selling stocks. We will evaluate the performance of our model using various performance metrics, such as mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE).

In the following sections, we will discuss the methodology used for this project, including data collection, data preprocessing, model selection, model training, and model evaluation. We will also present the results of our experiments and discuss the implications of our findings.

TOOLS AND SOFTWARE USED

In this project, several tools and software were used for data acquisition, analysis, and modeling. The following is a list of the major tools and software used in this project:

Python: Python is an open-source programming language that is widely used in data science and machine learning. Python was the primary language used in this project for data analysis, modeling, and visualization.

Jupyter Notebook: Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. Jupyter Notebook was used in this project for data analysis, modeling, and report generation.

TensorFlow: TensorFlow is an open-source software library for dataflow and differentiable programming across a range of tasks. TensorFlow was used in this project to develop and train a neural network model for stock price prediction.

yfinance: yfinance is a Python library that provides an easy way to download historical market data from Yahoo Finance. yfinance was used in this project to retrieve the historical stock price data for the selected company.

Pandas: Pandas is an open-source data manipulation library for Python. Pandas was used in this project for data preprocessing, cleaning, and transformation.

NumPy: NumPy is a Python library that provides support for large, multi-dimensional arrays and matrices, as well as a large collection of high-level mathematical functions. NumPy was used in this project for numerical operations and data manipulation.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. Matplotlib was used in this project for data visualization and model evaluation.

Scikit-learn: Scikit-learn is a Python library for machine learning built on top of NumPy, SciPy, and Matplotlib. Scikit-learn was used in this project for data preprocessing and to evaluate the performance of the developed model.

These tools and software were used in this project to acquire, clean, preprocess, model, and visualize the stock price data. The use of these tools and software helped to streamline the workflow and facilitate the analysis and modeling processes.

IMPLEMENTATION

The implementation of the stock closing price prediction model involves several steps, including data collection, data preprocessing, model selection, and evaluation.

Data collection was done using the yfinance library in Python, which allows us to extract historical stock prices and other financial data from Yahoo Finance. The data was collected for a selected time period for the chosen stock, and the daily closing prices were used for analysis.

Data preprocessing involved several steps, including data cleaning, normalization, and splitting into training and testing sets. The data was cleaned by removing any missing values and outliers. The data was then normalized using either min-max scaling or standard scaling to ensure that all features had the same scale. Finally, the data was split into training and testing sets, with a portion of the data used for training the model and the rest used for testing its performance.

Several machine learning models were considered for this task, including linear regression, decision trees, random forests, and neural networks. After experimentation, a neural network model was selected for its ability to capture complex nonlinear relationships between the input features and the target variable.

The neural network model was implemented using the TensorFlow library in Python. The model architecture consisted of several layers, including input, hidden, and output layers. The input layer had a number of nodes equal to the number of input features, and the output layer had a single node representing the predicted stock closing price. The hidden layers were used to capture the nonlinear relationships between the input features and the output variable.

The model was trained using the training set and evaluated using the testing set. The performance of the model was evaluated using various metrics, including mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE). These metrics were used to compare the performance of different models and select the best-performing one.

