In [1]:

```
#import all necessary Libraries

import pandas as pd
import numpy as np
import sklearn as sl
from sklearn.decomposition import PCA
```

In [2]:

```
pip install xgboost
```

```
Requirement already satisfied: xgboost in c:\programdata\ssh\lib\site-pack
ages (1.2.0)Note: you may need to restart the kernel to use updated packag
es.
Requirement already satisfied: numpy in c:\programdata\ssh\lib\site-packag
es (from xgboost) (1.18.5)
Requirement already satisfied: scipy in c:\programdata\ssh\lib\site-packag
es (from xgboost) (1.5.0)
```

In [3]:

```
import xgboost as xgb
```

In [4]:

```
#reading the datasets and explore it
#train dataset

df_train=pd.read_csv("train.csv")
df_train.describe()
```

Out[4]:

	ID	y	X10	X11	X12	X13	X1
count	4209.000000	4209.000000	4209.000000	4209.0	4209.000000	4209.000000	4209.00000(
mean	4205.960798	100.669318	0.013305	0.0	0.075077	0.057971	0.42813(
std	2437.608688	12.679381	0.114590	0.0	0.263547	0.233716	0.49486
min	0.000000	72.110000	0.000000	0.0	0.000000	0.000000	0.00000(
25%	2095.000000	90.820000	0.000000	0.0	0.000000	0.000000	0.00000(
50%	4220.000000	99.150000	0.000000	0.0	0.000000	0.000000	0.00000(
75%	6314.000000	109.010000	0.000000	0.0	0.000000	0.000000	1.00000(
max	8417.000000	265.320000	1.000000	0.0	1.000000	1.000000	1.00000(

8 rows × 370 columns

In [5]:

```
df_train.describe
```

Out[5]:

```
<bound method NDFrame.describe of         ID       y  X0 X1  X2 X3 X4  X5
X6 X8  ...  X375  X376  X377  X378  \
0         0  130.81   k  v  at  a  d   u  j  o  ...     0     0     1     0
1         6   88.53   k  t  av  e  d   y  l  o  ...     1     0     0     0
2         7   76.26  az  w   n  c  d   x  j  x  ...     0     0     0     0
3         9   80.62  az  t   n  f  d   x  l  e  ...     0     0     0     0
4        13   78.02  az  v   n  f  d   h  d  n  ...     0     0     0     0
...     ...     ...  .. ..  .. .. ..  .. .. ..  ...   ...   ...   ...   ...
4204  8405  107.39  ak  s  as  c  d  aa  d  q  ...     1     0     0     0
4205  8406  108.77   j  o   t  d  d  aa  h  h  ...     0     1     0     0
4206  8412  109.22  ak  v   r  a  d  aa  g  e  ...     0     0     1     0
4207  8415   87.48  al  r   e  f  d  aa  l  u  ...     0     0     0     0
4208  8417  110.85   z  r  ae  c  d  aa  g  w  ...     1     0     0     0

      X379  X380  X382  X383  X384  X385
0        0     0     0     0     0     0
1        0     0     0     0     0     0
2        0     0     1     0     0     0
3        0     0     0     0     0     0
4        0     0     0     0     0     0
...    ...   ...   ...   ...   ...   ...
4204     0     0     0     0     0     0
4205     0     0     0     0     0     0
4206     0     0     0     0     0     0
4207     0     0     0     0     0     0
4208     0     0     0     0     0     0

[4209 rows x 378 columns]>
```

In [6]:

```
df_train.dtypes
```

Out[6]:

```
ID        int64
y       float64
X0       object
X1       object
X2       object
         ...
X380      int64
X382      int64
X383      int64
X384      int64
X385      int64
Length: 378, dtype: object
```

In [7]:

```
print (df_train.shape)
```

```
(4209, 378)
```

In [8]:

```
#checking for the null value
df_train.isnull().sum()
```

Out[8]:

```
ID      0
y       0
X0      0
X1      0
X2      0
       ..
X380    0
X382    0
X383    0
X384    0
X385    0
Length: 378, dtype: int64
```

