

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