

R SATHISH (192324204)

DATE:11/02/2026

Experiment:26



```
[1]: import pandas as pd
from sklearn.linear_model import LinearRegression

df = pd.read_csv("housing_data.csv")
X,y = df[["Area","Bedrooms"]], df["Price"]

m = LinearRegression().fit(X,y)

a = float(input("Area: "))
b = int(input("Bedrooms: "))
print("Predicted Price:", round(m.predict([[a,b]])[0],2))

Area: 524
Bedrooms: 4
Predicted Price: 28.04
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LinearRegression.fit requires them.
warnings.warn(
```

EXP:27



```
[4]: import pandas as pd
from sklearn.linear_model import LogisticRegression

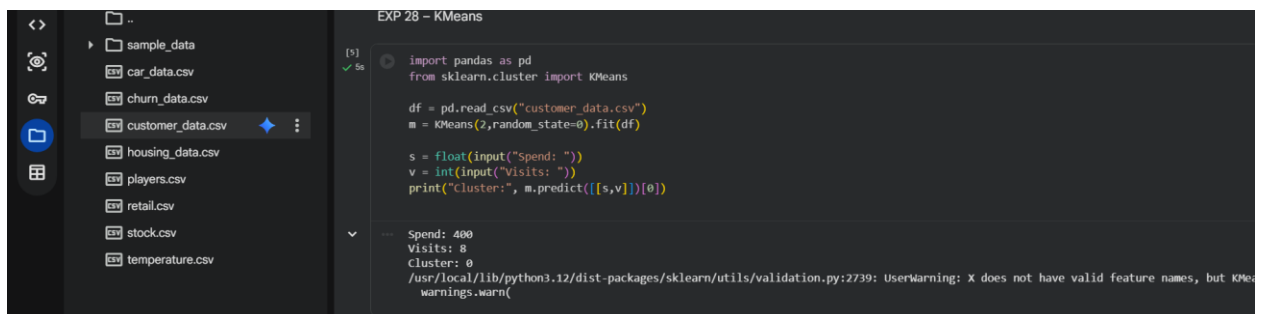
df = pd.read_csv("churn_data.csv")
X,y = df[["Minutes","Contract"]], df["Churn"]

m = LogisticRegression().fit(X,y)

x = float(input("Minutes: "))
y = int(input("Contract: "))
print("Churn:", m.predict([[x,y]])[0])

Minutes: 350
Contract: 8
Churn: 1
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but LogisticRegression.fit requires them.
warnings.warn(
```

EXP:28



```
[5]: import pandas as pd
from sklearn.cluster import KMeans

df = pd.read_csv("customer_data.csv")
m = KMeans(2,random_state=0).fit(df)

s = float(input("Spend: "))
v = int(input("Visits: "))
print("Cluster:", m.predict([[s,v]])[0])

Spend: 400
Visits: 8
Cluster: 0
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but KMeans.fit requires them.
warnings.warn(
```

EXP:29

```
EXP 29 – Evaluation Metric

[6] ✓ 0s
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import *

X,y = load_iris(return_X_y=True)
X1,X2,y1,y2 = train_test_split(X,y,test_size=0.3)

m = DecisionTreeClassifier().fit(X1,y1)
p = m.predict(X2)

print("Acc:",accuracy_score(y2,p))
print("Prec:",precision_score(y2,p,average='macro'))
print("Rec:",recall_score(y2,p,average='macro'))
print("F1:",f1_score(y2,p,average='macro'))

Acc: 0.9555555555555556
Prec: 0.9555555555555556
Rec: 0.9583333333333334
F1: 0.953968253968254
```

EXP:30

```
EXP 30 – CART (Decision Tree Regression)

[8] ✓ 4ms
import pandas as pd
from sklearn.tree import DecisionTreeRegressor

df = pd.read_csv("car_data.csv")
X,y = df[["Mileage","Age"]], df["Price"]
m = DecisionTreeRegressor().fit(X,y)

mi = float(input("Mileage: "))
ag = int(input("Age: "))
print("Price:", m.predict([[mi,ag]])[0])

... Mileage: 25000
Price: 750000.0
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Deci
warnings.warn(
```