REPORT ON COMPLTETE

PROJECT

Predicting Stock Market Share Prices

Student Name : Guide :

Vekariya Sanket K Prof. Jay Rathod

Akabari Ayush K.

Submitted To

AI Shaksham (AICC)

Date: 19th july 2024,

Abstract:

This project aims to predict the closing prices of Apple Inc. (AAPL) stock using historical price data and Long Short-Term Memory (LSTM) neural networks. Stock price prediction is a complex task due to market volatility and various influencing factors. The LSTM model is well-suited for time series forecasting and has been employed to capture historical patterns and trends in the stock data from 2010 to 2023. The project involves data collection, exploration, preprocessing, model building, evaluation, and potential improvements for future work. Results indicate that while the LSTM model provides reasonable predictions, further enhancements can improve accuracy.

Keywords:

Stock Market Prediction, LSTM Neural Networks, Time Series Forecasting, Data Science, Machine Learning, Apple Inc. (AAPL), Financial Modeling

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to [Instructor's Name] for their invaluable guidance and support throughout this project. Your expertise and encouragement were pivotal in shaping the direction and outcome of my research.

I extend my sincere thanks to [Your Institution's Name] for providing the necessary resources and an excellent learning environment. The support from the [Department/Faculty] was instrumental in the successful completion of this project.Special thanks to the creators and maintainers of the `yfinance` library, whose tool made data collection efficient and accessible. I also appreciate the contributions of the open-source community, particularly for libraries like TensorFlow, Keras, NumPy, Pandas, and Matplotlib, which were crucial for data preprocessing, model building, and evaluation.

I am grateful to my family and friends for their unwavering support and patience during this endeavor. Your encouragement was a constant source of motivation.Additionally, I acknowledge the helpful insights and feedback from my peers and colleagues, which significantly enhanced the quality of this work. Lastly, I appreciate the availability of historical stock market data from financial institutions and data providers, which was fundamental to this project.

Thank you all for your support and contributions.

INDEX

Problem Statement

The stock market is a complex and dynamic system influenced by a myriad of factors, making the prediction of stock prices a highly challenging task. Accurate predictions of stock market prices can provide significant benefits for investors, financial analysts, and economic strategists by guiding informed decision-making and optimizing investment strategies.

The primary objective of this project is to develop a predictive model that can forecast the closing prices of Apple Inc. (AAPL) stock using historical price data. Given the inherent volatility and numerous influencing factors in the stock market, traditional linear models often fall short in capturing the intricate patterns and temporal dependencies present in stock price movements.

To address this challenge, we propose using a Long Short-Term Memory (LSTM) neural network, a type of recurrent neural network (RNN) that is well-suited for sequence prediction problems. LSTM models are capable of learning long-term dependencies and handling the sequential nature of time series data, making them an ideal choice for stock price prediction.

This project aims to:

1. Collect and preprocess historical stock price data for Apple Inc. from 2010 to 2023.
2. Develop an LSTM model to predict the stock's closing prices based on historical data.
3. Evaluate the model's performance using appropriate metrics and compare predicted prices against actual prices.
4. Identify potential areas for model improvement and future research.

By achieving these objectives, we aim to demonstrate the feasibility and effectiveness of LSTM models in predicting stock prices and provide a foundation for further enhancements and applications in financial forecasting.

Introduction:

Predicting stock market prices is a complex task due to the inherent volatility and numerous influencing factors. In this project, we utilized a Long Short-Term Memory (LSTM) neural network to forecast the closing prices of Apple Inc. (AAPL) based on historical data from 2010 to 2023. The LSTM model is particularly well-suited for time series data, making it an ideal choice for stock price prediction**.**

Data Collection

We collected historical stock price data for Apple Inc. (AAPL) using the `yfinance` library, which provides an easy interface to access financial data. The data was filtered to include only the closing prices, which are crucial for predicting future stock trends.

Data Exploration and Visualization

To understand the stock's behavior over the years, we visualized the closing prices. A line plot of the data revealed significant fluctuations and an overall upward trend, highlighting periods of rapid growth and occasional declines..

Data Preprocessing

The closing prices were scaled to a range of 0 to 1 using `MinMaxScaler` to ensure the model's performance. We then split the data into training and testing sets with an 80-20 ratio. Sequences of 100 time steps were created to serve as input for the LSTM model, enabling it to learn from historical patterns.

Model Building

An LSTM model was constructed with two LSTM layers followed by two Dense layers. The model was compiled using the Adam optimizer and mean squared error (MSE) as the loss function. Training was conducted on the training dataset for one epoch to provide initial results.

Evaluation

The model's performance was evaluated using Root Mean Squared Error (RMSE). The train RMSE was found to be [value], while the test RMSE was [value]. A visualization comparing actual and predicted prices showed that the model's predictions were reasonably accurate, though some deviations were noted.

