ML Project 1 - Mercedes-Benz Greener Manufacturing

February 8, 2023

1 ML Project 1 - Mercedes-Benz Greener Manufacturing

[2]:
```
import pandas as pd
import numpy as np
```

[3]:
```
df_train = pd.read_csv('ml1train.csv')
```

[4]:
```
df_test = pd.read_csv('ml1test.csv')
```

[5]:
```
df_train.head()
```

[5]:
```
   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]
```

[6]:
```
df_train.shape
```

[6]:
```
(4209, 378)
```

[7]:
```
df_train.dtypes
```

[7]:
```
ID        int64
y       float64
X0       object
X1       object
X2       object
```

```
        …
X380      int64
X382      int64
X383      int64
X384      int64
X385      int64
Length: 378, dtype: object
```

[8]:
```
df_train.isnull().sum()
```

[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
```

[9]:
```
df_train.describe()
```

[9]:

	ID	y	X10	X11	X12	\
count	4209.000000	4209.000000	4209.000000	4209.0	4209.000000	
mean	4205.960798	100.669318	0.013305	0.0	0.075077	
std	2437.608688	12.679381	0.114590	0.0	0.263547	
min	0.000000	72.110000	0.000000	0.0	0.000000	
25%	2095.000000	90.820000	0.000000	0.0	0.000000	
50%	4220.000000	99.150000	0.000000	0.0	0.000000	
75%	6314.000000	109.010000	0.000000	0.0	0.000000	
max	8417.000000	265.320000	1.000000	0.0	1.000000	

	X13	X14	X15	X16	X17	…	\
count	4209.000000	4209.000000	4209.000000	4209.000000	4209.000000	…	
mean	0.057971	0.428130	0.000475	0.002613	0.007603	…	
std	0.233716	0.494867	0.021796	0.051061	0.086872	…	
min	0.000000	0.000000	0.000000	0.000000	0.000000	…	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	…	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	…	
75%	0.000000	1.000000	0.000000	0.000000	0.000000	…	
max	1.000000	1.000000	1.000000	1.000000	1.000000	…	

	X375	X376	X377	X378	X379	\
count	4209.000000	4209.000000	4209.000000	4209.000000	4209.000000	

```
mean      0.318841     0.057258     0.314802     0.020670     0.009503
std       0.466082     0.232363     0.464492     0.142294     0.097033
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       1.000000     0.000000     1.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

              X380         X382         X383         X384         X385
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000
mean      0.008078     0.007603     0.001663     0.000475     0.001426
std       0.089524     0.086872     0.040752     0.021796     0.037734
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

[8 rows x 370 columns]
```

[10]:
```
df_train = df_train.drop(['ID'],axis=1)
```

[11]:
```
df_train.head()
```

[11]:
```
        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]
```

[12]:
```
df_train.shape
```

[12]:
```
(4209, 377)
```

[13]:
```
df_cat = df_train.select_dtypes(include = np.object)
df_num = df_train.select_dtypes(exclude = np.object)
```

```
C:\Users\Vinosh\AppData\Local\Temp\ipykernel_18168\3134896722.py:1:
```

```
DeprecationWarning: `np.object` is a deprecated alias for the builtin `object`.
To silence this warning, use `object` by itself. Doing this will not modify any
behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  df_cat = df_train.select_dtypes(include = np.object)
C:\Users\Vinosh\AppData\Local\Temp\ipykernel_18168\3134896722.py:2:
DeprecationWarning: `np.object` is a deprecated alias for the builtin `object`.
To silence this warning, use `object` by itself. Doing this will not modify any
behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  df_num = df_train.select_dtypes(exclude = np.object)
```

[14]:
```
df_cat.head()
```

[14]:

	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

[15]:
```
df_num.head()
```

[15]:

	y	X10	X11	X12	X13	X14	X15	X16	X17	X18	…	X375	X376	X377	\
0	130.81	0	0	0	1	0	0	0	0	1	…	0	0	1	
1	88.53	0	0	0	0	0	0	0	0	1	…	1	0	0	
2	76.26	0	0	0	0	0	0	0	1	0	…	0	0	0	
3	80.62	0	0	0	0	0	0	0	0	0	…	0	0	0	
4	78.02	0	0	0	0	0	0	0	0	0	…	0	0	0	

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

[5 rows x 369 columns]

[16]:
```
df_num = df_num.drop(['y'],axis=1)
```

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

[17]:

	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	…	X375	X376	X377	\
0	0	0	0	1	0	0	0	0	1	0	…	0	0	1	
1	0	0	0	0	0	0	0	0	1	0	…	1	0	0	

