# ML Project:-> Mercedes-Benz Greener Manufacturing

## DESCRIPTION

Reduce the time a Mercedes-Benz spends on the test bench.

Problem Statement Scenario: Since the first automobile, the Benz Patent Motor Car in 1886, Mercedes-Benz has stood for important automotive innovations. These include the passenger safety cell with a crumple zone, the airbag, and intelligent assistance systems. Mercedes-Benz applies for nearly 2000 patents per year, making the brand the European leader among premium carmakers. Mercedes-Benz is the leader in the premium car industry. With a huge selection of features and options, customers can choose the customized Mercedes-Benz of their dreams.

To ensure the safety and reliability of every unique car configuration before they hit the road, the company's engineers have developed a robust testing system. As one of the world's biggest manufacturers of premium cars, safety and efficiency are paramount on Mercedes-Benz's production lines. However, optimizing the speed of their testing system for many possible feature combinations is complex and time-consuming without a powerful algorithmic approach.

You are required to reduce the time that cars spend on the test bench. Others will work with a dataset representing different permutations of features in a Mercedes-Benz car to predict the time it takes to pass testing. Optimal algorithms will contribute to faster testing, resulting in lower carbon dioxide emissions without reducing Mercedes-Benz's standards.

Following actions should be performed:

1. If for any column(s), the variance is equal to zero, then you need to remove those variable(s).
2. Check for null and unique values for test and train sets.
3. Apply label encoder.
4. Perform dimensionality reduction.
5. Predict your test_df values using XGBoost.

In [1]:

```
#Importing library

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
pd.set_option("display.max_columns", None)
import warnings;
warnings.simplefilter('ignore')
from xgboost import XGBRegressor

```

In [2]:

```
# Load train dataset here

df_train = pd.read_csv(r'C:\Users\Pavan Lande\Downloads\train\train.csv')
df_train.head()

```

Out[2]:

| | ID | y | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X: |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 130.81 | k | v | at | a | d | u | j | o | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 6 | 88.53 | k | t | av | e | d | y | l | o | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | |
| **2** | 7 | 76.26 | az | w | n | c | d | x | j | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |
| **3** | 9 | 80.62 | az | t | n | f | d | x | l | e | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |
| **4** | 13 | 78.02 | az | v | n | f | d | h | d | n | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | |

In [3]:

```
df_train.shape
```

Out[3]:

```
(4209, 378)
```

In [4]:

```
df_train.dtypes
```

Out[4]:

```
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 [5]:

```
# Check for null in training data set

df_train.isnull().sum()
```

Out[5]:

```
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 [6]:

```
df_train.describe()
```

Out[6]:

| | ID | y | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 4209.000000 | 4209.000000 | 4209.000000 | 4209.0 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.0 |
| **mean** | 4205.960798 | 100.669318 | 0.013305 | 0.0 | 0.075077 | 0.057971 | 0.428130 | 0.000475 | 0.002613 | 0.007603 | 0.007840 | 0.0 |
| **std** | 2437.608688 | 12.679381 | 0.114590 | 0.0 | 0.263547 | 0.233716 | 0.494867 | 0.021796 | 0.051061 | 0.086872 | 0.088208 | 0.2 |
| **min** | 0.000000 | 72.110000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 |
| **25%** | 2095.000000 | 90.820000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 |
| **50%** | 4220.000000 | 99.150000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 |
| **75%** | 6314.000000 | 109.010000 | 0.000000 | 0.0 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.0 |
| **max** | 8417.000000 | 265.320000 | 1.000000 | 0.0 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.0 |

In [7]:

```
df_train = df_train.drop('ID', axis=1)
```

In [8]:

```
df_train.head()
```

Out[8]:

| | y | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 130.81 | k | v | at | a | d | u | j | o | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 88.53 | k | t | av | e | d | y | l | o | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **2** | 76.26 | az | w | n | c | d | x | j | x | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| **3** | 80.62 | az | t | n | f | d | x | l | e | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| **4** | 78.02 | az | v | n | f | d | h | d | n | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

In [9]:

```
df_train.shape
```

Out[9]:

```
(4209, 377)
```

In [10]:

```
# Seperate the numerical and categorical columns for train data
df_cat = df_train.select_dtypes(include = np.object)
df_num = df_train.select_dtypes(exclude=np.object)
```

