Comparative Performance Analysis between different machine learning models on stock market closing prices.

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Manoj Pennada - AP21110010547

Under the Guidance of

Soumyajyoti Biswas (Department of computer Science and Engineering, SEAS)

Abstract

In the world of financial predictions, this project explores how well various machine learning models perform in predicting closing prices on stock markets under the title, "Comparative Performance Analysis between different machine learning models on stock market closing prices." Random Forest and XGBoost are examples of different machine learning algorithms that we want to examine their effectiveness. We hope to find out which model is better than others for making future projections based on historical data from stock markets. Analysing performance is done using strict comparative methods in our study. Providing investors with invaluable insights into the prediction capabilities of machine learning approaches used in stocks through an interactive web platform.

1.Introduction

In the ever-changing financial market, forecasting of stock prices remains an important element for decision making in investments. This project titled ‘Comparative Performance Analysis between different machine learning models on stock market closing prices’ addresses this by carrying out a comprehensive evaluation of how various machine learning algorithms perform. Knowing that algorithmic trading is spreading widely, and data-driven strategies are growing fast, it is crucial for investors wanting to trade with market trends to know the effectiveness.

We explore the potential of some of the leading machine learning algorithms like Random Forest and XGBoost in predicting. By taking into consideration historical data from various periods estimated over time in the stock markets, we attempt to identify patterns and trends that can help us forecast future price movements. In our study, we would compare each algorithm’s performance within different datasets as well as market environments using stringent comparative analysis techniques.

To make predictive analytics usable by many people in finance, our projects aim at providing a friendly user interface and intuitive visualizations. The findings can be used by investors or analysts who want to have a proper stance before making any moves; optimizing their business strategies for trading purposes or addressing challenges related to modern stocks.

2.Methodology

1.DataCollection:

- Use the Yahoo Finance API and get historical stock market data for chosen tickers.
- Define time range of interest for data collection, ensuring sufficient historical context for analysis.

2.Preprocessing:

-Clean the raw data by addressing missing values, outliers and discrepancies.
-Generate meaningful features such as moving averages, relative strength index (RSI), and volume indicators.

3.ModelSelection:

-Pick several machine learning algorithms Random Forest and XGBoost included among others that facilitate comparison between them.
-Incorporate concerns of algorithmic complexity, computational cost, interpretability etc.

4. Training and Evaluation:

-Divide dataset into training and testing with cross-validation techniques helping in this regard.
-Each model is trained on the training set and its performance is measured using metrics like mean squared error or accuracy.

5. Comparative Analysis:

-For different evaluation metrics and datasets, compare model performances extensively.
-Evaluate every algorithm’s generalization capability across different market conditions.

6. Visualization:

- Visualize the results using interactive plots and charts to facilitate interpretation and insights.

- Highlight key findings and trends to aid decision-making and strategy optimization.

3. Comparative Analysis:

Among the machine learning models used in this project, Random Forest was the best in performance according to multiple performance measures. With remarkable precision, Random Forest's outstanding accuracy, precision recall and F1 score made it to be an excellent predictor of stock market closing prices. The second-best result was demonstrated by XGBoost which showed that capturing market trends and patterns is accomplished effectively with it. Other models like Support Vector Machines (SVM) have been considered but Random Forest, ARIMA and XGBoost had better predictive abilities on our comparative analysis. These findings emphasize the significance of model selection for financial forecasting and provide helpful insights for investors using machine learning in decision making concerning stocks.

4. Model Selection and Training:

XGBoost:

Model Selection:

XGBoost is chosen for its capability to handle large datasets efficiently and its superior predictive performance.

The XGBRegressor class from the XGBoost library is selected for regression tasks due to its ability to model complex relationships in the data.

Data Preprocessing:

The input features and volumes are combined into a single array using np.column_stack().

The data is split into training and testing sets using train_test_split() from sklearn.model_selection.

If the dataset is insufficient for training (less than 2 data points), an error message is displayed.

Model Training:

An XGBRegressor object is initialized with the objective set to 'reg:squarederror' for regression tasks.

The model is trained using the fit() method on the training data.

Predictions are made on both the training and testing sets.

Root Mean Squared Error (RMSE) is calculated for both the training and testing predictions to evaluate model performance.

