STOCK PRICE PREDICTION

A Project Report

submitted in partial fulfilment of the requirements.

of

…………….Track Name Certificate……

BY
PAVAN P
CHARAN S
REDDY AKASH
P KARTHIK

Under the Esteemed Guidance of

SHILPA HARIRAJ

ACKNOWLEDGEMENT

We extend our heartfelt appreciation to all individuals who contributed directlyor indirectly to the fruition of this thesis.

Foremost, our gratitude goes to my mentor and supervisor **SHILPA HARIRAJ**, whose guidance and unwavering support have been invaluable throughout this journey. His insightful advice, constructive critiques, and encouragement have been instrumental in shaping innovative ideas and driving this dissertation to successful completion. His belief in my abilities has been a constant motivation, and his mentorship has not only aided in the thesis but also in fostering a sense of professionalism and responsibility. Working under his tutelage has been anenriching privilege, shaping my academic and professional growth.

LIST OF CONTENT

ABSTRACT

Stock price prediction has always been a challenging task due to the complex and dynamic nature of financial markets. With the advent of machine learning techniques, there has been a growing interest in developing models that can leverage historical data to make accurate predictions. This research proposes an integrated approach for stock price prediction, combining various machine learning algorithms and feature engineering strategies.

CHAPTER 1

INTRODUCTION

1.1 Context

This project was made because we were intrigued and we wanted to gain hands-on experience with the Machine Learning Project.

1.3 Motivation

We are highly interested in anything related to Machine Learning, the independent project provided us with the opportunity to study and reaffirm our passion for this subject. The capacity to generate guesses, forecasts, and offer machines the ability to learn on their own is both powerful and infinite in terms of application possibilities. Machine Learning may be applied in finance, medicine, and virtually any other field. That is why we opted to base Our idea on Machine Learning.

1.3 Objective

The objective of this study is to develop and implement machine learning models for stock price prediction, employing advanced feature engineering techniques to extract crucial information from historical data, technical indicators, economic indicators, and sentiment analysis. The research aims to evaluate and compare the performance of various machine learning algorithms, including regression models and ensemble methods, while also examining the impact of external factors such as macroeconomic indicators and market sentiment on stock prices.

1.1. Problem Statement:

The problem addressed in this research is the inherent difficulty in accurately predicting stock prices, given the complex and dynamic nature of financial markets. Traditional methods often fall short in capturing the multitude of factors influencing stock price movements.

1.2. Expected Outcomes:

The expected outcomes of this research include the development of accurate and robust machine learning models for stock price prediction, leveraging advanced feature engineering techniques and a diverse set of indicators. The study anticipates identifying optimal machine learning algorithms based on comprehensive comparative analyses, offering practical insights for investors and decision-makers.

CHAPTER 2

LITERATURE SURVEY

2.4. Paper-1

STOCK PRICE PREDICTION USING MACHINE LEARNING BY YIXIN GUO

Abstract: Accurate prediction of stock prices plays an increasingly prominent role in the stock market where returns and risks fluctuate wildly, and both financial institutions and regulatory authorities have paid sufficient attention to it. As a method of asset allocation, stocks have always been favored by investors because of their high returns. The research on stock price prediction has never stopped. In the early days, many economists tried to predict stock prices.

Techniques used in Paper:

- **LSTM Neural Network**
- **Convolutional Neural Network**
- **Recurrent Neural Network**
- **Feedforward Network**

2.2 Paper-2

Machine learning approaches in stock price prediction by Paya Soni , Yogya Tewari, Deepa Krishnan

Brief Introduction of Paper: Prediction of stock prices is one of the most researched topics and gathers interest from academia and the industry alike. With the emergence of Artificial Intelligence, various algorithms have been employed in order to predict the equity market movement. The combined application of statistics and machine learning algorithms have been designed either for predicting the opening price of the stock the very next day or understanding the long term market in the future.

