Project 1 - Mercedes-Benz Greener Manufacturing

November 15, 2022

1 Import Packages

[1]:
```
import pandas as pd
import numpy as np

from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()

from sklearn.model_selection import train_test_split

from sklearn.decomposition import PCA

import xgboost,time
from sklearn.metrics import r2_score
from sklearn.model_selection import GridSearchCV

import warnings
warnings.filterwarnings('ignore')
```

2 Import Data

[2]:
```
train = pd.read_csv('train.csv')
test = pd.read_csv('test.csv')
```

3 Check Data

[3]:
```
train.head()
```

[3]:
```
   ID       y  X0 X1  X2 X3 X4 X5 X6 X8  …  X375  X376  X377  X378  X379  \
0   0  130.81   k  v  at  a  d  u  j  o  …     0     0     1     0     0
1   6   88.53   k  t  av  e  d  y  l  o  …     1     0     0     0     0
2   7   76.26  az  w   n  c  d  x  j  x  …     0     0     0     0     0
3   9   80.62  az  t   n  f  d  x  l  e  …     0     0     0     0     0
```

```
4  13   78.02  az  v   n  f  d  h  d  n  …     0     0     0     0     0

   X380  X382  X383  X384  X385
0     0     0     0     0     0
1     0     0     0     0     0
2     0     1     0     0     0
3     0     0     0     0     0
4     0     0     0     0     0

[5 rows x 378 columns]
```

[4]:
```
test.head()
```

[4]:
```
   ID  X0 X1  X2 X3 X4 X5 X6 X8  X10  …  X375  X376  X377  X378  X379  X380  \
0   1  az  v   n  f  d  t  a  w    0  …     0     0     0     1     0     0
1   2   t  b  ai  a  d  b  g  y    0  …     0     0     1     0     0     0
2   3  az  v  as  f  d  a  j  j    0  …     0     0     0     1     0     0
3   4  az  l   n  f  d  z  l  n    0  …     0     0     0     1     0     0
4   5   w  s  as  c  d  y  i  m    0  …     1     0     0     0     0     0

   X382  X383  X384  X385
0     0     0     0     0
1     0     0     0     0
2     0     0     0     0
3     0     0     0     0
4     0     0     0     0

[5 rows x 377 columns]
```

[5]:
```
train.dtypes
```

[5]:
```
ID         int64
y        float64
X0        object
X1        object
X2        object
          …
X380       int64
X382       int64
X383       int64
X384       int64
X385       int64
Length: 378, dtype: object
```

[6]:
```
test.dtypes
```

[6]:
```
ID        int64
X0       object
X1       object
X2       object
X3       object
          …
X380      int64
X382      int64
X383      int64
X384      int64
X385      int64
Length: 377, dtype: object
```

4 Dropping irrelevant column

[7]:
```
train = train.drop('ID',axis =1)
test= test.drop('ID',axis=1)
```

[8]:
```
train.head()
```

[8]:
```
        y  X0 X1  X2 X3 X4 X5 X6 X8  X10  …  X375  X376  X377  X378  X379  \
0  130.81   k  v  at  a  d  u  j  o    0  …     0     0     1     0     0
1   88.53   k  t  av  e  d  y  l  o    0  …     1     0     0     0     0
2   76.26  az  w   n  c  d  x  j  x    0  …     0     0     0     0     0
3   80.62  az  t   n  f  d  x  l  e    0  …     0     0     0     0     0
4   78.02  az  v   n  f  d  h  d  n    0  …     0     0     0     0     0

   X380  X382  X383  X384  X385
0     0     0     0     0     0
1     0     0     0     0     0
2     0     1     0     0     0
3     0     0     0     0     0
4     0     0     0     0     0

[5 rows x 377 columns]
```

[9]:
```
test.head()
```

[9]:
```
   X0 X1  X2 X3 X4 X5 X6 X8  X10  X11  …  X375  X376  X377  X378  X379  \
0  az  v   n  f  d  t  a  w    0    0  …     0     0     0     1     0
1   t  b  ai  a  d  b  g  y    0    0  …     0     0     1     0     0
2  az  v  as  f  d  a  j  j    0    0  …     0     0     0     1     0
3  az  l   n  f  d  z  l  n    0    0  …     0     0     0     1     0
4   w  s  as  c  d  y  i  m    0    0  …     1     0     0     0     0
```