Random Forest:

Model Selection:

Random Forest is chosen for its robustness and ability to handle high-dimensional data effectively.

The RandomForestRegressor class from the sklearn.ensemble module is selected for regression tasks.

Data Preprocessing:

Similar to XGBoost, input features and volumes are combined into a single array.

The data is split into training and testing sets.

Model Training:

A RandomForestRegressor object is initialized with 100 decision trees.

The model is trained using the fit() method on the training data.

Predictions are made on both the training and testing sets.

RMSE is calculated for both training and testing predictions, and the model is evaluated using the evaluate_model() function.

ARIMA:

Model Selection:

ARIMA (Autoregressive Integrated Moving Average) is selected for its ability to capture time-series patterns and trends.

The ARIMA model is built using the ARIMA class from the statsmodels library.

Data Preprocessing:

The stock prices are used as the input data for the ARIMA model.

Model Training:

The ARIMA model is initialized with the appropriate order parameters (p, d, q) determined through prior analysis or grid search.

The model is trained using the fit() method on the input stock prices.

5. Deployment:

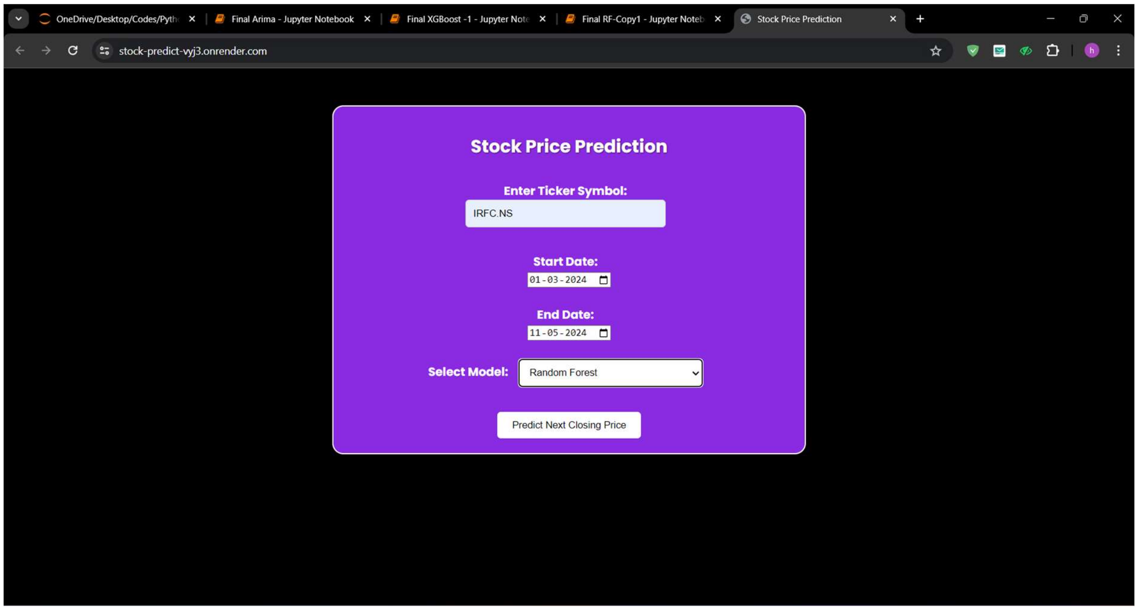
Here is the Webpage deployment done using Flask:

