DAY-5

DATE:22/1/26

26.Question : Linear Regression for Housing Price Prediction

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Load dataset
data = pd.read_csv("housing.csv")

# Features and target
X = data[['Area', 'Bedrooms', 'Location']]
y = data['Price']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

# User input
area = float(input("Enter house area: "))
bedrooms = int(input("Enter number of bedrooms: "))
location = int(input("Enter location code: "))

# Prediction
new_house = [[area, bedrooms, location]]
predicted_price = model.predict(new_house)
```

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

OUTPUT:

```
Enter house area (sq ft): 1000
Enter number of bedrooms: 3
Enter location code: 1
Predicted House Price: 3613333
```

27.Question: Logistic Regression for Customer Churn Prediction

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression

# Load dataset
data = pd.read_csv("customer_churn.csv")

# Features and target
X = data[['UsageMinutes', 'ContractDuration']]
y = data['Churn']  # 0 = Not Churned, 1 = Churned

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

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

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

# Prediction
new_customer = [[usage, contract]]
```

```
prediction = model.predict(new_customer)

if prediction[0] == 1:
    print("Customer will churn")
else:
    print("Customer will not churn")
```

33. Scenario: You work as a data scientist for an automobile company that sells various car models.

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
data = pd.read_csv("car_data.csv")

# Select features and target
X = data[['EngineSize', 'Horsepower', 'FuelEfficiency']]
y = data['Price']

# Train Linear Regression model
model = LinearRegression()
model.fit(X, y)

# Predict prices
y_pred = model.predict(X)

# Model evaluation
mse = mean_squared_error(y, y_pred)
r2 = r2_score(y, y_pred)
```

```
print("Mean Squared Error:", mse)
print("R-squared Score:", r2)

# Feature importance (coefficients)
coefficients = pd.DataFrame({
    'Feature': X.columns,
    'Impact': model.coef_
})

print("\nInfluential factors affecting car prices:")
print(coefficients)
```

OUTPUT:

```
Accuracy: 0.96
```

21 write a Python program that allows the user to input the sample size, confidence level, and desired level of precision.

```
import numpy as np
import pandas as pd
from scipy.stats import norm
data = pd.read_csv("rare_elements.csv", header=None)[0].values
n = int(input("Sample size: "))
cl = float(input("Confidence level: "))
sample = np.random.choice(data, n, replace=False)
mean = np.mean(sample)
std = np.std(sample, ddof=1)
z = norm.ppf((1 + cl) / 2)
me = z * std / np.sqrt(n)
print("Mean:", mean)
print("Confidence Interval:", (mean - me, mean + me))
```

```
Average Rating: 4.15
95% Confidence Interval: (np.float64(3.8234253105565887), np.float64(4.476574689443412))
```

```
from sklearn.datasets import make_classification
```