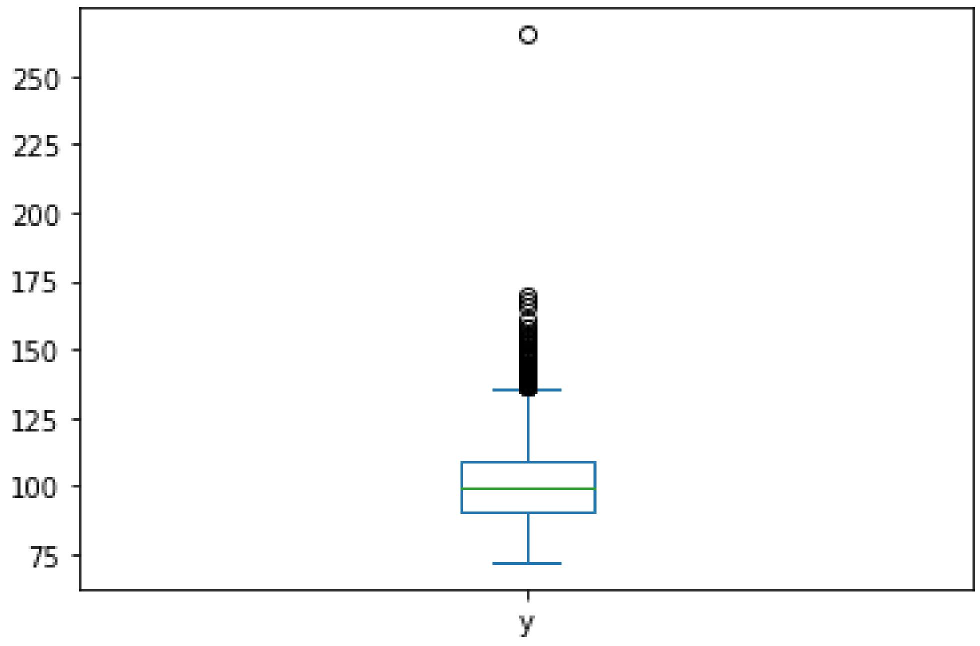
```
   X380  X382  X383  X384  X385
0     0     0     0     0     0
1     0     0     0     0     0
2     0     0     0     0     0
3     0     0     0     0     0
4     0     0     0     0     0

[5 rows x 376 columns]
```

5 Treat Target Column Outliers

[10]:
```python
train['y'].plot.box()
```

[10]:
```
<AxesSubplot:>
```



[11]:
```python
Q1 = train['y'].quantile(0.25)
Q3 = train['y'].quantile(0.75)

IQR = Q3 - Q1

lower_range = Q1 - 1.5 * IQR
upper_range = Q3 + 1.5 * IQR

print(lower_range)
print(upper_range)
```

```
63.534999999999975
136.29500000000002
```

[12]:
```
outlier_index = train[(train.y > upper_range)].index
outlier_index
```

[12]:
```
Int64Index([  43,  203,  216,  253,  342,  420,  429,  681,  846,  883,  889,
             900,  995,  998, 1033, 1036, 1060, 1141, 1203, 1205, 1269, 1279,
            1349, 1459, 1730, 2240, 2263, 2348, 2357, 2376, 2414, 2470, 2496,
            2735, 2736, 2852, 2887, 2888, 2905, 2983, 3028, 3090, 3133, 3177,
            3215, 3442, 3744, 3773, 3980, 4176],
           dtype='int64')
```

6 Drop Outliers from Train

[13]:
```
train = train.drop(outlier_index)
```

[14]:
```
train.shape
```

[14]:
```
(4159, 377)
```

7 Seprating categorical and numerical data types.

[15]:
```
df_num_train = train.select_dtypes(exclude = np.object)
df_cat_train = train.select_dtypes(include = np.object)

df_num_test = test.select_dtypes(exclude = np.object)
df_cat_test = test.select_dtypes(include = np.object)
```

[16]:
```
print('Shape of Cat. Test Data:',df_cat_test.shape)
print('Shape of Num. Test Data:',df_num_test.shape)
print('Shape of Cat. Train Data:',df_cat_train.shape)
print('Shape of Num. Train Data:',df_num_train.shape)
```

```
Shape of Cat. Test Data: (4209, 8)
Shape of Num. Test Data: (4209, 368)
Shape of Cat. Train Data: (4159, 8)
Shape of Num. Train Data: (4159, 369)
```

[17]:
```
df_num_test.head()
```

```
[17]:    X10  X11  X12  X13  X14  X15  X16  X17  X18  X19  …  X375  X376  X377  \
      0    0    0    0    0    0    0    0    0    0    0  …     0     0     0
      1    0    0    0    0    0    0    0    0    0    1  …     0     0     1
      2    0    0    0    0    1    0    0    0    0    0  …     0     0     0
      3    0    0    0    0    0    0    0    0    0    0  …     0     0     0
      4    0    0    0    0    1    0    0    0    0    0  …     1     0     0

         X378  X379  X380  X382  X383  X384  X385
      0     1     0     0     0     0     0     0
      1     0     0     0     0     0     0     0
      2     1     0     0     0     0     0     0
      3     1     0     0     0     0     0     0
      4     0     0     0     0     0     0     0

      [5 rows x 368 columns]
```

```
[18]: df_cat_train.head()
```

```
[18]:    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
```