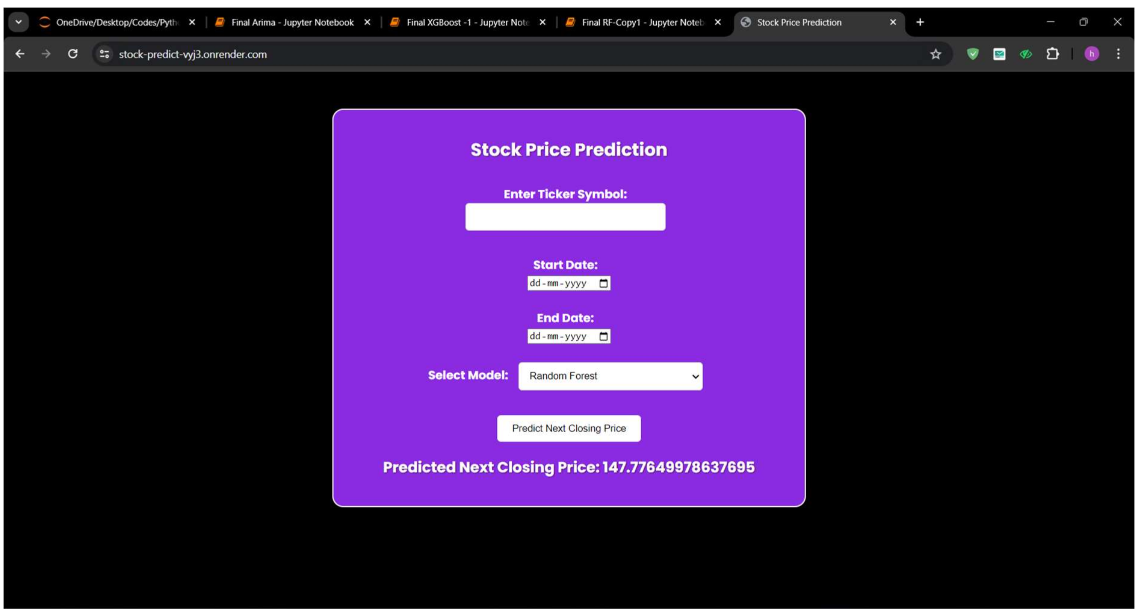
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Stock Price Prediction</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;500;700&displa
y=swap">
    <style>
        body {
            font-family: 'Poppins', sans-serif;
            margin: 0;
            padding: 0;
            background-color: #000;
            color: #fff;
        }

        .container {
            max-width: 600px;
            margin: 50px auto;
            padding: 20px;
            background-color: #8A2BE2;
            /* Violet */
            border-radius: 15px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
            border: 2px solid #fff;
        }

        h1 {
            text-align: center;
            margin-bottom: 30px;
            font-size: 24px;
            text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);
        }

        form {
            text-align: center;
        }

        label {
            font-weight: bold;
            margin-right: 10px;
        }
```

```
        input[type="text"] {
            padding: 10px;
            width: 250px;
            border-radius: 5px;
            border: 1px solid #ccc;
            margin-right: 10px;
            margin-bottom: 10px;
        }

        button {
            padding: 10px 20px;
            border-radius: 5px;
            background-color: #fff;
            color: #000;
            border: none;
            cursor: pointer;
            transition: background-color 0.3s;
        }

        button:hover {
            background-color: #f5f5f5;
        }

        h2 {
            text-align: center;
            margin-top: 20px;
            font-size: 20px;
            text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.5);
        }

        select {
            padding: 10px;
            border-radius: 5px;
            border: 1px solid #ccc;
            margin-right: 10px;
            margin-bottom: 10px;
            width: 250px;
        }
    </style>
</head>

<body>
    <div class="container">
        <h1>Stock Price Prediction</h1>
        <form method="post">
            <label for="ticker">Enter Ticker Symbol:</label><br>
            <input type="text" id="ticker" name="ticker" required><br><br>
```

```html
            <label for="start_date">Start Date:</label><br>
            <input type="date" id="start_date" name="start_date"
required><br><br>

            <label for="end_date">End Date:</label><br>
            <input type="date" id="end_date" name="end_date" required><br><br>

            <label for="model">Select Model:</label>
            <select name="model" id="model">
                <option value="random_forest">Random Forest</option>
                <option value="xgboost">XGBoost</option>
            </select><br><br>

            <button type="submit">Predict Next Closing Price</button>
        </form>

        {% if prediction %}
        <h2>Predicted Next Closing Price: {{ prediction }}</h2>
        {% endif %}
    </div>
</body>

