In [1]:

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

In [2]:

```
train_data=pd.read_csv("train.csv")
test_data=pd.read_csv("test.csv")
```

In [3]:

```
train_data.shape
```

Out[3]:

```
(4209, 378)
```

In [4]:

```
test_data.shape
```

Out[4]:

```
(4209, 377)
```

In [5]:

```
train_data
```

Out[5]:

| | ID | y | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 130.81 | k | v | at | a | d | u | j | o | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 6 | 88.53 | k | t | av | e | d | y | l | o | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 7 | 76.26 | az | w | n | c | d | x | j | x | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3** | 9 | 80.62 | az | t | n | f | d | x | l | e | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 13 | 78.02 | az | v | n | f | d | h | d | n | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4204** | 8405 | 107.39 | ak | s | as | c | d | aa | d | q | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4205** | 8406 | 108.77 | j | o | t | d | d | aa | h | h | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4206** | 8412 | 109.22 | ak | v | r | a | d | aa | g | e | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4207** | 8415 | 87.48 | al | r | e | f | d | aa | l | u | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4208** | 8417 | 110.85 | z | r | ae | c | d | aa | g | w | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

4209 rows × 378 columns

In [6]:

```
train_data.describe()
```

Out[6]:

| | ID | y | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | ... | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 4209.000000 | 4209.000000 | 4209.000000 | 4209.0 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | ... | 4209.( |
| **mean** | 4205.960798 | 100.669318 | 0.013305 | 0.0 | 0.075077 | 0.057971 | 0.428130 | 0.000475 | 0.002613 | 0.007603 | ... | 0.: |
| **std** | 2437.608688 | 12.679381 | 0.114590 | 0.0 | 0.263547 | 0.233716 | 0.494867 | 0.021796 | 0.051061 | 0.086872 | ... | 0.4 |
| **min** | 0.000000 | 72.110000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.( |
| **25%** | 2095.000000 | 90.820000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.( |
| **50%** | 4220.000000 | 99.150000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 0.( |
| **75%** | 6314.000000 | 109.010000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | ... | 1.( |
| **max** | 8417.000000 | 265.320000 | 1.000000 | 0.0 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | ... | 1.( |

8 rows × 370 columns

In [7]:

```
variance=pow(train_data.drop(columns={'ID','y'}).std(),2).to_dict()
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_9828\3076446643.py:1: FutureWarning: Dropping of nuisance columns in Dat
aFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError.  Select
only valid columns before calling the reduction.
  variance=pow(train_data.drop(columns={'ID','y'}).std(),2).to_dict()
```

In [8]:

```
variance
```

Out[8]:

```
{'X10': 0.013130924408766528,
 'X11': 0.0,
 'X12': 0.06945712925498301,
 'X13': 0.05462335372237543,
 'X14': 0.24489291460358809,
 'X15': 0.0004750593287785332,
 'X16': 0.002607236710760207,
 'X17': 0.007546747102668128,
 'X18': 0.007780719750453295,
 'X19': 0.08965996727995626,
 'X20': 0.12242957558806764,
 'X21': 0.0026072367107602044,
 'X22': 0.07941395271946106,
 'X23': 0.020247554805158056,
 'X24': 0.00189752720722469,
 'X26': 0.004965595180344528,
 'X27': 0.21671422906013918,
 'X28': 0.03149732557519826,
```

In [9]:

```
for key,value in variance.items():
    if(value==0):
        print('Name=',key)
```

```
Name= X11
Name= X93
Name= X107
Name= X233
Name= X235
Name= X268
Name= X289
Name= X290
Name= X293
Name= X297
Name= X330
Name= X347
```

In [10]:

```
# drop these columns
train_data=train_data.drop(columns={'X11','X93','X107','X233','X235','X268','X289','X290','X293','X297','X330','X347'})
```

In [11]:

```
train_data.shape
```

Out[11]:

```
(4209, 366)
```

In [12]:

```
# Create independent & dependent variable
train_data_feature=train_data.drop(columns={'y','ID'})
train_data_target=train_data.y
```