8 If for any column(s),Variance = 0, then remove those variable(s).

```
[19]: # for train data
      for i in df_num_train.columns :
          if df_num_train.var()[i] == 0:
              df_num_train.drop(i,axis=1,inplace=True)

      # for test data
      for i in df_num_test.columns :
          if df_num_test.var()[i] == 0:
              df_num_test.drop(i,axis=1,inplace=True)
```

```
[20]: print('Shape of Num. Test Data:',df_num_test.shape)
      print('Shape of Num. Train Data:',df_num_train.shape)
```

```
Shape of Num. Test Data: (4209, 363)
Shape of Num. Train Data: (4159, 356)
```

9 Check for null and unique values for test and train sets.

Concat. df_cat & df_num

```
[21]: newtrain = pd.concat([df_num_train,df_cat_train],axis=1)
      newtest = pd.concat([df_num_test,df_cat_test],axis=1)
```

```
[22]: print('Shape of New Test Data:',newtest.shape)
      print('Shape of New Train Data:',newtrain.shape)
```

```
Shape of New Test Data: (4209, 371)
Shape of New Train Data: (4159, 364)
```

Check for null values

```
[23]: for i in newtrain.columns :
          if newtrain[i].isna().value_counts() is True:
              print(newtrain[i])

      for i in newtest.columns :
          if newtest[i].isna().value_counts() is True:
              print(newtest[i])
```

