# PRCP-1021-InsCostPred

# PROBLEM STATEMENT

--> This project is about predicting medical expenses in order to determine premiums , to assess the risks and to ensure profitability. Here, predicting medical expenses depend on the factors like age, BMI, smoking habits, gender, dependents and regions.

--> The objective is to analyze the given dataset, uncover patterns and build machine learning models to predict individual charges. This involves addressing the challenges such as the influence of lifestyle, habits, regional differences by preparing the data, identifying key features and evaluating multiple ML models.

--> The results will contribute to minimizing financial risks, improving risk assessment, determining premiums in the insurance sector and enhancing customer benefits.

# DOMAIN ANALYSIS AND ATTRIBUTES

This project falls under "Health Insurance" domain focusing on predicting individual charges based on the key attributes such as,

1. Age : Represents the age of an insured person in years.
2. Sex : Indicates the gender of an insured person, categorized as male or female.
3. BMI : Provides the Body Mass Index (BMI) of each individual (Ideal BMI ranges from 18.5 to 24.9)
4. Children : Indicates the number of children covered by health insurance.
5. Smoker : This column will let us know whether an insured person is a smoker or non-smoker.
6. Region : Includes the geographical regions where the insured person resides, such as Northeast, Northwest, Southeast, and Southwest.
7. Charges : This is the target column which represents the insurance premium or charges that the insured person is required to pay.

