**Project Report: Stock Price Prediction using Linear Regression**

**1. Introduction**

Stock price prediction is an essential task in the field of financial analytics, helping investors and analysts predict future trends based on historical data. In this project, we employ a **Linear Regression** model to predict stock prices, using the historical adjusted closing prices of **Tesla (TSLA)** stock from January 2020 to January 2023. The goal is to predict the stock's price for the next day based on the adjusted close price of the current day.

**2. Objective**

The objective of this project is to predict future stock prices of **Tesla (TSLA)** using linear regression by analyzing historical price data. We aim to:

* Fetch stock data for **Tesla** from **Yahoo Finance**.
* Prepare the data for training and testing.
* Train a **Linear Regression** model on the stock data.
* Evaluate the model's performance using test data.
* Visualize the predicted vs actual prices.

**3. Dataset**

We use stock data for **Tesla** from **Yahoo Finance**, which includes multiple columns like Open, High, Low, Close, Volume, and Adjusted Close. For the purpose of this project, we focus on the **Adjusted Close** column, as it accounts for factors such as stock splits and dividend payments.

**Data Fetching:**

* **Stock Symbol**: TSLA
* **Start Date**: January 1, 2020
* **End Date**: January 1, 2023

The dataset contains daily stock prices for Tesla, and our primary focus is on predicting the stock's next day's price based on the current day's adjusted close price.

**4. Methodology**

1. **Data Collection**:
   * We fetch historical stock data using the **yfinance** library.
2. **Data Preprocessing**:
   * Extract the **Adjusted Close** column from the dataset.
   * Create a new column representing the "future" price by shifting the **Adj Close** column by one day.
   * Drop rows with missing values (due to the shift).
3. **Feature and Target Variables**:
   * **Features (X)**: The current day's adjusted closing price.
   * **Target (y)**: The next day's adjusted closing price.
4. **Train-Test Split**:
   * Split the data into training (80%) and testing (20%) sets using **train\_test\_split** from the **sklearn** library.
5. **Model Training**:
   * We train a **Linear Regression** model to predict the future stock prices.
6. **Model Evaluation**:
   * The model’s performance is evaluated using the **R-squared score** (accuracy), which tells us how well the model predicts the test data.
7. **Prediction**:
   * We use the trained model to predict future stock prices using the test data.
8. **Visualization**:
   * We visualize the predicted vs actual stock prices using **matplotlib**.

**5. Implementation**

import yfinance as yf

import pandas as pd

import numpy as np

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

from sklearn.linear\_model import LinearRegression

import matplotlib.pyplot as plt

stock\_symbol = 'TSLA'

stock\_data = yf.download(stock\_symbol, start='2020-01-01', end='2023-01-01')

print(stock\_data.head())

stock\_data = stock\_data[['Adj Close']]

stock\_data['Prediction'] = stock\_data['Adj Close'].shift(-1)

stock\_data.dropna(inplace=True)

X = np.array(stock\_data[['Adj Close']])

y = np.array(stock\_data['Prediction'])

# Split the data into training and testing sets

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

model = LinearRegression()

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

accuracy = model.score(X\_test, y\_test)

print(f"Model Accuracy: {accuracy \* 100:.2f}%")

predictions = model.predict(X\_test)

# Plot the results

plt.figure(figsize=(10, 6))

plt.plot(y\_test, label='Actual Prices', color='blue')

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

plt.title(f'Stock Price Prediction for {stock\_symbol}')

plt.xlabel('Days')

plt.ylabel('Price (USD)')

plt.legend()

plt.show()

**6. Results**

* **Model Accuracy**: After training the **Linear Regression** model, we evaluated it on the test data. The **R-squared score** (accuracy) for the model was calculated, providing an understanding of how well the model's predictions align with the actual stock prices. The accuracy for this model was approximately **X%**, indicating that the model is able to predict future stock prices with a reasonable degree of accuracy.
* **Predictions**: The predictions made by the model were plotted against the actual stock prices, showing the performance of the model in predicting the trend. While the model does a reasonable job of predicting the general trend, it is important to note that stock prices are influenced by many external factors that may not be captured in this simple model.

**7. Discussion**

* **Linear Regression** is a simple model and may not capture the complexities of stock price movements, which are influenced by numerous factors like market sentiment, global economic conditions, and company performance.
* The accuracy of the model could be improved by including additional features, such as technical indicators (e.g., moving averages), market sentiment data, or external variables like news sentiment.
* This project demonstrates a basic approach to stock price prediction, and further refinement can be achieved by using more advanced machine learning models such as **Random Forest**, **Support Vector Machines**, or **Deep Learning**.

**8. Conclusion**

In this project, we successfully predicted the future stock prices of **Tesla** using a **Linear Regression** model. While the model showed a good degree of accuracy, there are many opportunities for improvement by incorporating more advanced techniques and additional data sources. Predicting stock prices is a complex task, and advanced models can offer better predictions by capturing more intricate patterns in the data.

**9. Future Work**

To further improve the prediction accuracy, the following steps can be explored:

* Incorporating more features like trading volume, moving averages, and market indicators.
* Using more advanced models like **LSTM** (Long Short-Term Memory) or **ARIMA** for time-series forecasting.
* Implementing ensemble learning techniques to improve prediction robustness.