So no null values

Check for unique values

```
[24]: for i in newtrain.columns :
          print(i,':',end="")
          print(newtrain[i].unique())
```

```
y :[130.81  88.53  76.26 …  85.71 108.77  87.48]
X10 :[0 1]
X12 :[0 1]
X13 :[1 0]
X14 :[0 1]
X15 :[0 1]
X16 :[0 1]
X17 :[0 1]
X18 :[1 0]
X19 :[0 1]
X20 :[0 1]
X21 :[1 0]
X22 :[0 1]
X23 :[0 1]
X24 :[0 1]
X26 :[0 1]
X27 :[0 1]
X28 :[0 1]
```

```
X29 :[0 1]
X30 :[0 1]
X31 :[1 0]
X32 :[0 1]
X33 :[0 1]
X34 :[0 1]
X35 :[1 0]
X36 :[0 1]
X37 :[1 0]
X38 :[0 1]
X39 :[0 1]
X40 :[0 1]
X41 :[0 1]
X42 :[0 1]
X43 :[0 1]
X44 :[0 1]
X45 :[0 1]
X46 :[1 0]
X47 :[0 1]
X48 :[0 1]
X49 :[0 1]
X50 :[0 1]
X51 :[0 1]
X52 :[0 1]
X53 :[0 1]
X54 :[0 1]
X55 :[0 1]
X56 :[0 1]
X57 :[0 1]
X58 :[1 0]
X59 :[0 1]
X60 :[0 1]
X61 :[0 1]
X62 :[0 1]
X63 :[0 1]
X64 :[0 1]
X65 :[0 1]
X66 :[0 1]
X67 :[0 1]
X68 :[1 0]
X69 :[0 1]
X70 :[1 0]
X71 :[0 1]
X73 :[0 1]
X74 :[1 0]
X75 :[0 1]
X76 :[0 1]
X77 :[0 1]
```

```
X78 :[0 1]
X79 :[0 1]
X80 :[0 1]
X81 :[0 1]
X82 :[0 1]
X83 :[0 1]
X84 :[0 1]
X85 :[1 0]
X86 :[0 1]
X87 :[0 1]
X88 :[0 1]
X89 :[0 1]
X90 :[0 1]
X91 :[0 1]
X92 :[0 1]
X94 :[0 1]
X95 :[0 1]
X96 :[0 1]
X97 :[0 1]
X98 :[0 1]
X99 :[0 1]
X100 :[0 1]
X101 :[0 1]
X102 :[0 1]
X103 :[0 1]
X104 :[0 1]
X105 :[0 1]
X106 :[0 1]
X108 :[0 1]
X109 :[0 1]
X110 :[0 1]
X111 :[1 0]
X112 :[0 1]
X113 :[0 1]
X114 :[1 0]
X115 :[0 1]
X116 :[1 0]
X117 :[0 1]
X118 :[1 0]
X119 :[1 0]
X120 :[1 0]
X122 :[0 1]
X123 :[0 1]
X124 :[0 1]
X125 :[0 1]
X126 :[0 1]
X127 :[0 1]
X128 :[1 0]
```

```
X129 :[0 1]
X130 :[0 1]
X131 :[1 0]
X132 :[0 1]
X133 :[0 1]
X134 :[0 1]
X135 :[0 1]
X136 :[1 0]
X137 :[1 0]
X138 :[0 1]
X139 :[0 1]
X140 :[0 1]
X141 :[0 1]
X142 :[1 0]
X143 :[0 1]
X144 :[1 0]
X145 :[0 1]
X146 :[0 1]
X147 :[0 1]
X148 :[0 1]
X150 :[1 0]
X151 :[0 1]
X152 :[0 1]
X153 :[0 1]
X154 :[0 1]
X155 :[0 1]
X156 :[1 0]
X157 :[0 1]
X158 :[0 1]
X159 :[0 1]
X160 :[0 1]
X161 :[0 1]
X162 :[0 1]
X163 :[0 1]
X164 :[0 1]
X165 :[0 1]
X166 :[0 1]
X167 :[0 1]
X168 :[0 1]
X169 :[0 1]
X170 :[1 0]
X171 :[0 1]
X172 :[0 1]
X173 :[0 1]
X174 :[0 1]
X175 :[0 1]
X176 :[0 1]
X177 :[0 1]
```

```
X178 :[0 1]
X179 :[1 0]
X180 :[0 1]
X181 :[0 1]
X182 :[0 1]
X183 :[0 1]
X184 :[1 0]
X185 :[0 1]
X186 :[0 1]
X187 :[1 0]
X189 :[1 0]
X190 :[0 1]
X191 :[0 1]
X192 :[0 1]
X194 :[1 0]
X195 :[0 1]
X196 :[0 1]
X197 :[0 1]
X198 :[0 1]
X199 :[0 1]
X200 :[0 1]
X201 :[0 1]
X202 :[0 1]
X203 :[0 1]
X204 :[1 0]
X205 :[0 1]
X206 :[0 1]
X207 :[0 1]
X208 :[0 1]
X209 :[1 0]
X210 :[0 1]
X211 :[0 1]
X212 :[0 1]
X213 :[0 1]
X214 :[0 1]
X215 :[0 1]
X216 :[0 1]
X217 :[0 1]
X218 :[0 1]
X219 :[0 1]
X220 :[1 0]
X221 :[0 1]
X222 :[0 1]
X223 :[0 1]
X224 :[0 1]
X225 :[0 1]
X226 :[0 1]
X227 :[0 1]
```