In [13]:

```
train_data_feature.shape
```

Out[13]:

```
(4209, 364)
```

In [14]:

```
train_data_target.shape
```

Out[14]:

```
(4209,)
```

## Applying Label Encoder

In [15]:

```
train_data_feature.describe(include="object")
```

Out[15]:

| | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 |
|---|---|---|---|---|---|---|---|---|
| **count** | 4209 | 4209 | 4209 | 4209 | 4209 | 4209 | 4209 | 4209 |
| **unique** | 47 | 27 | 44 | 7 | 4 | 29 | 12 | 25 |
| **top** | z | aa | as | c | d | w | g | j |
| **freq** | 360 | 833 | 1659 | 1942 | 4205 | 231 | 1042 | 277 |

In [16]:

```
train_data_feature
```

Out[16]:

| | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | X12 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | k | v | at | a | d | u | j | o | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | k | t | av | e | d | y | l | o | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | az | w | n | c | d | x | j | x | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3** | az | t | n | f | d | x | l | e | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | az | v | n | f | d | h | d | n | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4204** | ak | s | as | c | d | aa | d | q | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4205** | j | o | t | d | d | aa | h | h | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4206** | ak | v | r | a | d | aa | g | e | 0 | 1 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4207** | al | r | e | f | d | aa | l | u | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4208** | z | r | ae | c | d | aa | g | w | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

4209 rows × 364 columns

In [17]:

```
train_data_target
```

Out[17]:

```
0       130.81
1        88.53
2        76.26
3        80.62
4        78.02
         ...
4204    107.39
4205    108.77
4206    109.22
4207     87.48
4208    110.85
Name: y, Length: 4209, dtype: float64
```

In [18]:

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

In [19]:

```
for i in train_data_feature.columns:
    data_type=train_data_feature[i].dtype
    if data_type=='object':
        train_data_feature[i]=le.fit_transform(train_data_feature[i])
```

In [20]:

```
train_data_feature
```

Out[20]:

| | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | X12 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 32 | 23 | 17 | 0 | 3 | 24 | 9 | 14 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 32 | 21 | 19 | 4 | 3 | 28 | 11 | 14 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 20 | 24 | 34 | 2 | 3 | 27 | 9 | 23 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3** | 20 | 21 | 34 | 5 | 3 | 27 | 11 | 4 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 20 | 23 | 34 | 5 | 3 | 12 | 3 | 13 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4204** | 8 | 20 | 16 | 2 | 3 | 0 | 3 | 16 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4205** | 31 | 16 | 40 | 3 | 3 | 0 | 7 | 7 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4206** | 8 | 23 | 38 | 0 | 3 | 0 | 6 | 4 | 0 | 1 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4207** | 9 | 19 | 25 | 5 | 3 | 0 | 11 | 20 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4208** | 46 | 19 | 3 | 2 | 3 | 0 | 6 | 22 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

4209 rows × 364 columns

## Performing Dimensionality reduction

In [21]:

```
from sklearn.decomposition import PCA
```

In [22]:

```
pca=PCA(n_components=0.95)
train_data_feature_trans=pca.fit_transform(train_data_feature)
```

In [23]:

```
train_data_feature_trans.shape
```

Out[23]:

```
(4209, 6)
```

In [24]:

```
# Split the dataset into train set & test set
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(train_data_feature_trans,train_data_target,test_size=.3,random_state=42)
```

In [25]:

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

```
(2946, 6)
(1263, 6)
(2946,)
(1263,)
```

## Model building

In [26]:

```
pip install xgboost
```

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

  Downloading xgboost-1.7.2-py3-none-win_amd64.whl (89.1 MB)
     ---------------------------------------- 89.1/89.1 MB 3.0 MB/s eta 0:00:00
Requirement already satisfied: scipy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.9.1)
Requirement already satisfied: numpy in c:\users\dell\anaconda3\lib\site-packages (from xgboost) (1.21.5)
Installing collected packages: xgboost
Successfully installed xgboost-1.7.2
```

In [27]:

```
import xgboost as xgb
```

In [28]:

```
# Train the model
xgb_reg=xgb.XGBRegressor()
model=xgb_reg.fit(X_train,y_train)
```

In [29]:

```
# predict the test data
y_pred=model.predict(X_test)
```

In [30]:

```
#Evaluate the model performance
from sklearn.metrics import mean_squared_error


```

In [31]:

```
print('RMSE=',np.sqrt(mean_squared_error(y_pred,y_test)))
```

```
RMSE= 11.813608278556764
```

In [32]:

```
#save the model
import joblib
```

In [33]:

```
joblib.dump(model,'xgbmodel.pkl')
```

Out[33]:

```
['xgbmodel.pkl']
```

In [36]:

```
#Load the model
loaded_model=joblib.load('xgbmodel.pkl')
print('model loaded successfully')
```

```
model loaded successfully
```

In [37]:

```
# Prediction on test data
test_data=test_data.drop(columns={'X11','X93','X107','X233','X235','X268','X289','X290','X293','X297','X330','X347'})
```

In [38]:

```
test_data.shape
```

Out[38]:

```
(4209, 365)
```

In [39]:

```
test_data
```

Out[39]:

| | ID | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | az | v | n | f | d | t | a | w | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 2 | t | b | ai | a | d | b | g | y | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 3 | az | v | as | f | d | a | j | j | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 4 | az | l | n | f | d | z | l | n | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 5 | w | s | as | c | d | y | i | m | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4204** | 8410 | aj | h | as | f | d | aa | j | e | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4205** | 8411 | t | aa | ai | d | d | aa | j | y | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4206** | 8413 | y | v | as | f | d | aa | d | w | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4207** | 8414 | ak | v | as | a | d | aa | c | q | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4208** | 8416 | t | aa | ai | c | d | aa | g | r | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

4209 rows × 365 columns

In [41]:

```
test_data.isna().sum().any()
```

Out[41]:

```
False
```

In [42]:

```
test_data.columns
```

Out[42]:

```
Index(['ID', 'X0', 'X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X8', 'X10',
       ...
       'X375', 'X376', 'X377', 'X378', 'X379', 'X380', 'X382', 'X383', 'X384',
       'X385'],
      dtype='object', length=365)
```

In [43]:

```
#create features
test_data_feature=test_data.drop(columns={'ID'})
```

In [44]:

```
test_data_feature.shape
```

Out[44]:

```
(4209, 364)
```

In [46]:

```
for i in test_data_feature.columns:
    data_type=test_data_feature[i].dtype
    if data_type=='object':
        test_data_feature[i]=le.fit_transform(test_data_feature[i])
```

In [47]:

```
test_data_feature
```

Out[47]:

| | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | X12 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 21 | 23 | 34 | 5 | 3 | 26 | 0 | 22 | 0 | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 42 | 3 | 8 | 0 | 3 | 9 | 6 | 24 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 21 | 23 | 17 | 5 | 3 | 0 | 9 | 9 | 0 | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 21 | 13 | 34 | 5 | 3 | 31 | 11 | 13 | 0 | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 45 | 20 | 17 | 2 | 3 | 30 | 8 | 12 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **4204** | 6 | 9 | 17 | 5 | 3 | 1 | 9 | 4 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4205** | 42 | 1 | 8 | 3 | 3 | 1 | 9 | 24 | 0 | 0 | ... | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4206** | 47 | 23 | 17 | 5 | 3 | 1 | 3 | 22 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4207** | 7 | 23 | 17 | 0 | 3 | 1 | 2 | 16 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4208** | 42 | 1 | 8 | 2 | 3 | 1 | 6 | 17 | 0 | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

4209 rows × 364 columns

In [48]:

```
pca.fit(test_data_feature)
```

Out[48]:

```
PCA(n_components=0.95)
```

In [49]:

```
test_data_feature_trans=pca.fit_transform(test_data_feature)
```

In [50]:

```
test_data_feature_trans.shape
```

Out[50]:

```
(4209, 6)
```

In [51]:

```
# prediction on test data with loaded model
test_pred=loaded_model.predict(test_data_feature_trans)
```

In [52]:

```
test_pred
```

Out[52]:

```
array([ 73.558525,  93.334915,  82.57792 , ..., 102.88445 , 105.04327 ,
        91.70111 ], dtype=float32)
```

In [ ]: