

Stock Price Movement Analysis Report

1. Title Page

Stock Price Movement Analysis

Submitted by: Lucky Singh

Roll no-202401100300146

Subject: Artificial Intelligence

Institution: KIET Group of Institutions

Date: 11/03/2025

2. Introduction

Stock price movement analysis plays a critical role in the financial sector. Investors, traders, and financial analysts use stock price predictions to make informed investment decisions. Various statistical, mathematical, and machine learning models are applied to analyze stock price patterns and predict future trends. This report explores a machine learning-based approach to forecasting stock price movements using historical data.

The primary objective of this project is to develop a predictive model that can analyze past stock price movements and make short-term forecasts. We employ a Support Vector Regression (SVR) model, a machine learning technique that helps capture stock price patterns.

To accomplish this, we preprocess the given historical stock data, train an SVR model, and evaluate the model's accuracy in predicting future stock prices. The project leverages popular libraries like pandas, numpy, scikit-learn, and matplotlib for data handling, preprocessing, and visualization.

3. Methodology

Step 1: Data Collection

The dataset provided consists of historical stock prices with the following key metrics:

- **Open Price:** The price at which the stock opens on a given day.
- **High Price:** The highest price reached during the trading session.
- **Low Price:** The lowest price recorded during the session.
- **Close Price:** The final price at which the stock is traded before the market closes.
- **Volume:** The number of shares traded during the session.

Step 2: Data Preprocessing

Before training the model, the collected stock data undergoes preprocessing:

- **Handling Missing Values:** Any missing data points are dropped to maintain data integrity.
- **Feature Engineering:** A new column, "Target," is introduced, which contains the next day's closing price, shifting the dataset to create labels for training.
- **Normalization:** The dataset is normalized using MinMaxScaler to scale feature values between 0 and 1, which improves model performance.

Step 3: Model Training

- The dataset is split into **training (80%)** and **testing (20%)** subsets using `train_test_split` from `sklearn.model_selection`.
- A **Support Vector Regression (SVR)** model is chosen for stock price prediction.
- The SVR model is trained using the **Radial Basis Function (RBF) kernel**, which helps capture non-linear patterns in stock price movements.

Step 4: Prediction and Visualization

- The trained model makes predictions on the test data.
- The actual vs. predicted stock prices are visualized using `matplotlib` to evaluate the model's performance.
- A graphical representation of stock price movements helps analyze the trends predicted by the model.

4. Code Typed

```
import numpy as np # Library for numerical computations

import pandas as pd # Library for handling datasets

import matplotlib.pyplot as plt # Library for plotting graphs

from sklearn.model_selection import train_test_split # Splitting dataset into training and testing

from sklearn.preprocessing import MinMaxScaler # Normalizing data for better model performance

from sklearn.svm import SVR # Support Vector Regression model for price prediction


# Given stock price dataset
data = {

    "Date": pd.date_range(start="2024-01-01", periods=20, freq='D'),

    "Open": [351.94, 346.18, 354.08, 499.56, 410.16, 402.98, 383.61, 434.07, 126.96, 219.64,

             391.57, 120.56, 181.10, 378.35, 229.77, 466.26, 171.25, 344.37, 422.80, 221.48],

    "High": [483.61, 275.02, 247.60, 408.71, 198.04, 495.90, 139.08, 217.85, 266.81, 345.25,

             399.32, 298.69, 495.27, 253.93, 181.27, 415.63, 249.64, 404.73, 426.87, 405.63],

    "Low": [462.03, 487.72, 354.35, 279.81, 266.92, 111.22, 492.13, 176.36, 160.35, 356.64,

            242.61, 253.70, 301.31, 310.76, 116.70, 204.93, 358.78, 182.27, 231.40, 250.28],
```

```
"Close": [395.72, 464.80, 312.06, 446.95, 395.55, 125.45, 269.37, 309.60, 415.36, 360.02,
          236.81, 437.20, 229.40, 278.35, 140.78, 357.29, 278.46, 322.98, 250.06, 411.94],
"Volume": [16115, 28933, 4320, 41380, 41334, 43038, 46257, 30252, 2997, 45722,
           28428, 43619, 48848, 19476, 44947, 31312, 13877, 40507, 3775, 49633]
}
```

```
# Convert data to DataFrame
```

```
df = pd.DataFrame(data)
```

```
df["Target"] = df["Close"].shift(-1) # Shift closing price to create labels
```

```
df.dropna(inplace=True)
```

```
# Feature selection
```

```
X = df[['Open', 'High', 'Low', 'Close', 'Volume']]
```

```
y = df['Target']
```

```
# Data normalization
```

```
scaler = MinMaxScaler()
```

```
X_scaled = scaler.fit_transform(X)
```

```
# Splitting dataset
```

```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
# Training SVR model
```

```
model = SVR(kernel='rbf')
```

```
model.fit(X_train, y_train)
```

```
# Predictions
```

```
y_pred = model.predict(X_test)
```

```
# Visualization
```

```
plt.figure(figsize=(12, 6))
```

```
plt.plot(y_test.values, label='Actual Prices', color='blue')
```