```
X228 :[0 1]
X229 :[0 1]
X230 :[0 1]
X231 :[0 1]
X232 :[0 1]
X234 :[1 0]
X236 :[0 1]
X237 :[1 0]
X238 :[0 1]
X239 :[0 1]
X240 :[0 1]
X241 :[0 1]
X242 :[0 1]
X243 :[0 1]
X244 :[0 1]
X245 :[0 1]
X246 :[0 1]
X247 :[0 1]
X248 :[0 1]
X249 :[0 1]
X250 :[0 1]
X251 :[0 1]
X252 :[0 1]
X253 :[0 1]
X254 :[0 1]
X255 :[0 1]
X256 :[0 1]
X257 :[0 1]
X258 :[0 1]
X259 :[0 1]
X260 :[0 1]
X261 :[0 1]
X262 :[1 0]
X263 :[1 0]
X264 :[0 1]
X265 :[0 1]
X266 :[1 0]
X267 :[0 1]
X269 :[0 1]
X270 :[0 1]
X271 :[0 1]
X272 :[0 1]
X273 :[1 0]
X274 :[0 1]
X275 :[1 0]
X276 :[0 1]
X277 :[0 1]
X278 :[0 1]
```

```
X279 :[0 1]
X280 :[0 1]
X281 :[0 1]
X282 :[0 1]
X283 :[0 1]
X284 :[0 1]
X285 :[1 0]
X286 :[0 1]
X287 :[0 1]
X288 :[0 1]
X291 :[0 1]
X292 :[0 1]
X294 :[0 1]
X295 :[0 1]
X296 :[0 1]
X298 :[0 1]
X299 :[0 1]
X300 :[0 1]
X301 :[0 1]
X302 :[0 1]
X304 :[0 1]
X305 :[0 1]
X306 :[1 0]
X307 :[0 1]
X308 :[0 1]
X309 :[0 1]
X310 :[0 1]
X311 :[0 1]
X312 :[0 1]
X313 :[0 1]
X314 :[0 1]
X315 :[0 1]
X316 :[1 0]
X317 :[0 1]
X318 :[0 1]
X319 :[0 1]
X320 :[0 1]
X321 :[0 1]
X322 :[0 1]
X323 :[0 1]
X324 :[1 0]
X325 :[0 1]
X326 :[0 1]
X327 :[1 0]
X328 :[0 1]
X329 :[1 0]
X331 :[0 1]
X332 :[0 1]
```

```
X333 :[0 1]
X334 :[1 0]
X335 :[0 1]
X336 :[0 1]
X337 :[0 1]
X338 :[0 1]
X340 :[0 1]
X341 :[0 1]
X342 :[0 1]
X343 :[0 1]
X344 :[0 1]
X345 :[0 1]
X346 :[0 1]
X348 :[0 1]
X349 :[0 1]
X350 :[0 1]
X351 :[0 1]
X352 :[0 1]
X353 :[0 1]
X354 :[1 0]
X355 :[0 1]
X356 :[0 1]
X357 :[0 1]
X358 :[0 1]
X359 :[0 1]
X360 :[0 1]
X361 :[1 0]
X362 :[0 1]
X363 :[0 1]
X364 :[0 1]
X365 :[0 1]
X366 :[0 1]
X367 :[0 1]
X368 :[0 1]
X369 :[0 1]
X370 :[0 1]
X371 :[0 1]
X372 :[0 1]
X373 :[0 1]
X374 :[0 1]
X375 :[0 1]
X376 :[0 1]
X377 :[1 0]
X378 :[0 1]
X379 :[0 1]
X380 :[0 1]
X382 :[0 1]
X383 :[0 1]
```

```
X384 :[0 1]
X385 :[0 1]
X0 :['k' 'az' 't' 'al' 'o' 'w' 'j' 'h' 's' 'n' 'ay' 'f' 'x' 'y' 'aj' 'ak' 'am'
 'z' 'q' 'at' 'ap' 'v' 'af' 'a' 'e' 'ai' 'd' 'aq' 'c' 'aa' 'ba' 'as' 'i'
 'r' 'b' 'ax' 'bc' 'u' 'ad' 'au' 'm' 'l' 'aw' 'ao' 'ac' 'g' 'ab']
X1 :['v' 't' 'w' 'b' 'r' 'l' 's' 'aa' 'c' 'a' 'e' 'h' 'z' 'j' 'o' 'u' 'p' 'n'
 'i' 'y' 'd' 'f' 'm' 'k' 'g' 'q' 'ab']
X2 :['at' 'av' 'n' 'e' 'as' 'aq' 'r' 'ai' 'ak' 'm' 'a' 'k' 'ae' 's' 'f' 'd'
 'ag' 'ay' 'ac' 'ap' 'g' 'i' 'aw' 'y' 'b' 'ao' 'al' 'x' 'au' 't' 'an' 'z'
 'ah' 'p' 'am' 'h' 'j' 'q' 'af' 'l' 'c' 'o' 'ar']
X3 :['a' 'e' 'c' 'f' 'd' 'b' 'g']
X4 :['d' 'b' 'c' 'a']
X5 :['u' 'y' 'x' 'h' 'g' 'f' 'j' 'i' 'd' 'c' 'af' 'ag' 'ab' 'ac' 'ad' 'ae'
 'ah' 'l' 'k' 'n' 'm' 'p' 'q' 's' 'r' 'v' 'w' 'o' 'aa']
X6 :['j' 'l' 'd' 'h' 'i' 'a' 'g' 'c' 'k' 'e' 'f' 'b']
X8 :['o' 'x' 'e' 'n' 's' 'a' 'h' 'p' 'm' 'k' 'd' 'i' 'v' 'j' 'b' 'q' 'w' 'g'
 'y' 'l' 'f' 'u' 'r' 't' 'c']
```

