Goal of the Project

This project is designed for you to practice and solve the activities that are based on the concepts covered in Simple linear regression & Model Evaluation:

Problem Statement

The most important factor for an Insurance Company is to determine what premium charges must be paid by an individual. The charges depend on various factors like age, gender, income, etc.

Build a model that is capable of predicting the insurance charges a person has to pay depending on his/her age using simple linear regression. Also, evaluate the accuracy of your model by calculating the value of error metrics such as R-squared, MSE, RMSE, and MAE.

List of Activities

Activity 1: Analysing the Dataset

Activity 2: Train-Test Split

Activity 3: Model Training

Activity 4: Model Prediction and Evaluation

Activity 1: Analysing the Dataset

- Create a Pandas DataFrame for **Insurance** dataset using the below link. This dataset consists of following columns:

Field	Description
age	Age of primary beneficiary
sex	Insurance contractor gender, female or male
bmi	Body mass index
children	Number of children covered by health insurance/number of dependents
region	Beneficiary's residential area in the US, northeast, southeast, southwest, northwest
charges	Individual medical costs billed by health insurance

Dataset Link: https://raw.githubusercontent.com/jiss-sngce/CO_3/main/insurance_dataset.csv

- Print the first five rows of the dataset. Check for null values and treat them accordingly.

- Create a regression plot with `age` on X-axis and `charges` on Y-axis to identify the relationship between these two attributes.

```
# Import modules
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt



# Load the dataset
df=pd.read_csv('https://raw.githubusercontent.com/jiss-sngce/CO_3/main/insurance_datase


# Print first five rows using head() function
df.head()

```

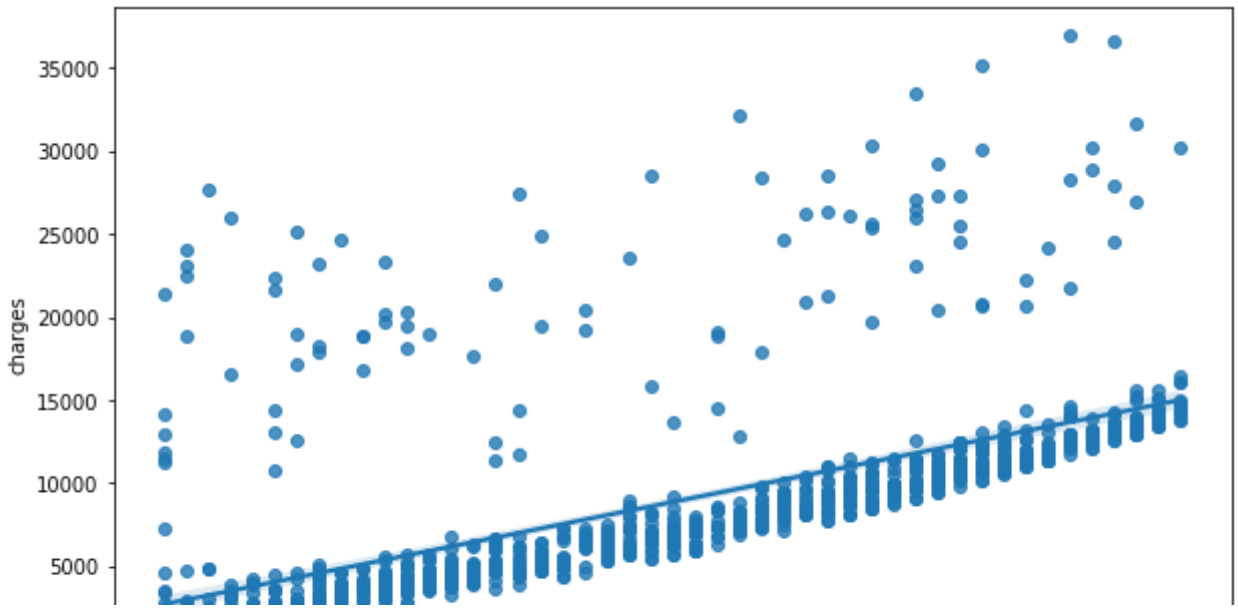
	age	sex	bmi	children	region	charges
0	18	male	33.770	1	southeast	1725.55230
1	28	male	33.000	3	southeast	4449.46200
2	33	male	22.705	0	northwest	21984.47061
3	32	male	28.880	0	northwest	3866.85520
4	31	female	25.740	0	southeast	3756.62160

```
# Check if there are any null values. If any column has null values, treat them accordi
df.isnull().sum()

```

```
age         0
sex         0
bmi         0
children    0
region      0
charges     0
dtype: int64
```

```
from seaborn.regression import regplot
# Create a regression plot between 'age' and 'charges'
plt.figure(figsize=(10,6))
sns=regplot(x='age',y='charges',data=df)
plt.show()
```



Activity 2: Train-Test Split

We have to determine the effect of `age` on insurance charges. Thus, `age` is the feature variable and `charges` is the target variable.

