```
#Import the required libraries
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA

#Read the data from train.csv
df_train = pd.read_csv('train.csv')

#Let us understand the data
print(df_train.head())
print(df_train.shape)
```

```
   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]
(4209, 378)
```

```
#Let us collect the y values into an array
#Seperate the y value from the data as we will use this value to learn
as prediction output
y_train = df_train['y'].values
y_train
```

```
array([130.81,  88.53,  76.26, ..., 109.22,  87.48, 110.85])
```

```
#Let us understand the data types we have

#Iterate through all the columns which has X in the name of the
columns
cols = [c for c in df_train.columns if 'X' in c]
print('Number of features: {}'.format(len(cols)))
```

```
print('Feature types')
df_train[cols].dtypes.value_counts()
```

```
Number of features: 376
Feature types

int64     368
object      8
dtype: int64
```

```
#Let us count the data in each column

counts = [[],[],[]]

for c in cols:
    typ = df_train[c].dtype
    uniq = len(np.unique(df_train[c]))
    if uniq == 1:
        counts[0].append(c)
    elif uniq == 2:
        counts[1].append(c)
    else:
        counts[2].append(c)

print('Constant features: {} Binary features: {} Categorical features:
{}\n'
      .format(*[len(c) for c in counts]))
print('Constant features:', counts[0])
print('Categorical features:', counts[2])
```

```
Constant features: 12 Binary features: 356 Categorical features: 8

Constant features: ['X11', 'X93', 'X107', 'X233', 'X235', 'X268',
'X289', 'X290', 'X293', 'X297', 'X330', 'X347']
Categorical features: ['X0', 'X1', 'X2', 'X3', 'X4', 'X5', 'X6', 'X8']
```

```
#Read the Test data
df_test = pd.read_csv('test.csv')

#Remove columns 'ID' and 'y' as they are not used for learning
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]

#Check for Null and unique values for train and test data set
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
```

```
#If for any of the columns variance is zero then we need to remove
those variables
#Apply LabelEncoder
for column in usable_columns:
    cardinality = len(np.unique(x_train[column]))
    if cardinality == 1:
        #Column with only one value is useless so we drop it
        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()
```

```
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/
ipykernel_2379/1425202713.py:11: 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#
returning-a-view-versus-a-copy
  x_train[column] = x_train[column].apply(mapper)
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:12: 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#
returning-a-view-versus-a-copy
  x_test[column] = x_test[column].apply(mapper)
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:11: 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#
returning-a-view-versus-a-copy
  x_train[column] = x_train[column].apply(mapper)
```

```
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:12: 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#
returning-a-view-versus-a-copy
  x_test[column] = x_test[column].apply(mapper)
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:11: 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#
returning-a-view-versus-a-copy
  x_train[column] = x_train[column].apply(mapper)
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:12: 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#
returning-a-view-versus-a-copy
  x_test[column] = x_test[column].apply(mapper)
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:11: 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#
returning-a-view-versus-a-copy
  x_train[column] = x_train[column].apply(mapper)
/var/folders/3_/_g3bl5vs0xlgybykjk6_xvwh0000gn/T/ipykernel_2379/142520
2713.py:12: 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#
returning-a-view-versus-a-copy
  x_test[column] = x_test[column].apply(mapper)

   X210  X116  X94  X76  X368  X141  X56  X346  X98  X44  ...  X102
X35  X17  \
0     0     1    0    0     0     0    0     0    0    0  ...     0
1    0
```

```
1     0     0    0    0     0     0    0     0    1    0  ...     0
1    0
2     0     0    0    1     0     0    0     0    1    0  ...     0
1    1
3     0     0    0    1     0     0    0     0    1    0  ...     0
1    0
4     0     0    0    1     0     0    0     0    1    0  ...     0
1    0

   X82  X385  X364  X80  X286  X47  X106
0    0     0     0    0     0    0     0
1    0     0     0    1     0    0     0
2    0     0     0    1     1    0     0
3    0     0     0    1     1    0     0
4    0     0     0    1     1    0     0

[5 rows x 376 columns]
```

```
#All Data is now changed to numericals
print('Feature types', x_train[cols].dtypes.value_counts())
```

```
Feature types int64    376
dtype: int64
```

```
#Performing dimensionality reduction using pricipal component analysis
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)
```

```
#Training using xgboost

import sys
!{sys.executable} -m pip install xgboost
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)
```

```
Requirement already satisfied: xgboost in
/Users/glikithvinayaka/opt/anaconda3/lib/python3.9/site-packages
(1.6.1)
Requirement already satisfied: numpy in
/Users/glikithvinayaka/opt/anaconda3/lib/python3.9/site-packages (from
xgboost) (1.21.5)
Requirement already satisfied: scipy in
/Users/glikithvinayaka/opt/anaconda3/lib/python3.9/site-packages (from
xgboost) (1.7.3)
```

```
import xgboost as xgb

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)
```

```
[16:00:59] WARNING: /Users/runner/work/xgboost/xgboost/python-
package/build/temp.macosx-10.9-x86_64-3.7/xgboost/src/objective/
regression_obj.cu:203: reg:linear is now deprecated in favor of
reg:squarederror.