# 1. IMPORTING LIBRARIES

```
pip install xgboost
```

```
Collecting xgboostNote: you may need to restart the kernel to use
updated packages.

  Downloading xgboost-3.0.0-py3-none-win_amd64.whl.metadata (2.1 kB)
Requirement already satisfied: numpy in c:\users\supri\anaconda3\lib\
site-packages (from xgboost) (1.26.4)
Requirement already satisfied: scipy in c:\users\supri\anaconda3\lib\
site-packages (from xgboost) (1.13.1)
Downloading xgboost-3.0.0-py3-none-win_amd64.whl (150.0 MB)
   ---------------------------------------- 0.0/150.0 MB ? eta -:--:--
   ---------------------------------------- 0.0/150.0 MB ? eta -:--:--
   ---------------------------------------- 0.0/150.0 MB 435.7 kB/s
eta 0:05:45
   ---------------------------------------- 0.1/150.0 MB 544.7 kB/s
eta 0:04:36
   ---------------------------------------- 0.2/150.0 MB 1.4 MB/s eta
0:01:48
   ---------------------------------------- 0.5/150.0 MB 2.4 MB/s eta
0:01:03
   ---------------------------------------- 0.8/150.0 MB 3.4 MB/s eta
0:00:44
   ---------------------------------------- 1.3/150.0 MB 4.5 MB/s eta
0:00:33
   ---------------------------------------- 1.7/150.0 MB 5.1 MB/s eta
0:00:30
    --------------------------------------- 2.0/150.0 MB 5.4 MB/s eta
0:00:28
    --------------------------------------- 2.4/150.0 MB 5.6 MB/s eta
0:00:27
    --------------------------------------- 2.7/150.0 MB 5.6 MB/s eta
0:00:27
    --------------------------------------- 3.0/150.0 MB 5.9 MB/s eta
0:00:25
    --------------------------------------- 3.4/150.0 MB 6.0 MB/s eta
0:00:25
   - -------------------------------------- 3.8/150.0 MB 6.2 MB/s eta
0:00:24
   - -------------------------------------- 4.1/150.0 MB 6.3 MB/s eta
0:00:24
   - -------------------------------------- 4.6/150.0 MB 6.5 MB/s eta
0:00:23
   - -------------------------------------- 4.8/150.0 MB 6.4 MB/s eta
0:00:23
   - -------------------------------------- 5.1/150.0 MB 6.4 MB/s eta
0:00:23
   - -------------------------------------- 5.5/150.0 MB 6.5 MB/s eta
0:00:23
   - -------------------------------------- 6.0/150.0 MB 6.7 MB/s eta
0:00:22
   - -------------------------------------- 6.3/150.0 MB 6.7 MB/s eta
```

```
0:00:22
   - -------------------------------------- 6.7/150.0 MB 6.8 MB/s eta
0:00:22
   - -------------------------------------- 7.2/150.0 MB 7.0 MB/s eta
0:00:21
   -- ------------------------------------- 7.7/150.0 MB 7.1 MB/s eta
0:00:21
   -- ------------------------------------- 8.1/150.0 MB 7.2 MB/s eta
0:00:20
   -- ------------------------------------- 8.4/150.0 MB 7.2 MB/s eta
0:00:20
   -- ------------------------------------- 8.9/150.0 MB 7.3 MB/s eta
0:00:20
   -- ------------------------------------- 9.3/150.0 MB 7.3 MB/s eta
0:00:20
   -- ------------------------------------- 9.8/150.0 MB 7.4 MB/s eta
0:00:19
   -- ------------------------------------- 10.1/150.0 MB 7.5 MB/s eta
0:00:19
   -- ------------------------------------- 10.6/150.0 MB 8.4 MB/s eta
0:00:17
   -- ------------------------------------- 11.0/150.0 MB 8.3 MB/s eta
0:00:17
   --- ------------------------------------ 11.4/150.0 MB 8.4 MB/s eta
0:00:17
   --- ------------------------------------ 11.8/150.0 MB 8.3 MB/s eta
0:00:17
   --- ------------------------------------ 12.2/150.0 MB 8.3 MB/s eta
0:00:17
   --- ------------------------------------ 12.7/150.0 MB 8.6 MB/s eta
0:00:16
   --- ------------------------------------ 12.8/150.0 MB 8.6 MB/s eta
0:00:16
   --- ------------------------------------ 12.8/150.0 MB 8.6 MB/s eta
0:00:16
   --- ------------------------------------ 14.2/150.0 MB 8.8 MB/s eta
0:00:16
   --- ------------------------------------ 14.6/150.0 MB 9.1 MB/s eta
0:00:15
   ---- ----------------------------------- 15.1/150.0 MB 9.2 MB/s eta
0:00:15
   ---- ----------------------------------- 15.6/150.0 MB 9.4 MB/s eta
0:00:15
   ---- ----------------------------------- 16.0/150.0 MB 9.4 MB/s eta
0:00:15
   ---- ----------------------------------- 16.5/150.0 MB 9.5 MB/s eta
0:00:15
   ---- ----------------------------------- 16.7/150.0 MB 9.6 MB/s eta
0:00:14
   ---- ----------------------------------- 16.9/150.0 MB 9.0 MB/s eta
```

```
0:00:15
   ---- ----------------------------------- 17.5/150.0 MB 9.4 MB/s eta
0:00:15
   ---- ----------------------------------- 18.0/150.0 MB 9.2 MB/s eta
0:00:15
   ---- ----------------------------------- 18.3/150.0 MB 9.1 MB/s eta
0:00:15
   ---- ----------------------------------- 18.5/150.0 MB 9.0 MB/s eta
0:00:15
   ----- ---------------------------------- 18.9/150.0 MB 9.0 MB/s eta
0:00:15
   ----- ---------------------------------- 19.4/150.0 MB 9.1 MB/s eta
0:00:15
   ----- ---------------------------------- 19.9/150.0 MB 9.0 MB/s eta
0:00:15
   ----- ---------------------------------- 20.2/150.0 MB 8.8 MB/s eta
0:00:15
   ----- ---------------------------------- 20.4/150.0 MB 8.8 MB/s eta
0:00:15
   ----- ---------------------------------- 20.7/150.0 MB 8.7 MB/s eta
0:00:15
   ----- ---------------------------------- 20.9/150.0 MB 8.5 MB/s eta
0:00:16
   ----- ---------------------------------- 21.1/150.0 MB 8.4 MB/s eta
0:00:16
   ----- ---------------------------------- 21.2/150.0 MB 8.2 MB/s eta
0:00:16
   ----- ---------------------------------- 21.3/150.0 MB 8.0 MB/s eta
0:00:17
   ----- ---------------------------------- 21.4/150.0 MB 7.7 MB/s eta
0:00:17
   ----- ---------------------------------- 21.6/150.0 MB 7.5 MB/s eta
0:00:18
   ----- ---------------------------------- 21.8/150.0 MB 7.4 MB/s eta
0:00:18
   ----- ---------------------------------- 22.2/150.0 MB 7.4 MB/s eta
0:00:18
   ----- ---------------------------------- 22.3/150.0 MB 7.3 MB/s eta
0:00:18
   ------ --------------------------------- 22.5/150.0 MB 7.2 MB/s eta
0:00:18
   ------ --------------------------------- 22.5/150.0 MB 7.2 MB/s eta
0:00:18
   ------ --------------------------------- 22.5/150.0 MB 7.2 MB/s eta
0:00:18
   ------ --------------------------------- 23.3/150.0 MB 7.2 MB/s eta
0:00:18
   ------ --------------------------------- 23.5/150.0 MB 7.0 MB/s eta
0:00:18
   ------ --------------------------------- 23.7/150.0 MB 6.8 MB/s eta
```

```
0:00:19
   ------ --------------------------------- 23.8/150.0 MB 6.5 MB/s eta
0:00:20
   ------ --------------------------------- 23.9/150.0 MB 6.4 MB/s eta
0:00:20
   ------ --------------------------------- 24.1/150.0 MB 6.2 MB/s eta
0:00:21
   ------ --------------------------------- 24.4/150.0 MB 6.1 MB/s eta
0:00:21
   ------ --------------------------------- 24.4/150.0 MB 6.0 MB/s eta
0:00:22
   ------ --------------------------------- 24.7/150.0 MB 5.9 MB/s eta
0:00:22
   ------ --------------------------------- 25.1/150.0 MB 5.8 MB/s eta
0:00:22
   ------ --------------------------------- 25.5/150.0 MB 5.8 MB/s eta
0:00:22
   ------ --------------------------------- 25.9/150.0 MB 5.8 MB/s eta
0:00:22
   ------ --------------------------------- 26.1/150.0 MB 5.7 MB/s eta
0:00:22
   ------- -------------------------------- 26.5/150.0 MB 5.7 MB/s eta
0:00:22
   ------- -------------------------------- 26.9/150.0 MB 5.6 MB/s eta
0:00:22
   ------- -------------------------------- 27.3/150.0 MB 5.8 MB/s eta
0:00:21
   ------- -------------------------------- 27.6/150.0 MB 5.7 MB/s eta
0:00:22
   ------- -------------------------------- 28.0/150.0 MB 5.7 MB/s eta
0:00:22
   ------- -------------------------------- 28.1/150.0 MB 5.6 MB/s eta
0:00:22
   ------- -------------------------------- 28.4/150.0 MB 5.6 MB/s eta
0:00:22
   ------- -------------------------------- 28.6/150.0 MB 5.5 MB/s eta
0:00:22
   ------- -------------------------------- 28.8/150.0 MB 5.5 MB/s eta
0:00:23
   ------- -------------------------------- 28.9/150.0 MB 5.5 MB/s eta
0:00:23
   ------- -------------------------------- 29.1/150.0 MB 5.3 MB/s eta
0:00:23
   ------- -------------------------------- 29.3/150.0 MB 5.2 MB/s eta
0:00:24
   ------- -------------------------------- 29.4/150.0 MB 5.2 MB/s eta
0:00:24
   ------- -------------------------------- 29.8/150.0 MB 5.1 MB/s eta
0:00:24
   ------- -------------------------------- 30.0/150.0 MB 5.1 MB/s eta
```

```
0:00:24
   -------- ------------------------------- 30.4/150.0 MB 5.1 MB/s eta
0:00:24
   -------- ------------------------------- 30.6/150.0 MB 5.0 MB/s eta
0:00:24
   -------- ------------------------------- 30.9/150.0 MB 5.2 MB/s eta
0:00:24
   -------- ------------------------------- 31.2/150.0 MB 5.1 MB/s eta
0:00:24
   -------- ------------------------------- 31.5/150.0 MB 5.2 MB/s eta
0:00:23
   -------- ------------------------------- 31.7/150.0 MB 5.4 MB/s eta
0:00:23
   -------- ------------------------------- 32.1/150.0 MB 5.5 MB/s eta
0:00:22
   -------- ------------------------------- 32.3/150.0 MB 5.5 MB/s eta
0:00:22
   -------- ------------------------------- 32.5/150.0 MB 5.4 MB/s eta
0:00:22
   -------- ------------------------------- 32.8/150.0 MB 5.9 MB/s eta
0:00:20
   -------- ------------------------------- 33.0/150.0 MB 5.7 MB/s eta
0:00:21
   -------- ------------------------------- 33.4/150.0 MB 5.6 MB/s eta
0:00:21
   -------- ------------------------------- 33.6/150.0 MB 5.6 MB/s eta
0:00:21
   --------- ------------------------------ 33.8/150.0 MB 5.6 MB/s eta
0:00:21
   --------- ------------------------------ 34.2/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 34.5/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 34.7/150.0 MB 6.0 MB/s eta
0:00:20
   --------- ------------------------------ 34.9/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 35.1/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 35.4/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 35.8/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 36.1/150.0 MB 5.7 MB/s eta
0:00:20
   --------- ------------------------------ 36.4/150.0 MB 5.8 MB/s eta
0:00:20
   --------- ------------------------------ 36.7/150.0 MB 5.7 MB/s eta
0:00:20
   --------- ------------------------------ 37.1/150.0 MB 5.8 MB/s eta
```

```
0:00:20
   --------- ------------------------------ 37.4/150.0 MB 5.8 MB/s eta
0:00:20
   ---------- ----------------------------- 37.9/150.0 MB 5.8 MB/s eta
0:00:20
   ---------- ----------------------------- 38.3/150.0 MB 5.8 MB/s eta
0:00:20
   ---------- ----------------------------- 38.7/150.0 MB 6.0 MB/s eta
0:00:19
   ---------- ----------------------------- 39.1/150.0 MB 6.2 MB/s eta
0:00:18
   ---------- ----------------------------- 39.4/150.0 MB 6.4 MB/s eta
0:00:18
   ---------- ----------------------------- 39.5/150.0 MB 6.2 MB/s eta
0:00:18
   ---------- ----------------------------- 39.6/150.0 MB 6.3 MB/s eta
0:00:18
   ---------- ----------------------------- 40.0/150.0 MB 6.4 MB/s eta
0:00:18
   ---------- ----------------------------- 40.3/150.0 MB 6.3 MB/s eta
0:00:18
   ---------- ----------------------------- 40.6/150.0 MB 6.3 MB/s eta
0:00:18
   ---------- ----------------------------- 40.9/150.0 MB 6.3 MB/s eta
0:00:18
   ----------- ---------------------------- 41.2/150.0 MB 6.4 MB/s eta
0:00:18
   ----------- ---------------------------- 41.6/150.0 MB 6.4 MB/s eta
0:00:17
   ----------- ---------------------------- 41.9/150.0 MB 6.5 MB/s eta
0:00:17
   ----------- ---------------------------- 42.3/150.0 MB 6.4 MB/s eta
0:00:17
   ----------- ---------------------------- 42.6/150.0 MB 6.5 MB/s eta
0:00:17
   ----------- ---------------------------- 43.1/150.0 MB 6.8 MB/s eta
0:00:16
   ----------- ---------------------------- 43.5/150.0 MB 6.8 MB/s eta
0:00:16
   ----------- ---------------------------- 43.8/150.0 MB 6.8 MB/s eta
0:00:16
   ----------- ---------------------------- 44.1/150.0 MB 6.9 MB/s eta
0:00:16
   ----------- ---------------------------- 44.4/150.0 MB 6.9 MB/s eta
0:00:16
   ----------- ---------------------------- 44.7/150.0 MB 6.9 MB/s eta
0:00:16
   ------------ --------------------------- 45.1/150.0 MB 7.1 MB/s eta
0:00:15
   ------------ --------------------------- 45.5/150.0 MB 7.3 MB/s eta
```

```
0:00:15
   ------------ --------------------------- 45.9/150.0 MB 7.3 MB/s eta
0:00:15
   ------------ --------------------------- 46.1/150.0 MB 7.1 MB/s eta
0:00:15
   ------------ --------------------------- 46.5/150.0 MB 7.2 MB/s eta
0:00:15
   ------------ --------------------------- 47.0/150.0 MB 7.4 MB/s eta
0:00:15
   ------------ --------------------------- 47.3/150.0 MB 7.4 MB/s eta
0:00:14
   ------------ --------------------------- 47.7/150.0 MB 7.4 MB/s eta
0:00:14
   ------------ --------------------------- 47.9/150.0 MB 7.2 MB/s eta
0:00:15
   ------------ --------------------------- 48.4/150.0 MB 7.3 MB/s eta
0:00:14
   ------------- -------------------------- 48.7/150.0 MB 7.2 MB/s eta
0:00:15
   ------------- -------------------------- 49.0/150.0 MB 7.1 MB/s eta
0:00:15
   ------------- -------------------------- 49.3/150.0 MB 7.1 MB/s eta
0:00:15
   ------------- -------------------------- 49.6/150.0 MB 7.0 MB/s eta
0:00:15
   ------------- -------------------------- 50.1/150.0 MB 7.4 MB/s eta
0:00:14
   ------------- -------------------------- 50.3/150.0 MB 7.4 MB/s eta
0:00:14
   ------------- -------------------------- 50.7/150.0 MB 7.4 MB/s eta
0:00:14
   ------------- -------------------------- 51.0/150.0 MB 7.5 MB/s eta
0:00:14
   ------------- -------------------------- 51.5/150.0 MB 7.5 MB/s eta
0:00:14
   ------------- -------------------------- 51.7/150.0 MB 7.4 MB/s eta
0:00:14
   ------------- -------------------------- 52.2/150.0 MB 7.6 MB/s eta
0:00:13
   -------------- ------------------------- 52.5/150.0 MB 7.7 MB/s eta
0:00:13
   -------------- ------------------------- 52.8/150.0 MB 7.5 MB/s eta
0:00:13
   -------------- ------------------------- 53.2/150.0 MB 7.4 MB/s eta
0:00:14
   -------------- ------------------------- 53.3/150.0 MB 7.4 MB/s eta
0:00:14
   -------------- ------------------------- 53.8/150.0 MB 7.4 MB/s eta
0:00:13
   -------------- ------------------------- 54.2/150.0 MB 7.4 MB/s eta
```

```
0:00:13
   -------------- ------------------------- 54.4/150.0 MB 7.4 MB/s eta
0:00:13
   -------------- ------------------------- 54.7/150.0 MB 7.4 MB/s eta
0:00:13
   -------------- ------------------------- 55.0/150.0 MB 7.4 MB/s eta
0:00:13
   -------------- ------------------------- 55.2/150.0 MB 7.3 MB/s eta
0:00:14
   -------------- ------------------------- 55.4/150.0 MB 7.2 MB/s eta
0:00:14
   -------------- ------------------------- 55.8/150.0 MB 7.3 MB/s eta
0:00:13
   -------------- ------------------------- 56.2/150.0 MB 7.3 MB/s eta
0:00:13
   --------------- ------------------------ 56.6/150.0 MB 7.4 MB/s eta
0:00:13
   --------------- ------------------------ 57.0/150.0 MB 7.3 MB/s eta
0:00:13
   --------------- ------------------------ 57.4/150.0 MB 7.3 MB/s eta
0:00:13
   --------------- ------------------------ 57.7/150.0 MB 7.3 MB/s eta
0:00:13
   --------------- ------------------------ 58.0/150.0 MB 7.2 MB/s eta
0:00:13
   --------------- ------------------------ 58.3/150.0 MB 7.2 MB/s eta
0:00:13
   --------------- ------------------------ 58.7/150.0 MB 7.1 MB/s eta
0:00:13
   --------------- ------------------------ 58.8/150.0 MB 7.0 MB/s eta
0:00:14
   --------------- ------------------------ 59.2/150.0 MB 7.1 MB/s eta
0:00:13
   --------------- ------------------------ 59.6/150.0 MB 7.1 MB/s eta
0:00:13
   --------------- ------------------------ 60.0/150.0 MB 7.2 MB/s eta
0:00:13
   ---------------- ----------------------- 60.4/150.0 MB 7.2 MB/s eta
0:00:13
   ---------------- ----------------------- 60.7/150.0 MB 7.2 MB/s eta
0:00:13
   ---------------- ----------------------- 60.9/150.0 MB 7.2 MB/s eta
0:00:13
   ---------------- ----------------------- 61.2/150.0 MB 7.1 MB/s eta
0:00:13
   ---------------- ----------------------- 61.6/150.0 MB 7.0 MB/s eta
0:00:13
   ---------------- ----------------------- 62.1/150.0 MB 7.2 MB/s eta
0:00:13
   ---------------- ----------------------- 62.5/150.0 MB 7.2 MB/s eta
```

```
0:00:13
   ---------------- ----------------------- 62.9/150.0 MB 7.2 MB/s eta
0:00:13
   ---------------- ----------------------- 63.3/150.0 MB 7.4 MB/s eta
0:00:12
   ---------------- ----------------------- 63.4/150.0 MB 7.3 MB/s eta
0:00:12
   ---------------- ----------------------- 63.4/150.0 MB 7.3 MB/s eta
0:00:12
   ----------------- ---------------------- 64.2/150.0 MB 7.2 MB/s eta
0:00:12
   ----------------- ---------------------- 64.5/150.0 MB 7.1 MB/s eta
0:00:13
   ----------------- ---------------------- 64.9/150.0 MB 7.3 MB/s eta
0:00:12
   ----------------- ---------------------- 65.2/150.0 MB 7.4 MB/s eta
0:00:12
   ----------------- ---------------------- 65.6/150.0 MB 7.6 MB/s eta
0:00:12
   ----------------- ---------------------- 65.9/150.0 MB 7.5 MB/s eta
0:00:12
   ----------------- ---------------------- 66.3/150.0 MB 7.5 MB/s eta
0:00:12
   ----------------- ---------------------- 66.4/150.0 MB 7.3 MB/s eta
0:00:12
   ----------------- ---------------------- 66.8/150.0 MB 7.3 MB/s eta
0:00:12
   ----------------- ---------------------- 67.0/150.0 MB 7.3 MB/s eta
0:00:12
   ----------------- ---------------------- 67.3/150.0 MB 7.1 MB/s eta
0:00:12
   ------------------ --------------------- 67.6/150.0 MB 7.1 MB/s eta
0:00:12
   ------------------ --------------------- 67.9/150.0 MB 7.0 MB/s eta
0:00:12
   ------------------ --------------------- 68.2/150.0 MB 7.2 MB/s eta
0:00:12
   ------------------ --------------------- 68.5/150.0 MB 7.2 MB/s eta
0:00:12
   ------------------ --------------------- 68.9/150.0 MB 7.1 MB/s eta
0:00:12
   ------------------ --------------------- 69.1/150.0 MB 7.2 MB/s eta
0:00:12
   ------------------ --------------------- 69.5/150.0 MB 7.1 MB/s eta
0:00:12
   ------------------ --------------------- 69.9/150.0 MB 7.1 MB/s eta
0:00:12
   ------------------ --------------------- 70.3/150.0 MB 7.2 MB/s eta
0:00:12
   ------------------ --------------------- 70.7/150.0 MB 7.1 MB/s eta
```

```
0:00:12
   ------------------ --------------------- 71.1/150.0 MB 7.4 MB/s eta
0:00:11
   ------------------- -------------------- 71.3/150.0 MB 7.3 MB/s eta
0:00:11
   ------------------- -------------------- 71.7/150.0 MB 7.3 MB/s eta
0:00:11
   ------------------- -------------------- 71.9/150.0 MB 7.1 MB/s eta
0:00:11
   ------------------- -------------------- 72.1/150.0 MB 7.0 MB/s eta
0:00:12
   ------------------- -------------------- 72.5/150.0 MB 7.0 MB/s eta
0:00:12
   ------------------- -------------------- 72.7/150.0 MB 6.9 MB/s eta
0:00:12
   ------------------- -------------------- 73.2/150.0 MB 6.9 MB/s eta
0:00:12
   ------------------- -------------------- 73.5/150.0 MB 6.8 MB/s eta
0:00:12
   ------------------- -------------------- 73.8/150.0 MB 7.3 MB/s eta
0:00:11
   ------------------- -------------------- 74.0/150.0 MB 7.0 MB/s eta
0:00:11
   ------------------- -------------------- 74.4/150.0 MB 6.8 MB/s eta
0:00:12
   ------------------- -------------------- 74.8/150.0 MB 7.0 MB/s eta
0:00:11
   -------------------- ------------------- 75.1/150.0 MB 6.8 MB/s eta
0:00:11
   -------------------- ------------------- 75.6/150.0 MB 6.9 MB/s eta
0:00:11
   -------------------- ------------------- 76.0/150.0 MB 7.0 MB/s eta
0:00:11
   -------------------- ------------------- 76.3/150.0 MB 7.0 MB/s eta
0:00:11
   -------------------- ------------------- 76.6/150.0 MB 7.1 MB/s eta
0:00:11
   -------------------- ------------------- 76.9/150.0 MB 7.0 MB/s eta
0:00:11
