```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, r2_score
```

```
from google.colab import files  # Use this if running on Google Colab

# Upload files
uploaded = files.upload()

# Load datasets
# Load datasets with the thousands parameter
train_data = pd.read_csv("Google_train_data.csv", thousands=',')
test_data = pd.read_csv("Google_test_data.csv", thousands=',')

# Display basic information
print("Train Data Info:")
print(train_data.info())

print("\nTest Data Info:")
print(test_data.info())

# Display first few rows
print("\nTrain Data Head:")
print(train_data.head())

print("\nTest Data Head:")
print(test_data.head())
```

Choose files 2 files

- **Google_test_data.csv**(text/csv) - 19908 bytes, last modified: 25/04/2025 - 100% done
- **Google_train_data.csv**(text/csv) - 62230 bytes, last modified: 25/04/2025 - 100% done

```
Saving Google_test_data.csv to Google_test_data.csv
Saving Google_train_data.csv to Google_train_data.csv
Train Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1258 entries, 0 to 1257
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    1258 non-null   object 
 1   Open    1258 non-null   float64
 2   High    1258 non-null   float64
 3   Low     1258 non-null   float64
 4   Close   1258 non-null   float64
 5   Volume  1258 non-null   int64  
dtypes: float64(4), int64(1), object(1)
memory usage: 59.1+ KB
None

Test Data Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252 entries, 0 to 251
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype  
---  ------     --------------  -----  
 0   Date       252 non-null    object 
 1   Open       252 non-null    float64
 2   High       252 non-null    float64
 3   Low        252 non-null    float64
 4   Close      252 non-null    float64
 5   Adj Close  252 non-null    float64
 6   Volume     252 non-null    int64  
dtypes: float64(5), int64(1), object(1)
memory usage: 13.9+ KB
None

Train Data Head:
       Date    Open    High     Low   Close    Volume
0  1/3/2012  325.25  332.83  324.97  663.59   7380500
1  1/4/2012  331.27  333.87  329.08  666.45   5749400
2  1/5/2012  329.83  330.75  326.89  657.21   6590300
3  1/6/2012  328.34  328.77  323.68  648.24   5405900
4  1/9/2012  322.04  322.29  309.46  620.76  11688800

Test Data Head:
         Date         Open         High          Low        Close  \
0  2018-08-13  1236.979980  1249.272949  1233.640991  1235.010010   
1  2018-08-14  1235.189941  1245.869995  1225.109985  1242.099976   
2  2018-08-15  1229.260010  1235.239990  1209.510010  1214.380005   
3  2018-08-16  1224.729980  1226.000000  1202.550049  1206.489990   
4  2018-08-17  1202.030029  1209.020020  1188.239990  1200.959961   

     Adj Close   Volume  
0  1235.010010   997300  
1  1242.099976  1348100  
2  1214.380005  1828800  
3  1206.489990  1343200  
4  1200.959961  1389600  
```

```
train_data = train_data.dropna()

# Feature Selection
features = ["Open", "High", "Low", "Volume"]
X = train_data[features]
y = train_data["Close"]

# Splitting Data
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize Models
models = {
    "Random Forest": RandomForestRegressor(n_estimators=100, random_state=42),
    "Decision Tree": DecisionTreeRegressor(random_state=42)
}
```

```
# Train & Evaluate Models
for name, model in models.items():
    model.fit(X_train, y_train)
    predictions = model.predict(X_val)
    mae = mean_absolute_error(y_val, predictions)
    r2 = r2_score(y_val, predictions)

    print(f"{name} - MAE: {mae:.2f}, R² Score: {r2:.2f}")
```

```
Random Forest - MAE: 31.16, R² Score: 0.74
Decision Tree - MAE: 28.70, R² Score: 0.55
```