In [11]:

```
# categorical data for train dataset
df_cat.head()
```

Out[11]:

| | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 |
|---|---|---|---|---|---|---|---|---|
| **0** | k | v | at | a | d | u | j | o |
| **1** | k | t | av | e | d | y | l | o |
| **2** | az | w | n | c | d | x | j | x |
| **3** | az | t | n | f | d | x | l | e |
| **4** | az | v | n | f | d | h | d | n |

In [12]:

```
# Numerical data for train dataset
df_num.head()
```

Out[12]:

| | y | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 130.81 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| **1** | 88.53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| **2** | 76.26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| **3** | 80.62 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |
| **4** | 78.02 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | |

In [13]:

```
# drop dependent variable from numerical data of train set
df_num = df_num.drop("y", axis = 1)
df_num.head()
```

Out[13]:

| | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

In [14]:

```
columns = df_num.columns
```

In [15]:

```
df_num.shape
```

Out[15]:

```
(4209, 368)
```

In [16]:

```
# Applying scaling technique for numerical data of train set
from sklearn.preprocessing import MinMaxScaler, StandardScaler
mn = MinMaxScaler()
```

In [17]:

```
df_mn = mn.fit_transform(df_num)
```

In [18]:

```
df_num_sc = pd.DataFrame(df_mn, index=df_num.index, columns=df_num.columns)
df_num_sc.head()
```

Out[18]:

| | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **1** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **2** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **3** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **4** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |

## TASK NO:- 01 If for any column(s), the variance is equal to zero, then you need to remove those variable(s).

In [19]:

```
# variance of numerical data of train set
variance_df_num = df_num.var()
```

## finding out the variance which are of zero in training set

In [20]:

```
variable_var_zero = [ ]

for i in range(0,len(variance_df_num)):
    if variance_df_num[i]==0: #checking if the variance for the df_num dataframe column has zero
        variable_var_zero.append(columns[i])
```

In [21]:

```
np.ravel(variable_var_zero)
```

Out[21]:

```
array(['X11', 'X93', 'X107', 'X233', 'X235', 'X268', 'X289', 'X290',
       'X293', 'X297', 'X330', 'X347'], dtype='<U4')
```

In [22]:

```
# features which are of Zero variance in training data set will be dropped
df_num_variance_with_zero_drop = df_num.drop(['X11', 'X93', 'X107', 'X233', 'X235', 'X268', 'X289', 'X290',
       'X293', 'X297', 'X330', 'X347'], axis = 1)
```

In [23]:

```
df_num_variance_with_zero_drop.head()
```

Out[23]:

| | X10 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 | X39 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

In [24]:

```
df_num_variance_with_zero_drop.describe()
```

Out[24]:

| | X10 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 | 4209.000000 |
| **mean** | 0.013305 | 0.075077 | 0.057971 | 0.428130 | 0.000475 | 0.002613 | 0.007603 | 0.007840 | 0.099549 | 0.142789 | 0.002613 |
| **std** | 0.114590 | 0.263547 | 0.233716 | 0.494867 | 0.021796 | 0.051061 | 0.086872 | 0.088208 | 0.299433 | 0.349899 | 0.051061 |
| **min** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **max** | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |

In [25]:

```
df_num_variance_with_zero_drop.shape
```

Out[25]:

```
(4209, 356)
```

In [26]:

```
df_train.shape
```

Out[26]:

```
(4209, 377)
```

In [27]:

```
df_cat.shape
```

Out[27]:

```
(4209, 8)
```

## TASK NO 02 :-> Check unique values for train sets.

In [29]:

```
# finding number of unique values in each feature
df_train.nunique()
```

Out[29]:

```
y       2545
X0        47
X1        27
X2        44
X3         7
        ...
X380       2
X382       2
X383       2
X384       2
X385       2
Length: 377, dtype: int64
```

