**Assignment No : 2**

import pandas as pd

file\_path = "/content/sample\_data/kaggle2.csv" # Replace with your file path

df = pd.read\_csv(file\_path)

# Assuming the units are grams (g), convert to kilograms (kg)

df['Brain\_Weight(kg)'] = df['brain\_weight'] / 1000 # Replace 'Brain Weight' with the exact column name

df = df[['body\_weight', 'Brain\_Weight(kg)']]

output\_path = "/content/sample\_data/converted\_file.csv" # Replace with your desired output file path

df.to\_csv(output\_path, index=False)

print(f"Conversion complete! File saved to {output\_path}")

**Output:** Conversion complete! File saved to /content/sample\_data/converted\_file.csv

import pandas as pd

from sklearn.model\_selection import KFold, cross\_val\_score

from sklearn.preprocessing import RobustScaler

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import make\_scorer, r2\_score

import numpy as np

file\_path = "/content/sample\_data/converted\_file.csv" # Replace with your dataset path

data = pd.read\_csv(file\_path)

print(data.head())

# Assume the dataset has columns 'Weight (kg)' and 'Brain Weight (kg)'

features = data[['body\_weight']] # Use only body weight as input

target = data['Brain\_Weight(kg)'] # Brain weight is the target

# Handling outliers using RobustScaler (scaled based on median and IQR)

scaler = RobustScaler()

features\_scaled = scaler.fit\_transform(features)

model = LinearRegression()

# K-Fold Cross-Validation

k = 2

kf = KFold(n\_splits=k, shuffle=True, random\_state=42)

scorer = make\_scorer(r2\_score)

r2\_scores = cross\_val\_score(model, features\_scaled, target, cv=kf, scoring=scorer)

# Output the results

print(f"R² Scores for each fold: {r2\_scores}")

print(f"Mean R² Score: {np.mean(r2\_scores):.4f}")

print(f"Standard Deviation of R² Scores: {np.std(r2\_scores):.4f}")

model.fit(features\_scaled, target)

try:

weight = float(input("Enter Weight (kg): ")) # User provides body weight

# Normalize the input

user\_input\_scaled = scaler.transform([[weight]])

# Predict brain weight

predicted\_brain\_weight = model.predict(user\_input\_scaled)

print(f"Predicted Brain Weight (kg): {predicted\_brain\_weight[0]:.4f}")

except ValueError:

print("Invalid input. Please enter a numerical value for Weight.")

**Output:** body\_weight Brain\_Weight(kg)

0 6654.000 5.7120

1 1.000 0.0066

2 3.385 0.0445

3 0.920 0.0057

4 2547.000 4.6030

R² Scores for each fold: [0.82263945 0.66804191]

Mean R² Score: 0.7453

Standard Deviation of R² Scores: 0.0773

Enter Weight (kg): 187

Predicted Brain Weight (kg): 0.2717