   -------------------- ------------------- 77.2/150.0 MB 7.0 MB/s eta
0:00:11
   -------------------- ------------------- 77.6/150.0 MB 7.2 MB/s eta
0:00:11
   -------------------- ------------------- 77.9/150.0 MB 7.2 MB/s eta
0:00:11
   -------------------- ------------------- 78.3/150.0 MB 7.2 MB/s eta
0:00:10
   -------------------- ------------------- 78.7/150.0 MB 7.4 MB/s eta
0:00:10
   --------------------- ------------------ 79.1/150.0 MB 7.4 MB/s eta
```

```
0:00:10
   --------------------- ------------------ 79.4/150.0 MB 7.4 MB/s eta
0:00:10
   --------------------- ------------------ 79.9/150.0 MB 7.4 MB/s eta
0:00:10
   --------------------- ------------------ 80.4/150.0 MB 7.5 MB/s eta
0:00:10
   --------------------- ------------------ 80.8/150.0 MB 7.5 MB/s eta
0:00:10
   --------------------- ------------------ 81.2/150.0 MB 7.5 MB/s eta
0:00:10
   --------------------- ------------------ 81.5/150.0 MB 7.5 MB/s eta
0:00:10
   --------------------- ------------------ 82.0/150.0 MB 7.6 MB/s eta
0:00:09
   --------------------- ------------------ 82.2/150.0 MB 7.6 MB/s eta
0:00:09
   ---------------------- ----------------- 82.5/150.0 MB 7.6 MB/s eta
0:00:09
   ---------------------- ----------------- 82.7/150.0 MB 7.5 MB/s eta
0:00:09
   ---------------------- ----------------- 83.2/150.0 MB 7.7 MB/s eta
0:00:09
   ---------------------- ----------------- 83.5/150.0 MB 7.6 MB/s eta
0:00:09
   ---------------------- ----------------- 83.9/150.0 MB 7.7 MB/s eta
0:00:09
   ---------------------- ----------------- 84.3/150.0 MB 7.8 MB/s eta
0:00:09
   ---------------------- ----------------- 84.6/150.0 MB 7.8 MB/s eta
0:00:09
   ---------------------- ----------------- 84.8/150.0 MB 7.6 MB/s eta
0:00:09
   ---------------------- ----------------- 85.1/150.0 MB 7.6 MB/s eta
0:00:09
   ---------------------- ----------------- 85.7/150.0 MB 7.7 MB/s eta
0:00:09
   ---------------------- ----------------- 86.1/150.0 MB 7.7 MB/s eta
0:00:09
   ----------------------- ---------------- 86.6/150.0 MB 7.8 MB/s eta
0:00:09
   ----------------------- ---------------- 87.0/150.0 MB 8.0 MB/s eta
0:00:08
   ----------------------- ---------------- 87.6/150.0 MB 8.1 MB/s eta
0:00:08
   ----------------------- ---------------- 88.0/150.0 MB 8.2 MB/s eta
0:00:08
   ----------------------- ---------------- 88.3/150.0 MB 8.3 MB/s eta
0:00:08
   ----------------------- ---------------- 88.7/150.0 MB 8.2 MB/s eta
```

```
0:00:08
   ----------------------- ---------------- 89.1/150.0 MB 8.2 MB/s eta
0:00:08
   ----------------------- ---------------- 89.6/150.0 MB 8.3 MB/s eta
0:00:08
   ----------------------- ---------------- 89.9/150.0 MB 8.4 MB/s eta
0:00:08
   ------------------------ --------------- 90.4/150.0 MB 8.2 MB/s eta
0:00:08
   ------------------------ --------------- 90.8/150.0 MB 8.2 MB/s eta
0:00:08
   ------------------------ --------------- 91.2/150.0 MB 8.2 MB/s eta
0:00:08
   ------------------------ --------------- 91.4/150.0 MB 8.1 MB/s eta
0:00:08
   ------------------------ --------------- 91.7/150.0 MB 8.2 MB/s eta
0:00:08
   ------------------------ --------------- 92.1/150.0 MB 8.0 MB/s eta
0:00:08
   ------------------------ --------------- 92.6/150.0 MB 8.3 MB/s eta
0:00:07
   ------------------------ --------------- 92.9/150.0 MB 8.4 MB/s eta
0:00:07
   ------------------------ --------------- 93.1/150.0 MB 8.2 MB/s eta
0:00:07
   ------------------------ --------------- 93.5/150.0 MB 8.3 MB/s eta
0:00:07
   ------------------------- -------------- 93.9/150.0 MB 8.3 MB/s eta
0:00:07
   ------------------------- -------------- 94.4/150.0 MB 8.3 MB/s eta
0:00:07
   ------------------------- -------------- 94.7/150.0 MB 8.4 MB/s eta
0:00:07
   ------------------------- -------------- 95.1/150.0 MB 8.6 MB/s eta
0:00:07
   ------------------------- -------------- 95.3/150.0 MB 8.5 MB/s eta
0:00:07
   ------------------------- -------------- 95.9/150.0 MB 8.5 MB/s eta
0:00:07
   ------------------------- -------------- 96.4/150.0 MB 8.5 MB/s eta
0:00:07
   ------------------------- -------------- 96.8/150.0 MB 8.5 MB/s eta
0:00:07
   ------------------------- -------------- 97.0/150.0 MB 8.3 MB/s eta
0:00:07
   ------------------------- -------------- 97.1/150.0 MB 8.1 MB/s eta
0:00:07
   ------------------------- -------------- 97.4/150.0 MB 8.0 MB/s eta
0:00:07
```

```
   -------------------------- ------------- 97.7/150.0 MB 7.9 MB/s eta
0:00:07
   -------------------------- ------------- 98.2/150.0 MB 7.9 MB/s eta
0:00:07
   -------------------------- ------------- 98.6/150.0 MB 7.9 MB/s eta
0:00:07
   -------------------------- ------------- 98.9/150.0 MB 7.8 MB/s eta
0:00:07
   -------------------------- ------------- 99.1/150.0 MB 7.7 MB/s eta
0:00:07
   -------------------------- ------------- 99.2/150.0 MB 7.7 MB/s eta
0:00:07
   -------------------------- ------------- 99.3/150.0 MB 7.4 MB/s eta
0:00:07
   -------------------------- ------------- 99.5/150.0 MB 7.2 MB/s eta
0:00:08
   -------------------------- ------------- 99.7/150.0 MB 7.0 MB/s eta
0:00:08
   -------------------------- ------------- 100.1/150.0 MB 7.0 MB/s
eta 0:00:08
   -------------------------- ------------- 100.4/150.0 MB 7.0 MB/s
eta 0:00:08
   -------------------------- ------------- 100.8/150.0 MB 7.0 MB/s
eta 0:00:08
   -------------------------- ------------- 101.2/150.0 MB 6.9 MB/s
eta 0:00:08
   --------------------------- ------------ 101.5/150.0 MB 7.0 MB/s
eta 0:00:07
   --------------------------- ------------ 101.9/150.0 MB 7.0 MB/s
eta 0:00:07
   --------------------------- ------------ 102.2/150.0 MB 7.0 MB/s
eta 0:00:07
   --------------------------- ------------ 102.6/150.0 MB 7.0 MB/s
eta 0:00:07
   --------------------------- ------------ 103.1/150.0 MB 7.0 MB/s
eta 0:00:07
   --------------------------- ------------ 103.5/150.0 MB 7.2 MB/s
eta 0:00:07
   --------------------------- ------------ 104.1/150.0 MB 7.3 MB/s
eta 0:00:07
   --------------------------- ------------ 104.5/150.0 MB 7.2 MB/s
eta 0:00:07
   --------------------------- ------------ 104.7/150.0 MB 7.3 MB/s
eta 0:00:07
   --------------------------- ------------ 104.9/150.0 MB 7.1 MB/s
eta 0:00:07
   ---------------------------- ----------- 105.5/150.0 MB 7.3 MB/s
eta 0:00:07
   ---------------------------- ----------- 105.9/150.0 MB 7.2 MB/s
```

```
eta 0:00:07
   ---------------------------- ----------- 106.3/150.0 MB 7.1 MB/s
eta 0:00:07
   ---------------------------- ----------- 106.8/150.0 MB 7.3 MB/s
eta 0:00:06
   ---------------------------- ----------- 107.2/150.0 MB 7.4 MB/s
eta 0:00:06
   ---------------------------- ----------- 107.7/150.0 MB 7.5 MB/s
eta 0:00:06
   ---------------------------- ----------- 108.1/150.0 MB 7.6 MB/s
eta 0:00:06
   ---------------------------- ----------- 108.4/150.0 MB 7.5 MB/s
eta 0:00:06
   ----------------------------- ---------- 108.8/150.0 MB 7.6 MB/s
eta 0:00:06
   ----------------------------- ---------- 109.1/150.0 MB 7.7 MB/s
eta 0:00:06
   ----------------------------- ---------- 109.5/150.0 MB 8.0 MB/s
eta 0:00:06
   ----------------------------- ---------- 109.9/150.0 MB 8.4 MB/s
eta 0:00:05
   ----------------------------- ---------- 110.3/150.0 MB 8.5 MB/s
eta 0:00:05
   ----------------------------- ---------- 110.6/150.0 MB 8.4 MB/s
eta 0:00:05
   ----------------------------- ---------- 111.0/150.0 MB 8.6 MB/s
eta 0:00:05
   ----------------------------- ---------- 111.2/150.0 MB 8.4 MB/s
eta 0:00:05
   ----------------------------- ---------- 111.7/150.0 MB 8.4 MB/s
eta 0:00:05
   ----------------------------- ---------- 111.9/150.0 MB 8.3 MB/s
eta 0:00:05
   ----------------------------- ---------- 112.4/150.0 MB 8.5 MB/s
eta 0:00:05
   ------------------------------ --------- 112.8/150.0 MB 8.5 MB/s
eta 0:00:05
   ------------------------------ --------- 113.0/150.0 MB 8.3 MB/s
eta 0:00:05
   ------------------------------ --------- 113.4/150.0 MB 8.2 MB/s
eta 0:00:05
   ------------------------------ --------- 113.6/150.0 MB 8.1 MB/s
eta 0:00:05
   ------------------------------ --------- 113.9/150.0 MB 8.0 MB/s
eta 0:00:05
   ------------------------------ --------- 114.3/150.0 MB 7.9 MB/s
eta 0:00:05
   ------------------------------ --------- 114.9/150.0 MB 8.0 MB/s
eta 0:00:05
```

```
   ------------------------------ --------- 115.3/150.0 MB 8.3 MB/s
eta 0:00:05
   ------------------------------ --------- 115.7/150.0 MB 8.2 MB/s
eta 0:00:05
   ------------------------------ --------- 116.1/150.0 MB 8.2 MB/s
eta 0:00:05
   ------------------------------- -------- 116.5/150.0 MB 8.2 MB/s
eta 0:00:05
   ------------------------------- -------- 116.9/150.0 MB 8.1 MB/s
eta 0:00:05
   ------------------------------- -------- 117.2/150.0 MB 8.1 MB/s
eta 0:00:05
   ------------------------------- -------- 117.5/150.0 MB 7.9 MB/s
eta 0:00:05
   ------------------------------- -------- 118.0/150.0 MB 8.0 MB/s
eta 0:00:05
   ------------------------------- -------- 118.4/150.0 MB 8.0 MB/s
eta 0:00:04
   ------------------------------- -------- 118.8/150.0 MB 8.1 MB/s
eta 0:00:04
   ------------------------------- -------- 119.1/150.0 MB 7.9 MB/s
eta 0:00:04
   ------------------------------- -------- 119.5/150.0 MB 7.9 MB/s
eta 0:00:04
   ------------------------------- -------- 119.8/150.0 MB 7.9 MB/s
eta 0:00:04
   -------------------------------- ------- 120.2/150.0 MB 7.9 MB/s
eta 0:00:04
   -------------------------------- ------- 120.5/150.0 MB 7.8 MB/s
eta 0:00:04
   -------------------------------- ------- 120.9/150.0 MB 7.9 MB/s
eta 0:00:04
   -------------------------------- ------- 121.3/150.0 MB 8.0 MB/s
eta 0:00:04
   -------------------------------- ------- 121.7/150.0 MB 8.1 MB/s
eta 0:00:04
   -------------------------------- ------- 122.0/150.0 MB 8.1 MB/s
eta 0:00:04
   -------------------------------- ------- 122.4/150.0 MB 8.0 MB/s
eta 0:00:04
   -------------------------------- ------- 122.8/150.0 MB 8.0 MB/s
eta 0:00:04
   -------------------------------- ------- 123.2/150.0 MB 8.1 MB/s
eta 0:00:04
   -------------------------------- ------- 123.6/150.0 MB 8.3 MB/s
eta 0:00:04
   --------------------------------- ------ 123.8/150.0 MB 8.2 MB/s
eta 0:00:04
   --------------------------------- ------ 124.3/150.0 MB 8.3 MB/s
```

```
eta 0:00:04
   --------------------------------- ------ 124.7/150.0 MB 8.2 MB/s
eta 0:00:04
   --------------------------------- ------ 125.1/150.0 MB 8.1 MB/s
eta 0:00:04
   --------------------------------- ------ 125.4/150.0 MB 8.0 MB/s
eta 0:00:04
   --------------------------------- ------ 125.7/150.0 MB 8.0 MB/s
eta 0:00:04
   --------------------------------- ------ 126.2/150.0 MB 8.0 MB/s
eta 0:00:03
   --------------------------------- ------ 126.6/150.0 MB 8.0 MB/s
eta 0:00:03
   --------------------------------- ------ 127.0/150.0 MB 8.0 MB/s
eta 0:00:03
   ---------------------------------- ----- 127.5/150.0 MB 8.3 MB/s
eta 0:00:03
   ---------------------------------- ----- 128.0/150.0 MB 8.3 MB/s
eta 0:00:03
   ---------------------------------- ----- 128.5/150.0 MB 8.2 MB/s
eta 0:00:03
   ---------------------------------- ----- 128.8/150.0 MB 8.3 MB/s
eta 0:00:03
   ---------------------------------- ----- 129.2/150.0 MB 8.4 MB/s
eta 0:00:03
   ---------------------------------- ----- 129.5/150.0 MB 8.3 MB/s
eta 0:00:03
   ---------------------------------- ----- 129.9/150.0 MB 8.4 MB/s
eta 0:00:03
   ---------------------------------- ----- 130.3/150.0 MB 8.4 MB/s
eta 0:00:03
   ---------------------------------- ----- 130.8/150.0 MB 8.6 MB/s
eta 0:00:03
   ----------------------------------- ---- 131.2/150.0 MB 8.5 MB/s
eta 0:00:03
   ----------------------------------- ---- 131.7/150.0 MB 8.6 MB/s
eta 0:00:03
   ----------------------------------- ---- 132.0/150.0 MB 8.5 MB/s
eta 0:00:03
   ----------------------------------- ---- 132.1/150.0 MB 8.3 MB/s
eta 0:00:03
   ----------------------------------- ---- 132.4/150.0 MB 8.3 MB/s
eta 0:00:03
   ----------------------------------- ---- 132.7/150.0 MB 8.2 MB/s
eta 0:00:03
   ----------------------------------- ---- 133.0/150.0 MB 8.2 MB/s
eta 0:00:03
   ----------------------------------- ---- 133.3/150.0 MB 8.0 MB/s
eta 0:00:03
```

```
   ----------------------------------- ---- 133.6/150.0 MB 8.0 MB/s
eta 0:00:03
   ----------------------------------- ---- 133.7/150.0 MB 7.8 MB/s
eta 0:00:03
   ----------------------------------- ---- 133.9/150.0 MB 7.5 MB/s
eta 0:00:03
   ----------------------------------- ---- 134.0/150.0 MB 7.5 MB/s
eta 0:00:03
   ----------------------------------- ---- 134.3/150.0 MB 7.4 MB/s
eta 0:00:03
   ----------------------------------- ---- 134.6/150.0 MB 7.4 MB/s
eta 0:00:03
   ----------------------------------- ---- 134.9/150.0 MB 7.4 MB/s
eta 0:00:03
   ------------------------------------ --- 135.1/150.0 MB 7.2 MB/s
eta 0:00:03
   ------------------------------------ --- 135.4/150.0 MB 7.1 MB/s
eta 0:00:03
   ------------------------------------ --- 135.7/150.0 MB 7.1 MB/s
eta 0:00:03
   ------------------------------------ --- 135.9/150.0 MB 7.1 MB/s
eta 0:00:02
   ------------------------------------ --- 136.4/150.0 MB 7.1 MB/s
eta 0:00:02
   ------------------------------------ --- 136.7/150.0 MB 7.0 MB/s
eta 0:00:02
   ------------------------------------ --- 137.1/150.0 MB 7.0 MB/s
eta 0:00:02
   ------------------------------------ --- 137.5/150.0 MB 7.0 MB/s
eta 0:00:02
   ------------------------------------ --- 137.7/150.0 MB 6.9 MB/s
eta 0:00:02
   ------------------------------------ --- 138.1/150.0 MB 6.8 MB/s
eta 0:00:02
   ------------------------------------ --- 138.6/150.0 MB 6.8 MB/s
eta 0:00:02
   ------------------------------------ --- 138.7/150.0 MB 6.8 MB/s
eta 0:00:02
   ------------------------------------ --- 138.7/150.0 MB 6.8 MB/s
eta 0:00:02
   ------------------------------------- -- 139.6/150.0 MB 6.7 MB/s
eta 0:00:02
   ------------------------------------- -- 139.9/150.0 MB 6.8 MB/s
eta 0:00:02
   ------------------------------------- -- 140.3/150.0 MB 6.7 MB/s
eta 0:00:02
   ------------------------------------- -- 140.6/150.0 MB 6.7 MB/s
eta 0:00:02
   ------------------------------------- -- 140.9/150.0 MB 6.7 MB/s
```

```
eta 0:00:02
   ------------------------------------- -- 141.3/150.0 MB 6.6 MB/s
eta 0:00:02
   ------------------------------------- -- 141.7/150.0 MB 6.5 MB/s
eta 0:00:02
   ------------------------------------- -- 142.0/150.0 MB 6.5 MB/s
eta 0:00:02
   ------------------------------------- -- 142.3/150.0 MB 6.6 MB/s
eta 0:00:02
   -------------------------------------- - 142.8/150.0 MB 6.8 MB/s
eta 0:00:02
   -------------------------------------- - 143.1/150.0 MB 6.8 MB/s
eta 0:00:02
   -------------------------------------- - 143.3/150.0 MB 6.7 MB/s
eta 0:00:01
   -------------------------------------- - 143.6/150.0 MB 6.8 MB/s
eta 0:00:01
   -------------------------------------- - 144.0/150.0 MB 7.0 MB/s
eta 0:00:01
   -------------------------------------- - 144.3/150.0 MB 7.3 MB/s
eta 0:00:01
   -------------------------------------- - 144.5/150.0 MB 7.2 MB/s
eta 0:00:01
   -------------------------------------- - 144.7/150.0 MB 7.1 MB/s
eta 0:00:01
   -------------------------------------- - 144.8/150.0 MB 7.0 MB/s
eta 0:00:01
   -------------------------------------- - 145.1/150.0 MB 7.0 MB/s
eta 0:00:01
   -------------------------------------- - 145.2/150.0 MB 6.9 MB/s
eta 0:00:01
   -------------------------------------- - 145.3/150.0 MB 6.8 MB/s
eta 0:00:01
   -------------------------------------- - 145.5/150.0 MB 6.6 MB/s
eta 0:00:01
   -------------------------------------- - 145.7/150.0 MB 6.8 MB/s
eta 0:00:01
   -------------------------------------- - 145.9/150.0 MB 6.6 MB/s
eta 0:00:01
   -------------------------------------- - 146.1/150.0 MB 6.6 MB/s
eta 0:00:01
   ---------------------------------------  146.5/150.0 MB 6.5 MB/s
eta 0:00:01
   ---------------------------------------  146.9/150.0 MB 6.5 MB/s
eta 0:00:01
   ---------------------------------------  147.0/150.0 MB 6.5 MB/s
eta 0:00:01
   ---------------------------------------  147.4/150.0 MB 6.5 MB/s
eta 0:00:01
```

```
   ----------------------------------------  147.4/150.0 MB 6.4 MB/s
eta 0:00:01
   ----------------------------------------  147.7/150.0 MB 6.2 MB/s
eta 0:00:01
   ----------------------------------------  147.9/150.0 MB 6.2 MB/s
eta 0:00:01
   ----------------------------------------  148.0/150.0 MB 6.1 MB/s
eta 0:00:01
   ----------------------------------------  148.3/150.0 MB 6.1 MB/s
eta 0:00:01
   ----------------------------------------  148.7/150.0 MB 6.0 MB/s
eta 0:00:01
   ----------------------------------------  149.0/150.0 MB 6.4 MB/s
eta 0:00:01
   ----------------------------------------  149.3/150.0 MB 6.2 MB/s
eta 0:00:01
   ----------------------------------------  149.6/150.0 MB 6.1 MB/s
eta 0:00:01
   ----------------------------------------  149.8/150.0 MB 6.0 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ----------------------------------------  150.0/150.0 MB 5.8 MB/s
```

```
eta 0:00:01
   ---------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ---------------------------------------  150.0/150.0 MB 5.8 MB/s
eta 0:00:01
   ---------------------------------------- 150.0/150.0 MB 4.1 MB/s
eta 0:00:00
Installing collected packages: xgboost
Successfully installed xgboost-3.0.0
```

```python
# Importing pandas library for working with datasets
import pandas as pd

# Importing numpy library for working with arrays
import numpy as np

# Importing matplotlib.pyplot for visualization
import matplotlib.pyplot as plt
%matplotlib inline

# Importing seaborn library for visualization
import seaborn as sns

# Importing warnings for disabling warnings from the code
import warnings
warnings.filterwarnings('ignore')

# Importing YData Profiling for generating an automatic exploratory
data analysis (EDA) report
from ydata_profiling import ProfileReport

# Importing OneHotEncoder for encoding
from sklearn.preprocessing import OneHotEncoder

# Importing Label-encoder for encoding
from sklearn.preprocessing import LabelEncoder

# Importing MinMaxScaler for feature scaling
from sklearn.preprocessing import MinMaxScaler

# Importing train_test_split for splitting data into training and
testing sets for model evaluation
from sklearn.model_selection import train_test_split

# Importing LinearRegression
from sklearn.linear_model import LinearRegression

# Importing DecisionTreeRegressor
from sklearn.tree import DecisionTreeRegressor

# Importing an ensemble model RandomForestRegressor
```

```
from sklearn.ensemble import RandomForestRegressor

# Importing an advanced boosting model GradientBoostingRegressor
from sklearn.ensemble import GradientBoostingRegressor

# Importing an efficient and optimized version of gradient- XGBoost
(Regression)
from xgboost import XGBRegressor

# Importing KNeighborsRegressor
from sklearn.neighbors import KNeighborsRegressor

# Importing RandomizedSearchCV for hyperparameter tuning using
randomized search
from sklearn.model_selection import RandomizedSearchCV

# Importing GridSearchCV for hyperparameter tuning
from sklearn.model_selection import GridSearchCV

# Importing Multi-layer Perceptron (MLP) Regressor
from sklearn.neural_network import MLPRegressor

# Importing performance metrics - Mean Absolute Error (MAE), Mean
Squared Error (MSE), Root Mean Squared Error (RMSE), R2_score
from sklearn.metrics import *
```