In [30]:

```
# returns the unique values in the training data set
train_feature_names = df_train[['y', 'X0', 'X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X8', 'X10', 'X11',
       'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20',
       'X21', 'X22', 'X23', 'X24', 'X26', 'X27', 'X28', 'X29', 'X30',
       'X31', 'X32', 'X33', 'X34', 'X35', 'X36', 'X37', 'X38', 'X39',
       'X40', 'X41', 'X42', 'X43', 'X44', 'X45', 'X46', 'X47', 'X48',
       'X49', 'X50', 'X51', 'X52', 'X53', 'X54', 'X55', 'X56', 'X57',
       'X58', 'X59', 'X60', 'X61', 'X62', 'X63', 'X64', 'X65', 'X66',
       'X67', 'X68', 'X69', 'X70', 'X71', 'X73', 'X74', 'X75', 'X76',
       'X77', 'X78', 'X79', 'X80', 'X81', 'X82', 'X83', 'X84', 'X85',
       'X86', 'X87', 'X88', 'X89', 'X90', 'X91', 'X92', 'X93', 'X94',
       'X95', 'X96', 'X97', 'X98', 'X99', 'X100', 'X101', 'X102', 'X103',
       'X104', 'X105', 'X106', 'X107', 'X108', 'X109', 'X110', 'X111',
       'X112', 'X113', 'X114', 'X115', 'X116', 'X117', 'X118', 'X119',
       'X120', 'X122', 'X123', 'X124', 'X125', 'X126', 'X127', 'X128',
       'X129', 'X130', 'X131', 'X132', 'X133', 'X134', 'X135', 'X136',
       'X137', 'X138', 'X139', 'X140', 'X141', 'X142', 'X143', 'X144',
       'X145', 'X146', 'X147', 'X148', 'X150', 'X151', 'X152', 'X153',
       'X154', 'X155', 'X156', 'X157', 'X158', 'X159', 'X160', 'X161',
       'X162', 'X163', 'X164', 'X165', 'X166', 'X167', 'X168', 'X169',
       'X170', 'X171', 'X172', 'X173', 'X174', 'X175', 'X176', 'X177',
       'X178', 'X179', 'X180', 'X181', 'X182', 'X183', 'X184', 'X185',
       'X186', 'X187', 'X189', 'X190', 'X191', 'X192', 'X194', 'X195',
       'X196', 'X197', 'X198', 'X199', 'X200', 'X201', 'X202', 'X203',
       'X204', 'X205', 'X206', 'X207', 'X208', 'X209', 'X210', 'X211',
       'X212', 'X213', 'X214', 'X215', 'X216', 'X217', 'X218', 'X219',
       'X220', 'X221', 'X222', 'X223', 'X224', 'X225', 'X226', 'X227',
       'X228', 'X229', 'X230', 'X231', 'X232', 'X233', 'X234', 'X235',
       'X236', 'X237', 'X238', 'X239', 'X240', 'X241', 'X242', 'X243',
       'X244', 'X245', 'X246', 'X247', 'X248', 'X249', 'X250', 'X251',
       'X252', 'X253', 'X254', 'X255', 'X256', 'X257', 'X258', 'X259',
       'X260', 'X261', 'X262', 'X263', 'X264', 'X265', 'X266', 'X267',
       'X268', 'X269', 'X270', 'X271', 'X272', 'X273', 'X274', 'X275',
       'X276', 'X277', 'X278', 'X279', 'X280', 'X281', 'X282', 'X283',
       'X284', 'X285', 'X286', 'X287', 'X288', 'X289', 'X290', 'X291',
       'X292', 'X293', 'X294', 'X295', 'X296', 'X297', 'X298', 'X299',
       'X300', 'X301', 'X302', 'X304', 'X305', 'X306', 'X307', 'X308',
       'X309', 'X310', 'X311', 'X312', 'X313', 'X314', 'X315', 'X316',
       'X317', 'X318', 'X319', 'X320', 'X321', 'X322', 'X323', 'X324',
       'X325', 'X326', 'X327', 'X328', 'X329', 'X330', 'X331', 'X332',
       'X333', 'X334', 'X335', 'X336', 'X337', 'X338', 'X339', 'X340',
       'X341', 'X342', 'X343', 'X344', 'X345', 'X346', 'X347', 'X348',
       'X349', 'X350', 'X351', 'X352', 'X353', 'X354', 'X355', 'X356',
       'X357', 'X358', 'X359', 'X360', 'X361', 'X362', 'X363', 'X364',
       'X365', 'X366', 'X367', 'X368', 'X369', 'X370', 'X371', 'X372',
       'X373', 'X374', 'X375', 'X376', 'X377', 'X378', 'X379', 'X380',
       'X382', 'X383', 'X384', 'X385']].values.ravel()
# train_feature_names
train_unique_values =  pd.unique(train_feature_names)
train_unique_values
```

