In [2]:

```
import numpy as np
import pandas as pd
from sklearn.decomposition import PCA
import xgboost as xgb
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
```

In [3]:

```
df_train = pd.read_csv(r"Z:\simp\ML\Mercedes-Benz_Greener_Manufacturing-master\train.csv")
df_train.head()
```

Out[3]:

| | ID | y | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 130.81 | k | v | at | a | d | u | j | o | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 6 | 88.53 | k | t | av | e | d | y | l | o | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 7 | 76.26 | az | w | n | c | d | x | j | x | ... | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| **3** | 9 | 80.62 | az | t | n | f | d | x | l | e | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 13 | 78.02 | az | v | n | f | d | h | d | n | ... | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 378 columns

In [4]:

```
df_test = pd.read_csv(r"Z:\simp\ML\Mercedes-Benz_Greener_Manufacturing-master\test.csv")
df_test.head()
```

Out[4]:

| | ID | X0 | X1 | X2 | X3 | X4 | X5 | X6 | X8 | X10 | ... | X375 | X376 | X377 | X378 | X379 | X380 | X382 | X383 | X384 | X385 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | az | v | n | f | d | t | a | w | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 2 | t | b | ai | a | d | b | g | y | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **2** | 3 | az | v | as | f | d | a | j | j | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **3** | 4 | az | l | n | f | d | z | l | n | 0 | ... | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 5 | w | s | as | c | d | y | i | m | 0 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

5 rows × 377 columns

In [5]:

```
y_train = df_train['y'].values
usable_columns = list(set(df_train.columns) - set(['ID', 'y']))
id_test = df_test['ID'].values
x_train = df_train[usable_columns]
x_test = df_test[usable_columns]
```

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

## Apply label encoder.

In [6]:

```
for column in usable_columns:
    cardinality = len(np.unique(x_train[column]))
    if cardinality == 1:
        x_train.drop(column, axis=1)
        x_test.drop(column, axis=1)
    if cardinality > 2:
        mapper = lambda x: sum([ord(digit) for digit in x])
        x_train[column] = x_train[column].apply(mapper)
        x_test[column] = x_test[column].apply(mapper)
x_train.head()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-
a-view-versus-a-copy)

C:\ProgramData\Anaconda3\lib\site-packages\ipykernel_launcher.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#re
turning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-
a-view-versus-a-copy)
  if __name__ == '__main__':
```

Out[6]:

| | X122 | X202 | X81 | X298 | X244 | X181 | X177 | X296 | X83 | X295 | ... | X75 | X47 | X358 | X336 | X147 | X21 | X266 | X34 | X158 | X92 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 0 | 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 | ... | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **4** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

5 rows × 376 columns

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

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

```
In [7]: def check_missing_values(df):
            if df.isnull().any().any():
                print("There are missing values in the dataframe")
            else:
                print("There are no missing values in the dataframe")
        check_missing_values(df_train)
        check_missing_values(df_test)
```

```
There are no missing values in the dataframe
There are no missing values in the dataframe
```

## Perform dimensionality reduction.

```
In [8]: pca = PCA(n_components=12, random_state=420)
        pca2_results_train = pca.fit_transform(x_train)
        pca2_results_test = pca.transform(x_test)
```