[25]:
```python
for i in newtest.columns :
    print(i,':',end="")
    print(newtest[i].unique())
```

```
X10 :[0 1]
X11 :[0 1]
X12 :[0 1]
X13 :[0 1]
X14 :[0 1]
X15 :[0 1]
X16 :[0 1]
X17 :[0 1]
X18 :[0 1]
X19 :[0 1]
X20 :[0 1]
X21 :[0 1]
X22 :[0 1]
X23 :[0 1]
X24 :[0 1]
X26 :[0 1]
X27 :[1 0]
X28 :[1 0]
X29 :[1 0]
X30 :[0 1]
X31 :[1 0]
X32 :[0 1]
X33 :[0 1]
X34 :[0 1]
X35 :[1 0]
X36 :[0 1]
```

```
X37 :[1 0]
X38 :[0 1]
X39 :[0 1]
X40 :[0 1]
X41 :[0 1]
X42 :[0 1]
X43 :[1 0]
X44 :[0 1]
X45 :[0 1]
X46 :[1 0]
X47 :[0 1]
X48 :[0 1]
X49 :[0 1]
X50 :[0 1]
X51 :[0 1]
X52 :[0 1]
X53 :[0 1]
X54 :[1 0]
X55 :[0 1]
X56 :[0 1]
X57 :[0 1]
X58 :[0 1]
X59 :[0 1]
X60 :[0 1]
X61 :[1 0]
X62 :[0 1]
X63 :[0 1]
X64 :[0 1]
X65 :[0 1]
X66 :[0 1]
X67 :[0 1]
X68 :[0 1]
X69 :[0 1]
X70 :[1 0]
X71 :[0 1]
X73 :[0 1]
X74 :[1 0]
X75 :[0 1]
X76 :[1 0]
X77 :[0 1]
X78 :[0 1]
X79 :[0 1]
X80 :[1 0]
X81 :[0 1]
X82 :[0 1]
X83 :[0 1]
X84 :[0 1]
X85 :[0 1]
```

```
X86 :[0 1]
X87 :[0 1]
X88 :[0 1]
X89 :[0 1]
X90 :[0 1]
X91 :[0 1]
X92 :[0 1]
X93 :[0 1]
X94 :[0 1]
X95 :[0 1]
X96 :[1 0]
X97 :[0 1]
X98 :[1 0]
X99 :[0 1]
X100 :[0 1]
X101 :[1 0]
X102 :[0 1]
X103 :[0 1]
X104 :[0 1]
X105 :[0 1]
X106 :[0 1]
X107 :[0 1]
X108 :[0 1]
X109 :[0 1]
X110 :[0 1]
X111 :[1 0]
X112 :[0 1]
X113 :[0 1]
X114 :[1 0]
X115 :[0 1]
X116 :[0 1]
X117 :[0 1]
X118 :[0 1]
X119 :[0 1]
X120 :[0 1]
X122 :[0 1]
X123 :[0 1]
X124 :[0 1]
X125 :[0 1]
X126 :[0 1]
X127 :[0 1]
X128 :[1 0]
X129 :[0 1]
X130 :[0 1]
X131 :[0 1]
X132 :[1 0]
X133 :[0 1]
X134 :[0 1]
```

```
X135 :[0 1]
X136 :[0 1]
X137 :[0 1]
X138 :[0 1]
X139 :[0 1]
X140 :[0 1]
X141 :[0 1]
X142 :[0 1]
X143 :[0 1]
X144 :[1 0]
X145 :[0 1]
X146 :[0 1]
X147 :[0 1]
X148 :[1 0]
X150 :[1 0]
X151 :[0 1]
X152 :[0 1]
X153 :[0 1]
X154 :[0 1]
X155 :[0 1]
X156 :[0 1]
X157 :[1 0]
X158 :[1 0]
X159 :[0 1]
X160 :[0 1]
X161 :[0 1]
X162 :[1 0]
X163 :[0 1]
X164 :[0 1]
X165 :[0 1]
X166 :[1 0]
X167 :[0 1]
X168 :[0 1]
X169 :[0 1]
X170 :[0 1]
X171 :[0 1]
X172 :[0 1]
X173 :[0 1]
X174 :[0 1]
X175 :[0 1]
X176 :[0 1]
X177 :[0 1]
X178 :[0 1]
X179 :[1 0]
X180 :[0 1]
X181 :[0 1]
X182 :[0 1]
X183 :[0 1]
```

```
X184 :[0 1]
X185 :[1 0]
X186 :[0 1]
X187 :[0 1]
X189 :[0 1]
X190 :[0 1]
X191 :[0 1]
X192 :[0 1]
X194 :[1 0]
X195 :[0 1]
X196 :[0 1]
X197 :[0 1]
X198 :[0 1]
X199 :[0 1]
X200 :[0 1]
X201 :[0 1]
X202 :[0 1]
X203 :[0 1]
X204 :[1 0]
X205 :[0 1]
X206 :[0 1]
X207 :[0 1]
X208 :[0 1]
X209 :[1 0]
X210 :[0 1]
X211 :[0 1]
X212 :[0 1]
X213 :[0 1]
X214 :[0 1]
X215 :[0 1]
X216 :[0 1]
X217 :[0 1]
X218 :[1 0]
X219 :[0 1]
X220 :[1 0]
X221 :[0 1]
X222 :[0 1]
X223 :[1 0]
X224 :[0 1]
X225 :[0 1]
X226 :[0 1]
X227 :[0 1]
X228 :[0 1]
X229 :[0 1]
X230 :[0 1]
X231 :[0 1]
X232 :[1 0]
X233 :[0 1]
```

```
X234 :[0 1]
X235 :[0 1]
X236 :[0 1]
X237 :[0 1]
X238 :[0 1]
X239 :[0 1]
X240 :[0 1]
X241 :[0 1]
X242 :[0 1]
X243 :[0 1]
X244 :[0 1]
X245 :[0 1]
X246 :[1 0]
X247 :[0 1]
X248 :[0 1]
X249 :[0 1]
X250 :[1 0]
X251 :[0 1]
X252 :[0 1]
X253 :[0 1]
X254 :[0 1]
X255 :[0 1]
X256 :[1 0]
X259 :[0 1]
X260 :[0 1]
X261 :[0 1]
X262 :[0 1]
X263 :[0 1]
X264 :[0 1]
X265 :[0 1]
X266 :[0 1]
X267 :[0 1]
X268 :[0 1]
X269 :[0 1]
X270 :[0 1]
X271 :[0 1]
X272 :[1 0]
X273 :[1 0]
X274 :[0 1]
X275 :[0 1]
X276 :[1 0]
X277 :[0 1]
X278 :[0 1]
X279 :[1 0]
X280 :[0 1]
X281 :[0 1]
X282 :[0 1]
X283 :[0 1]
```

```
X284 :[0 1]
X285 :[0 1]
X286 :[1 0]
X287 :[0 1]
X288 :[0 1]
X289 :[0 1]
X290 :[0 1]
X291 :[0 1]
X292 :[0 1]
X293 :[0 1]
X294 :[0 1]
X297 :[0 1]
X298 :[0 1]
X299 :[0 1]
X300 :[0 1]
X301 :[0 1]
X302 :[0 1]
X304 :[1 0]
X305 :[0 1]
X306 :[0 1]
X307 :[0 1]
X308 :[0 1]
X309 :[0 1]
X310 :[0 1]
X311 :[0 1]
X312 :[0 1]
X313 :[0 1]
X314 :[0 1]
X315 :[0 1]
X316 :[0 1]
X317 :[0 1]
X318 :[0 1]
X319 :[0 1]
X320 :[0 1]
X321 :[0 1]
X322 :[0 1]
X323 :[0 1]
X324 :[0 1]
X325 :[0 1]
X326 :[0 1]
X327 :[0 1]
X328 :[1 0]
X329 :[0 1]
X330 :[0 1]
X331 :[0 1]
X332 :[0 1]
X333 :[0 1]
X334 :[1 0]
```

```
X335 :[0 1]
X336 :[0 1]
X337 :[0 1]
X338 :[0 1]
X339 :[0 1]
X340 :[0 1]
X341 :[0 1]
X342 :[0 1]
X343 :[0 1]
X344 :[0 1]
X345 :[0 1]
X346 :[0 1]
X347 :[0 1]
X348 :[1 0]
X349 :[0 1]
X350 :[1 0]
X351 :[0 1]
X352 :[0 1]
X353 :[0 1]
X354 :[0 1]
X355 :[0 1]
X356 :[0 1]
X357 :[0 1]
X358 :[1 0]
X359 :[0 1]
X360 :[0 1]
X361 :[1 0]
X362 :[0 1]
X363 :[1 0]
X364 :[0 1]
X365 :[0 1]
X366 :[0 1]
X367 :[0 1]
X368 :[0 1]
X370 :[0 1]
X371 :[0 1]
X372 :[0 1]
X373 :[0 1]
X374 :[0 1]
X375 :[0 1]
X376 :[0 1]
X377 :[0 1]
X378 :[1 0]
X379 :[0 1]
X380 :[0 1]
X382 :[0 1]
X383 :[0 1]
X384 :[0 1]
```

```
X385 :[0 1]
X0 :['az' 't' 'w' 'y' 'x' 'f' 'ap' 'o' 'ay' 'al' 'h' 'z' 'aj' 'd' 'v' 'ak'
 'ba' 'n' 'j' 's' 'af' 'ax' 'at' 'aq' 'av' 'm' 'k' 'a' 'e' 'ai' 'i' 'ag'
 'b' 'am' 'aw' 'as' 'r' 'ao' 'u' 'l' 'c' 'ad' 'au' 'bc' 'g' 'an' 'ae' 'p'
 'bb']
X1 :['v' 'b' 'l' 's' 'aa' 'r' 'a' 'i' 'p' 'c' 'o' 'm' 'z' 'e' 'h' 'w' 'g' 'k'
 'y' 't' 'u' 'd' 'j' 'q' 'n' 'f' 'ab']
X2 :['n' 'ai' 'as' 'ae' 's' 'b' 'e' 'ak' 'm' 'a' 'aq' 'ag' 'r' 'k' 'aj' 'ay'
 'ao' 'an' 'ac' 'af' 'ax' 'h' 'i' 'f' 'ap' 'p' 'au' 't' 'z' 'y' 'aw' 'd'
 'at' 'g' 'am' 'j' 'x' 'ab' 'w' 'q' 'ah' 'ad' 'al' 'av' 'u']
X3 :['f' 'a' 'c' 'e' 'd' 'g' 'b']
X4 :['d' 'b' 'a' 'c']
X5 :['t' 'b' 'a' 'z' 'y' 'x' 'h' 'g' 'f' 'j' 'i' 'd' 'c' 'af' 'ag' 'ab' 'ac'
 'ad' 'ae' 'ah' 'l' 'k' 'n' 'm' 'p' 'q' 's' 'r' 'v' 'w' 'o' 'aa']
X6 :['a' 'g' 'j' 'l' 'i' 'd' 'f' 'h' 'c' 'k' 'e' 'b']
X8 :['w' 'y' 'j' 'n' 'm' 's' 'a' 'v' 'r' 'o' 't' 'h' 'c' 'k' 'p' 'u' 'd' 'g'
 'b' 'q' 'e' 'l' 'f' 'i' 'x']
```