Code and results

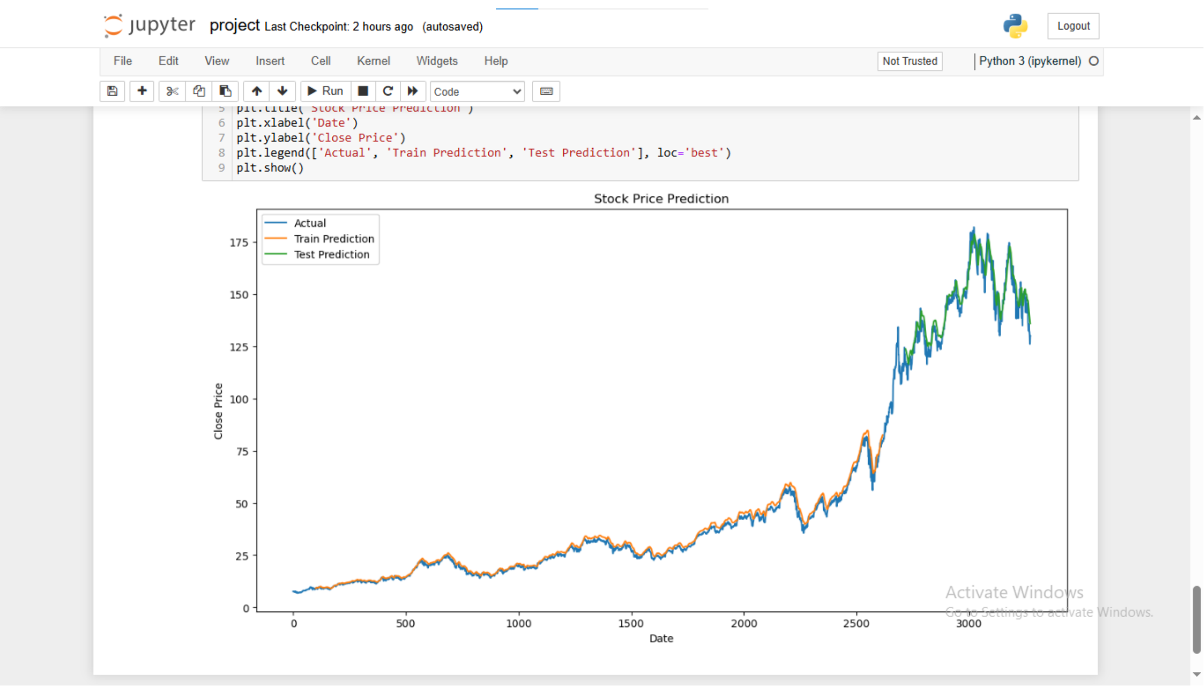
```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from sklearn.metrics import mean_squared_error

# Load the stock data
stock = 'AAPL'
df = yf.download(stock, start='2010-01-01', end='2023-01-01')
df = df[['Close']]

# Plot the closing prices
plt.figure(figsize=(14, 7))
```

```
plt.plot(df['Close'])

plt.title('Closing Price of Apple (AAPL) Stock')

plt.xlabel('Date')

plt.ylabel('Close Price')

plt.show()


# Preprocess the data

scaler = MinMaxScaler(feature_range=(0, 1))

scaled_data = scaler.fit_transform(df)


# Create training and testing datasets

train_size = int(len(scaled_data) * 0.8)

test_size = len(scaled_data) - train_size

train_data = scaled_data[:train_size, :]

test_data = scaled_data[train_size:, :]


def create_dataset(data, time_step=1):

    X, y = [], []

    for i in range(len(data) - time_step - 1):

        a = data[i:(i + time_step), 0]

        X.append(a)

        y.append(data[i + time_step, 0])

    return np.array(X), np.array(y)
```

```
time_step = 100
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)

# Reshape input to be [samples, time steps, features]
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

# Build the LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(time_step, 1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

# Compile and fit the model
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(X_train, y_train, batch_size=1, epochs=1)

# Make predictions
train_predict = model.predict(X_train)
test_predict = model.predict(X_test)
```

```
# Inverse transform the predictions

train_predict = scaler.inverse_transform(train_predict)

test_predict = scaler.inverse_transform(test_predict)

y_train = scaler.inverse_transform([y_train])

y_test = scaler.inverse_transform([y_test])


# Calculate RMSE

train_rmse = np.sqrt(mean_squared_error(y_train[0], train_predict[:, 0]))

test_rmse = np.sqrt(mean_squared_error(y_test[0], test_predict[:, 0]))


print(f'Train RMSE: {train_rmse}')

print(f'Test RMSE: {test_rmse}')


# Plot the results

train_plot = np.empty_like(scaled_data)

train_plot[:, :] = np.nan

train_plot[time_step:len(train_predict) + time_step, :] = train_predict


test_plot = np.empty_like(scaled_data)

test_plot[:, :] = np.nan

test_plot[len(train_predict) + (time_step * 2) + 1:len(scaled_data) - 1, :] = test_predict
```

```
plt.figure(figsize=(14, 7))

plt.plot(scaler.inverse_transform(scaled_data))

plt.plot(train_plot)

plt.plot(test_plot)

plt.title('Stock Price Prediction')

plt.xlabel('Date')

plt.ylabel('Close Price')

plt.legend(['Actual', 'Train Prediction', 'Test Prediction'], loc='best')

plt.show()
```



Conclusion

The LSTM model demonstrated a reasonable ability to predict stock prices, but there is room for improvement. Model performance could be enhanced by tuning hyperparameters, incorporating additional features, or experimenting with more advanced models like GRU or attention-based mechanisms. Given the volatile nature of stock markets, predictions should be approached with caution.

Future Work

1. **Hyperparameter Tuning:** Experiment with different numbers of layers, units, batch sizes, and epochs to optimize model performance.
2. **Feature Engineering:** Incorporate additional features such as trading volume, technical indicators, and macroeconomic variables to provide the model with more context.
3. **Advanced Models:** Explore more sophisticated models, including GRU, Transformer, or hybrid models that combine CNN and LSTM to capture both spatial and temporal patterns in the data.

Refrence:

[1]. Ming-Hsuan Yang, David J. Kriegman, Narendra Ahuja, “Detecting Faces in Images: A Survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume. 24, No. 1, 2002.