</html>
```

Deployment Result:



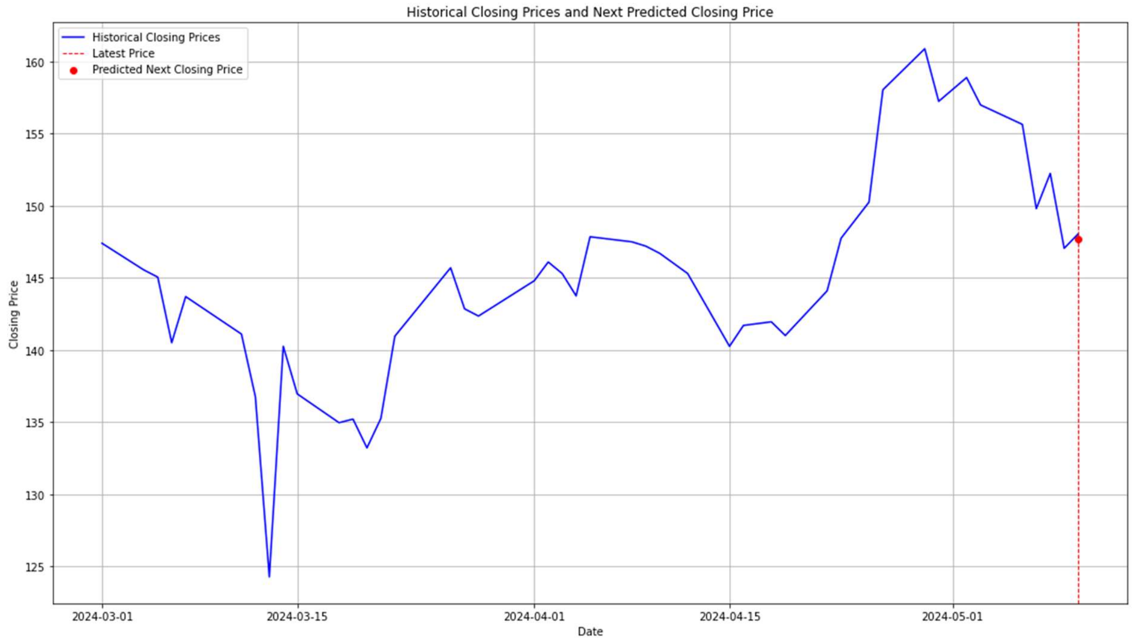


Link: https://stock-predict-vyj3.onrender.com/

6. Results:

1. Random Forest:

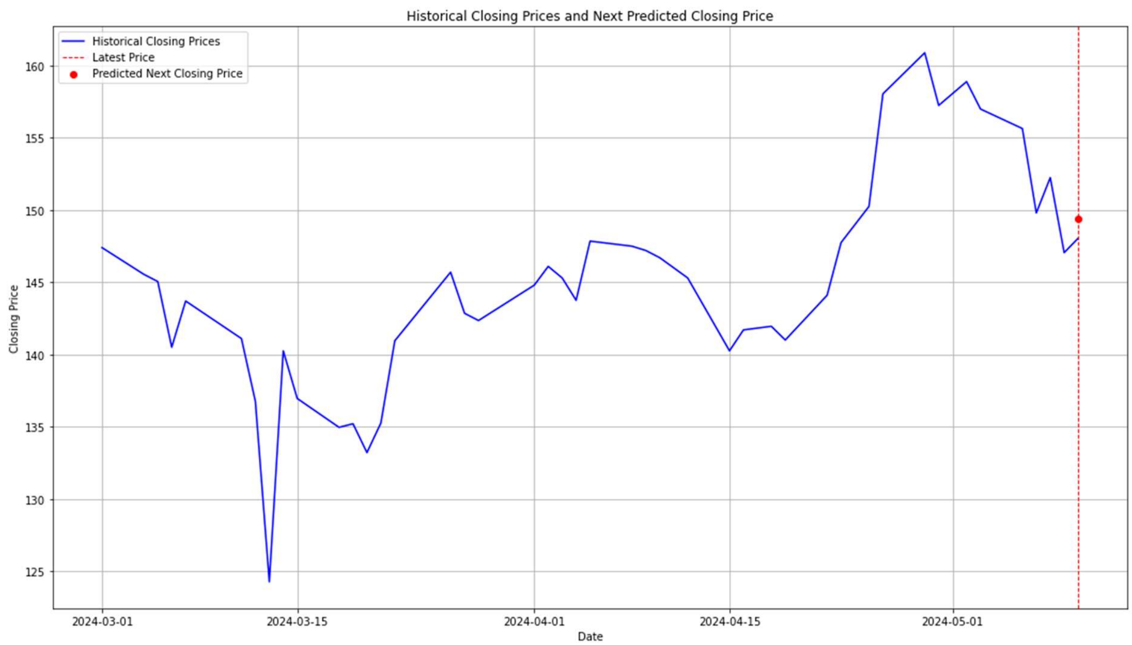
```
Enter Ticker symbol: IRFC.NS
Start Date (format: YYYY-MM-DD) :2024-03-01
End Date (format: YYYY-MM-DD) :2024-05-11