```
2    0    0    0    0    0    0    0    1    0    0  …     0     0     0
3    0    0    0    0    0    0    0    0    0    0  …     0     0     0
4    0    0    0    0    0    0    0    0    0    0  …     0     0     0

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

[5 rows x 368 columns]
```

[18]:
```
columns = df_num.columns
```

[19]:
```
df_num.shape
```

[19]:
```
(4209, 368)
```

[20]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler
```

[21]:
```
mn = MinMaxScaler()
```

[22]:
```
df_mn = mn.fit_transform(df_num)
```

[23]:
```
df_num_sc = pd.DataFrame(df_mn, index=df_num.index,columns=df_num.columns)
```

[24]:
```
df_num_sc.head()
```

[24]:
```
   X10  X11  X12  X13  X14  X15  X16  X17  X18  X19  …  X375  X376  X377  \
0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  1.0  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  1.0  0.0  …   1.0   0.0   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
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
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

   X378  X379  X380  X382  X383  X384  X385
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
2   0.0   0.0   0.0   1.0   0.0   0.0   0.0
3   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 368 columns]
```

[25]:
```
variance_df_num = df_num.var()
```

[26]:
```
variable_var_zero = []

for i in range(0,len(variance_df_num)):
    if variance_df_num[i] == 0:
        variable_var_zero.append(columns[i])
```

[27]:
```
np.ravel(variable_var_zero)
```

[27]:
```
array(['X11', 'X93', 'X107', 'X233', 'X235', 'X268', 'X289', 'X290',
       'X293', 'X297', 'X330', 'X347'], dtype='<U4')
```

[28]:
```
df_num_variance_with_zero_drop = df_num.drop(['X11', 'X93', 'X107', 'X233',␣
 ↪'X235', 'X268', 'X289', 'X290',
       'X293', 'X297', 'X330', 'X347'],axis=1)
```

[29]:
```
df_num_variance_with_zero_drop.head()
```

[29]:

	X10	X12	X13	X14	X15	X16	X17	X18	X19	X20	…	X375	X376	X377	\
0	0	0	1	0	0	0	0	1	0	0	…	0	0	1	
1	0	0	0	0	0	0	0	1	0	0	…	1	0	0	
2	0	0	0	0	0	0	1	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	0	0	0	0	0	0	…	0	0	0	

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

[5 rows x 356 columns]

[30]:
```
df_num_variance_with_zero_drop.shape
```

[30]:
```
(4209, 356)
```

[31]:
```
df_num_variance_with_zero_drop.describe()
```

[31]:

	X10	X12	X13	X14	X15	\
count	4209.000000	4209.000000	4209.000000	4209.000000	4209.000000	
mean	0.013305	0.075077	0.057971	0.428130	0.000475	
std	0.114590	0.263547	0.233716	0.494867	0.021796	
min	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	0.000000	0.000000	0.000000	0.000000	0.000000	
50%	0.000000	0.000000	0.000000	0.000000	0.000000	
75%	0.000000	0.000000	0.000000	1.000000	0.000000	
max	1.000000	1.000000	1.000000	1.000000	1.000000	

```
              X16          X17          X18          X19          X20  …  \
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000  …
mean      0.002613     0.007603     0.007840     0.099549     0.142789  …
std       0.051061     0.086872     0.088208     0.299433     0.349899  …
min       0.000000     0.000000     0.000000     0.000000     0.000000  …
25%       0.000000     0.000000     0.000000     0.000000     0.000000  …
50%       0.000000     0.000000     0.000000     0.000000     0.000000  …
75%       0.000000     0.000000     0.000000     0.000000     0.000000  …
max       1.000000     1.000000     1.000000     1.000000     1.000000  …

              X375         X376         X377         X378         X379  \
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000
mean      0.318841     0.057258     0.314802     0.020670     0.009503
std       0.466082     0.232363     0.464492     0.142294     0.097033
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       1.000000     0.000000     1.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

              X380         X382         X383         X384         X385
count  4209.000000  4209.000000  4209.000000  4209.000000  4209.000000
mean      0.008078     0.007603     0.001663     0.000475     0.001426
std       0.089524     0.086872     0.040752     0.021796     0.037734
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       1.000000     1.000000     1.000000     1.000000     1.000000

[8 rows x 356 columns]
```

[32]:
```
df_train.shape
```

[32]:
```
(4209, 377)
```

[33]:
```
df_num.shape
```

[33]:
```
(4209, 368)
```

[34]:
```
df_cat.shape
```

[34]:
```
(4209, 8)
```

[35]:
```
df_train.nunique()
```

[35]:
```
y       2545
X0        47
X1        27
X2        44
X3         7
         …
X380       2
X382       2
X383       2
X384       2
X385       2
Length: 377, dtype: int64
```

[36]:
```
train_feature_names = df_train.values.ravel()
```

[37]:
```
train_unique_values = pd.unique(train_feature_names)
```

[38]:
```
train_unique_values
```

[38]:
```
array([130.81, 'k', 'v', …, 85.71, 108.77, 87.48], dtype=object)
```

[39]:
```
from sklearn.preprocessing import OneHotEncoder
```

[40]:
```
ohe = OneHotEncoder(handle_unknown='ignore')
```

[41]:
```
df_cat_dum = ohe.fit_transform(df_cat).toarray()
```

[42]:
```
col_names = ohe.get_feature_names()
```

```
C:\Users\Vinosh\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87:
FutureWarning: Function get_feature_names is deprecated; get_feature_names is
deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out
instead.
  warnings.warn(msg, category=FutureWarning)
```

[43]:
```
col_names = np.array(col_names).ravel()
```

[44]:
```
df_cat_oh = pd.DataFrame(df_cat_dum,columns=col_names)
```

[45]:
```
df_cat_oh.head()
```

[45]:

	x0_a	x0_aa	x0_ab	x0_ac	x0_ad	x0_af	x0_ai	x0_aj	x0_ak	x0_al	…	\
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	…	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	…	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	…	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	…	

```
   x7_p  x7_q  x7_r  x7_s  x7_t  x7_u  x7_v  x7_w  x7_x  x7_y
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
2   0.0   0.0   0.0   0.0   0.0   0.0   0.0   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
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 195 columns]
```

[46]:
```
df_cat_oh.shape
```

[46]:
```
(4209, 195)
```

[47]:
```
df_train_final = pd.concat([df_num_variance_with_zero_drop,df_cat_oh], axis=1)
```

[48]:
```
df_train_final.head()
```

[48]:
```
   X10  X12  X13  X14  X15  X16  X17  X18  X19  X20  …  x7_p  x7_q  x7_r  \
0    0    0    1    0    0    0    0    1    0    0  …   0.0   0.0   0.0
1    0    0    0    0    0    0    0    1    0    0  …   0.0   0.0   0.0
2    0    0    0    0    0    0    1    0    0    0  …   0.0   0.0   0.0
3    0    0    0    0    0    0    0    0    0    0  …   0.0   0.0   0.0
4    0    0    0    0    0    0    0    0    0    0  …   0.0   0.0   0.0

   x7_s  x7_t  x7_u  x7_v  x7_w  x7_x  x7_y
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
2   0.0   0.0   0.0   0.0   0.0   1.0   0.0
3   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 551 columns]
```