Split the dataset into training set and test set such that the training set contains 67% of the instances and the remaining instances will become the test set.

```
1 from os import X_OK
2 # Split the DataFrame into the training and test sets.
3 from sklearn.model_selection import train_test_split
4 x=df['age']
5 y=df['charges']
6 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.33,random_state=42)
7 x_train
8
9
```

```
239     49
361     56
211     19
976     40
594     45
        ..
330     45
466     39
121     37
1044    18
860     47
Name: age, Length: 712, dtype: int64
```

Activity 3: Model Training

Implement simple linear regression using `sklearn` module in the following way:

1. Reshape the feature and the target variable arrays into two-dimensional arrays by using `reshape(-1, 1)` function of numpy module.
2. Deploy the model by importing the `LinearRegression` class and create an object of this class.
3. Call the `fit()` function on the LinearRegression object and print the slope and intercept values of the best fit line.

```
# 1. Create two-dimensional NumPy arrays for the feature and target variables.
x_test_res=x_test.values.reshape(-1,1)
x_train_res=x_train.values.reshape(-1,1)
y_train_res=y_train.values.reshape(-1,1)
y_test_res=y_test.values.reshape(-1,1)



# Print the shape or dimensions of these reshaped arrays
print(x_test_res.shape)
print(x_train_res.shape)
print(y_train_res.shape)
print(y_test_res.shape)
```

```
(352, 1)
(712, 1)
(712, 1)
(352, 1)
```

```
# 2. Deploy linear regression model using the 'sklearn.linear_model' module.
from sklearn.linear_model import LinearRegression


# Create an object of the 'LinearRegression' class.
lin_reg=LinearRegression()


# 3. Call the 'fit()' function
lin_reg.fit(x_train_res,y_train_res)



# Print the slope and intercept values
print(lin_reg.coef_)
```

```
16 print(lin_reg.intercept_)
17
```

```
[[258.95102199]]
```

Activity 4: Model Prediction and Evaluation

Predict the values for both training and test sets by calling the `predict()` function on the LinearRegression object. Also, calculate the R^2, MSE, RMSE and MAE values to evaluate the accuracy of your model.

```
 1 from typing import Set
 2 from math import sqrt
 3 # Predict the target variable values for both training set and test set
 4 from sklearn .metrics import r2_score,mean_squared_error,mean_absolute_error
 5 y_train_pred=lin_reg.predict(x_train_res)
 6 y_test_pred=lin_reg.predict(x_test_res)
 7
 8
 9
10
11 # Call 'r2_score', 'mean_squared_error' & 'mean_absolute_error' functions of the 'sklea
12
13 # Print these values for both training set and test set
14 print("Train Set")
15 print('R_Squared:',r2_score(y_train_res,y_train_pred))
16 print('Root Mean Squared Error:',mean_squared_error(y_train_res,y_train_pred))
17 print('Root Mean Squared Error:',np.sqrt(mean_squared_error(y_train_res,y_train_pred)))
18 print('Mean Absolute Error:',mean_absolute_error(y_train_res,y_train_pred))
19
20 print('Testset')
21 print('R_Squared:',r2_score(y_test_res,y_test_pred))
22 print('Root Mean Squared Error:',mean_squared_error(y_test_res,y_test_pred))
23 print('Root Mean Squared Error:',np.sqrt(mean_squared_error(y_test_res,y_test_pred)))
24 print('Mean Absolute Error:',mean_absolute_error(y_test_res,y_test_pred))
25
26
27
28
```

```
Train Set
R_Squared: 0.37535676235236426
Root Mean Squared Error: 22108233.951971985
Root Mean Squared Error: 4701.939381996751
Mean Absolute Error: 2505.0324439849537
Testset
R_Squared: 0.429634671852961
Root Mean Squared Error: 21039679.07376732
Root Mean Squared Error: 4586.902993716711
Mean Absolute Error: 2649.282252871463
```