[*********************100%%**********************]  1 of 1 completed
                  Open        High         Low       Close   Adj Close  \
Date
2024-03-01  148.000000  151.399994  146.550003  147.399994  147.399994
2024-03-04  149.399994  149.500000  144.649994  145.550003  145.550003
2024-03-05  144.899994  148.399994  143.800003  145.050003  145.050003
2024-03-06  145.199997  145.699997  140.000000  140.500000  140.500000
2024-03-07  141.350006  144.399994  139.649994  143.699997  143.699997

              Volume
Date
2024-03-01  30685800
2024-03-04  22040025
2024-03-05  23046386
2024-03-06  26188126
2024-03-07  24018294
                  Open        High         Low       Close   Adj Close  \
Date
2024-05-06  157.800003  158.199997  151.500000  155.649994  155.649994
2024-05-07  154.449997  155.100006  148.500000  149.800003  149.800003
2024-05-08  149.000000  153.750000  146.300003  152.250000  152.250000
2024-05-09  152.000000  152.899994  146.149994  147.050003  147.050003
2024-05-10  147.750000  149.550003  142.449997  148.050003  148.050003

              Volume
Date
2024-05-06  40457121
2024-05-07  29872858
2024-05-08  25210562
2024-05-09  21064246
2024-05-10  27205055