# 2. LOADING THE DATA

```
data = pd.read_csv("Insurance.csv")
data
```

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| 0 | 19 | female | 27.900 | 0 | yes | southwest | 16884.92400 |
| 1 | 18 | male | 33.770 | 1 | no | southeast | 1725.55230 |
| 2 | 28 | male | 33.000 | 3 | no | southeast | 4449.46200 |
| 3 | 33 | male | 22.705 | 0 | no | northwest | 21984.47061 |
| 4 | 32 | male | 28.880 | 0 | no | northwest | 3866.85520 |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 1333 | 50 | male | 30.970 | 3 | no | northwest | 10600.54830 |
| 1334 | 18 | female | 31.920 | 0 | no | northeast | 2205.98080 |
| 1335 | 18 | female | 36.850 | 0 | no | southeast | 1629.83350 |
| 1336 | 21 | female | 25.800 | 0 | no | southwest | 2007.94500 |
| 1337 | 61 | female | 29.070 | 0 | yes | northwest | 29141.36030 |

[1338 rows x 7 columns]

# 3. BASIC CHECKS

```
# Visualizing the first 10 rows of the data
data.head(10)
```

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| 0 | 19 | female | 27.900 | 0 | yes | southwest | 16884.92400 |
| 1 | 18 | male | 33.770 | 1 | no | southeast | 1725.55230 |
| 2 | 28 | male | 33.000 | 3 | no | southeast | 4449.46200 |
| 3 | 33 | male | 22.705 | 0 | no | northwest | 21984.47061 |
| 4 | 32 | male | 28.880 | 0 | no | northwest | 3866.85520 |
| 5 | 31 | female | 25.740 | 0 | no | southeast | 3756.62160 |
| 6 | 46 | female | 33.440 | 1 | no | southeast | 8240.58960 |
| 7 | 37 | female | 27.740 | 3 | no | northwest | 7281.50560 |
| 8 | 37 | male | 29.830 | 2 | no | northeast | 6406.41070 |
| 9 | 60 | female | 25.840 | 0 | no | northwest | 28923.13692 |

```
# Visualizing the last 10 rows of the data
data.tail(10)
```

| | age | sex | bmi | children | smoker | region | charges |
|---|---|---|---|---|---|---|---|
| 1328 | 23 | female | 24.225 | 2 | no | northeast | 22395.74424 |
| 1329 | 52 | male | 38.600 | 2 | no | southwest | 10325.20600 |
| 1330 | 57 | female | 25.740 | 2 | no | southeast | 12629.16560 |
| 1331 | 23 | female | 33.400 | 0 | no | southwest | 10795.93733 |
| 1332 | 52 | female | 44.700 | 3 | no | southwest | 11411.68500 |
| 1333 | 50 | male | 30.970 | 3 | no | northwest | 10600.54830 |
| 1334 | 18 | female | 31.920 | 0 | no | northeast | 2205.98080 |
| 1335 | 18 | female | 36.850 | 0 | no | southeast | 1629.83350 |
| 1336 | 21 | female | 25.800 | 0 | no | southwest | 2007.94500 |
| 1337 | 61 | female | 29.070 | 0 | yes | northwest | 29141.36030 |

```
# Checking the column names of the data
data.columns
```

```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region',
'charges'], dtype='object')
```

```
# Numerical columns of the data
numerical_cols = data.select_dtypes(include=["int","float"]).columns
numerical_cols
```

```
Index(['age', 'bmi', 'children', 'charges'], dtype='object')
```

```
# Categorical columns of the data
categorical_cols = data.select_dtypes(include=["object"]).columns
categorical_cols
```

```
Index(['sex', 'smoker', 'region'], dtype='object')
```

```python
# Checking the index of the data
data.index
```

```
RangeIndex(start=0, stop=1338, step=1)
```

```python
# Checking the number of rows and columns of the data
data.shape
```

```
(1338, 7)
```

```python
# Checking the basic information of the data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1338 non-null   int64  
 1   sex       1338 non-null   object 
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64  
 4   smoker    1338 non-null   object 
 5   region    1338 non-null   object 
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

```python
# Checking the statistical information of the numerical column in a data
data.describe()
```

```
               age          bmi     children       charges
count  1338.000000  1338.000000  1338.000000   1338.000000
mean     39.207025    30.663397     1.094918  13270.422265
std      14.049960     6.098187     1.205493  12110.011237
min      18.000000    15.960000     0.000000   1121.873900
25%      27.000000    26.296250     0.000000   4740.287150
50%      39.000000    30.400000     1.000000   9382.033000
75%      51.000000    34.693750     2.000000  16639.912515
max      64.000000    53.130000     5.000000  63770.428010
```

```python
# Checking the statistical information of the categorical column in a data
data.describe(include="O")
```

```
         sex smoker     region
count   1338   1338       1338
unique     2      2          4
top     male     no  southeast
freq     676   1064        364
```

```python
# Checking the unique values for all the columns in the data
for i in data:
    print(f"                                {i.title()}
")
    print("\n")
    print(f"* The number of unique values in {i}
column :",data[i].nunique())
    print(">>",data[i].unique())

print('===============================================================
===============')
    print("\n")
```

```
                                Age


* The number of unique values in age column : 47
>> [19 18 28 33 32 31 46 37 60 25 62 23 56 27 52 30 34 59 63 55 22 26
35 24
 41 38 36 21 48 40 58 53 43 64 20 61 44 57 29 45 54 49 47 51 42 50 39]
======================================================================
========


                                Sex


* The number of unique values in sex column : 2
>> ['female' 'male']
======================================================================
========


                                Bmi


* The number of unique values in bmi column : 548
>> [27.9   33.77  33.    22.705 28.88  25.74  33.44  27.74  29.83
25.84
 26.22  26.29  34.4   39.82  42.13  24.6   30.78  23.845 40.3   35.3
 36.005 32.4   34.1   31.92  28.025 27.72  23.085 32.775 17.385 36.3
 35.6   26.315 28.6   28.31  36.4   20.425 32.965 20.8   36.67  39.9
 26.6   36.63  21.78  30.8   37.05  37.3   38.665 34.77  24.53  35.2
 35.625 33.63  28.    34.43  28.69  36.955 31.825 31.68  22.88  37.335
 27.36  33.66  24.7   25.935 22.42  28.9   39.1   36.19  23.98  24.75
 28.5   28.1   32.01  27.4   34.01  29.59  35.53  39.805 26.885 38.285
 37.62  41.23  34.8   22.895 31.16  27.2   26.98  39.49  24.795 31.3
 38.28  19.95  19.3   31.6   25.46  30.115 29.92  27.5   28.4   30.875
 27.94  35.09  29.7   35.72  32.205 28.595 49.06  27.17  23.37  37.1
 23.75  28.975 31.35  33.915 28.785 28.3   37.4   17.765 34.7   26.505
```

```
 22.04  35.9   25.555 28.05  25.175 31.9   36.    32.49  25.3   29.735
 38.83  30.495 37.73  37.43  24.13  37.145 39.52  24.42  27.83  36.85
 39.6   29.8   29.64  28.215 37.    33.155 18.905 41.47  30.3   15.96
 33.345 37.7   27.835 29.2   26.41  30.69  41.895 30.9   32.2   32.11
 31.57  26.2   30.59  32.8   18.05  39.33  32.23  24.035 36.08  22.3
 26.4   31.8   26.73  23.1   23.21  33.7   33.25  24.64  33.88  38.06
 41.91  31.635 36.195 17.8   24.51  22.22  38.39  29.07  22.135 26.8
 30.02  35.86  20.9   17.29  34.21  25.365 40.15  24.415 25.2   26.84
 24.32  42.35  19.8   32.395 30.2   29.37  34.2   27.455 27.55  20.615
 24.3   31.79  21.56  28.12  40.565 27.645 31.2   26.62  48.07  36.765
 33.4   45.54  28.82  22.99  27.7   25.41  34.39  22.61  37.51  38.
 33.33  34.865 33.06  35.97  31.4   25.27  40.945 34.105 36.48  33.8
 36.7   36.385 34.5   32.3   27.6   29.26  35.75  23.18  25.6   35.245
 43.89  20.79  30.5   21.7   21.89  24.985 32.015 30.4   21.09  22.23
 32.9   24.89  31.46  17.955 30.685 43.34  39.05  30.21  31.445 19.855
 31.02  38.17  20.6   47.52  20.4   38.38  24.31  23.6   21.12  30.03
 17.48  20.235 17.195 23.9   35.15  35.64  22.6   39.16  27.265 29.165
 16.815 33.1   26.9   33.11  31.73  46.75  29.45  32.68  33.5   43.01
 36.52  26.695 25.65  29.6   38.6   23.4   46.53  30.14  30.    38.095
 28.38  28.7   33.82  24.09  32.67  25.1   32.56  41.325 39.5   34.3
 31.065 21.47  25.08  43.4   25.7   27.93  39.2   26.03  30.25  28.93
 35.7   35.31  31.    44.22  26.07  25.8   39.425 40.48  38.9   47.41
 35.435 46.7   46.2   21.4   23.8   44.77  32.12  29.1   37.29  43.12
 36.86  34.295 23.465 45.43  23.65  20.7   28.27  35.91  29.    19.57
 31.13  21.85  40.26  33.725 29.48  32.6   37.525 23.655 37.8   19.
 21.3   33.535 42.46  38.95  36.1   29.3   39.7   38.19  42.4   34.96
 42.68  31.54  29.81  21.375 40.81  17.4   20.3   18.5   26.125 41.69
 24.1   36.2   40.185 39.27  34.87  44.745 29.545 23.54  40.47  40.66
 36.6   35.4   27.075 28.405 21.755 40.28  30.1   32.1   23.7   35.5
 29.15  27.    37.905 22.77  22.8   34.58  27.1   19.475 26.7   34.32
 24.4   41.14  22.515 41.8   26.18  42.24  26.51  35.815 41.42  36.575
 42.94  21.01  24.225 17.67  31.5   31.1   32.78  32.45  50.38  47.6
 25.4   29.9   43.7   24.86  28.8   29.5   29.04  38.94  44.    20.045
 40.92  35.1   29.355 32.585 32.34  39.8   24.605 33.99  28.2   25.
 33.2   23.2   20.1   32.5   37.18  46.09  39.93  35.8   31.255 18.335
 42.9   26.79  39.615 25.9   25.745 28.16  23.56  40.5   35.42  39.995
 34.675 20.52  23.275 36.29  32.7   19.19  20.13  23.32  45.32  34.6
 18.715 21.565 23.    37.07  52.58  42.655 21.66  32.    18.3   47.74
 22.1   19.095 31.24  29.925 20.35  25.85  42.75  18.6   23.87  45.9
 21.5   30.305 44.88  41.1   40.37  28.49  33.55  40.375 27.28  17.86
 33.3   39.14  21.945 24.97  23.94  34.485 21.8   23.3   36.96  21.28
 29.4   27.3   37.9   37.715 23.76  25.52  27.61  27.06  39.4   34.9
 22.    30.36  27.8   53.13  39.71  32.87  44.7   30.97 ]
======================================================================
========


                                Children
```

```
* The number of unique values in children column : 6
>> [0 1 3 2 5 4]
==========================================================================
========


                                Smoker


* The number of unique values in smoker column : 2
>> ['yes' 'no']
==========================================================================
========


                                Region


* The number of unique values in region column : 4
>> ['southwest' 'southeast' 'northwest' 'northeast']
==========================================================================
========


                                Charges


* The number of unique values in charges column : 1337
>> [16884.924   1725.5523  4449.462  ...  1629.8335  2007.945
29141.3603]
==========================================================================
========
```

```python
# Checking the value_counts
colls = ["children","sex","smoker","region"]
for i in data[colls]:
    print(data[i].value_counts())
    print('=============================')
    print("\n")
```

```
children
0    574
1    324
2    240
3    157
4     25
5     18
Name: count, dtype: int64
============================
```

```
sex
male      676
female    662
Name: count, dtype: int64
============================


smoker
no     1064
yes     274
Name: count, dtype: int64
============================


region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
============================
```

### INSIGHTS

1. There are 1338 rows and 7 columns (4 numerical and 3 categorical) in the dataset.
2. The value count of 'sex' column shows there are 676 males and 662 females.
3. There are 1064 non-smokers and 274 smokers invloved in this dataset.

# 4. EXPLORATORY DATA ANALYSIS (EDA)

## 4.1. PROFILE REPORT

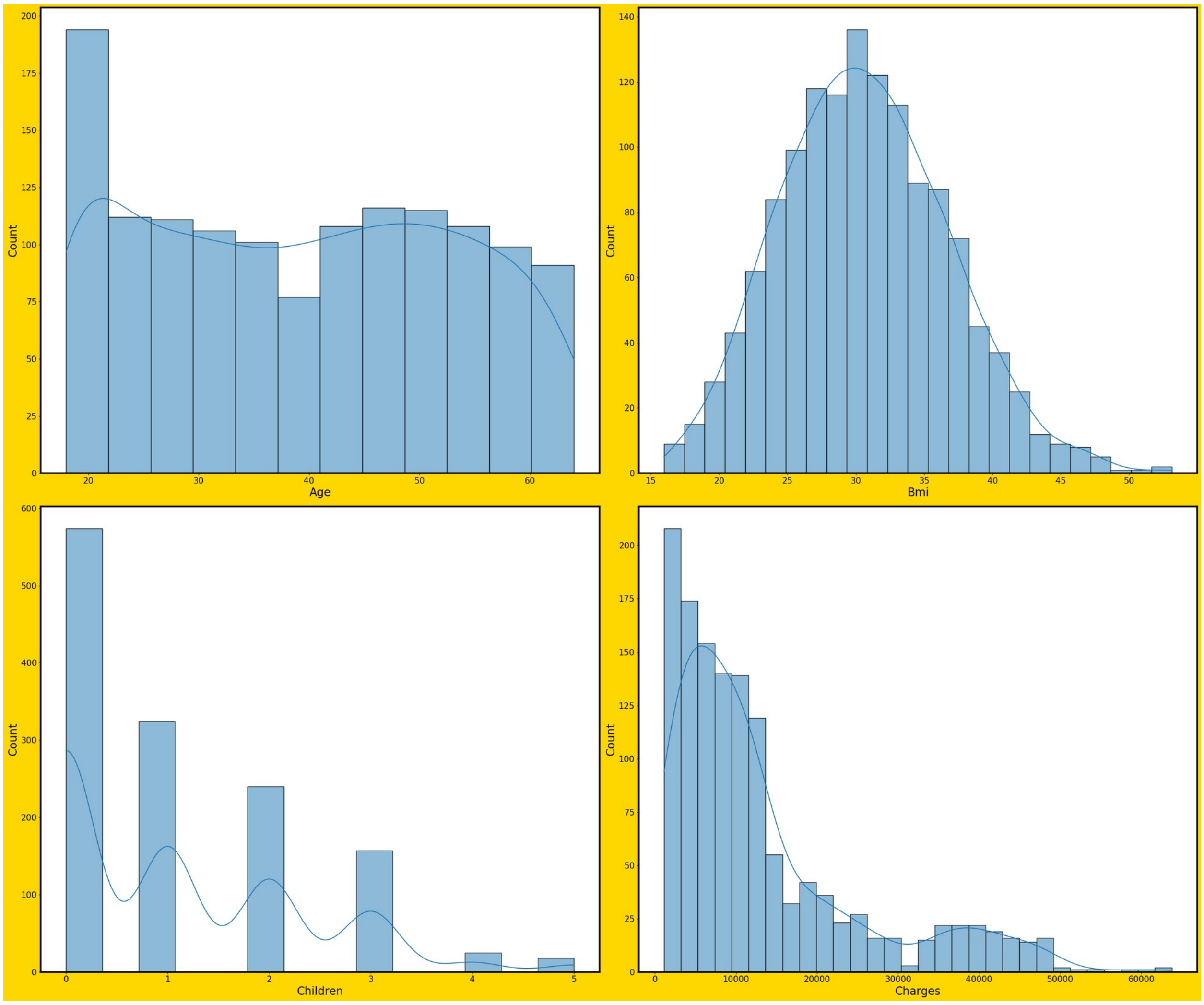
```
report = ProfileReport(data, title="EDA")
report
```


```
{"model_id":"6edcd35d081240b6959d6e337e201da7","version_major":2,"version_minor":0}

{"model_id":"80ec8f5d281a4142b1bd4b3f5909a7f2","version_major":2,"version_minor":0}

{"model_id":"b9589efe85d2446490874e046cd9925d","version_major":2,"version_minor":0}
```


```
<IPython.core.display.HTML object>
```

## 4.2. UNIVARIATE ANALYSIS FOR NUMERICAL COLUMN

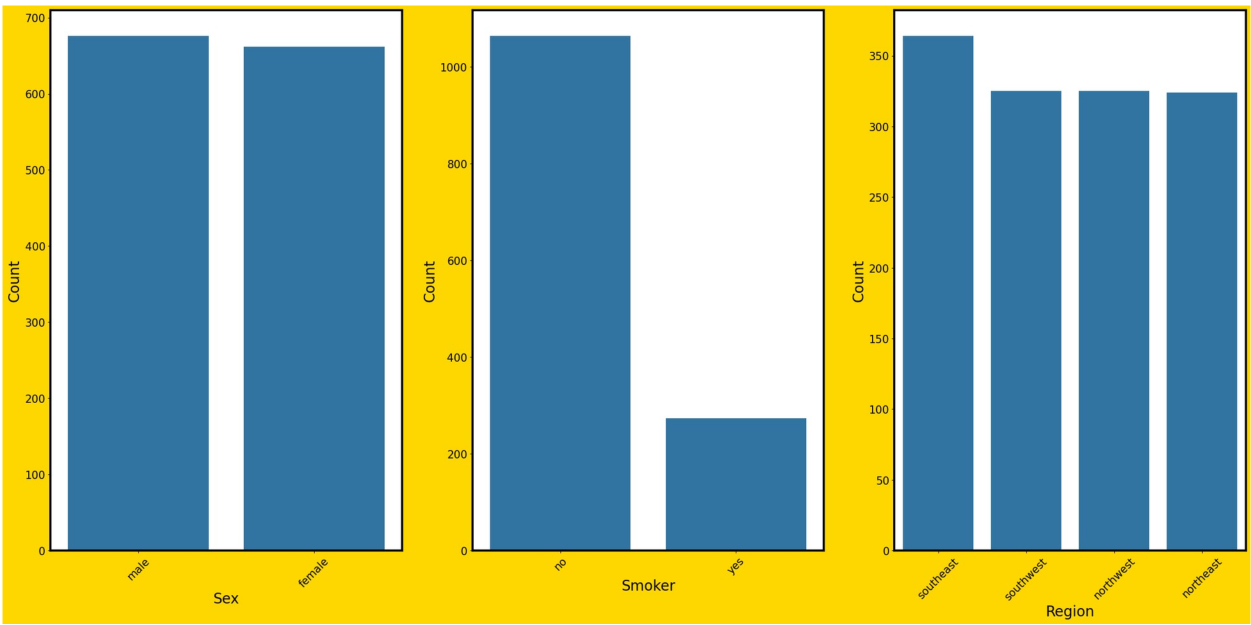
```
plt.figure(figsize=(30,25),facecolor="gold")
pltnum =1

for i in numerical_cols:
    if pltnum <=4:
        sp = plt.subplot(2,2,pltnum)

        sns.histplot(data[i],kde=True)

        plt.xlabel(i.title(),fontsize=20)
        plt.ylabel("Count",fontsize=20)
        plt.xticks(fontsize=15)
        plt.yticks(fontsize=15)

        sp.spines['top'].set_linewidth(3)
        sp.spines['bottom'].set_linewidth(3)
        sp.spines['left'].set_linewidth(3)
        sp.spines['right'].set_linewidth(3)

    pltnum +=1

plt.tight_layout()
plt.show()
```




## 4.3. UNIVARIATE ANALYSIS FOR CATEGORICAL COLUMNS

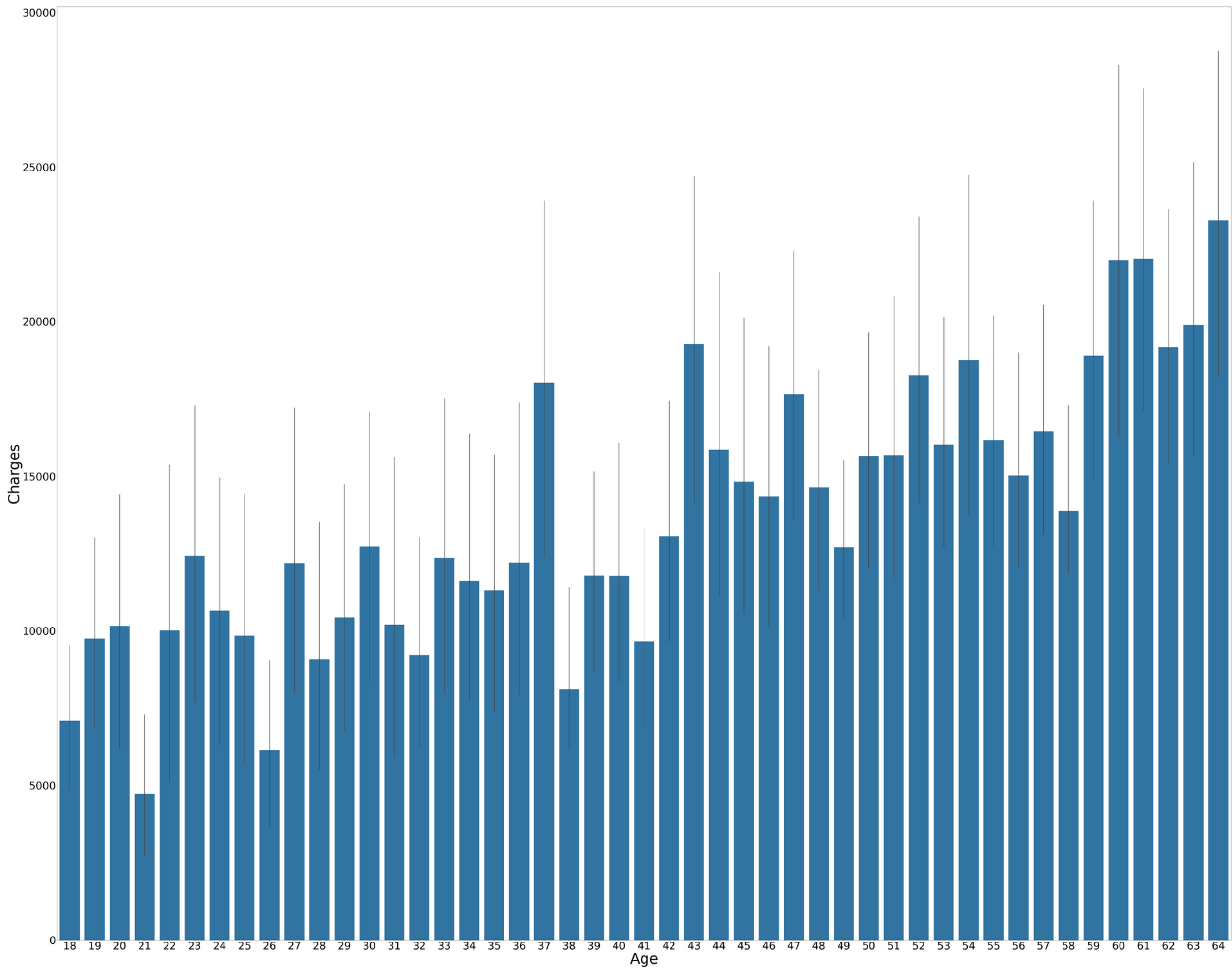
```
plt.figure(figsize=(30,30),facecolor="gold")

plttnum=1

for i in categorical_cols:
    if plttnum <= 3:
        sp = plt.subplot(2, 3, plttnum)

        sns.barplot(x=data[i].value_counts().index,
y=data[i].value_counts().values)

        plt.xlabel(i.title(), fontsize=20)
        plt.ylabel("Count", fontsize=20)
        plt.xticks(rotation=45, fontsize=15)
        plt.yticks(fontsize=15)
```

```
        sp.spines['top'].set_linewidth(3)
        sp.spines['bottom'].set_linewidth(3)
        sp.spines['left'].set_linewidth(3)
        sp.spines['right'].set_linewidth(3)

    plttnum += 1


plt.show()
```

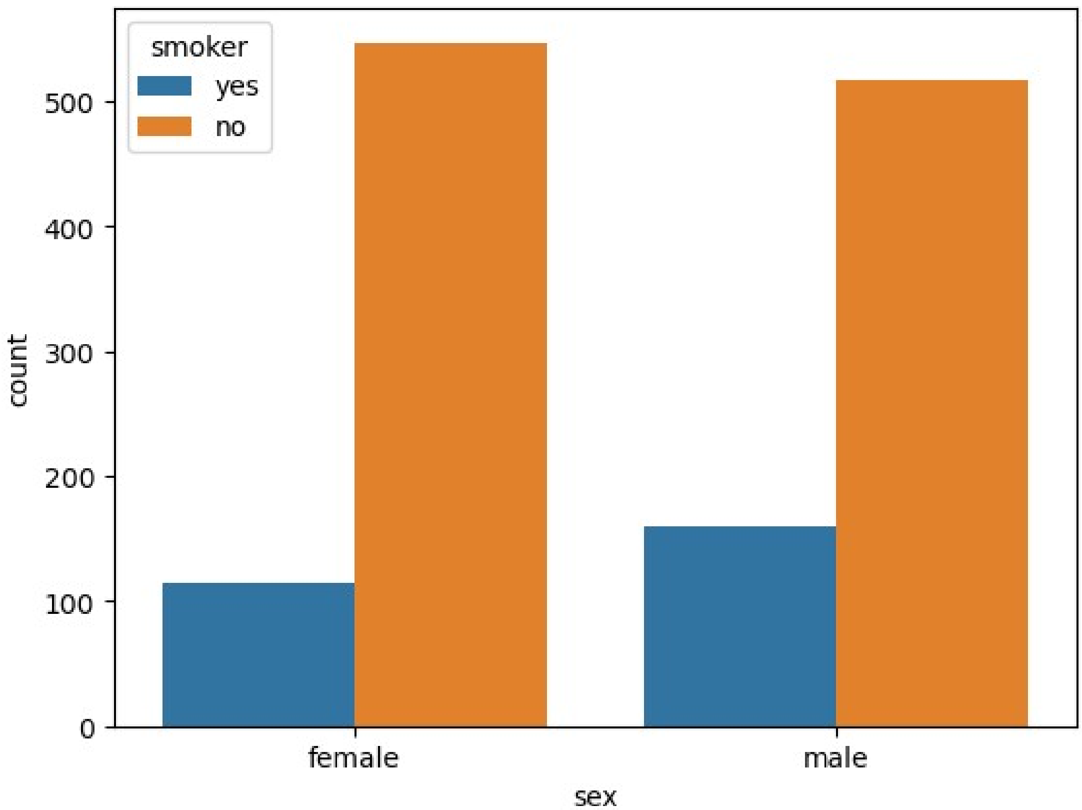



# 4.4. BIVARIATE ANALYSIS

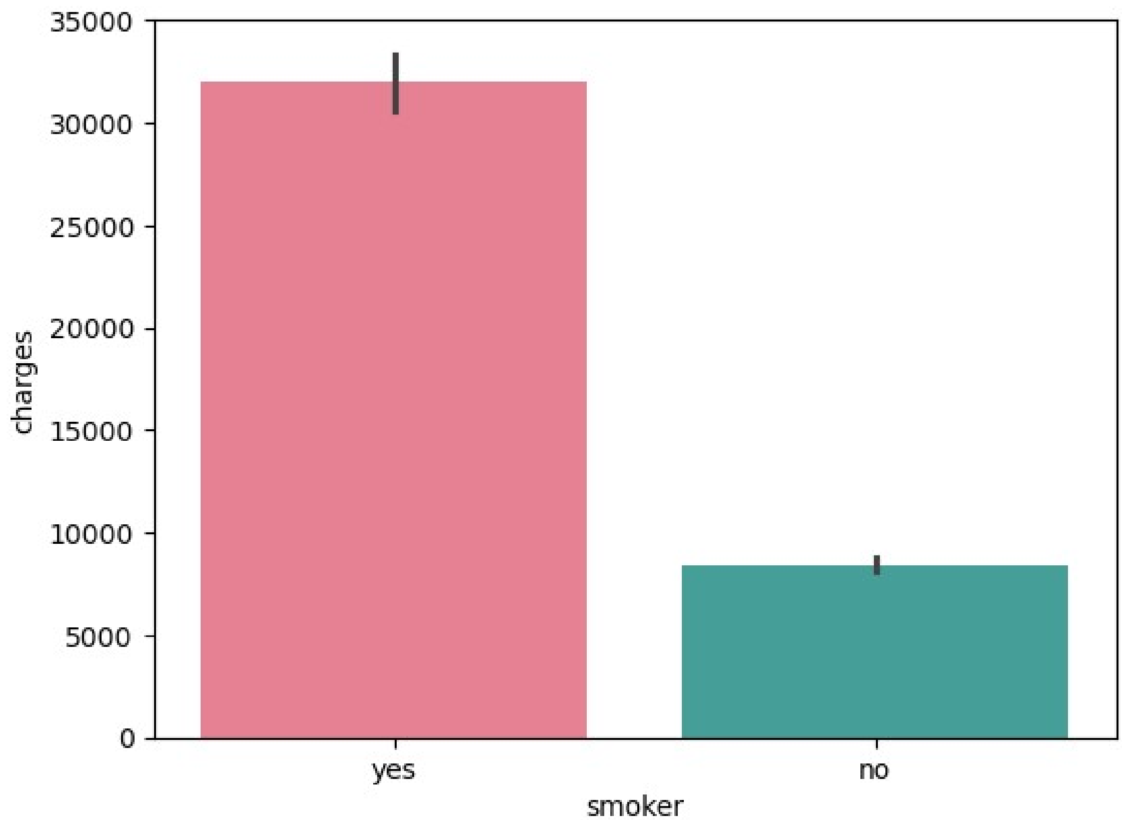
```
# Age vs Charges
plt.figure(figsize=(100,80))

sns.barplot(data,x="age",y="charges")

plt.xlabel("Age",fontsize=70)
plt.ylabel("Charges",fontsize=70)
plt.xticks(fontsize=50)
plt.yticks(fontsize=50)
plt.show()
```



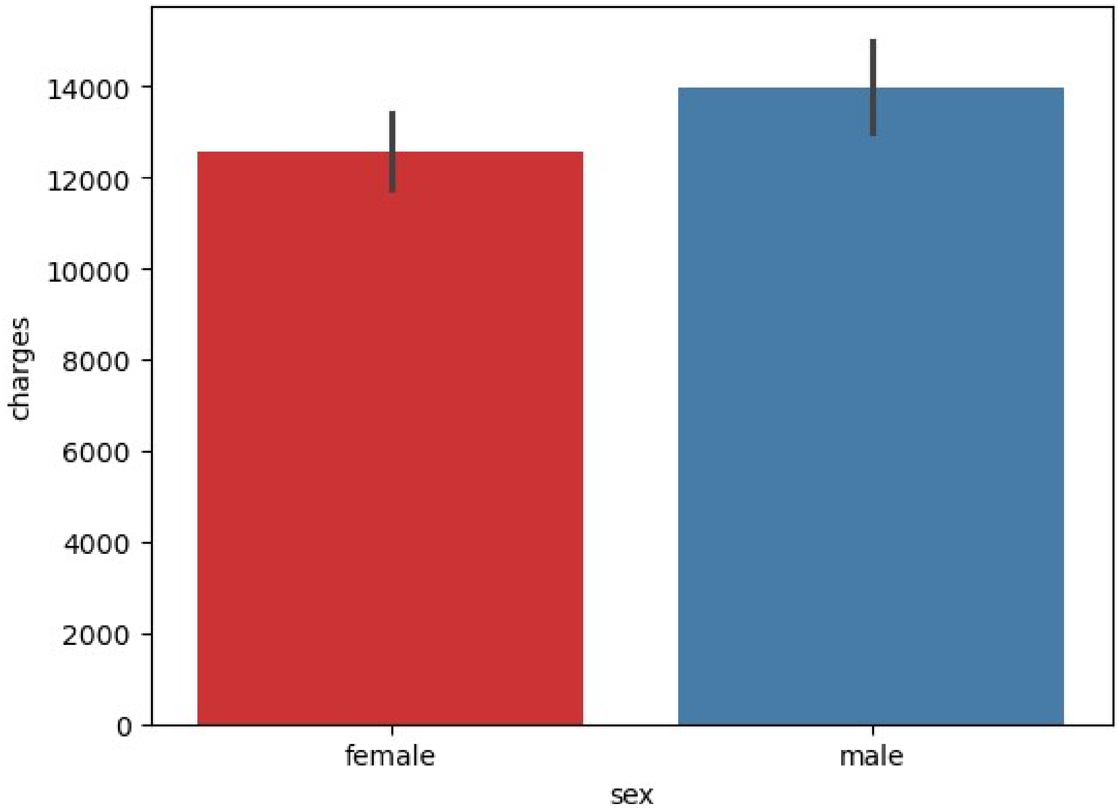

```
# Visualizing the number of smokers in each gender
sns.countplot(x='sex',hue='smoker',data=data)
plt.show()
```




```
# Number of female smokers in the dataset- 115
len(data[(data["sex"]=="female") & (data["smoker"]=="yes")])