[0]   train-rmse:99.14834   train-r2:-58.35295    valid-rmse:98.26297
      valid-r2:-67.63754
[10]  train-rmse:81.27653   train-r2:-38.88428    valid-rmse:80.36433
      valid-r2:-44.91014
[20]  train-rmse:66.71610   train-r2:-25.87403    valid-rmse:65.77334
      valid-r2:-29.75260
[30]  train-rmse:54.86915   train-r2:-17.17724    valid-rmse:53.89120
      valid-r2:-19.64513
[40]  train-rmse:45.24564   train-r2:-11.36018    valid-rmse:44.22231
      valid-r2:-12.90160
[50]  train-rmse:37.44741   train-r2:-7.46672     valid-rmse:36.37758
      valid-r2:-8.40697

/Users/glikithvinayaka/opt/anaconda3/lib/python3.9/site-packages/
xgboost/core.py:525: FutureWarning: Pass `evals` as keyword args.
Passing these as positional arguments will be considered as error in
future releases.
  warnings.warn(
/Users/glikithvinayaka/opt/anaconda3/lib/python3.9/site-packages/xgboo
st/training.py:35: UserWarning: `feval` is deprecated, use
`custom_metric` instead.  They have different behavior when custom
objective is also used.See
https://xgboost.readthedocs.io/en/latest/tutorials/custom_metric_obj.h
```

```
tml for details on the `custom_metric`.
  warnings.warn(

[60]  train-rmse:31.15105   train-r2:-4.85891    valid-rmse:30.01771
      valid-r2:-5.40526
[70]  train-rmse:26.08767   train-r2:-3.10906    valid-rmse:24.90843
      valid-r2:-3.41037
[80]  train-rmse:22.04897   train-r2:-1.93527    valid-rmse:20.82554
      valid-r2:-2.08301
[90]  train-rmse:18.84730   train-r2:-1.14472    valid-rmse:17.59520
      valid-r2:-1.20075
[100] train-rmse:16.33662   train-r2:-0.61137    valid-rmse:15.07626
      valid-r2:-0.61573
[110] train-rmse:14.39752   train-r2:-0.25155    valid-rmse:13.14546
      valid-r2:-0.22838
[120] train-rmse:12.92282   train-r2:-0.00829    valid-rmse:11.68457
      valid-r2:0.02947
[130] train-rmse:11.81491   train-r2:0.15718 valid-rmse:10.61418
      valid-r2:0.19914
[140] train-rmse:10.98412   train-r2:0.27155 valid-rmse:9.85128
      valid-r2:0.31013
[150] train-rmse:10.37286   train-r2:0.35037 valid-rmse:9.31602
      valid-r2:0.38306
[160] train-rmse:9.92385    train-r2:0.40539 valid-rmse:8.95286
      valid-r2:0.43022
[170] train-rmse:9.59446    train-r2:0.44421 valid-rmse:8.70782
      valid-r2:0.46098
[180] train-rmse:9.35086    train-r2:0.47207 valid-rmse:8.54704
      valid-r2:0.48071
[190] train-rmse:9.16592    train-r2:0.49275 valid-rmse:8.44115
      valid-r2:0.49349
[200] train-rmse:9.02125    train-r2:0.50863 valid-rmse:8.37700
      valid-r2:0.50116
[210] train-rmse:8.91196    train-r2:0.52047 valid-rmse:8.34192
      valid-r2:0.50533
[220] train-rmse:8.82786    train-r2:0.52948 valid-rmse:8.31911
      valid-r2:0.50803