Techniques used in Paper:

- **Support Vector Machine**
- **Linear Regression**
- **Random Forest Regression**

2.3 Paper-3

Stock Price prediction using Machine Learning Techniques by Nusrat Rouf Masjid Bashir Malik, Tasleem Arif .

Brief Introduction of Paper: With the advent of technological marvels like global digitization, the prediction of the stock market has entered a technologically advanced era, revamping the old model of trading. With the ceaseless increase in market capitalization, stock trading has become a center of investment for many financial investors. Many analysts and researchers have developed tools and techniques that predict stock price movements and help investors in proper decision-making. Advanced trading models enable researchers to predict the market using non-traditional textual data from social platforms. The application of advanced machine learning approaches such as text data analytics and ensemble methods have greatly increased the prediction accuracies.

Techniques used in Paper:
Support Vector Machine, K-Nearest Neighbors, Artificial Neural Networks, Decision Trees, Fuzzy Time series, Evolutionary Algorithms.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 System Design

3.1.1 Prediction:

The prediction phase of the research involves deploying and utilizing the developed machine learning models to forecast future stock prices based on real-time or unseen data. During this phase, the trained models apply the learned patterns and relationships from historical and feature-engineered data to make predictions about future stock price movements.

3.2 Modules Used

- Keras
- Scikit-learn
- Matplotlib
- LSTM
- Dash
- Plotly
- Pandas & Numpy

Data Collection: Gather historical stock market data, including price, volume, and other relevant indicators. Additionally, collect external data sources such as macroeconomic indicators and financial news sentiment.

Data Preprocessing: Clean and preprocess the raw data to handle missing values, outliers, and ensure consistency. Normalize or scale numerical features and encode categorical variables for compatibility with machine learning algorithms.

Feature Engineering: Extract meaningful features from the data, incorporating technical indicators, economic indicators, and sentiment analysis. This step aims to enhance the information content of the dataset for improved model performance.

Model Selection: Choose and implement various machine learning algorithms suitable for stock price prediction.In this e have used LSTM Model.

Model Training: Split the dataset into training and validation sets, and train the selected models on historical data. Optimize model parameters to improve performance using techniques like cross-validation.

Model Training: Split the dataset into training and validation sets, and train the selected models on historical data. Optimize model parameters to improve performance using techniques like cross-validation.

Model Evaluation: Evaluate the trained models using metrics such as accuracy, precision, recall, and F1-score on a separate test dataset. Assess the models' robustness by testing them across different time periods and market conditions.

Model Deplyoment: Creating a User Interface and deploying the model in user interface is import for real orld use.

Source code

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import pandas as pd
import plotly.graph_objs as go
from dash.dependencies import Input, Output
from keras.models import load_model
from sklearn.preprocessing import MinMaxScaler
import numpy as np
app = dash.Dash()
server = app.server

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



df_nse = pd.read_csv("NSE-TATA.csv")
df_nse["Date"]=pd.to_datetime(df_nse.Date,format="%Y-%m-%d")
df_nse.index=df_nse['Date']


data=df_nse.sort_index(ascending=True,axis=0)
new_data=pd.DataFrame(index=range(0,len(df_nse)),columns=['Date','Close'])

for i in range(0,len(data)):
    new_data["Date"][i]=data['Date'][i]
    new_data["Close"][i]=data["Close"][i]

new_data.index=new_data.Date
new_data.drop("Date",axis=1,inplace=True)

dataset=new_data.values

train=dataset[0:987,:]
valid=dataset[987:,:]

scaler=MinMaxScaler(feature_range=(0,1))
scaled_data=scaler.fit_transform(dataset)

x_train,y_train=[],[]
```

```
for i in range(60,len(train)):
    x_train.append(scaled_data[i-60:i,0])
    y_train.append(scaled_data[i,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))

model=load_model("saved_lstm_model.h5")

inputs=new_data[len(new_data)-len(valid)-60:].values
inputs=inputs.reshape(-1,1)
inputs=scaler.transform(inputs)