Out[30]:

```
array([130.81, 'k', 'v', ..., 85.71, 108.77, 87.48], dtype=object)
```

## TASK NO 03:-> Apply Label Encoder

In [31]:

```
# df_cat_dum = pd.get_dummies(df_cat)
# apply OHE - One Hot Encoding
from sklearn.preprocessing import OneHotEncoder
```

In [32]:

```
ohe = OneHotEncoder(handle_unknown = "ignore")
```

In [33]:

```
df_cat_dum = ohe.fit_transform(df_cat).toarray()
col_names = ohe.get_feature_names()
col_names = np.array(col_names).ravel()
df_cat_oh  =pd.DataFrame(df_cat_dum, columns=col_names)
```

In [34]:

```
df_cat_oh.head()
```

Out[34]:

| | x0_a | x0_aa | x0_ab | x0_ac | x0_ad | x0_af | x0_ai | x0_aj | x0_ak | x0_al | x0_am | x0_ao | x0_ap | x0_aq | x0_as | x0_at | x0_au | x0_aw | x0_ax | x0_ay | x0_az |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **1** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **2** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| **3** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| **4** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |

In [35]:

```
df_cat_oh.shape
```

Out[35]:

```
(4209, 195)
```

In [36]:

```
# Concatenate categorical and numerical data into one data frame of training data
df_train_final = pd.concat([df_num_variance_with_zero_drop, df_cat_oh], axis = 1)
```

In [37]:

```
df_train_final.head()
```

Out[37]:

| | X10 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 | X39 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

In [38]:

```
df_train_final.shape
```

Out[38]:

```
(4209, 551)
```

## TASK NO 04 :-> Perform dimensionality reduction.

In [39]:

```
from sklearn.decomposition import PCA
pca = PCA(n_components=24)
```

In [40]:

```
df_train.dtypes
```

Out[40]:

```
y       float64
X0       object
X1       object
X2       object
X3       object
         ...
X380      int64
X382      int64
X383      int64
X384      int64
X385      int64
Length: 377, dtype: object
```

In [41]:

```
x_pca = pca.fit_transform(df_train_final)
```

In [42]:

```
df_train_final.shape
```

Out[42]:

```
(4209, 551)
```

In [43]:

```
df_pca = pd.DataFrame(x_pca)
```

In [44]:

```
df_pca.head()
```

Out[44]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.850248 | -1.252515 | 2.021640 | 0.865224 | 1.592171 | -0.056847 | 0.563839 | -1.030707 | 0.205181 | -0.264499 | -1.753130 | -0.771407 | 0.050662 | 0.107751 | 0 |
| **1** | -0.109302 | -1.299662 | -0.045801 | -0.796931 | 0.277976 | 0.140880 | 1.108070 | -0.726632 | -0.032186 | 0.612273 | -0.004161 | 1.040539 | -0.074579 | 0.499581 | -0 |
| **2** | -0.673653 | -2.367697 | 1.787792 | 2.345645 | 0.356806 | 3.753878 | -1.188808 | 0.679649 | -0.924717 | -0.215851 | 0.360034 | -0.326273 | 0.258768 | 0.199129 | 0 |
| **3** | -0.480940 | -2.695789 | 0.524340 | 2.881771 | -0.485304 | 3.765186 | -0.307379 | -0.014647 | -1.239946 | 0.254645 | 0.275336 | 0.172502 | -0.345946 | 0.763346 | 0 |
| **4** | -0.516369 | -2.692792 | 0.334140 | 3.103397 | -0.723453 | 3.866238 | -0.451954 | 0.151803 | -1.801277 | -0.298117 | 0.098126 | 0.327385 | -0.084511 | 0.030923 | -0 |

In [45]:

```
pca.explained_variance_ratio_
```

Out[45]:

```
array([0.11327864, 0.07799109, 0.07358181, 0.05848106, 0.04943089,
       0.04191889, 0.03310021, 0.0282729 , 0.02515469, 0.02153505,
       0.02077602, 0.01725079, 0.01505285, 0.01435205, 0.01385206,
       0.01296764, 0.01205455, 0.01092876, 0.00984213, 0.00913206,
       0.00883394, 0.00843642, 0.00823214, 0.00772568])
```

## load the test set here

In [46]:

```
df_test = pd.read_csv(r'C:\Users\Pavan Lande\Downloads\test\test.csv')
df_test.head()
```