115

# Number of male smokers in the dataset - 159
len(data[(data["sex"]=="male") & (data["smoker"]=="yes")])

159

# The highest charge value in the target column -  63770.42801
data["charges"].max()

63770.42801

# The lowest charge value in the target column -  1121.8739
data["charges"].min()

1121.8739

# smoker vs charges
sns.barplot(data,x="smoker",y="charges",palette="husl")
plt.show()
```

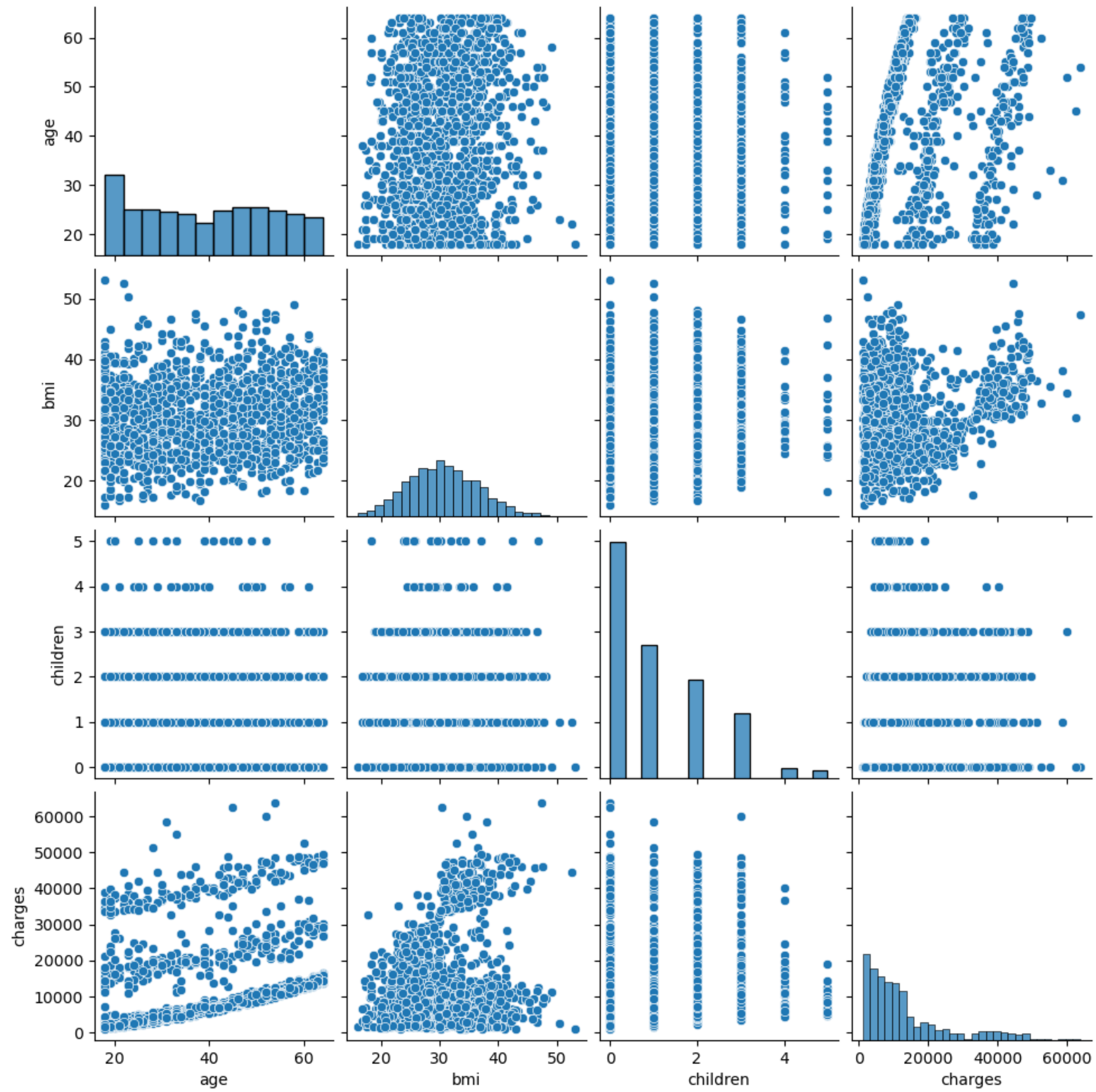



```python
# sex vs charges
sns.barplot(data,x="sex",y="charges",palette="Set1")
plt.show()
```




# 4.5. MULTIVARIATE ANALYSIS

```
sns.pairplot(data)
plt.show()
```




## INSIGHTS FROM EDA

1. There are 91 individuals with a BMI over 40, classified as Obesity Class 3 (Extremely Obese), putting them at a higher risk of developing conditions such as Type 2 diabetes, hypertension, and heart disease. Due to their increased likelihood of requiring medical expense support earlier than others, the company can adjust premiums accordingly to maintain profitability

2. The dataset consists of 20.5% smokers and 79.5% non-smokers, indicating that the number of non-smokers is significantly higher than that of smokers

3. The 'age' column is highly correlated with 'charges', meaning that as age increases, medical charges also tend to increase

4. There are 159 male smokers and 115 female smokers in the dataset, indicating that the number of male smokers is higher than that of female smokers

5. The highest insurance charge in the dataset is 63,770.42801, while the lowest is 1,121.8739

6. There is a significant correlation between smoking and medical charges, indicating that smokers tend to have higher medical expenses due to increased health risks

# 5. DATA PREPROCESSING

## 5.1. NULL VALUES CHECKING

```
data.isnull().sum()

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

## 5.2. DUPLICATE VALUES CHECKING

```
data.duplicated().sum()

1

dup = data[data.duplicated(keep=False)]
dup
# rows located in the index of 195 and 581 are duplicated

     age   sex    bmi  children smoker     region    charges
195   19  male  30.59         0     no  northwest  1639.5631
581   19  male  30.59         0     no  northwest  1639.5631

# Dropping the duplicates
data.drop_duplicates(inplace=True)

# Verification
data.duplicated().sum()

0

data.shape # One duplicate removed out of 1338

(1337, 7)
```

## 5.3. CHECK FOR CORRUPTED VALUES

### AGE

```
data[data["age"]==0]
```

```
Empty DataFrame
Columns: [age, sex, bmi, children, smoker, region, charges]
Index: []
```

### BMI

```
data[data["bmi"]==0]
```

```
Empty DataFrame
Columns: [age, sex, bmi, children, smoker, region, charges]
Index: []
```

### CHARGES

```
data[data["charges"]==0]
```

```
Empty DataFrame
Columns: [age, sex, bmi, children, smoker, region, charges]
Index: []
```

```
# Check for negative values in numerical columns
for i in numerical_cols:

    negative=data[data[i]<0]

    if negative.empty:
        print(f'{i}: There is no negative values in this column')
    else:
        print(negative)
```

```
age: There is no negative values in this column
bmi: There is no negative values in this column
children: There is no negative values in this column
charges: There is no negative values in this column
```

### Insights

1. There are no null values present in this dataset
2. There is only one duplicate row in the dataset and has been removed successfully
3. The 'age', 'bmi', 'charges' columns doesn't have the value '0'
4. The 'age', 'bmi', 'charges' and 'children' doesn't have any negative values

Therefore, Points 3.& 4. infers that there are no corrupted values in the dataset

# 5.4. ENCODING CATEGORICAL COLUMNS

## 5.4.1. ONE HOT ENCODING

SEX

```
# Initializing OneHotEncoder
ohe_sex = OneHotEncoder()

# Fitting and transforming the "Sex" column
sex_encoded = ohe_sex.fit_transform(data[['sex']]).toarray()
sex_encoded
```

```
array([[1., 0.],
       [0., 1.],
       [0., 1.],
       ...,
       [1., 0.],
       [1., 0.],
       [1., 0.]])
```

```
# Getting OneHotEncoder output features names
ohe_sex.get_feature_names_out()
```

```
array(['sex_female', 'sex_male'], dtype=object)
```

```
# Creating a DataFrame of encoded array
sex_encoded_df = pd.DataFrame(sex_encoded.astype(int),
index=data.index, columns=['sex_female', 'sex_male'])
sex_encoded_df
```

```
      sex_female  sex_male
0              1         0
1              0         1
2              0         1
3              0         1
4              0         1
...          ...       ...
1333           0         1
1334           1         0
1335           1         0
1336           1         0
1337           1         0

[1337 rows x 2 columns]
```

```
# Dropping first column from sex_encoded_df
sex_encoded_df.drop('sex_female',axis=1,inplace=True)

sex_encoded_df
```

```
      sex_male
0            0
1            1
2            1
3            1
4            1
...        ...
1333         1
1334         0
1335         0
1336         0
1337         0

[1337 rows x 1 columns]

# Concatenating "sex_encoded_df" to the original dataframe
data = pd.concat([sex_encoded_df,data],axis=1)

data

      sex_male  age     sex     bmi  children smoker     region
charges
0            0   19  female  27.900         0    yes  southwest
16884.92400
1            1   18    male  33.770         1     no  southeast
1725.55230
2            1   28    male  33.000         3     no  southeast
4449.46200
3            1   33    male  22.705         0     no  northwest
21984.47061
4            1   32    male  28.880         0     no  northwest
3866.85520
...        ...  ...     ...     ...       ...    ...        ...
...
1333         1   50    male  30.970         3     no  northwest
10600.54830
1334         0   18  female  31.920         0     no  northeast
2205.98080
1335         0   18  female  36.850         0     no  southeast
1629.83350
1336         0   21  female  25.800         0     no  southwest
2007.94500
1337         0   61  female  29.070         0    yes  northwest
29141.36030

[1337 rows x 8 columns]

# Removing the 'Sex' column from the data
data.drop("sex",axis=1,inplace=True)

data
```

```
      sex_male  age     bmi  children smoker     region      charges
0            0   19  27.900         0    yes  southwest  16884.92400
1            1   18  33.770         1     no  southeast   1725.55230
2            1   28  33.000         3     no  southeast   4449.46200
3            1   33  22.705         0     no  northwest  21984.47061
4            1   32  28.880         0     no  northwest   3866.85520
...        ...  ...     ...       ...    ...        ...          ...
1333         1   50  30.970         3     no  northwest  10600.54830
1334         0   18  31.920         0     no  northeast   2205.98080
1335         0   18  36.850         0     no  southeast   1629.83350
1336         0   21  25.800         0     no  southwest   2007.94500
1337         0   61  29.070         0    yes  northwest  29141.36030

[1337 rows x 7 columns]
```

REGION

```
# Initializing OneHotEncoder
ohe_region = OneHotEncoder()

# Fit and transform of "Region" column
region_encoded = ohe_region.fit_transform(data[["region"]]).toarray()
region_encoded

array([[0., 0., 0., 1.],
       [0., 0., 1., 0.],
       [0., 0., 1., 0.],
       ...,
       [0., 0., 1., 0.],
       [0., 0., 0., 1.],
       [0., 1., 0., 0.]])

# Get OneHotEncoder output features names
ohe_region.get_feature_names_out()

array(['region_northeast', 'region_northwest', 'region_southeast',
       'region_southwest'], dtype=object)

# Creating a DataFrame of encoded array
region_encoded_df = pd.DataFrame(region_encoded.astype(int),
index=data.index, columns=['region_northeast', 'region_northwest',
'region_southeast',
       'region_southwest'])
region_encoded_df

      region_northeast  region_northwest  region_southeast
region_southwest
0                    0                 0                 0
1
1                    0                 0                 1
0
```

```
2                     0                  0                  1
0
3                     0                  1                  0
0
4                     0                  1                  0
0
...                 ...                ...                ...
...
1333                  0                  1                  0
0
1334                  1                  0                  0
0
1335                  0                  0                  1
0
1336                  0                  0                  0
1
1337                  0                  1                  0
0

[1337 rows x 4 columns]

# Dropping first column from region_encoded_df
region_encoded_df.drop('region_northeast',axis=1,inplace=True)

region_encoded_df

      region_northwest  region_southeast  region_southwest
0                    0                 0                 1
1                    0                 1                 0
2                    0                 1                 0
3                    1                 0                 0
4                    1                 0                 0
...                ...               ...               ...
1333                 1                 0                 0
1334                 0                 0                 0
1335                 0                 1                 0
1336                 0                 0                 1
1337                 1                 0                 0

[1337 rows x 3 columns]

# Concatenating "region_encoded_df" to the original dataframe
data = pd.concat([region_encoded_df,data],axis=1)

data

      region_northwest  region_southeast  region_southwest  sex_male
age  \
0                    0                 0                 1         0
19
1                    0                 1                 0         1
```

```
18
2                     0                  1                  0         1
28
3                     1                  0                  0         1
33
4                     1                  0                  0         1
32
...                 ...                ...                ...       ...
...
1333                  1                  0                  0         1
50
1334                  0                  0                  0         0
18
1335                  0                  1                  0         0
18
1336                  0                  0                  1         0
21
1337                  1                  0                  0         0
61

         bmi  children smoker     region      charges
0     27.900         0    yes  southwest  16884.92400
1     33.770         1     no  southeast   1725.55230
2     33.000         3     no  southeast   4449.46200
3     22.705         0     no  northwest  21984.47061
4     28.880         0     no  northwest   3866.85520
...      ...       ...    ...        ...          ...
1333  30.970         3     no  northwest  10600.54830
1334  31.920         0     no  northeast   2205.98080
1335  36.850         0     no  southeast   1629.83350
1336  25.800         0     no  southwest   2007.94500
1337  29.070         0    yes  northwest  29141.36030

[1337 rows x 10 columns]

# Removing the 'Region' column from the data
data.drop("region",axis=1,inplace=True)

data

      region_northwest  region_southeast  region_southwest  sex_male
age  \
0                     0                  0                  1         0
19
1                     0                  1                  0         1
18
2                     0                  1                  0         1
28
3                     1                  0                  0         1
33
```

```
4                      1                  0                  0          1
32
...                  ...                ...                ...        ...
...
1333                   1                  0                  0          1
50
1334                   0                  0                  0          0
18
1335                   0                  1                  0          0
18
1336                   0                  0                  1          0
21
1337                   1                  0                  0          0
61

         bmi  children smoker      charges
0     27.900         0    yes  16884.92400
1     33.770         1     no   1725.55230
2     33.000         3     no   4449.46200
3     22.705         0     no  21984.47061
4     28.880         0     no   3866.85520
...      ...       ...    ...          ...
1333  30.970         3     no  10600.54830
1334  31.920         0     no   2205.98080
1335  36.850         0     no   1629.83350
1336  25.800         0     no   2007.94500
1337  29.070         0    yes  29141.36030

[1337 rows x 9 columns]
```

## 5.4.2 LABEL ENCODING

SMOKER

```
# Initializing LabelEncoder
le_smoker = LabelEncoder()

# Fitting and transforming the 'Smoker' column
data["smoker"] = le_smoker.fit_transform(data["smoker"])

data

      region_northwest  region_southeast  region_southwest  sex_male
age  \
0                    0                 0                 1         0
19
1                    0                 1                 0         1
18
2                    0                 1                 0         1
28
```

```
3                     1                  0                  0          1
33
4                     1                  0                  0          1
32
...                 ...                ...                ...        ...
...
1333                  1                  0                  0          1
50
1334                  0                  0                  0          0
18
1335                  0                  1                  0          0
18
1336                  0                  0                  1          0
21
1337                  1                  0                  0          0
61

         bmi  children  smoker      charges
0     27.900         0       1  16884.92400
1     33.770         1       0   1725.55230
2     33.000         3       0   4449.46200
3     22.705         0       0  21984.47061
4     28.880         0       0   3866.85520
...      ...       ...     ...          ...
1333  30.970         3       0  10600.54830
1334  31.920         0       0   2205.98080
1335  36.850         0       0   1629.83350
1336  25.800         0       0   2007.94500
1337  29.070         0       1  29141.36030

[1337 rows x 9 columns]
```

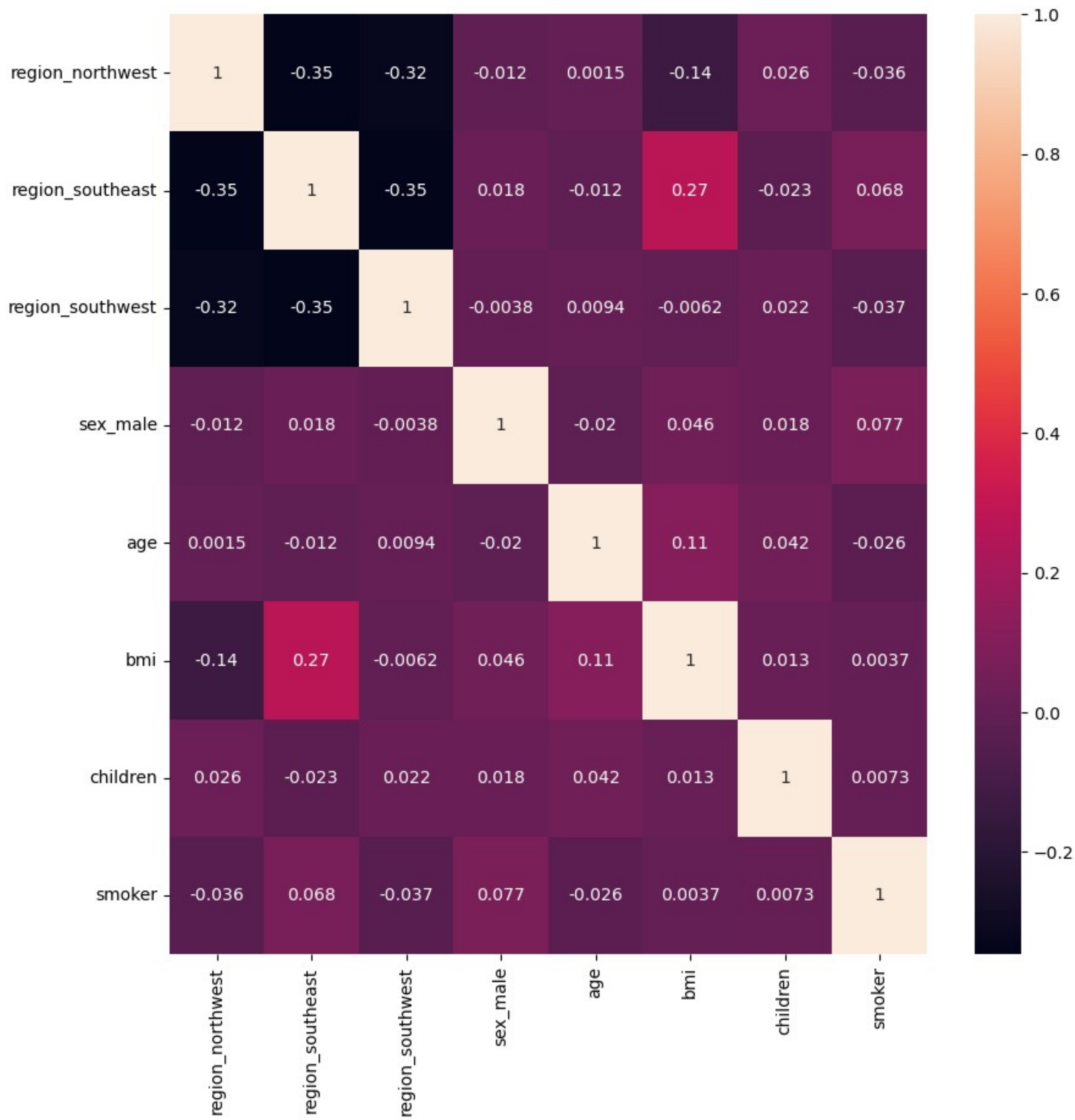
# 5.5. SCALING FOR NUMERICAL COLUMNS

```
# Using MinmaxScaler

min_max = MinMaxScaler()

data[numerical_cols] = min_max.fit_transform(data[numerical_cols])

data

      region_northwest  region_southeast  region_southwest
sex_male  \
0                     0                  0                  1          0

1                     0                  1                  0          1

2                     0                  1                  0          1
```

```
3                      1                  0                  0          1

4                      1                  0                  0          1

...                  ...                ...                ...        ...

1333                   1                  0                  0          1

1334                   0                  0                  0          0

1335                   0                  1                  0          0

1336                   0                  0                  1          0

1337                   1                  0                  0          0


           age       bmi  children  smoker   charges
0     0.021739  0.321227       0.0       1  0.251611
1     0.000000  0.479150       0.2       0  0.009636
2     0.217391  0.458434       0.6       0  0.053115
3     0.326087  0.181464       0.0       0  0.333010
4     0.304348  0.347592       0.0       0  0.043816
...        ...       ...       ...     ...       ...
1333  0.695652  0.403820       0.6       0  0.151299
1334  0.000000  0.429379       0.0       0  0.017305
1335  0.000000  0.562012       0.0       0  0.008108
1336  0.065217  0.264730       0.0       0  0.014144
1337  0.934783  0.352704       0.0       1  0.447249

[1337 rows x 9 columns]
```

## 5.6. CORRELATION

```
# Check for correlation to avoid multicollinearity
plt.figure(figsize=(10,10))
sns.heatmap(data=data.iloc[:,:-1].corr(),annot=True)
plt.show()
```




```
data.iloc[:,:-1].corr()
```

```
                   region_northwest  region_southeast  region_southwest  \
region_northwest           1.000000         -0.345909         -0.320493

region_southeast          -0.345909          1.000000         -0.346614

region_southwest          -0.320493         -0.346614          1.000000

sex_male                  -0.012482          0.017578         -0.003767
```

```
age                            0.001495            -0.012311            0.009415

bmi                           -0.136138             0.270057           -0.006211

children                       0.026044            -0.023492            0.021538

smoker                        -0.036321             0.068282           -0.037168


                  sex_male       age       bmi  children    smoker
region_northwest -0.012482  0.001495 -0.136138  0.026044 -0.036321
region_southeast  0.017578 -0.012311  0.270057 -0.023492  0.068282
region_southwest -0.003767  0.009415 -0.006211  0.021538 -0.037168
sex_male          1.000000 -0.019814  0.046397  0.017848  0.076596
age              -0.019814  1.000000  0.109344  0.041536 -0.025587
bmi               0.046397  0.109344  1.000000  0.012755  0.003746
children          0.017848  0.041536  0.012755  1.000000  0.007331
smoker            0.076596 -0.025587  0.003746  0.007331  1.000000
```

# 6. DATA SPLITTING

```
x = data.iloc[:,:-1] # Independent variables
x

      region_northwest  region_southeast  region_southwest
sex_male  \
0                    0                 0                 1         0

1                    0                 1                 0         1

2                    0                 1                 0         1

3                    1                 0                 0         1

4                    1                 0                 0         1

...                ...               ...               ...       ...

1333                 1                 0                 0         1

1334                 0                 0                 0         0

1335                 0                 1                 0         0

1336                 0                 0                 1         0

1337                 1                 0                 0         0
```

```
           age       bmi  children  smoker
0     0.021739  0.321227       0.0       1
1     0.000000  0.479150       0.2       0
2     0.217391  0.458434       0.6       0
3     0.326087  0.181464       0.0       0
4     0.304348  0.347592       0.0       0
...        ...       ...       ...     ...
1333  0.695652  0.403820       0.6       0
1334  0.000000  0.429379       0.0       0
1335  0.000000  0.562012       0.0       0
1336  0.065217  0.264730       0.0       0
1337  0.934783  0.352704       0.0       1

[1337 rows x 8 columns]
```

```
y = data.iloc[:,-1] # Target variable
y
```

```
0       0.251611
1       0.009636
2       0.053115
3       0.333010
4       0.043816
          ...
1333    0.151299
1334    0.017305
1335    0.008108
1336    0.014144
1337    0.447249
Name: charges, Length: 1337, dtype: float64
```

```
# train_test_split for splitting the data
x_train, x_test, y_train, y_test = train_test_split(x,y,train_size =
0.8, random_state = 11)
```

```
x_train.shape
```

```
(1069, 8)
```

```
x_test.shape
```

```
(268, 8)
```

```
y_train.shape
```

```
(1069,)
```

```
y_test.shape
```

```
(268,)
```

# 7. MODEL CREATION

## 7.1. LINEAR REGRESSION

```
lr_model = LinearRegression()

lr_model.fit(x_train, y_train)

LinearRegression()

y_pred_linear = lr_model.predict(x_test)
y_pred_linear

array([ 0.43307651,  0.06485189,  0.15529651,  0.00325287,
0.18265638,
        0.15950923,  0.10594521,  0.23201013,  0.21238288,
0.22242137,
        0.10958839,  0.22910189,  0.04412948,  0.01613748,
0.08706258,
        0.55079696,  0.04034759,  0.1136747 ,  0.11212339,
0.18890999,
        0.15881654,  0.03305363,  0.0317627 ,  0.04692156,
0.04919854,
        0.22864428,  0.07036419,  0.11840696,  0.40755256,
0.03118554,
        0.51768944,  0.0087606 ,  0.01767745, -0.02274578,
0.57102328,
        0.49293264,  0.0749543 ,  0.40794263,  0.11159944,
0.12606644,
        0.08026446,  0.19343267,  0.41580639, -0.00209118,
0.12516938,
        0.16204242,  0.18030471,  0.02992358,  0.12902902,
0.41151532,
        0.19552163,  0.45114212,  0.22713368,  0.03670732,
0.17611673,
        0.02759561,  0.04478799,  0.11396797,  0.51336184,
0.07033212,
        0.13529423,  0.19366186,  0.10416806,  0.22639027,
0.06843883,
        0.10504687,  0.17891029,  0.41876246,  0.18517551,
0.0065349 ,
        0.51665328,  0.14320974,  0.07663294,  0.09862289,
0.09019788,
       -0.01442756,  0.15644795,  0.1576996 ,  0.04396406,
0.01643294,
        0.0860725 ,  0.07061395,  0.01896179,  0.52659898,
0.44484003,
        0.09363113,  0.21295854, -0.05345342,  0.53859066,
0.19401039,
        0.23397863,  0.01406549,  0.15508185,  0.07145579,
```

```
0.15707264,
        0.00654601,  0.09428648,  0.29947898,  0.17235815,
0.5129711 ,
        0.57676129,  0.26054737,  0.06013779, -0.0161384 ,
0.07623354,
        0.13007476,  0.10561992,  0.17423859,  0.10146481,
0.23744354,
        0.08755066,  0.19457898,  0.12643957,  0.01422011,
0.55780675,
       -0.00305631,  0.07590167,  0.20143198,  0.13625522,
0.10241698,
        0.05109323,  0.0125483 ,  0.4140685 ,  0.17647765,
0.4836091 ,
        0.07093277,  0.11435964,  0.0221663 , -0.01638612,
0.05629887,
        0.02790713,  0.17025962,  0.09779545,  0.02707649,
0.44019508,
        0.05766714,  0.04953177,  0.42615775,  0.37351711,
0.00631423,
        0.09189707,  0.2212557 ,  0.17341703,  0.15216527,
0.19507089,
        0.12071745,  0.14160155,  0.04608313,  0.24565782,
0.5183326 ,
        0.46773408,  0.07352921,  0.05053146,  0.0992339 ,
0.22396471,
        0.4870612 ,  0.10089043,  0.23527779,  0.0840032 ,
0.28058071,
        0.01538247,  0.0073356 ,  0.10980642,  0.15434467,
0.05629887,
        0.193182  ,  0.41787017,  0.59878786,  0.19857373, -
0.02126867,
        0.51893711,  0.02271802,  0.07410584,  0.13438625,
0.39482641,
        0.03889744,  0.02747143,  0.06689157,  0.1509471 ,
0.17495353,
        0.07145916, -0.00083346,  0.52619652,  0.14307882,
0.0940363 ,
        0.24164716,  0.08831905,  0.14572699,  0.06759049,
0.14497417,
        0.62304892,  0.14541535,  0.10677563,  0.15129412,
0.1329025 ,
        0.45087172,  0.41364326,  0.05273397,  0.17344957,
0.11732533,
        0.61423318,  0.25851927,  0.59933362,  0.21844493,
0.02042875,
        0.07957968,  0.55402445,  0.22776128,  0.08067789,
0.53704544,
        0.18828029,  0.1963738 ,  0.18414001,  0.20792133,
0.18317443,
```

```
        0.1482773 ,  0.45438436,  0.13727843,  0.1034826 ,
0.37984034,
        0.09672258,  0.40992089, -0.02142015,  0.23960967,
0.23721358,
        0.20417354,  0.18614004,  0.40831929,  0.18411662,
0.06953007,
        0.03248225,  0.10812962,  0.23553772,  0.11087326,
0.14886855,
        0.15571079,  0.24757148,  0.21863073,  0.19020286,
0.05986923,
        0.0921232 ,  0.11933765,  0.01812196,  0.57044578,
0.5319589 ,
        0.06733356,  0.21981618,  0.24336801,  0.07139214,
0.09336398,
        0.46985086,  0.12597855,  0.0735869 ,  0.00601657,
0.16636826,
        0.48764177,  0.08786907,  0.48875207,  0.4075527 ,
0.04545544,
        0.11818595,  0.23790509,  0.22352277,  0.02321887,
0.13642489,
        0.07098239,  0.49361395,  0.14355369])

linear_reg_r2 = r2_score(y_test, y_pred_linear)
print(f'r2_score (Linear Regression) : {linear_reg_r2}')

r2_score (Linear Regression) : 0.8141312870011562
```

## 7.2. DECISION TREE

```
DTR=DecisionTreeRegressor()
model=DTR.fit(x_train,y_train)
model

DecisionTreeRegressor()

y_pred_dtr=model.predict(x_test)
print(f'R2_score (Decision Tree) : {r2_score(y_test,y_pred_dtr)}')
print(f'mean_squared_error       :
{mean_squared_error(y_test,y_pred_dtr)}')
print(f'mean_absolute_error      :
{mean_absolute_error(y_test,y_pred_dtr)}')
print(f'root_mean_square_error   :
{np.sqrt(mean_squared_error(y_test,y_pred_dtr))}')

R2_score (Decision Tree) : 0.6608437925413919
mean_squared_error       : 0.011328934160013193
mean_absolute_error      : 0.05196216834715074
root_mean_square_error   : 0.106437465960127
```

### 7.2.1. HYPERPARAMETER TUNING (DECISION TREE)

```
params={'splitter':["best", "random"],
        'criterion':["squared_error", "absolute_error"],
        'max_depth':list(range(1,20)),
        'min_samples_split':list(range(1,10)),
        'min_samples_leaf':list(range(1,10))}

tree_reg=DecisionTreeRegressor()
tree_cv=RandomizedSearchCV(tree_reg,param_distributions=params,scoring
='r2',n_jobs=-1,verbose=3,cv=3,n_iter=300)
tree_cv.fit(x_train,y_train)
```

```
Fitting 3 folds for each of 300 candidates, totalling 900 fits
```

```
RandomizedSearchCV(cv=3, estimator=DecisionTreeRegressor(),
n_iter=300,
                   n_jobs=-1,
                   param_distributions={'criterion': ['squared_error',

'absolute_error'],
                                        'max_depth': [1, 2, 3, 4, 5,
6, 7, 8, 9,
                                                      10, 11, 12, 13,
14, 15,
                                                      16, 17, 18, 19],
                                        'min_samples_leaf': [1, 2, 3,
4, 5, 6,
                                                             7, 8, 9],
                                        'min_samples_split': [1, 2, 3,
4, 5, 6,
                                                              7, 8,
9],
                                        'splitter': ['best',
'random']},
                   scoring='r2', verbose=3)
```

```
best_params=tree_cv.best_params_
print(f'best parameters:{best_params}')
```

```
best parameters:{'splitter': 'best', 'min_samples_split': 9,
'min_samples_leaf': 4, 'max_depth': 15, 'criterion': 'absolute_error'}
```

```
dt=DecisionTreeRegressor(splitter='best', min_samples_split=9,
min_samples_leaf=4, max_depth=14, criterion='absolute_error')
dt.fit(x_train,y_train)
y_pred=dt.predict(x_test)
decisiontree_reg_r2=r2_score(y_test,y_pred)
print(f'r2_score (decision tree):{decisiontree_reg_r2}')
```

```
r2_score (decision tree):0.8903590019951106
```

## 7.3. RANDOM FOREST

```
RFR=RandomForestRegressor()

RFR.fit(x_train,y_train)

RandomForestRegressor()

y_pred_rfr=RFR.predict(x_test)

print(f'r2_score (random forest) : {r2_score(y_test,y_pred_rfr)}')
print(f'mean_squared_error       :
{mean_squared_error(y_test,y_pred_rfr)}')
print(f'mean_absolute_error      :
{mean_absolute_error(y_test,y_pred_rfr)}')
print(f'root_mean_squared_error  :
{np.sqrt(mean_squared_error(y_test,y_pred_rfr))}')

r2_score (random forest) : 0.8848414210565999
mean_squared_error       : 0.003846675750346344
mean_absolute_error      : 0.038296208953598386
root_mean_squared_error  : 0.0620215748779918
```

