26.Linear Regression Model for Housing Price Prediction

Program:-

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import csv
# Create a dummy housing_data.csv file for demonstration
with open('housing_data.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(['square_footage', 'bedrooms', 'price'])
    writer.writerow([1500, 3, 250000])
    writer.writerow([2000, 4, 350000])
    writer.writerow([1200, 2, 200000])
    writer.writerow([1800, 3, 300000])
    writer.writerow([2200, 4, 380000])
data = pd.read_csv("housing_data.csv")
X = data.drop("price", axis=1)
y = data["price"]
model = LinearRegression()
model.fit(X, y)
print("Enter new house details:")
user_input = []
for col in X.columns:
    val = float(input(f"{col}: "))
    user_input.append(val)
prediction = model.predict([user_input])
print("Predicted House Price:", prediction[0])
```

Output:-

```
Enter new house details:

square_footage: 1700

bedrooms: 4

Predicted House Price: 298241.7582417583
```

27.Logistic Regression Model for Customer Churn Prediction

Program:-

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
import csv
# Create a dummy churn_data.csv file for demonstration
with open('churn_data.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(['customer_id', 'age', 'tenure', 'monthly_charges', 'churn'])
    writer.writerow([1, 30, 12, 50.0, 0])
    writer.writerow([2, 45, 24, 75.0, 0])
    writer.writerow([3, 22, 5, 60.0, 1])
    writer.writerow([4, 50, 36, 80.0, 0])
    writer.writerow([5, 28, 8, 55.0, 1])
data = pd.read_csv("churn_data.csv")
X = data.drop("churn", axis=1)
y = data["churn"]
model = LogisticRegression()
model.fit(X, y)
print("Enter new customer details:")
new = []
for col in X.columns:
    val = float(input(f"{col}: "))
    new.append(val)
pred = model.predict([new])
if pred[0] == 1:
    print("Prediction: Customer WILL churn")
else:
    print("Prediction: Customer will NOT churn")
```

Output:-

```
Enter new customer details:

customer_id: 3

age: 18

tenure: 6

monthly_charges: 60

Prediction: Customer WILL churn
```

28.K-Means Clustering for Customer Segmentation

Program:-

```
import pandas as pd
from sklearn.cluster import KMeans
import csv
# Create a dummy customer_segments.csv file for demonstration
with open('customer_segments.csv', 'w', newline='') as f:
    writer = csv.writer(f)
    writer.writerow(['age', 'income', 'spending_score'])
    writer.writerow([30, 50000, 70])
    writer.writerow([45, 75000, 40])
    writer.writerow([22, 30000, 85])
    writer.writerow([50, 100000, 20])
    writer.writerow([35, 60000, 65])
    writer.writerow([28, 40000, 90])
    writer.writerow([55, 120000, 15])
    writer.writerow([40, 70000, 50])
data = pd.read_csv("customer_segments.csv")
X = data
kmeans = KMeans(n_clusters=4, random_state=42, n_init='auto') # Added random_state and n_init for reproducibility and to suppress future warnings
kmeans.fit(X)
print("Enter new customer details:")
user = []
for col in X.columns:
    val = float(input(f"{col}: "))
    user.append(val)
cluster = kmeans.predict([user])
print("Customer belongs to Segment:", cluster[0])
```

Output:-

```
Enter new customer details:

age: 22

income: 35000

spending_score: 87

Customer belongs to Segment: 0
```

29.Model Evaluation Using Accuracy, Precision, Recall, and F1-score

Program:-

```
from sklearn.datasets import load_breast_cancer

from sklearn.model_selection import train_test_split

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

from sklearn.tree import DecisionTreeClassifier

import pandas as pd

data = load_breast_cancer()

df = pd.DataFrame(data.data, columns=data.feature_names)

df['target'] = data.target

features = input("Enter feature names (comma separated): ").split(",")

target = "target"

X = df[features]

y = df[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = DecisionTreeClassifier()

model.fit(X_train, y_train)

pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, pred))

print("Precision:", precision_score(y_test, pred))

print("Recall:", recall_score(y_test, pred))

print("F1 Score:", f1_score(y_test, pred))
```

Output:-

```
Enter feature names (comma separated): mean radius

Accuracy: 0.8245614035087719

Precision: 0.9

Recall: 0.7941176470588235

F1 Score: 0.84375
```

30.CART Decision Tree Model for Used Car Price Prediction

Program:-

```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor, export_text
import csv
# Create a dummy car_data.csv file for demonstration
with open('car_data.csv', 'w', newline=") as f:
    writer = csv.writer(f)
    writer.writerow(['engine_size', 'horsepower', 'price'])
    writer.writerow([1.6, 120, 18000])
    writer.writerow([2.0, 150, 25000])
    writer.writerow([1.8, 130, 20000])
    writer.writerow([2.5, 180, 32000])
    writer.writerow([1.5, 110, 16000])
data = pd.read_csv("car_data.csv")
X = data.drop("price", axis=1)
y = data["price"]
model = DecisionTreeRegressor()
model.fit(X, y)
print("Enter new car details:")
user_input = []
for col in X.columns:
    val = float(input(f"{col}: "))
    user_input.append(val)
prediction = model.predict([user_input])
print("Predicted Car Price:", prediction[0])
tree_rules = export_text(model, feature_names=list(X.columns))
print("\nDecision Path:\n", tree_rules)
```

Output:-

```
Enter new car details:

engine_size: 2

horsepower: 150

Predicted Car Price: 25000.0

Decision Path:

 |--- engine_size <= 1.90

|   |--- horsepower <= 115.00

|   |   |--- value: [16000.00]

|   |--- horsepower >  115.00

|   |   |--- engine_size <= 1.70

|   |   |   |--- value: [18000.00]

|   |   |--- engine_size >  1.70

|   |   |   |--- value: [20000.00]

|--- engine_size >  1.90

|   |--- engine_size <= 2.25

|   |   |--- value: [25000.00]

|   |--- engine_size >  2.25

|   |   |--- value: [32000.00]
```