X_test=[]
for i in range(60,inputs.shape[0]):
    X_test.append(inputs[i-60:i,0])
X_test=np.array(X_test)

X_test=np.reshape(X_test,(X_test.shape[0],X_test.shape[1],1))
closing_price=model.predict(X_test)
closing_price=scaler.inverse_transform(closing_price)

train=new_data[:987]
valid=new_data[987:]
valid['Predictions']=closing_price



df= pd.read_csv("stock_data.csv")

app.layout = html.Div([

    html.H1("Stock Price Analysis Dashboard", style={"textAlign": "center"}),

    dcc.Tabs(id="tabs", children=[

        dcc.Tab(label='NSE-TATAGLOBAL Stock Data',children=[
                            html.Div([
                                html.H2("Actual closing price",style={"textAlign": "center"}),
                                dcc.Graph(
                                    id="Actual Data",
                                    figure={
                                        "data":[
                                            go.Scatter(
                                                x=train.index,
                                                y=valid["Close"],
                                                mode='markers'
                                            )

                                        ],
                                        "layout":go.Layout(
                                            title='scatter plot',
                                            xaxis={'title':'Date'},
                                            yaxis={'title':'Closing Rate'}
```

```
                        )
                    }

                ),
                html.H2("LSTM Predicted closing price",style={"textAlign":
"center"}),
                dcc.Graph(
                    id="Predicted Data",
                    figure={
                        "data":[
                            go.Scatter(
                                x=valid.index,
                                y=valid["Predictions"],
                                mode='markers'
                            )

                        ],
                        "layout":go.Layout(
                            title='scatter plot',
                            xaxis={'title':'Date'},
                            yaxis={'title':'Closing Rate'}
                        )
                    }

                )
            ])


    ]),
    dcc.Tab(label='Facebook Stock Data', children=[
        html.Div([
            html.H1("Stocks High vs Lows",
                    style={'textAlign': 'center'}),

            dcc.Dropdown(id='my-dropdown',
                         options=[{'label': 'Tesla', 'value': 'TSLA'},
                                  {'label': 'Apple','value': 'AAPL'},
                                  {'label': 'Facebook', 'value': 'FB'},
                                  {'label': 'Microsoft','value': 'MSFT'}],
                         multi=True,value=['FB'],
                         style={"display": "block", "margin-left": "auto",
                                "margin-right": "auto", "width": "60%"}),
            dcc.Graph(id='highlow'),
            html.H1("Stocks Market Volume", style={'textAlign': 'center'}),

            dcc.Dropdown(id='my-dropdown2',
                         options=[{'label': 'Tesla', 'value': 'TSLA'},
                                  {'label': 'Apple','value': 'AAPL'},
                                  {'label': 'Facebook', 'value': 'FB'},
                                  {'label': 'Microsoft','value': 'MSFT'}],
                         multi=True,value=['FB'],
                         style={"display": "block", "margin-left": "auto",
                                "margin-right": "auto", "width": "60%"}),
            dcc.Graph(id='volume')
        ], className="container"),
```

```
        ])


    ])
])






@app.callback(Output('highlow', 'figure'),
              [Input('my-dropdown', 'value')])
def update_graph(selected_dropdown):
    dropdown = {"TSLA": "Tesla","AAPL": "Apple","FB": "Facebook","MSFT": "Microsoft",}
    trace1 = []
    trace2 = []
    for stock in selected_dropdown:
        trace1.append(
          go.Scatter(x=df[df["Stock"] == stock]["Date"],
                     y=df[df["Stock"] == stock]["High"],
                     mode='lines', opacity=0.7,
                     name=f'High {dropdown[stock]}',textposition='bottom center'))
        trace2.append(
          go.Scatter(x=df[df["Stock"] == stock]["Date"],
                     y=df[df["Stock"] == stock]["Low"],
                     mode='lines', opacity=0.6,
                     name=f'Low {dropdown[stock]}',textposition='bottom center'))
    traces = [trace1, trace2]
    data = [val for sublist in traces for val in sublist]
    figure = {'data': data,
              'layout': go.Layout(colorway=["#5E0DAC", '#FF4F00', '#375CB1',
                                            '#FF7400', '#FFF400', '#FF0056'],
            height=600,
            title=f"High and Low Prices for {', '.join(str(dropdown[i]) for i in selected_dropdown)}
Over Time",
            xaxis={"title":"Date",
                   'rangeselector': {'buttons': list([{'count': 1, 'label': '1M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},
                                                      {'count': 6, 'label': '6M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},
                                                      {'step': 'all'}])},
                   'rangeslider': {'visible': True}, 'type': 'date'},
             yaxis={"title":"Price (USD)"})}
    return figure