## Training using xgboost

In [9]:

```
x_train, x_valid, y_train, y_valid = train_test_split(pca2_results_train,y_train,test_size=0.2,random_state=4242

d_train = xgb.DMatrix(x_train, label=y_train)
d_valid = xgb.DMatrix(x_valid, label=y_valid)
d_test = xgb.DMatrix(pca2_results_test)

params = {}
params['objective'] = 'reg:linear'
params['eta'] = 0.02
params['max_depth'] = 4

def xgb_r2_score(preds, dtrain):
    labels = dtrain.get_label()
    return 'r2', r2_score(labels, preds)

watchlist = [(d_train, 'train'), (d_valid, 'valid')]

clf = xgb.train(params, d_train,
                1000, watchlist, early_stopping_rounds=50,
                feval=xgb_r2_score, maximize=True, verbose_eval=10)
```

```
[22:07:19] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.2.0/src/objective/regression_obj.
cu:174: reg:linear is now deprecated in favor of reg:squarederror.
[0]     train-rmse:99.14835     valid-rmse:98.26297     train-r2:-58.35295      valid-r2:-67.63754
Multiple eval metrics have been passed: 'valid-r2' will be used for early stopping.

Will train until valid-r2 hasn't improved in 50 rounds.
[10]    train-rmse:81.27651     valid-rmse:80.36433     train-r2:-38.88428      valid-r2:-44.91014
[20]    train-rmse:66.71610     valid-rmse:65.77334     train-r2:-25.87403      valid-r2:-29.75260
[30]    train-rmse:54.86956     valid-rmse:53.88973     train-r2:-17.17752      valid-r2:-19.64401
[40]    train-rmse:45.24491     valid-rmse:44.21971     train-r2:-11.35979      valid-r2:-12.89996
[50]    train-rmse:37.44729     valid-rmse:36.37237     train-r2:-7.46666       valid-r2:-8.40428
[60]    train-rmse:31.14750     valid-rmse:30.01872     train-r2:-4.85757       valid-r2:-5.40569
[70]    train-rmse:26.08663     valid-rmse:24.90881     train-r2:-3.10873       valid-r2:-3.41050
[80]    train-rmse:22.04641     valid-rmse:20.83260     train-r2:-1.93459       valid-r2:-2.08510
[90]    train-rmse:18.84416     valid-rmse:17.60380     train-r2:-1.14400       valid-r2:-1.20290
[100]   train-rmse:16.33653     valid-rmse:15.09088     train-r2:-0.61135       valid-r2:-0.61887
[110]   train-rmse:14.40015     valid-rmse:13.16003     train-r2:-0.25200       valid-r2:-0.23111
[120]   train-rmse:12.92382     valid-rmse:11.70215     train-r2:-0.00845       valid-r2:0.02655
[130]   train-rmse:11.81076     valid-rmse:10.62816     train-r2:0.15778        valid-r2:0.19703
[140]   train-rmse:10.98284     valid-rmse:9.86719      train-r2:0.27172        valid-r2:0.30790
```

```
[150]	train-rmse:10.37413	valid-rmse:9.33344	train-r2:0.35021	valid-r2:0.38075
[160]	train-rmse:9.92138	valid-rmse:8.97024	train-r2:0.40569	valid-r2:0.42801
[170]	train-rmse:9.58916	valid-rmse:8.72554	train-r2:0.44482	valid-r2:0.45879
[180]	train-rmse:9.33795	valid-rmse:8.57520	train-r2:0.47353	valid-r2:0.47728
[190]	train-rmse:9.15204	valid-rmse:8.47333	train-r2:0.49428	valid-r2:0.48962
[200]	train-rmse:9.00611	valid-rmse:8.40128	train-r2:0.51028	valid-r2:0.49827
[210]	train-rmse:8.90500	valid-rmse:8.36383	train-r2:0.52122	valid-r2:0.50273
[220]	train-rmse:8.82961	valid-rmse:8.33971	train-r2:0.52929	valid-r2:0.50559
[230]	train-rmse:8.76812	valid-rmse:8.33155	train-r2:0.53582	valid-r2:0.50656
[240]	train-rmse:8.72120	valid-rmse:8.31928	train-r2:0.54078	valid-r2:0.50801
[250]	train-rmse:8.68162	valid-rmse:8.31601	train-r2:0.54494	valid-r2:0.50840
[260]	train-rmse:8.63817	valid-rmse:8.31023	train-r2:0.54948	valid-r2:0.50908