### 7.3.1. HYPERPARAMETER TUNING (RANDOM FOREST)

```
params={'n_estimators':[100,150,200,250,300],
        'max_features':['auto','sqrt','log2'],
        'max_depth':list(range(1,20)),
        'min_samples_split':list(range(1,10)),
        'min_samples_leaf':list(range(1,10)),
        'criterion': ['squared_error',"friedman_mse",
"absolute_error","poisson"]}

rf_cv=RandomizedSearchCV(RFR,param_distributions=params,scoring='r2',n
_jobs=-1,verbose=3,cv=5,n_iter=300)
rf_cv.fit(x_train,y_train)

Fitting 5 folds for each of 300 candidates, totalling 1500 fits

RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(),
n_iter=300,
                   n_jobs=-1,
                   param_distributions={'criterion': ['squared_error',
                                                      'friedman_mse',

'absolute_error',
                                                      'poisson'],
                                        'max_depth': [1, 2, 3, 4, 5,
6, 7, 8, 9,
                                                      10, 11, 12, 13,
14, 15,
```

```
                                                                        16, 17, 18, 19],
                                                      'max_features': ['auto',
'sqrt',
                                                                            'log2'],
                                                      'min_samples_leaf': [1, 2, 3,
4, 5, 6,
                                                                                 7, 8, 9],
                                                      'min_samples_split': [1, 2, 3,
4, 5, 6,
                                                                                  7, 8,
9],
                                                      'n_estimators': [100, 150,
200, 250,
                                                                            300]},
                          scoring='r2', verbose=3)

best_params=rf_cv.best_params_
print(f'best parameters:{best_params}')

best parameters:{'n_estimators': 200, 'min_samples_split': 3,
'min_samples_leaf': 1, 'max_features': 'log2', 'max_depth': 9,
'criterion': 'absolute_error'}

rf_best=RandomForestRegressor(n_estimators=250, min_samples_split=4,
min_samples_leaf=2, max_features='log2', max_depth=10,
criterion='absolute_error')
rf_best.fit(x_train,y_train)
y_pred_rfr_best=rf_best.predict(x_test)
randomforest_reg_r2=r2_score(y_test,y_pred_rfr_best)
print(randomforest_reg_r2)

0.9180899952300885
```

# 7.4. GRADIENT BOOSTING

```
GBR=GradientBoostingRegressor()
GBR.fit(x_train,y_train)
y_pred_gbr=GBR.predict(x_test)
print(f' r2_score(gradient boosting) : {r2_score(y_test,y_pred_gbr)}')
print(f' mean_absolute_error         :
{mean_absolute_error(y_test,y_pred_gbr)}')
print(f' mean_squared_error          :
{mean_squared_error(y_test,y_pred_gbr)}')
print(f' root_mean_squared_error     :
{mean_squared_error(y_test,y_pred_gbr,squared=False)}')

 r2_score(gradient boosting) : 0.9041690121331395
 mean_absolute_error         : 0.035548572215303345
 mean_squared_error          : 0.0032010705632306152
 root_mean_squared_error     : 0.056578004235132005
```

### 7.4.1 HYPERPARAMETER TUNING (GRADIENT BOOSTING)

```
params={'n_estimators':[100,200,300],
        'learning_rate':[0.001,0.01,0.02,0.03,0.1],
        'max_depth':list(range(1,20)),
        'min_samples_split':list(range(1,10)),
        'min_samples_leaf':list(range(1,10)),
       }

gb_cv=RandomizedSearchCV(estimator=GBR,scoring='r2',param_distribution
s=params,cv=5,verbose=2,n_jobs=-1,n_iter=300)
gb_cv.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 300 candidates, totalling 1500 fits
```

```
RandomizedSearchCV(cv=5, estimator=GradientBoostingRegressor(),
n_iter=300,
                   n_jobs=-1,
                   param_distributions={'learning_rate': [0.001, 0.01,
0.02,
                                                          0.03, 0.1],
                                        'max_depth': [1, 2, 3, 4, 5,
6, 7, 8, 9,
                                                      10, 11, 12, 13,
14, 15,
                                                      16, 17, 18, 19],
                                        'min_samples_leaf': [1, 2, 3,
4, 5, 6,
                                                             7, 8, 9],
                                        'min_samples_split': [1, 2, 3,
4, 5, 6,
                                                              7, 8,
9],
                                        'n_estimators': [100, 200,
300]},
                   scoring='r2', verbose=2)
```

```
best_params=gb_cv.best_params_
print(f'best_params:{best_params}')
```

```
best_params:{'n_estimators': 200, 'min_samples_split': 3,
'min_samples_leaf': 5, 'max_depth': 3, 'learning_rate': 0.03}
```

```
gbr_best=GradientBoostingRegressor(n_estimators=200,
min_samples_split=4, min_samples_leaf=7, max_depth=3,
learning_rate=0.03)
gbr_best.fit(x_train,y_train)
y_pred_gbr_best=gbr_best.predict(x_test)
gradboost_reg_r2=r2_score(y_test,y_pred_gbr_best)
print(gradboost_reg_r2)
```

```
0.9157548130632605
```

## 7.5. XGBOOSTING

```
xgb=XGBRegressor()
xgb.fit(x_train,y_train)
y_pred_xgb=xgb.predict(x_test)
print(f' r2_score (xgboost)      : {r2_score(y_test,y_pred_xgb)}')
print(f' mean_absolute_error     :
{mean_absolute_error(y_test,y_pred_xgb)}')
print(f' mean_squared_error      :
{mean_squared_error(y_test,y_pred_xgb)}')
print(f' root_mean_squared_error :
{mean_squared_error(y_test,y_pred_xgb,squared=False)}')
```

```
 r2_score (xgboost)      : 0.8306180117472496
 mean_absolute_error     : 0.04737986420799072
 mean_squared_error      : 0.00565791617728751
 root_mean_squared_error : 0.07521912108824132
```

### 7.5.1 HYPERPARAMETER TUNING (XGBOOST)

```
xg_param_grid = {"gamma":[0,0.1,0.2,0.4],
             "learning_rate":[0.01,0.02,0.03,0.04,0.05,0.06,0.1],
             "max_depth":list(range(1,11)),
             "n_estimators":[50,65,80,100,150],
             "alpha":[0,0.1,0.5,1],
            }

XGB = XGBRegressor()
rcv = GridSearchCV(estimator=XGB, scoring="r2",
param_grid=xg_param_grid , cv=5, verbose=3,n_jobs=-1)

rcv.fit(x_train, y_train)
```

```
Fitting 5 folds for each of 5600 candidates, totalling 28000 fits
```

```
GridSearchCV(cv=5,
             estimator=XGBRegressor(base_score=None, booster=None,
                                    callbacks=None,
colsample_bylevel=None,
                                    colsample_bynode=None,
                                    colsample_bytree=None,
device=None,
                                    early_stopping_rounds=None,
                                    enable_categorical=False,
eval_metric=None,
                                    feature_types=None,
feature_weights=None,
                                    gamma=None, grow_policy=None,
```

```
                                    importance_type=None,
                                    interaction_constraints=None...
                                    min_child_weight=None,
missing=nan,
                                    monotone_constraints=None,
                                    multi_strategy=None,
n_estimators=None,
                                    n_jobs=None,
num_parallel_tree=None, ...),
             n_jobs=-1,
             param_grid={'alpha': [0, 0.1, 0.5, 1], 'gamma': [0, 0.1,
0.2, 0.4],
                         'learning_rate': [0.01, 0.02, 0.03, 0.04,
0.05, 0.06,
                                           0.1],
                         'max_depth': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'n_estimators': [50, 65, 80, 100, 150]},
             scoring='r2', verbose=3)

cv_best_params = rcv.best_params_
print(f"Best parameters: {cv_best_params}")

Best parameters: {'alpha': 0, 'gamma': 0, 'learning_rate': 0.06,
'max_depth': 3, 'n_estimators': 80}

xgb_model=XGBRegressor(alpha=0, gamma=0,learning_rate=0.06,
max_depth=3, n_estimators=80)
xgb_model.fit(x_train,y_train)
xg1_y_pred=xgb_model.predict(x_test)
xgb_reg_r2=(r2_score(y_test,xg1_y_pred))
print(xgb_reg_r2)

0.9190729318474143
```

## 7.6. K-NEAREST NEIGHBORS (KNN)

```
knn=KNeighborsRegressor(n_neighbors=5)
knn.fit(x_train,y_train)
y_pred_knn=knn.predict(x_test)
from sklearn.metrics import *
print(f' r2_score (knn)          : {r2_score(y_test,y_pred_knn)}')
print(f' mean_absolute_error     :
{mean_absolute_error(y_test,y_pred_knn)}')
print(f' mean_squared_error      :
{mean_squared_error(y_test,y_pred_knn)}')
print(f' root_mean_squared_error :
{mean_squared_error(y_test,y_pred_knn,squared=False)}')

 r2_score (knn)          : 0.8059815601669482
 mean_absolute_error     : 0.050012039880432374
```

```
mean_squared_error      : 0.006480854787142232
root_mean_squared_error : 0.08050375635423625
```

### 7.6.1. HYPERPARAMETER TUNING (KNN)

```
param_grid={'n_neighbors':[3,5,7,9,11],
            'weights':['uniform','distance'],
            'algorithm':['auto','ball_tree','kd_tree','brute'],
            'p':[1,2]}
knn1=RandomizedSearchCV(estimator=KNeighborsRegressor(),param_distribu
tions=param_grid,scoring='r2',n_jobs=-1,verbose=3,cv=3,n_iter=3000)
knn1.fit(x_train,y_train)
```

```
Fitting 3 folds for each of 80 candidates, totalling 240 fits
```

```
RandomizedSearchCV(cv=3, estimator=KNeighborsRegressor(), n_iter=3000,
                   n_jobs=-1,
                   param_distributions={'algorithm': ['auto',
'ball_tree',
                                                      'kd_tree',
'brute'],
                                        'n_neighbors': [3, 5, 7, 9,
11],
                                        'p': [1, 2],
                                        'weights': ['uniform',
'distance']},
                   scoring='r2', verbose=3)
```

```
best_params=knn1.best_params_
print(f'best_parameters:{best_params}')
```

```
best_parameters:{'weights': 'distance', 'p': 1, 'n_neighbors': 7,
'algorithm': 'auto'}
```

```
knn_model=KNeighborsRegressor(weights='distance', p= 1, n_neighbors=7,
algorithm='auto')
knn_model.fit(x_train,y_train)
y_pred=knn_model.predict(x_test)
knn_reg_r2=r2_score(y_test,y_pred)
print(knn_reg_r2)
```

```
0.8414907733117032
```

## 7.7. ARTIFICIAL NEURAL NETWORK (ANN)

```
model_ann=MLPRegressor()
model_ann.fit(x_train,y_train)
y_pred=model_ann.predict(x_test)

MLP_reg_r2=r2_score(y_test,y_pred)
print('The R2_score is ',MLP_reg_r2)
```

```
The R2_score is  0.8244277403626056
```

### 7.7.1. HYPERPARAMETER TUNING (ANN)

```
params_ann={'learning_rate_init':
[0.001,0.002,0.005,0.01,0.1,0.2,0.5,1],
        'max_iter':[100,200,300,400,500,1000,2000,3000]
        }

ann_cv=GridSearchCV(estimator=model_ann,param_grid=params_ann,scoring=
'r2',n_jobs=-1,cv=5,verbose=3)
ann_cv.fit(x_train,y_train)

Fitting 5 folds for each of 64 candidates, totalling 320 fits

GridSearchCV(cv=5, estimator=MLPRegressor(), n_jobs=-1,
             param_grid={'learning_rate_init': [0.001, 0.002, 0.005,
0.01, 0.1,
                                                0.2, 0.5, 1],
                         'max_iter': [100, 200, 300, 400, 500, 1000,
2000,
                                      3000]},
             scoring='r2', verbose=3)

ann_best_params=ann_cv.best_params_
print(f'best_parameters:{ann_best_params}')

best_parameters:{'learning_rate_init': 0.005, 'max_iter': 300}

ann_model=MLPRegressor(learning_rate_init=0.01, max_iter=3000)
ann_model.fit(x_train,y_train)

MLPRegressor(learning_rate_init=0.01, max_iter=3000)

ann_y_pred=ann_model.predict(x_test)
ann_reg_r2=r2_score(y_test,ann_y_pred)
print(ann_reg_r2)

0.8706969693170691
```

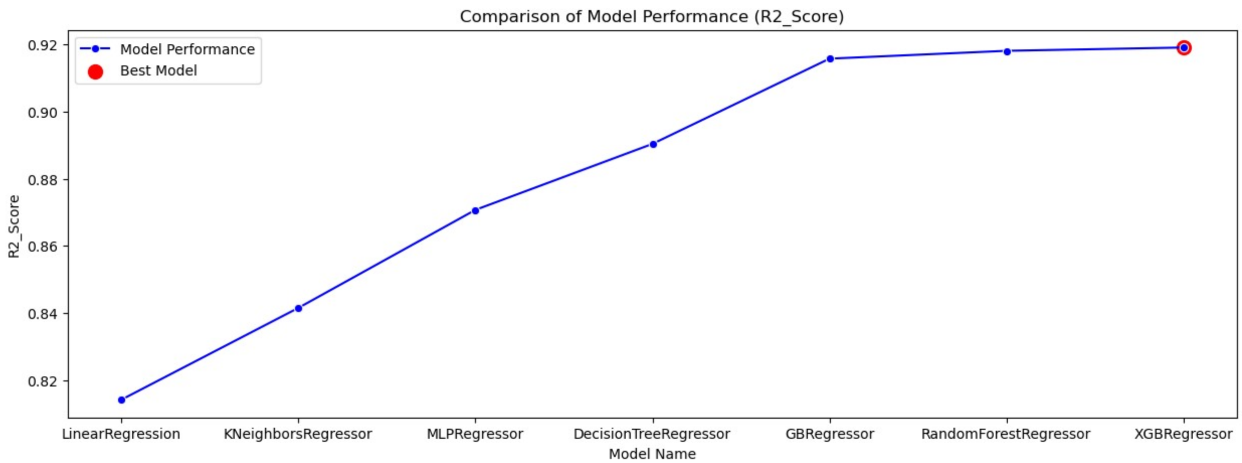
# 8. MODEL COMPARISON REPORT

```
comparison_dict = {
    'Model': ['LinearRegression', 'DecisionTreeRegressor',
'RandomForestRegressor', 'GBRegressor',
'XGBRegressor','KNeighborsRegressor', 'MLPRegressor'],
    'R2_score': [linear_reg_r2, decisiontree_reg_r2,
randomforest_reg_r2, gradboost_reg_r2, xgb_reg_r2,knn_reg_r2,
ann_reg_r2]
}
```

```python
# Creating DataFrame
comparison_df = pd.DataFrame(comparison_dict)
print(comparison_df)
```

```
                   Model  R2_score
0       LinearRegression  0.814131
1  DecisionTreeRegressor  0.890359
2  RandomForestRegressor  0.918090
3            GBRegressor  0.915755
4           XGBRegressor  0.919073
5    KNeighborsRegressor  0.841491
6           MLPRegressor  0.870697
```

```python
comparison_df = comparison_df.sort_values(by='R2_score',
ascending=True)


max_index = comparison_df['R2_score'].idxmax()  # This returns the row
index having highest r2_score


plt.figure(figsize=(15, 5))
sns.lineplot(data=comparison_df, x='Model', y='R2_score', marker='o',
color='blue', label="Model Performance")


plt.scatter(comparison_df.loc[max_index, 'Model'],
comparison_df.loc[max_index, 'R2_score'],color='red', s=100,
label="Best Model")


plt.xlabel("Model Name")
plt.ylabel("R2_Score")

plt.legend()
plt.title("Comparison of Model Performance (R2_Score)")
plt.show()
```




# 9. CONCLUSION

- From the above lineplot, models such as GBRegressor, XGBRegressor and RandomForestRegressor almost gives similar r2_score which works well with this regression problem compared to other models.
- Among all the models, 'XGBRegressor' has the highest predictive performance. Therefore, we can infer that 'XGBRegressor' will be the most suitable model for the given dataset.