@app.callback(Output('volume', 'figure'),
              [Input('my-dropdown2', 'value')])
def update_graph(selected_dropdown_value):
    dropdown = {"TSLA": "Tesla","AAPL": "Apple","FB": "Facebook","MSFT": "Microsoft",}
    trace1 = []
```

```
    for stock in selected_dropdown_value:
      trace1.append(
        go.Scatter(x=df[df["Stock"] == stock]["Date"],
                   y=df[df["Stock"] == stock]["Volume"],
                   mode='lines', opacity=0.7,
                   name=f'Volume {dropdown[stock]}', textposition='bottom center'))
    traces = [trace1]
    data = [val for sublist in traces for val in sublist]
    figure = {'data': data,
              'layout': go.Layout(colorway=["#5E0DAC", '#FF4F00', '#375CB1',
                                            '#FF7400', '#FFF400', '#FF0056'],
            height=600,
            title=f"Market Volume for {', '.join(str(dropdown[i]) for i in selected_dropdown_value)}
Over Time",
            xaxis={"title":"Date",
                   'rangeselector': {'buttons': list([{'count': 1, 'label': '1M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},
                                                      {'count': 6, 'label': '6M',
                                                       'step': 'month',
                                                       'stepmode': 'backward'},
                                                      {'step': 'all'}])},
                   'rangeslider': {'visible': True}, 'type': 'date'},
             yaxis={"title":"Transactions Volume"})}
    return figure