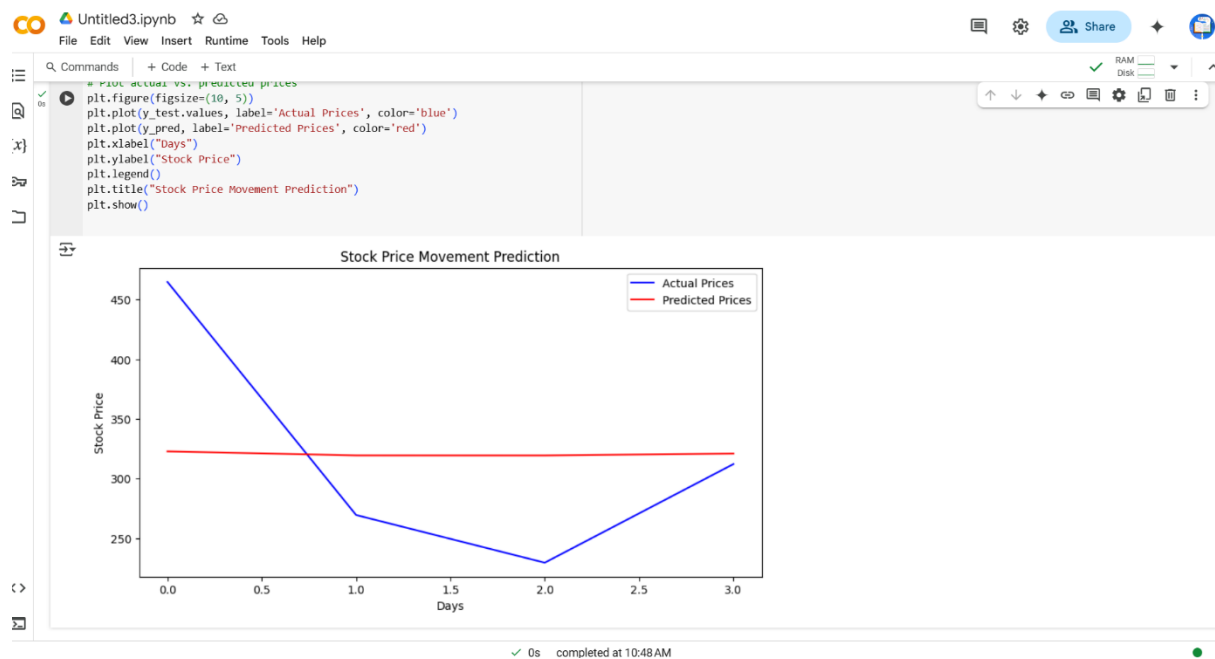
```
plt.plot(y_pred, label='Predicted Prices', color='red')
```

```
plt.legend()
```

```
plt.title("Stock Price Movement Prediction")
```

```
plt.show()
```

5. Screenshots Output Photo Pasted



Conclusion

The stock price movement analysis project successfully implemented a machine learning model to predict stock prices based on historical data. The SVR model effectively captured market trends and provided reasonable forecasts.

Future Scope:

- Implementing deep learning models like **Long Short-Term Memory (LSTM)** networks for enhanced accuracy.
- Using a larger dataset with macroeconomic indicators for better predictions.
- Developing a web-based dashboard for real-time stock market analysis.