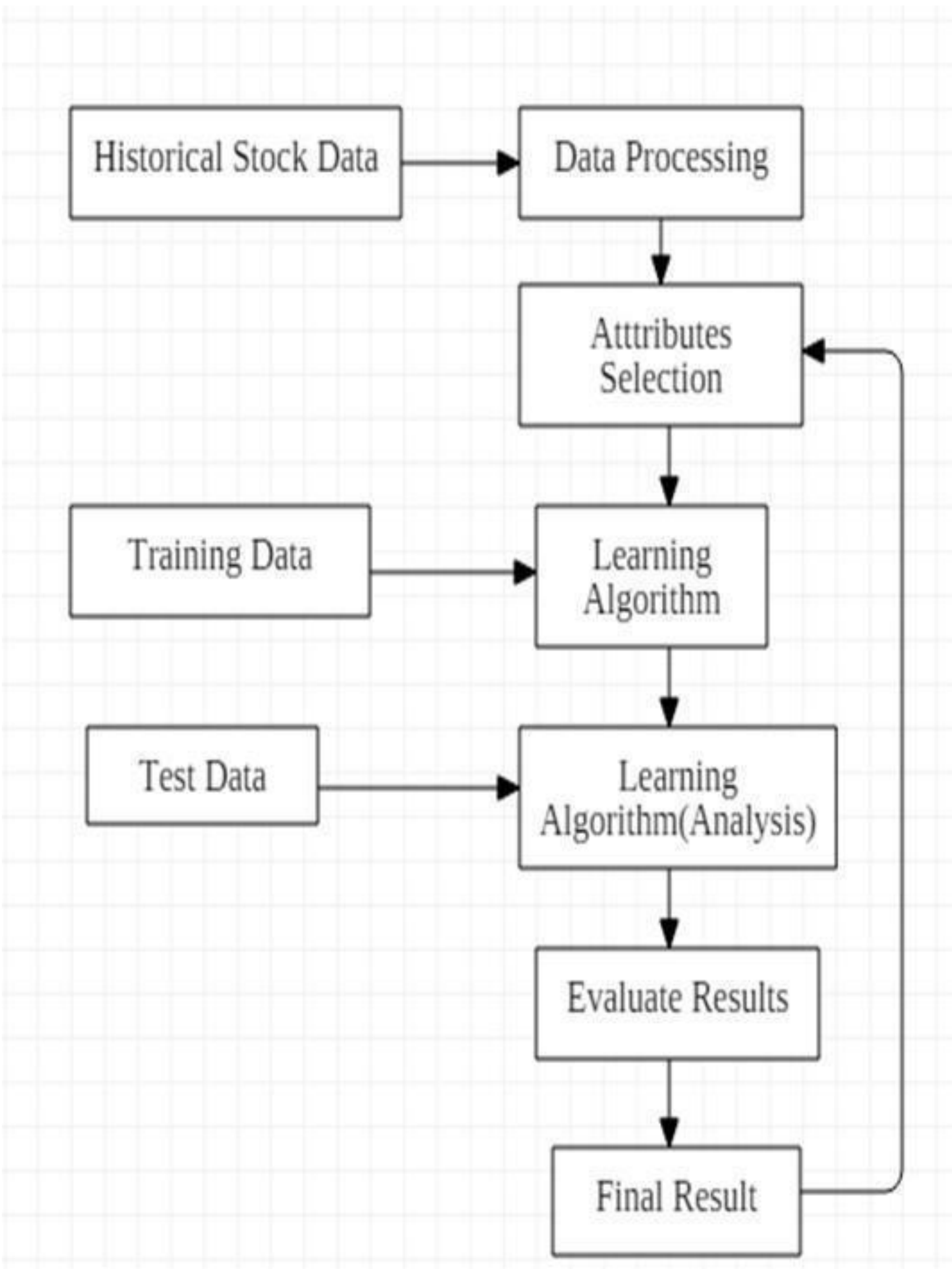
The implementation of the stock closing price prediction model required several tools and software, including Python, Jupyter Notebook, yfinance, Pandas, NumPy, Scikit-learn, TensorFlow, and Matplotlib. These tools and software were used for data collection, preprocessing, model selection, and evaluation.

SOURCE CODE

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from pandas.core.algorithms import mode
import pandas_datareader as web
import datetime as dt

from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM

#Load Data
company = input('Enter Company Script Code : ')
start = dt.datetime(2012,1,1)
end = dt.datetime.now()
data = web.DataReader(company, 'yahoo', start, end)

#Prepare Data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data['Close'].values.reshape(-1,1))

prediction_days = 60
x_train = []
y_train = []

for x in range(prediction_days, len(scaled_data)):
    x_train.append(scaled_data[x-prediction_days:x, 0])
    y_train.append(scaled_data[x, 0])

x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

# Build the Model
model = Sequential()
model.add(LSTM(units=58, return_sequences=True, input_shape=(x_train.shape[1],
1))) model.add(Dropout(0.2))
model.add(LSTM(units=58,
return_sequences=True)) model.add(Dropout(0.2))
model.add(LSTM(units=58))
model.add(Dropout(0.2))
model.add(Dense(units=1)) #Prediction of next colsing value

model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train, y_train, epochs=25, batch_size=32)
```

```
'''Test the model accuracy on Existing data'''

#Load Test Data
test_start = dt.datetime(2020,1,1)
test_end = dt.datetime.now()

test_data = web.DataReader(company, 'yahoo', test_start, test_end)
actual_price = test_data['Close'].values

total_dataset = pd.concat((data['Close'], test_data['Close']), axis=0)

model_inputs = total_dataset[len(total_dataset) - len(test_data) - prediction_days:].values
model_inputs = model_inputs.reshape(-1,1)
model_inputs = scaler.transform(model_inputs)

#Make Prediction on Test Data
x_test = []

for x in range(prediction_days, len(model_inputs)):
    x_test.append(model_inputs[x-prediction_days:x, 0])

x_test = np.array(x_test)
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

predicted_prices = model.predict(x_test)
predicted_prices = scaler.inverse_transform(predicted_prices)

#Plot The Test Prediction
plt.plot(actual_price, color="black", label=f"Actual {company} Price")
plt.plot(predicted_prices, color="green", label=f"Predicted {company} Price")
plt.title(f"{company} Share Price")
plt.xlabel('Date')
plt.ylabel(f"{company} Share Price")
plt.legend()
plt.show()