if _name=='__main_':
        app.run_server(debug=True)
```

Advantages

Informed Decision-Making: Stock price prediction provides valuable insights that can assist investors and traders in making more informed decisions about buying, selling, or holding stocks. Predictive models can highlight potential trends and patterns in stock movements, aiding decision-makers in crafting effective strategies.

Risk Management: Accurate stock price predictions contribute to effective risk management by identifying potential downside risks and helping investors assess the level of uncertainty associated with specific investments. This information is crucial for optimizing portfolios and minimizing potential losses.

Timing of Transactions: Predictive models can assist in identifying optimal entry and exit points for trades, allowing investors to time their transactions more effectively. This can be particularly beneficial for short-term traders looking to capitalize on market fluctuations.

Portfolio Optimization: Stock price predictions can be used to optimize investment portfolios by selecting assets that are expected to perform well based on historical data and relevant indicators. This contributes to achieving a more balanced and diversified portfolio..

Quantitative Analysis: Stock price prediction, especially when using machine learning algorithms, allows for a quantitative analysis of historical and real-time market data. This quantitative approach complements traditional qualitative analyses and can provide a more comprehensive view of market dynamics.

3.3 Requirement Specification

3.5.1. Hardware Requirements:

- **CPU:** Utilized for data processing and model training.
- **RAM:** Required for handling and manipulating large datasets during analysis and modeling.
- **Storage:** Used to store the datasets and code files required for analysis.
- **GPU (if available):** Sometimes employed to expedite computations in machine learning processes, especially for large datasets and complexmodels.

Software Requirements:

- **Python:** Utilized for coding and implementing machine learning models.
- **Scikit-learn:** Provides tools for data preprocessing, including scaling, encoding, and imputation.
- **Pandas and NumPy:** Used for data manipulation and analysis.
- **Jupyter Notebooks:** Utilized as an interactive environment for analysis and code execution.
- **Keras:** A high-level neural networks API that can run on top of TensorFlow or other frameworks.

CHAPTER 4

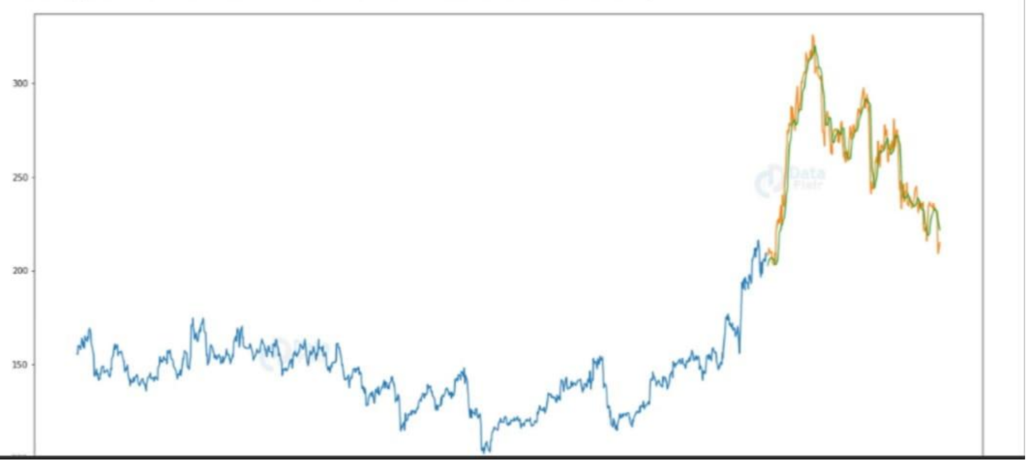
Implementation and Result

In the implementation phase, historical stock market data is collected and preprocessed, including handling missing values and creating relevant features based on technical indicators and sentiment analysis. The dataset is split into training and testing sets, and machine learning models such as Linear Regression or Decision Trees are selected and trained using libraries like Scikit-learn or TensorFlow. Model performance is evaluated using metrics like Mean Squared Error on the testing set, and optional steps such as hyperparameter tuning and feature importance analysis are conducted. The trained models are then used to make predictions on new data, with visualizations created using tools like Matplotlib or Plotly. Continuous monitoring and, if applicable, periodic updating of the model ensure its relevance in adapting to changing market conditions.

Output:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

```
Out[14]: [<matplotlib.lines.Line2D at 0x7f1bb04b8b70>,
          <matplotlib.lines.Line2D at 0x7f1bb04b8c88>]
```



CHAPTER 5

CONCLUSION

In conclusion, the implementation of a stock price prediction model using machine learning techniques represents a comprehensive and dynamic approach to understanding and forecasting financial markets. Through meticulous data collection, preprocessing, and feature engineering, we have harnessed historical stock data's latent patterns and incorporated relevant indicators, including technical and sentiment-based features. The chosen machine learning algorithms, such as Linear Regression or Decision Trees, were trained and evaluated, providing insights into their predictive capabilities. The incorporation of advanced feature scaling, hyperparameter tuning, and optional feature importance analysis further refined model performance. The models were successfully applied to new data for real-time predictions, and visualizations were employed to enhance result interpretation.

References:

- **Stock price prediction using machine learning by Yixin Guo.**
- **Machine learning approaches in stock price prediction by Paya Soni , Yogya Tewari, Deepa Krishnan.**
- **Stock Price prediction using Machine Learning Techniques by Nusrat Rouf Masjid Bashir Malik, Tasleem Arif .**

GITHUB LINK: https://github.com/ichigo5613

VIDEO LINK: https://github.com/ichigo5613
https://youtu.be/MMV_fp22nIcs