[230] train-rmse:8.76426    train-r2:0.53623 valid-rmse:8.30225
      valid-r2:0.51002
[240] train-rmse:8.71793    train-r2:0.54112 valid-rmse:8.29535
      valid-r2:0.51084
[250] train-rmse:8.67415    train-r2:0.54572 valid-rmse:8.29048
      valid-r2:0.51141
[260] train-rmse:8.63753    train-r2:0.54955 valid-rmse:8.28707
      valid-r2:0.51182
[270] train-rmse:8.60657    train-r2:0.55277 valid-rmse:8.28586
      valid-r2:0.51196
[280] train-rmse:8.57964    train-r2:0.55556 valid-rmse:8.28621
      valid-r2:0.51192
[290] train-rmse:8.54983    train-r2:0.55865 valid-rmse:8.28461
```

```
      valid-r2:0.51211
[300] train-rmse:8.52775    train-r2:0.56092 valid-rmse:8.28311
      valid-r2:0.51228
[310] train-rmse:8.49954    train-r2:0.56382 valid-rmse:8.28367
      valid-r2:0.51222
[320] train-rmse:8.47391    train-r2:0.56645 valid-rmse:8.28258
      valid-r2:0.51234
[330] train-rmse:8.45065    train-r2:0.56883 valid-rmse:8.27759
      valid-r2:0.51293
[340] train-rmse:8.41910    train-r2:0.57204 valid-rmse:8.27638
      valid-r2:0.51307
[350] train-rmse:8.39611    train-r2:0.57437 valid-rmse:8.27474
      valid-r2:0.51327
[360] train-rmse:8.36699    train-r2:0.57732 valid-rmse:8.27190
      valid-r2:0.51360
[370] train-rmse:8.34324    train-r2:0.57972 valid-rmse:8.27179
      valid-r2:0.51361
[380] train-rmse:8.31800    train-r2:0.58226 valid-rmse:8.27477
      valid-r2:0.51326
[390] train-rmse:8.29621    train-r2:0.58444 valid-rmse:8.27299
      valid-r2:0.51347
[400] train-rmse:8.27222    train-r2:0.58684 valid-rmse:8.27025
      valid-r2:0.51380
[410] train-rmse:8.24145    train-r2:0.58991 valid-rmse:8.27188
      valid-r2:0.51360
[420] train-rmse:8.21912    train-r2:0.59213 valid-rmse:8.27078
      valid-r2:0.51373
[430] train-rmse:8.19356    train-r2:0.59466 valid-rmse:8.26983
      valid-r2:0.51384
[440] train-rmse:8.16556    train-r2:0.59743 valid-rmse:8.26848
      valid-r2:0.51400
[450] train-rmse:8.14935    train-r2:0.59903 valid-rmse:8.26765
      valid-r2:0.51410
[460] train-rmse:8.12534    train-r2:0.60138 valid-rmse:8.26885
      valid-r2:0.51396
[470] train-rmse:8.09629    train-r2:0.60423 valid-rmse:8.26988
      valid-r2:0.51384
[480] train-rmse:8.07847    train-r2:0.60597 valid-rmse:8.27178
      valid-r2:0.51362
[490] train-rmse:8.05210    train-r2:0.60854 valid-rmse:8.27163
      valid-r2:0.51363
[500] train-rmse:8.03028    train-r2:0.61066 valid-rmse:8.27094
      valid-r2:0.51371
[503] train-rmse:8.02424    train-r2:0.61124 valid-rmse:8.27013
      valid-r2:0.51381
```

```
#Predict the test_df values using xgboost
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()
```

	ID	y
0	1	82.688263
1	2	97.398170
2	3	83.233788
3	4	77.283653
4	5	112.651413