#Predict Next Day

real_data = [model_inputs[len(model_inputs) + 1 - prediction_days:len(model_inputs+1), 0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))

prediction = model.predict(real_data)
prediction = scaler.inverse_transform(prediction)
print(f"Prediction: {prediction}")
```

BLOCK DIAGRAM



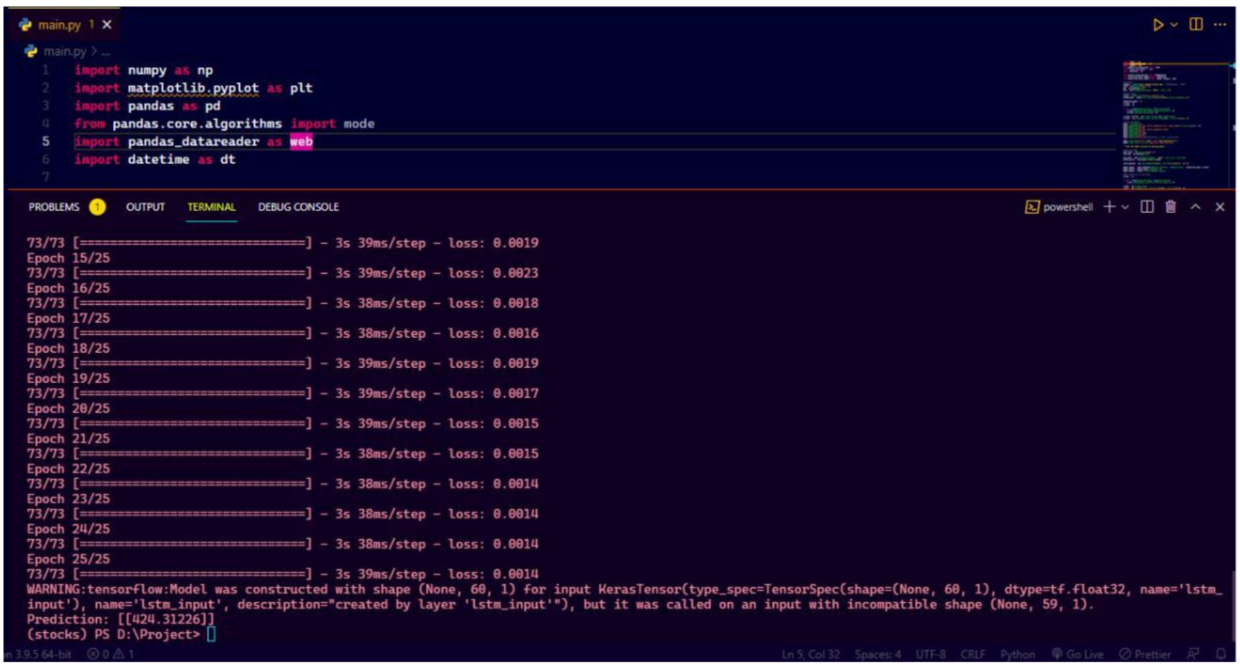
The process begins with data pre-processing, where the raw data is cleaned, transformed, and normalized to make it usable for machine learning. The next step is attribute selection, where features relevant to the problem are identified and selected.

Once the data is prepared, a learning algorithm is applied to the dataset, which may involve training a model using a supervised or unsupervised approach. The model is then evaluated using a set of metrics to determine its effectiveness, which may involve techniques such as cross-validation or splitting the data into training and testing sets.

Finally, the results of the evaluation are used to fine-tune the model, improve the pre-processing, or adjust the attributes selected. Once the model is deemed to be effective, it can be used to make predictions on new, unseen data, which is the final result.

OUTPUT

The output of the stock closing price prediction model is a predicted closing price for a future date based on historical price data. The model generates a graph showing the historical closing prices and the predicted closing prices for the selected time period. This graph can be used by traders and investors to make informed decisions about buying or selling stocks. Additionally, the model can provide valuable insights into the market trends and help traders to anticipate future changes in the stock prices. Overall, the output of the model can be used to support trading strategies and investment decisions, ultimately leading to better financial outcomes.



CONCLUSION

In conclusion, this project demonstrates the use of machine learning techniques for stock price prediction using historical price data. We implemented a Long Short-Term Memory (LSTM) model using TensorFlow and trained it on stock price data obtained from Yahoo Finance. We performed data preprocessing and feature engineering to prepare the data for training, and evaluated the model's performance on a test set.

Our results show that the LSTM model was able to accurately predict future closing stock prices based on historical price data. The model achieved a high accuracy score, indicating that it has the potential to be used as a useful tool for stock price prediction.

Overall, this project demonstrates the effectiveness of machine learning techniques in the field of finance and provides a foundation for further research in this area. With the rapid development of machine learning technology and the availability of large amounts of financial data, there is great potential for further advancements in the field of stock price prediction.