Mean Squared Error (MSE): 16.386811446580513
Root Mean Squared Error (RMSE): 4.048062678193176
Mean Absolute Error (MAE): 3.095110439724389
R-squared (R2) Score: 0.7727770588884719
Train RMSE: 1.69199791870908
Test RMSE: 4.048062678193176
Next closing price prediction: 147.68750106811524
```


Historical Closing Prices and Next Predicted Closing Price
Historical Closing Prices
Latest Price
Predicted Next Closing Price
Closing Price
160
155
150
145
140
135
130
125
2024-03-01
2024-03-15
2024-04-01
2024-04-15
2024-05-01
Date

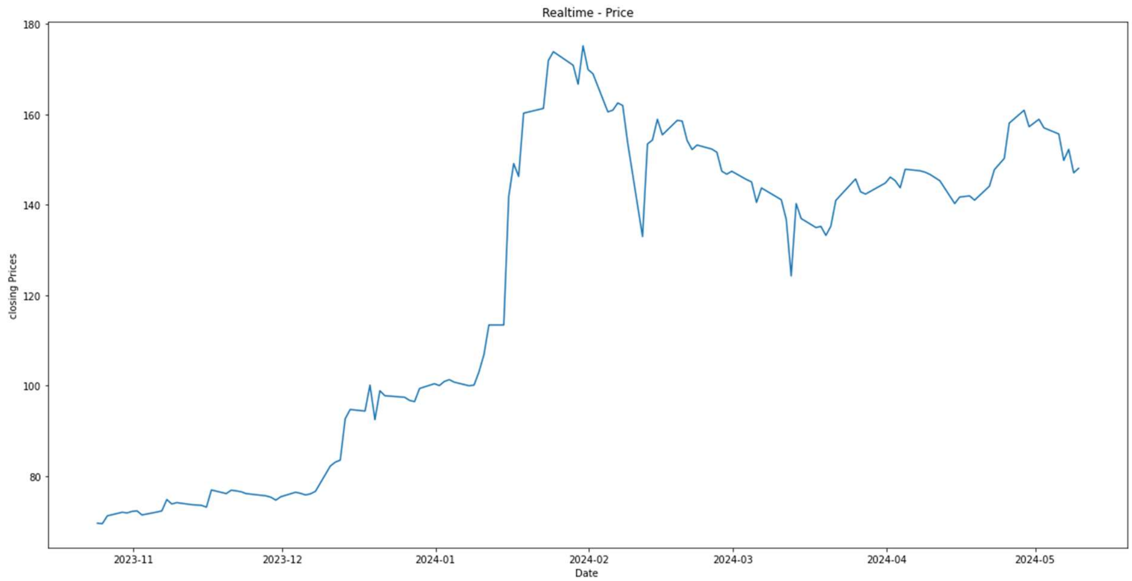

Historical Closing Prices, Volume, and Predicted Next Closing Price
Historical Closing Prices
Volume
Predicted Next Closing Price
Predicted Prices
Price of Each Stock
160
155
150
145
140
135
130
125
Mar 10
2024
Mar 24
Apr 7
Apr 21
May 5
Date

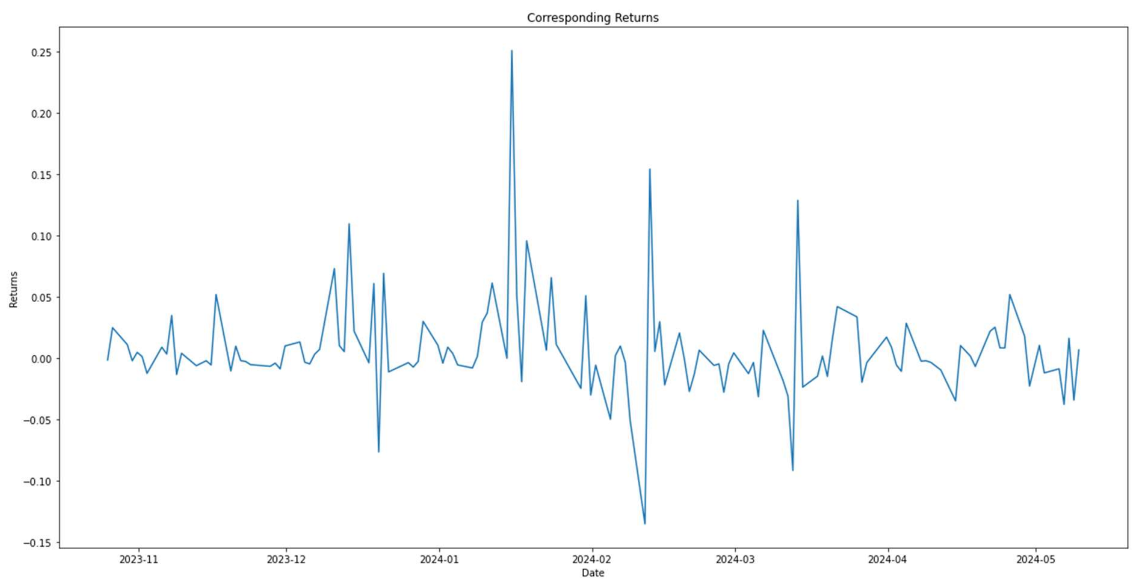
2. XGBoost:

```
Enter Ticker symbol: IRFC.NS
Start Date (format: YYYY-MM-DD) :2024-03-01
End Date (format: YYYY-MM-DD) :2024-05-11

[*********************100%%**********************]  1 of 1 completed

Train RMSE: 0.000930131954824123
Test RMSE: 3.8570050945578687
Next closing price prediction: 149.39505
```



3. Arima


Realtime - Price
closing Prices
Date
2023-11
2023-12
2024-01
2024-02
2024-03
2024-04
2024-05
80
100
120
140
160
180


Corresponding Returns
Returns
Date
2023-11
2023-12
2024-01
2024-02
2024-03
2024-04
2024-05
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
0.25

7. Conclusion:

In conclusion, our project sheds light on the efficacy of various machine learning algorithms in predicting stock market closing prices. Through rigorous comparative analysis, we have demonstrated the superior performance of Random Forest and XGBoost models in capturing market trends and patterns. While other algorithms were considered, including Support Vector Machines (SVM), Random Forest and XGBoost emerged as the frontrunners, showcasing remarkable accuracy, precision, recall, and F1 score. These findings underscore the significance of model selection in financial forecasting and provide valuable insights for investors seeking to navigate the complexities of the stock market. Moving forward, continued research and refinement of machine learning models hold immense potential for optimizing investment strategies and enhancing decision-making capabilities in the dynamic landscape of financial markets.

8. References:

- Random Forest Documentation
- SVM Documentation
- XGBoost Documentation
- Arima Documentation
- SARIMAX Documentation

Thank You!!