STOCK MARKET PREDICTION

Table of Content

|  |  |
| --- | --- |
| Sl No | Topic |
| 1 | Introduction |
| 1.1 | Background |
| 2 | Data Loading and Exploration |
| 2.1 | Data Loading |
| 2.2 | Exploring the Dataset |
| 2.2.1 | Dataset Overview |
| 2.2.2 | Handling Missing Values |
| 2.2.3 | Correlation Matrix |
| 3 | .Data Visualization |
| 3.1 | Stock Price Trends Over Time |
| 3.2 | Volume Trends Over Time |
| 3.3 | Moving Average of Closing Prices |
| 4 | Predictive Modeling |
| 4.1 | Data Preparation for Modeling |
| 4.2 | Building and Evaluating the Linear Regression Model |
| 5 | Conclusion |

1. Introduction

1.1 Background

The purpose of this project is to analyze historical stock data and predict stock prices using machine learning. In this documentation, we will walk through the various steps involved in the analysis, starting from data exploration and visualization to building a predictive model.

2. Data Loading and Exploration

2.1 Data Loading

The first step involves loading the stock data from the provided CSV file. We utilize the pandas library to create a DataFrame, enabling easy manipulation and analysis of the dataset.

import pandas as pd

df = pd.read\_csv('stock data.csv')

2.2 Exploring the Dataset

2.2.1 Dataset Overview

df.shape

df.columns

df.info()

df.head()

df.describe()

2.2.2 Handling Missing Values

df.isnull().sum()

df.dropna(inplace=True)

df.isnull().sum()

2.2.3 Correlation Matrix

correlation\_matrix = df.corr()

3. Data Visualization

3.1 Stock Price Trends Over Time

import matplotlib.pyplot as plt

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

plt.plot(df['date'], df['close'], label='Close Price')

plt.xlabel('Date')

plt.ylabel('Close Price')

plt.title('Stock Prices Over Time')

plt.legend()

plt.show()

3.2 Volume Trends Over Time

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

plt.plot(df['date'], df['volume'], label='Volume', color='orange')

plt.xlabel('Date')

plt.ylabel('Volume')

plt.title('Volume Trends Over Time')

plt.legend()

plt.show()

3.3 Moving Average of Closing Prices

window\_size = 15

df['MA\_close'] = df['close'].rolling(window=window\_size).mean()

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

plt.plot(df['date'], df['close'], label='Close Price')

plt.plot(df['date'], df['MA\_close'], label=f'{window\_size}-Day Moving Average', linestyle='--', color='red')

plt.xlabel('Date')

plt.ylabel('Close Price')

plt.title('Closing Prices with Moving Average')

plt.legend()

plt.show()

4. Predictive Modeling

4.1 Data Preparation for Modeling

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import StandardScaler

features = df[['open', 'high', 'low', 'volume', 'RSIadjclose15', 'RSIvolume15']]

target = df['TARGET']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

4.2 Building and Evaluating the Linear Regression Model

model = LinearRegression()

model.fit(X\_train\_scaled, y\_train)

predictions = model.predict(X\_test\_scaled)

print(predictions)

plt.scatter(y\_test, predictions)

plt.xlabel('Actual Values')

plt.ylabel('Predicted Values')

plt.title('Actual vs Predicted Values')

plt.show()

5. Conclusion

In this documentation, we covered the entire process of stock data analysis, from data loading and exploration to visualization and predictive modeling. The code provided is well-documented and includes visualizations to enhance understanding. Additionally, a linear regression model was built and evaluated for stock price prediction.