[49]:
```
df_train_final.shape
```

[49]:
```
(4209, 551)
```

[50]:
```
from sklearn.decomposition import PCA
```

[51]:
```
pca = PCA(n_components=25)
```

[52]:
```
x_pca = pca.fit_transform(df_train_final)
```

[53]:
```
df_pca = pd.DataFrame(x_pca)
```

[54]:
```
df_pca.head()
```

[54]:
```
          0         1         2         3         4         5         6   \
0  0.850248 -1.252515  2.021640  0.865224  1.592171 -0.056846  0.563839
1 -0.109302 -1.299662 -0.045801 -0.796931  0.277976  0.140880  1.108070
2 -0.673653 -2.367697  1.787792  2.345645  0.356806  3.753878 -1.188809
3 -0.480940 -2.695789  0.524340  2.881771 -0.485304  3.765186 -0.307380
4 -0.516369 -2.692792  0.334140  3.103397 -0.723453  3.866238 -0.451954

          7         8         9   …        15        16        17        18  \
0 -1.030708  0.205188 -0.264541  …  0.295399 -0.519060 -0.475316 -0.524516
1 -0.726633 -0.032185  0.612266  … -0.647857 -0.005580  0.096066  0.853962
2  0.679650 -0.924718 -0.215836  …  0.844761 -0.353158 -0.827082  0.561851
3 -0.014646 -1.239946  0.254648  …  0.359271  0.274448 -0.778751  0.822115
4  0.151801 -1.801270 -0.298125  … -0.216463 -0.090146 -0.205178  0.415562

          19        20        21        22        23        24
0  0.402055 -0.331132  1.112317 -0.203295  1.330357 -0.684745
1 -0.188607 -0.878619  0.620477 -0.344099  1.366941  0.030507
2  0.593954  0.883888 -0.554307  0.559493  0.674682 -0.043273
3  0.624387 -0.353793 -0.313110  0.247941 -0.215715 -1.146009
4  0.163971 -0.024790  0.409919  0.333957  0.269841 -0.290321

[5 rows x 25 columns]
```

[55]:
```
pca.explained_variance_ratio_
```

[55]:
```
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, 0.00984214, 0.00913174,
       0.00883412, 0.00843679, 0.00822998, 0.00772725, 0.00743289])
```

[56]:
```
df_test
```

[56]:
```
        ID  X0  X1  X2 X3 X4  X5 X6 X8  X10  …  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]
```

[57]:
```
df_test.isnull().sum()
```

[57]:
```
ID      0
X0      0
X1      0
X2      0
X3      0
       ..
X380    0
X382    0
X383    0
X384    0
X385    0
Length: 377, dtype: int64
```

[58]:
```
df_test.nunique()
```

[58]:
```
ID      4209
X0        49
X1        27
X2        45
X3         7
        …
X380       2
X382       2
X383       2
X384       2
X385       2
Length: 377, dtype: int64
```

[59]:
```
test_feature_names = df_test.values.ravel()
```

[60]:
```
test_unique_values = pd.unique(test_feature_names)
```

[61]:
```
test_unique_values
```

[61]:
```
array([1, 'az', 'v', …, 8413, 8414, 8416], dtype=object)
```

[62]:
```
df_test.shape
```

[62]:
```
(4209, 377)
```

[63]:
```
df_test.dtypes
```

[63]:
```
ID       int64
X0      object
X1      object
X2      object
X3      object
         …
X380     int64
X382     int64
X383     int64
X384     int64
X385     int64
Length: 377, dtype: object
```

[64]:
```
df_test_cat = df_test.select_dtypes(include = object)
df_test_num = df_test.select_dtypes(exclude = object)
```

[65]:
```
df_test_num.head()
```

[65]:

	ID	X10	X11	X12	X13	X14	X15	X16	X17	X18	…	X375	X376	X377	\
0	1	0	0	0	0	0	0	0	0	0	…	0	0	0	
1	2	0	0	0	0	0	0	0	0	0	…	0	0	1	
2	3	0	0	0	0	1	0	0	0	0	…	0	0	0	
3	4	0	0	0	0	0	0	0	0	0	…	0	0	0	
4	5	0	0	0	0	1	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 369 columns]

[66]:
```
df_test_cat.head()
```

[66]:

	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
```

[67]:
```
df_test_num = df_test_num.drop(['ID'],axis=1)
```

[68]:
```
df_test_num.head()
```

[68]:
```
   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]
```

[69]:
```
df_test_num.shape
```

[69]:
```
(4209, 368)
```

[70]:
```
test_columns = df_test_num.columns
```

[71]:
```
test_columns
```

[71]:
```
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)
```

[72]:
```
test_df_num_sc = mn.fit_transform(df_test_num)
```

[73]:
```
test_df_num_df = pd.DataFrame(test_df_num_sc,columns=df_test_num.
↪columns,index=df_test_num.index)
```

[74]:
```
test_df_num_df.head()
```

[74]:
```
   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  0.0  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  1.0  …   0.0   0.0   1.0
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
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
4  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  …   1.0   0.0   0.0

   X378  X379  X380  X382  X383  X384  X385
0   1.0   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
2   1.0   0.0   0.0   0.0   0.0   0.0   0.0
3   1.0   0.0   0.0   0.0   0.0   0.0   0.0
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 368 columns]
```

[75]:
```
test_df_num_df.shape
```

[75]:
```
(4209, 368)
```

[76]:
```
test_variance_df_num = df_test_num.var()
```

[77]:
```
test_variance_var_zero = []
```

[78]:
```
for i in range(0,len(test_variance_df_num)):
    if test_variance_df_num[i]==0:
        test_variance_var_zero.append(test_columns[i])
```

[79]:
```
test_variance_var_zero
```

[79]:
```
['X257', 'X258', 'X295', 'X296', 'X369']
```

[80]:
```
test_df_num_variance_with_zero_drop = df_test_num.drop(['X257', 'X258', 'X295',␣
↪'X296', 'X369'],axis=1)
```

[81]:
```
test_df_num_variance_with_zero_drop.head()
```

[81]:
```
   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 363 columns]

[82]:
```
test_df_num_variance_with_zero_drop.shape
```

[82]: (4209, 363)

[83]:
```
test_df_cat_dum = ohe.transform(df_test_cat).toarray()
```

[84]:
```
test_col_names = ohe.get_feature_names()
```

```
C:\Users\Vinosh\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87:
FutureWarning: Function get_feature_names is deprecated; get_feature_names is
deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out
instead.
  warnings.warn(msg, category=FutureWarning)
```

[85]:
```
test_col_names = np.array(test_col_names).ravel()
```

[86]:
```
test_df_cat_oh = pd.DataFrame(test_df_cat_dum,columns=test_col_names)
```

[87]:
```
test_df_cat_oh.head()
```

[87]:

	x0_a	x0_aa	x0_ab	x0_ac	x0_ad	x0_af	x0_ai	x0_aj	x0_ak	x0_al	…	\
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	…	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	…	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	…	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	…	

	x7_p	x7_q	x7_r	x7_s	x7_t	x7_u	x7_v	x7_w	x7_x	x7_y
0	0.0	0.0	0.0	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	0.0	1.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[5 rows x 195 columns]

[88]:
```
df_test_final = pd.
↪concat([test_df_num_variance_with_zero_drop,test_df_cat_oh],axis=1)
```

[89]:
```
df_test_final.head()
```

[89]:

	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	…	x7_p	x7_q	x7_r	\
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	1	…	0.0	0.0	0.0	
2	0	0	0	0	1	0	0	0	0	0	…	0.0	0.0	0.0	

```
3    0    0    0    0    0    0    0    0    0    0  …   0.0   0.0   0.0
4    0    0    0    0    1    0    0    0    0    0  …   0.0   0.0   0.0

   x7_s  x7_t  x7_u  x7_v  x7_w  x7_x  x7_y
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   1.0
2   0.0   0.0   0.0   0.0   0.0   0.0   0.0
3   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 558 columns]
```

[90]:
```
df_test_final.shape
```

[90]:
```
(4209, 558)
```

[91]:
```
df_train_final.shape
```

[91]:
```
(4209, 551)
```

[92]:
```
test_df_newdata = df_test_final.reindex(labels=df_train_final.columns,axis=1)
```

[93]:
```
test_df_newdata.head()
```

[93]:
```
   X10  X12  X13  X14  X15  X16  X17  X18  X19  X20  …  x7_p  x7_q  x7_r  \
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    1    0  …   0.0   0.0   0.0
2    0    0    0    1    0    0    0    0    0    0  …   0.0   0.0   0.0
3    0    0    0    0    0    0    0    0    0    0  …   0.0   0.0   0.0
4    0    0    0    1    0    0    0    0    0    0  …   0.0   0.0   0.0

   x7_s  x7_t  x7_u  x7_v  x7_w  x7_x  x7_y
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   1.0
2   0.0   0.0   0.0   0.0   0.0   0.0   0.0
3   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 551 columns]
```

[94]:
```
test_variance_var_zero
```

[94]:
```
['X257', 'X258', 'X295', 'X296', 'X369']
```

[95]:
```
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)
```

[96]:
```
test_df_newdata.head()
```

[96]:
```
   X10  X12  X13  X14  X15  X16  X17  X18  X19  X20  …  x7_p  x7_q  x7_r  \
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    1    0  …   0.0   0.0   0.0
2    0    0    0    1    0    0    0    0    0    0  …   0.0   0.0   0.0
3    0    0    0    0    0    0    0    0    0    0  …   0.0   0.0   0.0
4    0    0    0    1    0    0    0    0    0    0  …   0.0   0.0   0.0

   x7_s  x7_t  x7_u  x7_v  x7_w  x7_x  x7_y
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   1.0
2   0.0   0.0   0.0   0.0   0.0   0.0   0.0
3   0.0   0.0   0.0   0.0   0.0   0.0   0.0
4   0.0   0.0   0.0   0.0   0.0   0.0   0.0

[5 rows x 551 columns]
```

[97]:
```
test_x_pca = pca.transform(test_df_newdata)
```

[98]:
```
X_train = df_train_final
y_train = df_train['y']
```

[99]:
```
X_test = test_df_newdata
```

[100]:
```
from xgboost import XGBRegressor
```

[101]:
```
xgb = XGBRegressor()
```

[102]:
```
xgb.fit(X_train,y_train)
```

[102]:
```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=100, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, …)
```

[103]:
```
pred = xgb.predict(X_test)
```

[104]:
```
pred
```

[104]:
```
array([ 95.92638, 112.90855,  99.74303, …,  96.50017, 107.51481,
        90.8429 ], dtype=float32)
```

[105]:
```
df_res = pd.DataFrame(pred, columns=['Predicted'])
```

[106]:
```
df_res
```

[106]:

	Predicted
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 x 1 columns]

[108]:
```
df_res.head(10)
```

[108]:

	Predicted
0	95.926376
1	112.908546
2	99.743027
3	79.599861
4	112.196259
5	92.122078
6	109.448975
7	98.710701
8	117.565399
9	92.402451

[109]:
```
df_res.tail(10)
```

[109]:

	Predicted
4199	86.859184
4200	88.972832
4201	92.266197
4202	106.903854
4203	106.822060
4204	107.167992
4205	90.772079
4206	96.500168
4207	107.514809

```
4208   90.842903
```

[ ]: