Source code :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from google.colab import files
import io

# Upload your stock data CSV file (with columns like: Date, Close)
print("Please upload your historical stock prices CSV file.")
uploaded = files.upload()
file_name = next(iter(uploaded))
df = pd.read_csv(io.BytesIO(uploaded[file_name]))

# Ensure 'Date' is datetime and sort by it
df['Date'] = pd.to_datetime(df['Date'])
df = df.sort_values('Date')

# Use only 'Date' and 'Close' for simplicity
df = df[['Date', 'Close']].dropna()

# Create lag features for time series prediction
def create_lag_features(data, lags=5):
    for lag in range(1, lags + 1):
        data[f'lag_{lag}'] = data['Close'].shift(lag)
    return data.dropna()

df_lagged = create_lag_features(df.copy(), lags=5)

# Features and labels
X = df_lagged[[f'lag_{i}' for i in range(1, 6)]]
y = df_lagged['Close']

# Train/test split (last 20% as test)
split_index = int(len(df_lagged) * 0.8)
X_train, X_test = X[:split_index], X[split_index:]
y_train, y_test = y[:split_index], y[split_index:]
dates_test = df_lagged['Date'][split_index:]

# Model training
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"\nModel Performance:")
print(f"Mean Squared Error: {mse:.2f}")
print(f"R^2 Score: {r2:.2f}")

# Plot actual vs predicted
plt.figure(figsize=(12, 6))
plt.plot(dates_test, y_test.values, label='Actual Price', color='blue')
plt.plot(dates_test, y_pred, label='Predicted Price', color='orange')
plt.title('Stock Price Prediction vs Actual')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```