In [1]:

```
# Step1: Import the required libraries

# linear algebra
import numpy as np
# data processing, CSV file I/O (e.g. pd.read_csv)
import pandas as pd
# for dimensionality reduction
from sklearn.decomposition import PCA
```

In [2]:

```
# Step2: Read the data from train.csv
df_train = pd.read_csv('train.csv')
df_train.head()
```

Out[2]:

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

5 rows × 378 columns

In [5]:

```
df_test = pd.read_csv('test.csv')
```

In [6]:

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

x_train = df_train[usable_columns]
x_test = df_test[usable_columns]
```

In [7]:

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

for column in usable_columns:
    cardinality = len(np.unique(x_train[column]))
    if cardinality == 1:
        x_train.drop(column, axis=1) # Column with only one
        # value is useless so we drop it
        x_test.drop(column, axis=1)
    if cardinality > 2: # Column is categorical
        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:\Users\Sumit\AppData\Local\Temp/ipykernel_10904/2608306690.py:13: SettingW
ithCopyWarning:
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#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  x_train[column] = x_train[column].apply(mapper)
C:\Users\Sumit\AppData\Local\Temp/ipykernel_10904/2608306690.py:14: SettingW
ithCopyWarning:
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#returning-a-view-versus-a-copy (https://pand
as.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-v
ersus-a-copy)
  x_test[column] = x_test[column].apply(mapper)
```

Out[7]:

	X183	X51	X151	X182	X10	X3	X104	X293	X350	X191	...	X50	X174	X333	X70	X9
0	0	0	0	0	0	97	0	0	0	0	...	0	0	0	1	
1	0	1	0	0	0	101	0	0	0	0	...	0	0	0	1	
2	0	1	0	0	0	99	0	0	1	0	...	0	0	0	1	
3	0	0	0	0	0	102	0	0	1	0	...	0	1	0	1	
4	0	1	0	0	0	102	0	0	1	0	...	0	0	0	1	

5 rows × 376 columns

In [8]:

```
# Perform dimensionality reduction
# Linear dimensionality reduction using Singular Value Decomposition of
# the data to project it to a lower dimensional space.
n_comp = 12
pca = PCA(n_components=n_comp, random_state=420)
pca2_results_train = pca.fit_transform(x_train)
pca2_results_test = pca.transform(x_test)
```

