PROJECT - MERCEDES BENZ GREENER MANUFACTURING

**# Importing Libraries**

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.decomposition import PCA

import xgboost as xgb

from sklearn.model\_selection import train\_test\_split as tts

from sklearn.metrics mean\_squared\_error as mse

from math import sqrt

import warnings

import seaborn as sns

warnings.filterwarnings('ignore')

**# Importing Datasets**

trn= pd.read\_csv('mercedestrain.csv')

tst= pd.read\_csv('mercedestest.csv')

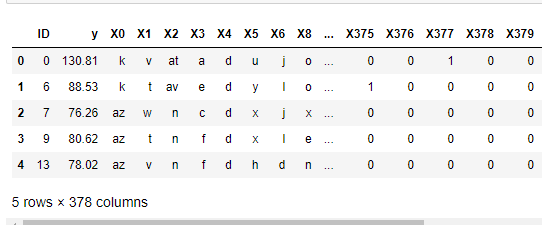
print(trn.shape)

print(tst.shape)

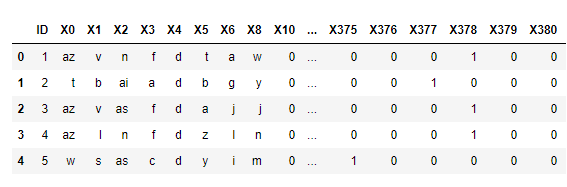
(4209, 378)

(4209, 377)

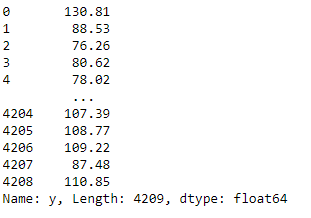
tr.head()



tst.head()



y = trn.y



**# dropping y and ID columns**

trn.drop(columns=['y','ID'],inplace = True)

tst.drop('ID',axis = 1,inplace = True)

print(y.shape)

print(trn.shape)

print(tst.shape)

(4209,)

(4209, 376)

(4209, 376)

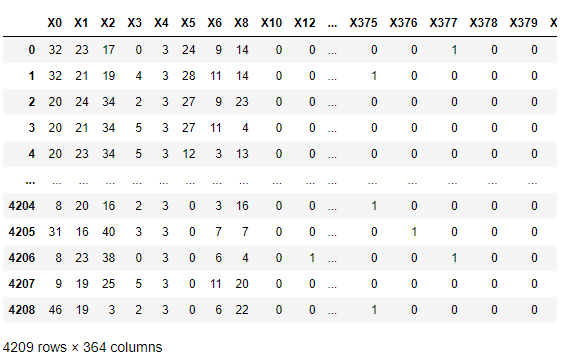
**# label encoding the categorical variables**

lbl = LabelEncoder()

for i in trn.loc[:,'X0':'X8']:

trn.loc[:,i] = lbl.fit\_transform(trn.loc[:,i])

trn



**# removing zero variant columns**

zero\_var = []

for i in trn.columns[8:] :

if trn[i].var()==0:

zero\_var.append(i)

zero\_var

['X11',

'X93',

'X107',

'X233',

'X235',

'X268',

'X289',

'X290',

'X293',

'X297',

'X330',

'X347']

trn.drop(zero\_var, axis = 1 , inplace =True)

**# working with pca**

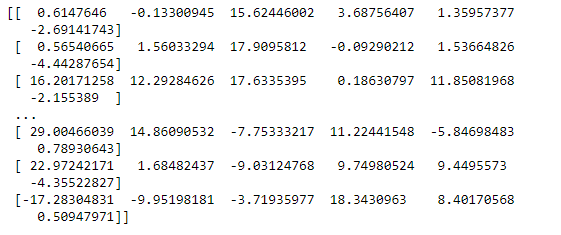
pca = PCA(n\_components = 0.95)

pca.fit(trn,y)

PCA(n\_components=0.95)

trnpca = pca.fit\_transform(trn)

print(trnpca)



trnpca.shape

(4209, 364)

**# splitting the dataset into train and test**

x\_train,x\_test,y\_train,y\_test = tts(trnpca,y,test\_size = 0.3,random\_state = 7)

print(x\_train.shape)

print(x\_test.shape)

print(y\_train.shape)

print(y\_test.shape)

(2946, 6)

(1263, 6)

(2946,)

(1263,)

**# working with XGBRegressor**

xgb\_reg = xgb.XGBRegressor(objective='reg:linear')

model = xgb\_reg.fit(x\_train,y\_train)

sqrt(mse(model.predict(x\_test),y\_test))

11.095226864027202

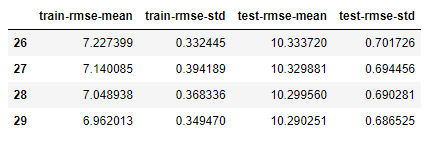
**# working with folds XGB Cross fold**

matrix = xgb.DMatrix(data =trnpca , label = y)

params = {'objective':'reg:linear'}

model\_cv = xgb.cv(dtrain= matrix , params=params, nfold=3, num\_boost\_round=50, early\_stopping\_rounds=10, metrics="rmse", as\_pandas=True, seed=7)

model\_cv.tail(4)



# prediction on test set

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

lbl = LabelEncoder()

for i in tst.loc[:,'X0':'X8']:

tst.loc[:,i] = lbl.fit\_transform(tst.loc[:,i])

tstpca = pca.fit\_transform(tst)

print(tstpca.shape)

tst\_pred = model.predict(tstpca)

tst\_pred

array([ 77.65116, 95.14801, 83.37289, ..., 100.17626, 120.56959,

95.79432], dtype=float32)