Out[46]:

| X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 | X39 | X40 | X41 | X42 | X43 | X44 | X45 | X46 | X47 | X48 | X49 | X50 | X51 | X52 | X53 | X54 | X55 | X56 | X57 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

## TASK NO 02:-> Check for null in test set

In [47]:

```
# Check for null in test set
df_test.isnull().sum()
```

Out[47]:

```
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 [48]:

```
df_test.nunique()
```

Out[48]:

```
ID      4209
X0        49
X1        27
X2        45
X3         7
        ... 
X380       2
X382       2
X383       2
X384       2
X385       2
Length: 377, dtype: int64
```

## TASK NO 02:-> unique values for test sets.

In [49]:

```
test_feature_values = df_test[['ID', 'X0', 'X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X8', 'X10', 'X11',
       'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19', 'X20',
       'X21', 'X22', 'X23', 'X24', 'X26', 'X27', 'X28', 'X29', 'X30',
       'X31', 'X32', 'X33', 'X34', 'X35', 'X36', 'X37', 'X38', 'X39',
       'X40', 'X41', 'X42', 'X43', 'X44', 'X45', 'X46', 'X47', 'X48',
       'X49', 'X50', 'X51', 'X52', 'X53', 'X54', 'X55', 'X56', 'X57',
       'X58', 'X59', 'X60', 'X61', 'X62', 'X63', 'X64', 'X65', 'X66',
       'X67', 'X68', 'X69', 'X70', 'X71', 'X73', 'X74', 'X75', 'X76',
       'X77', 'X78', 'X79', 'X80', 'X81', 'X82', 'X83', 'X84', 'X85',
       'X86', 'X87', 'X88', 'X89', 'X90', 'X91', 'X92', 'X93', 'X94',
       'X95', 'X96', 'X97', 'X98', 'X99', 'X100', 'X101', 'X102', 'X103',
       'X104', 'X105', 'X106', 'X107', 'X108', 'X109', 'X110', 'X111',
       'X112', 'X113', 'X114', 'X115', 'X116', 'X117', 'X118', 'X119',
       'X120', 'X122', 'X123', 'X124', 'X125', 'X126', 'X127', 'X128',
       'X129', 'X130', 'X131', 'X132', 'X133', 'X134', 'X135', 'X136',
       'X137', 'X138', 'X139', 'X140', 'X141', 'X142', 'X143', 'X144',
       'X145', 'X146', 'X147', 'X148', 'X150', 'X151', 'X152', 'X153',
       'X154', 'X155', 'X156', 'X157', 'X158', 'X159', 'X160', 'X161',
       'X162', 'X163', 'X164', 'X165', 'X166', 'X167', 'X168', 'X169',
       'X170', 'X171', 'X172', 'X173', 'X174', 'X175', 'X176', 'X177',
       'X178', 'X179', 'X180', 'X181', 'X182', 'X183', 'X184', 'X185',
       'X186', 'X187', 'X189', 'X190', 'X191', 'X192', 'X194', 'X195',
       'X196', 'X197', 'X198', 'X199', 'X200', 'X201', 'X202', 'X203',
       'X204', 'X205', 'X206', 'X207', 'X208', 'X209', 'X210', 'X211',
       'X212', 'X213', 'X214', 'X215', 'X216', 'X217', 'X218', 'X219',
       'X220', 'X221', 'X222', 'X223', 'X224', 'X225', 'X226', 'X227',
       'X228', 'X229', 'X230', 'X231', 'X232', 'X233', 'X234', 'X235',
       'X236', 'X237', 'X238', 'X239', 'X240', 'X241', 'X242', 'X243',
       'X244', 'X245', 'X246', 'X247', 'X248', 'X249', 'X250', 'X251',
       'X252', 'X253', 'X254', 'X255', 'X256', 'X257', 'X258', 'X259',
       'X260', 'X261', 'X262', 'X263', 'X264', 'X265', 'X266', 'X267',
       'X268', 'X269', 'X270', 'X271', 'X272', 'X273', 'X274', 'X275',
       'X276', 'X277', 'X278', 'X279', 'X280', 'X281', 'X282', 'X283',
       'X284', 'X285', 'X286', 'X287', 'X288', 'X289', 'X290', 'X291',
       'X292', 'X293', 'X294', 'X295', 'X296', 'X297', 'X298', 'X299',
       'X300', 'X301', 'X302', 'X304', 'X305', 'X306', 'X307', 'X308',
       'X309', 'X310', 'X311', 'X312', 'X313', 'X314', 'X315', 'X316',
       'X317', 'X318', 'X319', 'X320', 'X321', 'X322', 'X323', 'X324',
       'X325', 'X326', 'X327', 'X328', 'X329', 'X330', 'X331', 'X332',
       'X333', 'X334', 'X335', 'X336', 'X337', 'X338', 'X339', 'X340',
       'X341', 'X342', 'X343', 'X344', 'X345', 'X346', 'X347', 'X348',
       'X349', 'X350', 'X351', 'X352', 'X353', 'X354', 'X355', 'X356',
       'X357', 'X358', 'X359', 'X360', 'X361', 'X362', 'X363', 'X364',
       'X365', 'X366', 'X367', 'X368', 'X369', 'X370', 'X371', 'X372',
       'X373', 'X374', 'X375', 'X376', 'X377', 'X378', 'X379', 'X380',
       'X382', 'X383', 'X384', 'X385']].values.ravel()
test_unique_values =  pd.unique(test_feature_values)
test_unique_values
```

Out[49]:

```
array([1, 'az', 'v', ..., 8413, 8414, 8416], dtype=object)
```

In [50]:

```
df_test.shape
```

Out[50]:

```
(4209, 377)
```

In [51]:

```
df_test.dtypes
```

Out[51]:

```
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 [52]:

```
# Seperate the numerical and categorical columns for test data
test_df_cat = df_test.select_dtypes(include = np.object)
test_df_num = df_test.select_dtypes(exclude = np.object)
```

In [53]:

```
test_df_cat.head()
```

Out[53]:

| | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 |
|---|---|---|---|---|---|---|---|---|
| **0** | az | v | n | f | d | t | a | w |
| **1** | t | b | ai | a | d | b | g | y |
| **2** | az | v | as | f | d | a | j | j |
| **3** | az | l | n | f | d | z | l | n |
| **4** | w | s | as | c | d | y | i | m |

In [54]:

```
test_df_num.head()
```

Out[54]:

| | ID | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | |
| **1** | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | |
| **2** | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | |
| **3** | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | |
| **4** | 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

In [55]:

```
test_df_num = test_df_num.drop("ID", axis = 1)
test_df_num.head()
```

Out[55]:

| | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In [56]:

```
test_df_num.shape
```

Out[56]:

```
(4209, 368)
```

In [57]:

```
test_columns = test_df_num.columns
test_columns
```

Out[57]:

```
Index(['X10', 'X11', 'X12', 'X13', 'X14', 'X15', 'X16', 'X17', 'X18', 'X19',
       ...
       'X375', 'X376', 'X377', 'X378', 'X379', 'X380', 'X382', 'X383', 'X384',
       'X385'],
      dtype='object', length=368)
```

In [58]:

```
# Apply scaling for test set
test_df_num_sc = mn.transform(test_df_num)
test_df_num_df = pd.DataFrame(test_df_num_sc, index = test_df_num.index, columns=test_df_num.columns)
test_df_num_df.head()
```

Out[58]:

| | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **1** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **2** | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **3** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 1.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 |
| **4** | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

## Test Set - If for any column(s), the variance is equal to zero, then you need to remove those variable(s)

In [59]:

```
test_variance_df_num = test_df_num.var()
```

In [60]:

```
test_variable_var_zero = [ ]

for i in range(0,len(test_variance_df_num)):
    if test_variance_df_num[i]==0: #checking if the variance for the df_num dataframe column has zero
        test_variable_var_zero.append(test_columns[i])
```

In [61]:

```
np.ravel(test_variable_var_zero)
```

Out[61]:

```
array(['X257', 'X258', 'X295', 'X296', 'X369'], dtype='<U4')
```

In [62]:

```
test_df_num_variance_with_zero_drop = test_df_num.drop(['X257', 'X258', 'X295', 'X296', 'X369'], axis = 1)
```

In [63]:

```
test_df_num_variance_with_zero_drop.head()
```

Out[63]:

| | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In [64]:

```
test_df_num_variance_with_zero_drop.shape
```

Out[64]:

```
(4209, 363)
```

In [65]:

```
#### Apply ONE HOT encoder for test set
test_df_cat_dum = ohe.transform(test_df_cat).toarray()
test_col_names = ohe.get_feature_names()
test_col_names = np.array(test_col_names).ravel()
test_df_cat_oh  =pd.DataFrame(test_df_cat_dum, columns=test_col_names)
test_df_cat_oh.head()
```

Out[65]:

| | x0_a | x0_aa | x0_ab | x0_ac | x0_ad | x0_af | x0_ai | x0_aj | x0_ak | x0_al | x0_am | x0_ao | x0_ap | x0_aq | x0_as | x0_at | x0_au | x0_aw | x0_ax | x0_ay | x0_az |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| **1** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **2** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| **3** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 1.0 |
| **4** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

In [66]:

```
# concatenate the both categorical and numerical features of test set
df_test_final = pd.concat([test_df_num_variance_with_zero_drop, test_df_cat_oh], axis = 1)
```

In [67]:

```
df_test_final.head()
```

Out[67]:

| | X10 | X11 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **2** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| **4** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In [68]:

```
print(df_train_final.shape)
print(df_test_final.shape)
```

```
(4209, 551)
(4209, 558)
```

In [69]:

```
# while dropping columns with 0 variance for train and test data sets feature results are different,
# hence to balance the feature in train and test sets, added dropped dummy columns with NAN values to apply PCA
# reset the test data features to align with train features
test_df_newdata = df_test_final.reindex(labels=df_train_final.columns,axis=1)
test_df_newdata.head()
```

Out[69]:

| | X10 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 | X39 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

In [70]:

```
# fill the NAN values with 0 to fit to PCA
test_df_newdata["X257"] = test_df_newdata["X257"].fillna(0)
test_df_newdata["X258"] = test_df_newdata["X258"].fillna(0)
test_df_newdata["X295"] = test_df_newdata["X295"].fillna(0)
test_df_newdata["X296"] = test_df_newdata["X296"].fillna(0)
test_df_newdata["X369"] = test_df_newdata["X369"].fillna(0)
test_df_newdata.head()
```

Out[70]:

| | X10 | X12 | X13 | X14 | X15 | X16 | X17 | X18 | X19 | X20 | X21 | X22 | X23 | X24 | X26 | X27 | X28 | X29 | X30 | X31 | X32 | X33 | X34 | X35 | X36 | X37 | X38 | X39 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **3** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| **4** | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Apply PCA for test dataset

In [71]:

```
test_x_pca = pca.transform(test_df_newdata)
```

In [72]:

```
# X_train and y Values of train data set
X_train = df_train_final
y_train = df_train['y']
```

In [73]:

```
# X_test values of test data set
X_test = test_df_newdata
```

## TASK NO 05 :-> Predict your test_df values using XGBoost.

In [74]:

```
xgb = XGBRegressor()
```

In [75]:

```
xgb.fit(X_train, y_train)
```

Out[75]:

```
XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
             colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
             early_stopping_rounds=None, enable_categorical=False,
             eval_metric=None, feature_types=None, gamma=0, gpu_id=-1,
             grow_policy='depthwise', importance_type=None,
             interaction_constraints='', learning_rate=0.300000012, max_bin=256,
             max_cat_threshold=64, max_cat_to_onehot=4, max_delta_step=0,
             max_depth=6, max_leaves=0, min_child_weight=1, missing=nan,
             monotone_constraints='()', n_estimators=100, n_jobs=0,
             num_parallel_tree=1, predictor='auto', random_state=0, ...)
```

In [76]:

```
pred = xgb.predict(X_test)
```

In [77]:

```
pred
```

Out[77]:

```
array([ 95.92638, 112.90855,  99.74303, ...,  96.50017, 107.51481,
        90.8429 ], dtype=float32)
```

In [78]:

```
df_res = pd.DataFrame(pred, columns = ["yHat"])
df_res
```

Out[78]:

|  | yHat |
|---|---|
| **0** | 95.926376 |
| **1** | 112.908546 |
| **2** | 99.743027 |
| **3** | 79.599861 |
| **4** | 112.196259 |
| **...** | ... |
| **4204** | 107.167992 |
| **4205** | 90.772079 |
| **4206** | 96.500168 |
| **4207** | 107.514809 |
| **4208** | 90.842903 |

4209 rows × 1 columns

In [79]:

```
df_res.to_csv('submission.csv',index=False)
```

In [ ]:

```
## END ###
```