In [9]:

```
#test set exploring
df_test=pd.read_csv("test.csv")
print (df_test.head)
```

```
<bound method NDFrame.head of         ID  X0  X1  X2 X3 X4  X5 X6 X8  X1
0  ...  X375  X376  X377  X378  \
0         1  az   v   n  f  d   t  a  w    0  ...     0     0     0     1
1         2   t   b  ai  a  d   b  g  y    0  ...     0     0     1     0
2         3  az   v  as  f  d   a  j  j    0  ...     0     0     0     1
3         4  az   l   n  f  d   z  l  n    0  ...     0     0     0     1
4         5   w   s  as  c  d   y  i  m    0  ...     1     0     0     0
...     ...  ..  ..  .. .. ..  .. .. ..  ...  ...   ...   ...   ...   ...
4204  8410  aj   h  as  f  d  aa  j  e    0  ...     0     0     0     0
4205  8411   t  aa  ai  d  d  aa  j  y    0  ...     0     1     0     0
4206  8413   y   v  as  f  d  aa  d  w    0  ...     0     0     0     0
4207  8414  ak   v  as  a  d  aa  c  q    0  ...     0     0     1     0
4208  8416   t  aa  ai  c  d  aa  g  r    0  ...     1     0     0     0

      X379  X380  X382  X383  X384  X385
0        0     0     0     0     0     0
1        0     0     0     0     0     0
2        0     0     0     0     0     0
3        0     0     0     0     0     0
4        0     0     0     0     0     0
...    ...   ...   ...   ...   ...   ...
4204     0     0     0     0     0     0
4205     0     0     0     0     0     0
4206     0     0     0     0     0     0
4207     0     0     0     0     0     0
4208     0     0     0     0     0     0

[4209 rows x 377 columns]>
```

In [10]:

```
df_test.dtypes
```

Out[10]:

```
ID       int64
X0      object
X1      object
X2      object
X3      object
         ...
X380     int64
X382     int64
X383     int64
X384     int64
X385     int64
Length: 377, dtype: object
```

In [11]:

```
#checking for any null values
df_test.isnull().sum()
```

Out[11]:

```
ID      0
X0      0
X1      0
X2      0
X3      0
       ..
X380    0
X382    0
X383    0
X384    0
X385    0
Length: 377, dtype: int64
```

In [12]:

```
# merging the two datasets of test and train for easy outputs

df_mercedes=pd.concat([df_train,df_test])
df_mercedes.describe()
```

Out[12]:

	ID	y	X10	X11	X12	X13	
count	8418.000000	4209.000000	8418.000000	8418.000000	8418.000000	8418.000000	8418.(
mean	4208.500000	100.669318	0.016156	0.000119	0.074721	0.059515	0.
std	2430.211616	12.679381	0.126082	0.010899	0.262956	0.236601	0.
min	0.000000	72.110000	0.000000	0.000000	0.000000	0.000000	0.(
25%	2104.250000	90.820000	0.000000	0.000000	0.000000	0.000000	0.(
50%	4208.500000	99.150000	0.000000	0.000000	0.000000	0.000000	0.(
75%	6312.750000	109.010000	0.000000	0.000000	0.000000	0.000000	1.(
max	8417.000000	265.320000	1.000000	1.000000	1.000000	1.000000	1.(

8 rows × 370 columns

In [13]:

```
#Question2= checking the null values
df_mercedes.isnull().sum()
```

Out[13]:

```
ID         0
y       4209
X0         0
X1         0
X2         0
        ...
X380       0
X382       0
X383       0
X384       0
X385       0
Length: 378, dtype: int64
```

In [14]:

```
#splitting the dataset into X and y variable
from sklearn.model_selection import train_test_split
X=df_mercedes
y=df_mercedes
```

In [15]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=21)
```

In [16]:

```
print(X_train.shape)
print (X_test.shape)
print (y_train.shape)
print(y_test.shape)
```

```
(6734, 378)
(1684, 378)
(6734, 378)
(1684, 378)
```

In [17]:

```
#checking the unique values
Counts= df_mercedes.nunique()
```

In [18]:

