# DAY – 6

# DSA0410 – Fundamentals of Data Science

## Lab Experiments:

Name: B Harish Balaji

Reg no: 192424386

Slot: D

26.Question 3: Linear Regression for Housing Price Prediction

You are a real estate analyst trying to predict housing prices based on various features of the

houses, such as area, number of bedrooms, and location. You have collected a dataset of houses

with their respective prices.

Write a Python program that allows the user to input the features (area, number of bedrooms, etc.)

of a new house. The program should use linear regression from scikit-learn to predict the price of

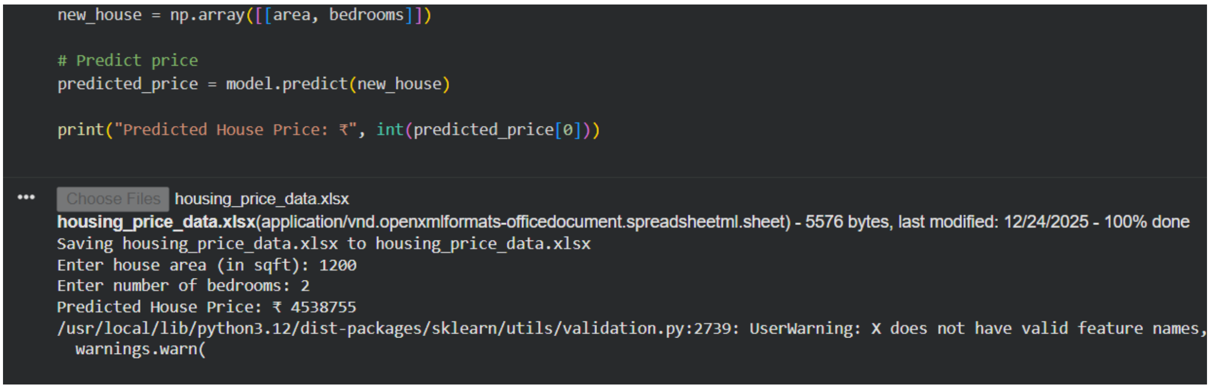
the new house based on the input features.

CODE:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression

# Load dataset from Excel
df = pd.read_excel("housing_price_data.xlsx")

# Features and target
X = df[["Area_sqft", "Bedrooms"]]
y = df["Price"]

# Train Linear Regression model
```

```
model = LinearRegression()

model.fit(X, y)


# User input

area = float(input("Enter house area (in sqft): "))

bedrooms = int(input("Enter number of bedrooms: "))


new_house = np.array([[area, bedrooms]])


# Predict price

predicted_price = model.predict(new_house)


print("Predicted House Price: ₹", int(predicted_price[0]))
```

Sample output:



```
new_house = np.array([[area, bedrooms]])

# Predict price
predicted_price = model.predict(new_house)

print("Predicted House Price: ₹", int(predicted_price[0]))
```

```
housing_price_data.xlsx
housing_price_data.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 5576 bytes, last modified: 12/24/2025 - 100% done
Saving housing_price_data.xlsx to housing_price_data.xlsx
Enter house area (in sqft): 1200
Enter number of bedrooms: 2
Predicted House Price: ₹ 4538755
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names,
  warnings.warn(
```

27.Question: Logistic Regression for Customer Churn Prediction

You are working for a telecommunications company, and you want to predict whether a customer will churn (leave the company) based on their usage patterns and demographic data. You have

collected a dataset of past customers with their churn status (0 for not churned, 1 for churned) and

various features.

Write a Python program that allows the user to input the features (e.g., usage minutes, contract

duration) of a new customer. The program should use logistic regression from scikit-learn to predict

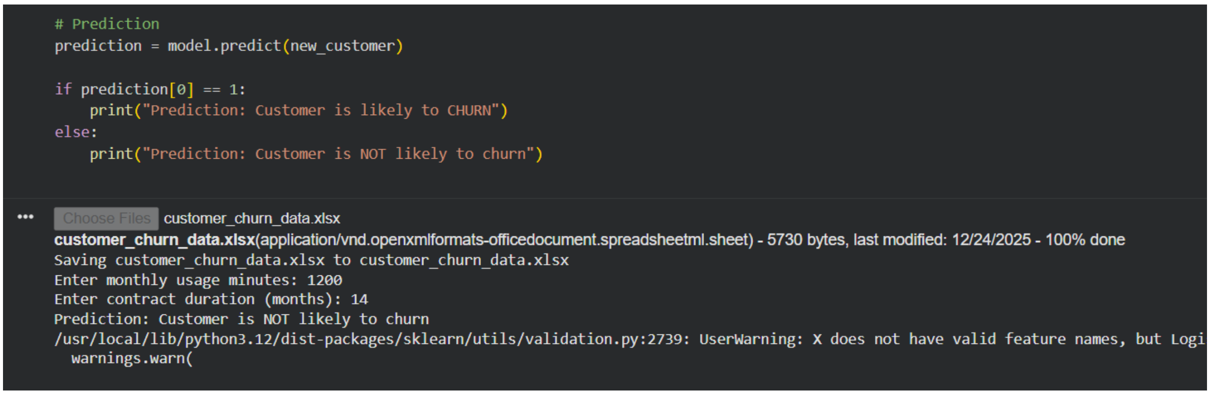
whether the new customer will churn or not based on the input features.

CODE:

```
import pandas as pd

import numpy as np

from sklearn.linear_model import LogisticRegression


# Load dataset from Excel

df = pd.read_excel("customer_churn_data.xlsx")


# Features and target

X = df[["Usage_Minutes", "Contract_Duration"]]

y = df["Churn"]
```

```
# Train Logistic Regression model
model = LogisticRegression()
model.fit(X, y)

# User input
usage_minutes = float(input("Enter monthly usage minutes: "))
contract_duration = int(input("Enter contract duration (months): "))

new_customer = np.array([[usage_minutes, contract_duration]])

# Prediction
prediction = model.predict(new_customer)

if prediction[0] == 1:
    print("Prediction: Customer is likely to CHURN")
else:
    print("Prediction: Customer is NOT likely to churn")
```

Sample output:



```
# Prediction
prediction = model.predict(new_customer)

if prediction[0] == 1:
    print("Prediction: Customer is likely to CHURN")
else:
    print("Prediction: Customer is NOT likely to churn")
```

```
customer_churn_data.xlsx
customer_churn_data.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 5730 bytes, last modified: 12/24/2025 - 100% done
Saving customer_churn_data.xlsx to customer_churn_data.xlsx
Enter monthly usage minutes: 1200
Enter contract duration (months): 14
Prediction: Customer is NOT likely to churn
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Logi
  warnings.warn(
```

## 28.Question: K-Means Clustering for Customer Segmentation

You are working for an e-commerce company and want to segment your customers into distinct

groups based on their purchasing behavior. You have collected a dataset of customer data with

various shopping-related features.

Write a Python program that allows the user to input the shopping-related features of a new

customer. The program should use K-Means clustering from scikit-learn to assign the new customer

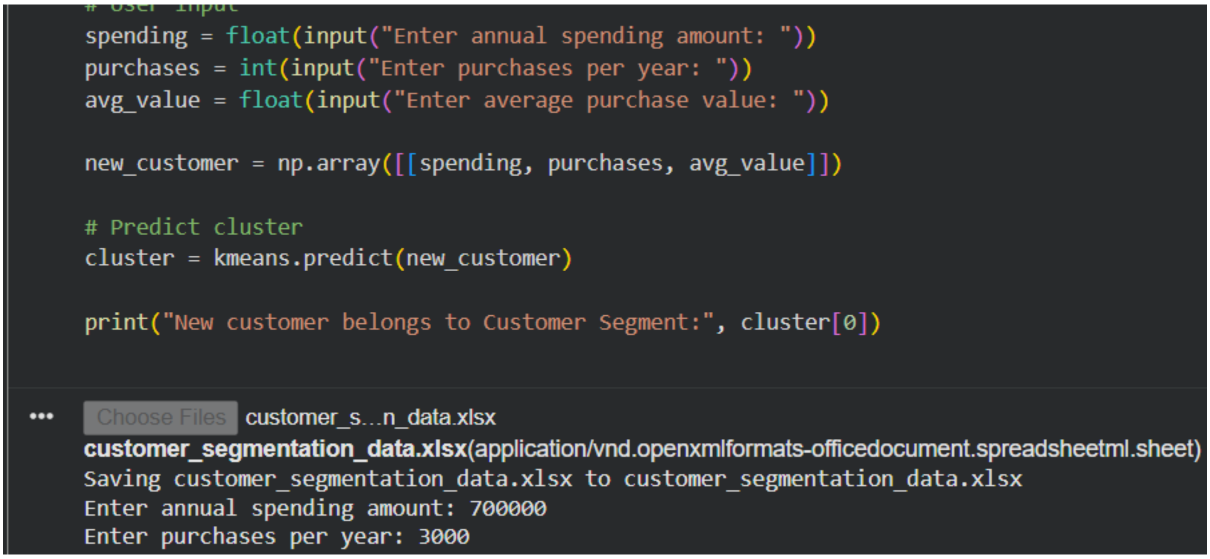
to one of the existing segments based on the input features.

CODE:

```
import pandas as pd

import numpy as np

from sklearn.cluster import KMeans


# Load dataset from Excel

df = pd.read_excel("customer_segmentation_data.xlsx")


X = df[["Annual_Spending", "Purchases_Per_Year", "Avg_Purchase_Value"]]


# Train K-Means model

kmeans = KMeans(n_clusters=3, random_state=42)
```

```
kmeans.fit(X)

# User input
spending = float(input("Enter annual spending amount: "))
purchases = int(input("Enter purchases per year: "))
avg_value = float(input("Enter average purchase value: "))

new_customer = np.array([[spending, purchases, avg_value]])

# Predict cluster
cluster = kmeans.predict(new_customer)

print("New customer belongs to Customer Segment:", cluster[0])
```

Sample output:



```
# User input
spending = float(input("Enter annual spending amount: "))
purchases = int(input("Enter purchases per year: "))
avg_value = float(input("Enter average purchase value: "))

new_customer = np.array([[spending, purchases, avg_value]])

# Predict cluster
cluster = kmeans.predict(new_customer)

print("New customer belongs to Customer Segment:", cluster[0])
```

```
customer_s...n_data.xlsx
customer_segmentation_data.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet)
Saving customer_segmentation_data.xlsx to customer_segmentation_data.xlsx
Enter annual spending amount: 700000
Enter purchases per year: 3000
```

29.Question: Evaluation Metrics for Model Performance

You have trained a machine learning model on a dataset, and now you want to evaluate its

performance using various metrics.

Write a Python program that loads a dataset and trained model from scikit-learn. The program

should ask the user to input the names of the features and the target variable they want to use for

evaluation. The program should then calculate and display common evaluation metrics such as

accuracy, precision, recall, and F1-score for the model's predictions on the test data.

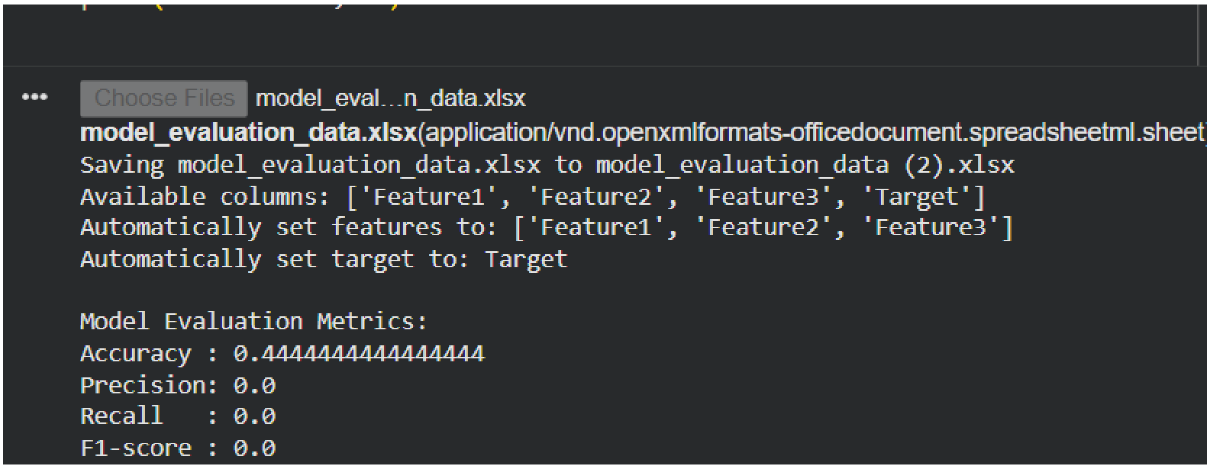
CODE:

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score


# Load dataset from Excel

df = pd.read_excel("model_evaluation_data.xlsx")


print("Available columns:", list(df.columns))
```

```
# User input for features and target
features = input("Enter feature column names separated by comma: ").split(",")
target = input("Enter target column name: ")

features = [f.strip() for f in features]

X = df[features]
y = df[target]

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# Train model
model = LogisticRegression()
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)

# Evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
```

```
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Output
print("\nModel Evaluation Metrics:")
print("Accuracy :", accuracy)
print("Precision:", precision)
print("Recall   :", recall)
print("F1-score :", f1)
```

Sample output:



```
model_eval...n_data.xlsx
model_evaluation_data.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
Saving model_evaluation_data.xlsx to model_evaluation_data (2).xlsx
Available columns: ['Feature1', 'Feature2', 'Feature3', 'Target']
Automatically set features to: ['Feature1', 'Feature2', 'Feature3']
Automatically set target to: Target

Model Evaluation Metrics:
Accuracy : 0.4444444444444444
Precision: 0.0
Recall   : 0.0
F1-score : 0.0
```

30.Question: Classification and Regression Trees (CART) for Car Price Prediction

You are working for a car dealership, and you want to predict the price of used cars based on various features such as the car's mileage, age, brand, and engine type. You have collected a dataset

of used cars with their respective prices.