10 Apply label encoder

[26]:
```
# for train data
for i in df_cat_train.columns:
    df_cat_train[i] = labelencoder.fit_transform(df_cat_train[i])
df_cat_train
```

[26]:
```
      X0  X1  X2  X3  X4  X5  X6  X8
0     32  23  16   0   3  24   9  14
1     32  21  18   4   3  28  11  14
2     20  24  33   2   3  27   9  23
3     20  21  33   5   3  27  11   4
4     20  23  33   5   3  12   3  13
…     ..  ..  ..  ..  ..  ..  ..  ..
4204   8  20  15   2   3   0   3  16
4205  31  16  39   3   3   0   7   7
4206   8  23  37   0   3   0   6   4
4207   9  19  24   5   3   0  11  20
4208  46  19   2   2   3   0   6  22

[4159 rows x 8 columns]
```

[27]:
```
# # for test data
for i in df_cat_test.columns:
    df_cat_test[i] = labelencoder.fit_transform(df_cat_test[i])
df_cat_test
```

[27]:
```
      X0  X1  X2  X3  X4  X5  X6  X8
0     21  23  34   5   3  26   0  22
1     42   3   8   0   3   9   6  24
2     21  23  17   5   3   0   9   9
3     21  13  34   5   3  31  11  13
4     45  20  17   2   3  30   8  12
…    ..  ..  ..  ..  ..  ..  ..  ..
4204   6   9  17   5   3   1   9   4
4205  42   1   8   3   3   1   9  24
4206  47  23  17   5   3   1   3  22
4207   7  23  17   0   3   1   2  16
4208  42   1   8   2   3   1   6  17

[4209 rows x 8 columns]
```

[28]:
```
newtrain1 = pd.concat([df_num_train,df_cat_train],axis=1)
newtest1 = pd.concat([df_num_test,df_cat_test],axis=1)
```

11 Divide Data into features and target

[29]:
```
features = newtrain1.drop('y',axis=1)
target = newtrain1['y']
# X_test = newtest1
```

12 Train & Validation Split

[30]:
```
X_train, X_test, y_train, y_test = train_test_split(features, target,␣
 ↪train_size = 0.70, random_state = 3)
print('Shape of X_train:',X_train.shape)
print('Shape of y_train:',y_train.shape)
print('Shape of X_test:',X_test.shape)
print('Shape of y_test:',y_test.shape)
```

```
Shape of X_train: (2911, 363)
Shape of y_train: (2911,)
Shape of X_test: (1248, 363)
Shape of y_test: (1248,)
```

13 Perform Dimensionality Reduction (PCA)

[31]:
```
pca = PCA(n_components=0.95)
```

[32]:
```
pca.fit(X_train)
```

[32]:
```
PCA(n_components=0.95)
```

[33]:
```
pca.explained_variance_ratio_
```

[33]:
```
array([0.38545701, 0.21176206, 0.13354339, 0.11817431, 0.09057302,
       0.01645431])
```

[34]:
```
np.sum(pca.explained_variance_ratio_)
```

[34]:
```
0.9559640947644873
```

[35]:
```
X_train_transformed = pca.transform(X_train)
X_test_transformed = pca.transform(X_test)
```

[36]:
```
pd.DataFrame(X_train_transformed)
```

[36]:
```
              0          1          2          3         4         5
0     23.561022  15.422601  -9.550391  -3.074573  3.436354  4.176733
1     -3.825024   0.776333   1.042777 -12.679915 -0.189836  3.678164
2    -14.442352  -5.225660  12.982459 -10.424020  9.888473  6.303518
3     -2.813386   3.263954  -0.921325  11.784038  7.090589 -3.781918
4      3.144741  -9.597645  12.576378  -5.245294 -8.473391  0.034307
…           …          …          …          …         …         …
2906  19.399721  -7.716407 -10.862942  -0.190089  1.057341 -1.199447
2907   5.954690  14.740205 -10.709911   2.746565 -6.159345  1.491938
2908  20.931689  -8.400373  -9.206365  -3.825675  9.945546 -2.348915
2909 -14.709168 -11.149776   3.800036   9.644795 -4.540294  5.558305
2910  21.838512  -7.953137  -4.700524  -9.153851  5.653786  0.459855

[2911 rows x 6 columns]
```

14 xgboost

[37]:
```
xgb_clf = xgboost.XGBRegressor(colsample_bytree= 0.7,
                               learning_rate= 0.03,
                               max_depth= 5,
                               min_child_weight= 4,
                               n_estimators= 200,
                               nthread= 4,
                                   subsample= 0.9)
```

[38]:
```
start = time.time()

xgb_clf.fit(X_train,y_train)

end = time.time()
time_elapsed = end - start
print('Time taken:',time_elapsed)

y_pred = xgb_clf.predict(X_test)
```

```
Time taken: 3.322930097579956
```

15 Check R Squared

[39]:
```
r2_score(y_test,y_pred)
```

[39]:
```
0.6334253267224981
```