[270]	train-rmse:8.61100	valid-rmse:8.31065	train-r2:0.55231	valid-r2:0.50903
[280]	train-rmse:8.57987	valid-rmse:8.30896	train-r2:0.55554	valid-r2:0.50923
[290]	train-rmse:8.55442	valid-rmse:8.30815	train-r2:0.55817	valid-r2:0.50933
[300]	train-rmse:8.52829	valid-rmse:8.30642	train-r2:0.56087	valid-r2:0.50953
[310]	train-rmse:8.50598	valid-rmse:8.30821	train-r2:0.56316	valid-r2:0.50932
[320]	train-rmse:8.47949	valid-rmse:8.30541	train-r2:0.56588	valid-r2:0.50965
[330]	train-rmse:8.45156	valid-rmse:8.30209	train-r2:0.56873	valid-r2:0.51004
[340]	train-rmse:8.43065	valid-rmse:8.30466	train-r2:0.57087	valid-r2:0.50974
[350]	train-rmse:8.40660	valid-rmse:8.30248	train-r2:0.57331	valid-r2:0.51000
[360]	train-rmse:8.38291	valid-rmse:8.30133	train-r2:0.57571	valid-r2:0.51013
[370]	train-rmse:8.35779	valid-rmse:8.29834	train-r2:0.57825	valid-r2:0.51049
[380]	train-rmse:8.33932	valid-rmse:8.29755	train-r2:0.58011	valid-r2:0.51058
[390]	train-rmse:8.31494	valid-rmse:8.29479	train-r2:0.58256	valid-r2:0.51091
[400]	train-rmse:8.28566	valid-rmse:8.29395	train-r2:0.58550	valid-r2:0.51100
[410]	train-rmse:8.26047	valid-rmse:8.29258	train-r2:0.58801	valid-r2:0.51117
[420]	train-rmse:8.23583	valid-rmse:8.29097	train-r2:0.59047	valid-r2:0.51136
[430]	train-rmse:8.20714	valid-rmse:8.28946	train-r2:0.59332	valid-r2:0.51153
[440]	train-rmse:8.18002	valid-rmse:8.28622	train-r2:0.59600	valid-r2:0.51192
[450]	train-rmse:8.15989	valid-rmse:8.28679	train-r2:0.59799	valid-r2:0.51185
[460]	train-rmse:8.13480	valid-rmse:8.28290	train-r2:0.60046	valid-r2:0.51231
[470]	train-rmse:8.11224	valid-rmse:8.28347	train-r2:0.60267	valid-r2:0.51224
[480]	train-rmse:8.08773	valid-rmse:8.28296	train-r2:0.60507	valid-r2:0.51230
[490]	train-rmse:8.06191	valid-rmse:8.28530	train-r2:0.60758	valid-r2:0.51202
[500]	train-rmse:8.04176	valid-rmse:8.28578	train-r2:0.60954	valid-r2:0.51197
[510]	train-rmse:8.02518	valid-rmse:8.28552	train-r2:0.61115	valid-r2:0.51200
[520]	train-rmse:8.00256	valid-rmse:8.28299	train-r2:0.61334	valid-r2:0.51230
[530]	train-rmse:7.98200	valid-rmse:8.28026	train-r2:0.61532	valid-r2:0.51262
[540]	train-rmse:7.96034	valid-rmse:8.27910	train-r2:0.61741	valid-r2:0.51275
[550]	train-rmse:7.94252	valid-rmse:8.27807	train-r2:0.61912	valid-r2:0.51287
[560]	train-rmse:7.92634	valid-rmse:8.27955	train-r2:0.62067	valid-r2:0.51270
[570]	train-rmse:7.90057	valid-rmse:8.28088	train-r2:0.62313	valid-r2:0.51255
```

```
[580]	train-rmse:7.88431	valid-rmse:8.27999	train-r2:0.62468	valid-r2:0.51265
[590]	train-rmse:7.86093	valid-rmse:8.28033	train-r2:0.62691	valid-r2:0.51261
[600]	train-rmse:7.83955	valid-rmse:8.27933	train-r2:0.62893	valid-r2:0.51273
Stopping. Best iteration:
[550]	train-rmse:7.94252	valid-rmse:8.27807	train-r2:0.61912	valid-r2:0.51287

[22:07:21] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.2.0/src/objective/regression_obj.
cu:174: reg:linear is now deprecated in favor of reg:squarederror.
```

# Predict your test_df values using XGBoost.

In [12]:
```
p_test = clf.predict(d_test)

sub = pd.DataFrame()
sub['ID'] = id_test
sub['y'] = p_test
sub.to_csv('predict.csv', index=False)

sub
```

Out[12]:
```
        ID           y
   0     1   82.695930
   1     2   97.201721
   2     3   83.188972
   3     4   76.915253
   4     5  112.542229
 ...   ...         ...
4204  8410  109.579132
4205  8411  100.319641
4206  8413   99.032097
4207  8414  106.875366
4208  8416   96.754906

4209 rows × 2 columns
```