```
print(Counts)
```

```
ID      8418
y       2545
X0        53
X1        27
X2        50
        ...
X380       2
X382       2
X383       2
X384       2
X385       2
Length: 378, dtype: int64
```

In [19]:

```
df_mercedes.dtypes
```

Out[19]:

```
ID        int64
y       float64
X0       object
X1       object
X2       object
         ...
X380      int64
X382      int64
X383      int64
X384      int64
X385      int64
Length: 378, dtype: object
```

In [20]:

```
#labe encodcer

from sklearn.preprocessing import LabelEncoder
```

In [21]:

```
le=LabelEncoder()
```

In [31]:

```
df_mercedes['y']=le.fit_transform(df_mercedes['y'])
```

In [28]:

```
df_mercedes.dtypes
```

Out[28]:

```
ID       int64
y        int64
X0      object
X1      object
X2      object
         ...
X380     int64
X382     int64
X383     int64
X384     int64
X385     int64
Length: 378, dtype: object
```

In [32]:

```
df_mercedes.head()
```

Out[32]:

	ID	y	X0	X1	X2	X3	X4	X5	X6	X8	...	X375	X376	X377	X378	X379	X380	X
0	0	2466	k	v	at	a	d	u	j	o	...	0	0	1	0	0	0	
1	6	366	k	t	av	e	d	y	l	o	...	1	0	0	0	0	0	
2	7	69	az	w	n	c	d	x	j	x	...	0	0	0	0	0	0	
3	9	133	az	t	n	f	d	x	l	e	...	0	0	0	0	0	0	
4	13	106	az	v	n	f	d	h	d	n	...	0	0	0	0	0	0	

5 rows × 378 columns

In [39]:

```
df_mercedes.apply(LabelEncoder().fit_transform)
```

Out[39]:

	ID	y	X0	X1	X2	X3	X4	X5	X6	X8	...	X375	X376	X377	X378	X379	X3
0	0	2466	37	23	20	0	3	27	9	14	...	0	0	1	0	0	
1	6	366	37	21	22	4	3	31	11	14	...	1	0	0	0	0	
2	7	69	24	24	38	2	3	30	9	23	...	0	0	0	0	0	
3	9	133	24	21	38	5	3	30	11	4	...	0	0	0	0	0	
4	13	106	24	23	38	5	3	14	3	13	...	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4204	8410	3951	9	9	19	5	3	1	9	4	...	0	0	0	0	0	
4205	8411	3952	46	1	9	3	3	1	9	24	...	0	1	0	0	0	
4206	8413	3953	51	23	19	5	3	1	3	22	...	0	0	0	0	0	
4207	8414	5340	10	23	19	0	3	1	2	16	...	0	0	1	0	0	
4208	8416	6753	46	1	9	2	3	1	6	17	...	1	0	0	0	0	

8418 rows × 378 columns

In [58]:

```
#dimentionality reduction
#pca library already imported
```

In [59]:

```
#prediction with xgboost

df_test=pd.read_csv("test.csv")
```

In [62]:

```
print (df_test.keys())
```

```
Index(['ID', 'X0', 'X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X8', 'X10',
       ...
       'X375', 'X376', 'X377', 'X378', 'X379', 'X380', 'X382', 'X383', 'X3
84',
       'X385'],
      dtype='object', length=377)
```

In [63]:

```
from sklearn.model_selection import train_test_split
```

In [83]:

```
X_train,X_test,y_train,y_test= train_test_split(X,y,test_size=0.2,random_state=21)
print (X_train.shape)
print(X_test.shape)
print (y_train.shape)
print(y_test.shape)
```

```
(6734, 378)
(1684, 378)
(6734, 378)
(1684, 378)
```

In [92]:

```
xg_reg=xgb.XGBRegressor(objective='reg:linear',colsample_bytree=0.3,learning_rate=0.1,m
ax_depth=5,alpha=10,n_estimator=10)
```

In [85]:

```
from sklearn.linear_model import LinearRegression
model_lr=LinearRegression()
```

In [ ]:

```

```

In [ ]:

```

```