Write a Python program that loads the car dataset and allows the user to input the features of a new

car they want to sell. The program should use the Classification and Regression Trees (CART)

algorithm from scikit-learn to predict the price of the new car based on the input features.

The CART algorithm will create a tree-based model that will split the data into subsets based on the

chosen features and their values, leading to a decision path that eventually predicts the price of the

car. The program should output the predicted price and display the decision path (the sequence of

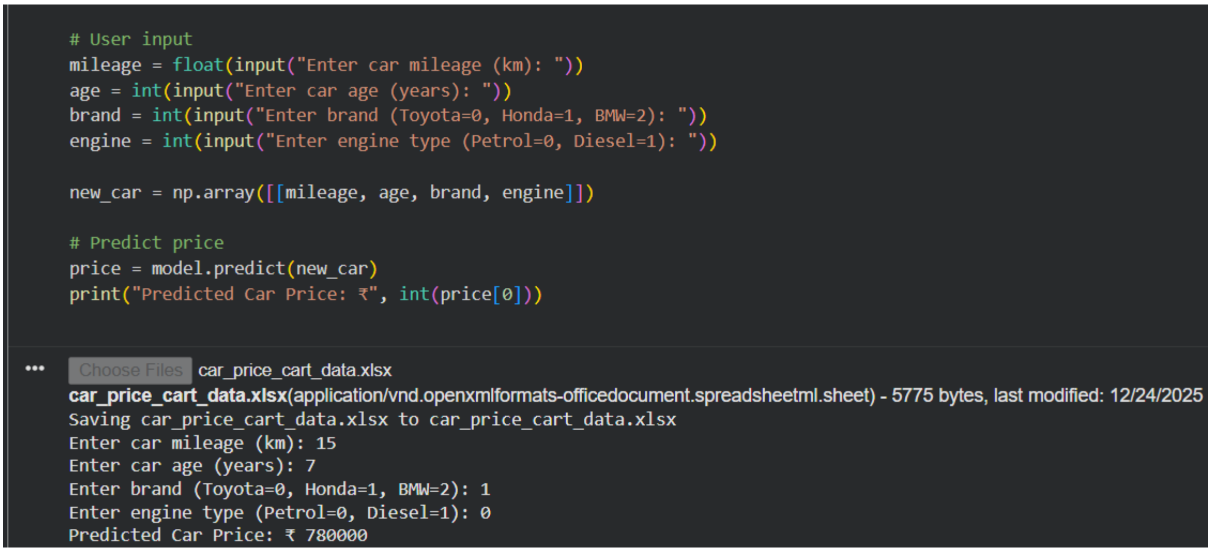
conditions leading to the prediction) for the new car.

CODE:

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeRegressor

# Load dataset from Excel
df = pd.read_excel("car_price_cart_data.xlsx")

X = df[["Mileage", "Age", "Brand", "Engine"]]
```

```
y = df["Price"]

# Train CART model
model = DecisionTreeRegressor(random_state=42)
model.fit(X, y)

# User input
mileage = float(input("Enter car mileage (km): "))
age = int(input("Enter car age (years): "))
brand = int(input("Enter brand (Toyota=0, Honda=1, BMW=2): "))
engine = int(input("Enter engine type (Petrol=0, Diesel=1): "))

new_car = np.array([[mileage, age, brand, engine]])
price = model.predict(new_car)
print("Predicted Car Price: ₹", int(price[0]))
```

Sample output:



```
# User input
mileage = float(input("Enter car mileage (km): "))
age = int(input("Enter car age (years): "))
brand = int(input("Enter brand (Toyota=0, Honda=1, BMW=2): "))
engine = int(input("Enter engine type (Petrol=0, Diesel=1): "))

new_car = np.array([[mileage, age, brand, engine]])

# Predict price
price = model.predict(new_car)
print("Predicted Car Price: ₹", int(price[0]))
```

```
car_price_cart_data.xlsx
car_price_cart_data.xlsx(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 5775 bytes, last modified: 12/24/2025
Saving car_price_cart_data.xlsx to car_price_cart_data.xlsx
Enter car mileage (km): 15
Enter car age (years): 7
Enter brand (Toyota=0, Honda=1, BMW=2): 1
Enter engine type (Petrol=0, Diesel=1): 0
Predicted Car Price: ₹ 780000
```