In [13]:

```
# Step11: Training using xgboost

import xgboost as xgb
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split

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(x_test)
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)
```

```
[19:17:53] WARNING: d:\bld\xgboost-split_1645118015404\work\src\objective
\regression_obj.cu:188: reg:linear is now deprecated in favor of reg:squar
ederror.
[0]     train-rmse:99.14835     train-r2:-58.35295      valid-rmse:98.2629
7       valid-r2:-67.63754
[10]    train-rmse:81.27653     train-r2:-38.88428      valid-rmse:80.3643
3       valid-r2:-44.91014
[20]    train-rmse:66.71610     train-r2:-25.87403      valid-rmse:65.7733
4       valid-r2:-29.75260
[30]    train-rmse:54.86956     train-r2:-17.17751      valid-rmse:53.8896
3       valid-r2:-19.64393
[40]    train-rmse:45.24492     train-r2:-11.35979      valid-rmse:44.2199
5       valid-r2:-12.90012
[50]    train-rmse:37.44735     train-r2:-7.46669       valid-rmse:36.3745
6       valid-r2:-8.40541
[60]    train-rmse:31.14759     train-r2:-4.85761       valid-rmse:30.0207
2       valid-r2:-5.40655
[70]    train-rmse:26.08677     train-r2:-3.10877       valid-rmse:24.9106
2       valid-r2:-3.41114
[80]    train-rmse:22.04666     train-r2:-1.93465       valid-rmse:20.8324
0       valid-r2:-2.08504
[90]    train-rmse:18.84413     train-r2:-1.14399       valid-rmse:17.6057
2       valid-r2:-1.20338
[100]   train-rmse:16.34036     train-r2:-0.61211       valid-rmse:15.0841
7       valid-r2:-0.61743
[110]   train-rmse:14.40185     train-r2:-0.25230       valid-rmse:13.1489
5       valid-r2:-0.22903
[120]   train-rmse:12.92203     train-r2:-0.00817       valid-rmse:11.6896
```

```
9       valid-r2:0.02862
[130]   train-rmse:11.81350     train-r2:0.15738        valid-rmse:10.6155
0       valid-r2:0.19894
[140]   train-rmse:10.98284     train-r2:0.27172        valid-rmse:9.84830
valid-r2:0.31055
[150]   train-rmse:10.37527     train-r2:0.35007        valid-rmse:9.31465
valid-r2:0.38324
[160]   train-rmse:9.93136      train-r2:0.40449        valid-rmse:8.95036
valid-r2:0.43054
[170]   train-rmse:9.59197      train-r2:0.44450        valid-rmse:8.71045
valid-r2:0.46066
[180]   train-rmse:9.34686      train-r2:0.47252        valid-rmse:8.55318
valid-r2:0.47996
[190]   train-rmse:9.15743      train-r2:0.49369        valid-rmse:8.44958
valid-r2:0.49248
[200]   train-rmse:9.01297      train-r2:0.50954        valid-rmse:8.38462
valid-r2:0.50026
[210]   train-rmse:8.90998      train-r2:0.52068        valid-rmse:8.34134
valid-r2:0.50540
[220]   train-rmse:8.83071      train-r2:0.52917        valid-rmse:8.32256
valid-r2:0.50763
[230]   train-rmse:8.76606      train-r2:0.53604        valid-rmse:8.31029
valid-r2:0.50908
[240]   train-rmse:8.72189      train-r2:0.54070        valid-rmse:8.30562
valid-r2:0.50963
[250]   train-rmse:8.68375      train-r2:0.54471        valid-rmse:8.30231
valid-r2:0.51002
[260]   train-rmse:8.64870      train-r2:0.54838        valid-rmse:8.29922
valid-r2:0.51038
[270]   train-rmse:8.61395      train-r2:0.55200        valid-rmse:8.29619
valid-r2:0.51074
[280]   train-rmse:8.58595      train-r2:0.55491        valid-rmse:8.29806
valid-r2:0.51052
[290]   train-rmse:8.55738      train-r2:0.55787        valid-rmse:8.29592
valid-r2:0.51077
[300]   train-rmse:8.53586      train-r2:0.56009        valid-rmse:8.29735
valid-r2:0.51060
[310]   train-rmse:8.51569      train-r2:0.56216        valid-rmse:8.29896
valid-r2:0.51041
[320]   train-rmse:8.48662      train-r2:0.56515        valid-rmse:8.29763
valid-r2:0.51057
[330]   train-rmse:8.46170      train-r2:0.56770        valid-rmse:8.29202
valid-r2:0.51123
[340]   train-rmse:8.43840      train-r2:0.57008        valid-rmse:8.29166
valid-r2:0.51128
[350]   train-rmse:8.41204      train-r2:0.57276        valid-rmse:8.29100
valid-r2:0.51135
[360]   train-rmse:8.38977      train-r2:0.57502        valid-rmse:8.28751
valid-r2:0.51176
[370]   train-rmse:8.36800      train-r2:0.57722        valid-rmse:8.28999
valid-r2:0.51147
[380]   train-rmse:8.34389      train-r2:0.57965        valid-rmse:8.28952
valid-r2:0.51153
[390]   train-rmse:8.31818      train-r2:0.58224        valid-rmse:8.28946
valid-r2:0.51153
[400]   train-rmse:8.28785      train-r2:0.58528        valid-rmse:8.28059
valid-r2:0.51258
[410]   train-rmse:8.26540      train-r2:0.58752        valid-rmse:8.27831
valid-r2:0.51285
[420]   train-rmse:8.24268      train-r2:0.58979        valid-rmse:8.27517
valid-r2:0.51322
```

```
[430]	train-rmse:8.21670	train-r2:0.59237	valid-rmse:8.27563
valid-r2:0.51316
[440]	train-rmse:8.18860	train-r2:0.59515	valid-rmse:8.27617
valid-r2:0.51310
[450]	train-rmse:8.16267	train-r2:0.59771	valid-rmse:8.27586
valid-r2:0.51314
[460]	train-rmse:8.13848	train-r2:0.60009	valid-rmse:8.27684
valid-r2:0.51302
[470]	train-rmse:8.11594	train-r2:0.60231	valid-rmse:8.27722
valid-r2:0.51297
[472]	train-rmse:8.11375	train-r2:0.60252	valid-rmse:8.27739
valid-r2:0.51296
```

In [14]:

```
p_test = clf.predict(d_test)

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

sub.head()
```

Out[14]:

	ID	y
0	1	83.168739
1	2	97.533386
2	3	83.400864
3	4	77.144096
4	5	112.598930

In [12]:

```
```

In [ ]:

```
```