In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

pd.pandas.set_option('display.max_columns',None)

import warnings
warnings.filterwarnings("ignore")
```

In [21]:

```
train_df = pd.read_csv("train.csv")
test_df = pd.read_csv("test.csv")
```

In [4]:

```
train_df.head()
```

Out[4]:

	ID	y	X0	X1	X2	X3	X4	X5	X6	X8	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X26	X27	X28
0	0	130.81	k	v	at	a	d	u	j	o	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0
1	6	88.53	k	t	av	e	d	y	l	o	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
2	7	76.26	az	w	n	c	d	x	j	x	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
3	9	80.62	az	t	n	f	d	x	l	e	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	13	78.02	az	v	n	f	d	h	d	n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

In [5]:

```
print("The training data has {} rows and {} columns".format(*train_df.shape))
```

```
The training data has 4209 rows and 378 columns
```

In [7]:

```
y_train = train_df["y"].values
```

In [12]:

```
columns = [c for c in train_df.columns if "X" in c]
```

In [15]:

```
print("The total features are {}".format(len(columns)))
```

```
The total features are 376
```

In [20]:

```
train_df[columns].dtypes.value_counts()
```

Out[20]:

```
int64     368
object      8
dtype: int64
```

In [22]:

```
columns_usable = list(set(train_df.columns) - set(['ID', 'y']))
```

In [25]:

```
y_train = train_df['y'].values
id_test = test_df['ID'].values

X_train = train_df[columns_usable]
X_test = test_df[columns_usable]
```

Checking for Null values and Unique Values

In [26]:

```
def null_fn(df):
    if df.isna().sum().any()== True:
        print(" There are missing values")
    else:
        print("There are no missing values")
```

In [27]:

```
null_fn(X_train)
```

```
There are no missing values
```

In [28]:

```
null_fn(X_test)
```

```
There are no missing values
```

In [47]:

```python
for column in train_df.columns:
    print(train_df[column].name,train_df[column].unique())
```

```
ID [   0    6    7 ... 8412 8415 8417]
y [130.81  88.53  76.26 ...  85.71 108.77  87.48]
X0 ['k' 'az' 't' 'al' 'o' 'w' 'j' 'h' 's' 'n' 'ay' 'f' 'x' 'y' 'aj' 'ak' 'am'
 'z' 'q' 'at' 'ap' 'v' 'af' 'a' 'e' 'ai' 'd' 'aq' 'c' 'aa' 'ba' 'as' 'i'
 'r' 'b' 'ax' 'bc' 'u' 'ad' 'au' 'm' 'l' 'aw' 'ao' 'ac' 'g' 'ab']
X1 ['v' 't' 'w' 'b' 'r' 'l' 's' 'aa' 'c' 'a' 'e' 'h' 'z' 'j' 'o' 'u' 'p' 'n'
 'i' 'y' 'd' 'f' 'm' 'k' 'g' 'q' 'ab']
X2 ['at' 'av' 'n' 'e' 'as' 'aq' 'r' 'ai' 'ak' 'm' 'a' 'k' 'ae' 's' 'f' 'd'
 'ag' 'ay' 'ac' 'ap' 'g' 'i' 'aw' 'y' 'b' 'ao' 'al' 'h' 'x' 'au' 't' 'an'
 'z' 'ah' 'p' 'am' 'j' 'q' 'af' 'l' 'aa' 'c' 'o' 'ar']
X3 ['a' 'e' 'c' 'f' 'd' 'b' 'g']
X4 ['d' 'b' 'c' 'a']
X5 ['u' 'y' 'x' 'h' 'g' 'f' 'j' 'i' 'd' 'c' 'af' 'ag' 'ab' 'ac' 'ad' 'ae'
 'ah' 'l' 'k' 'n' 'm' 'p' 'q' 's' 'r' 'v' 'w' 'o' 'aa']
X6 ['j' 'l' 'd' 'h' 'i' 'a' 'g' 'c' 'k' 'e' 'f' 'b']
X8 ['o' 'x' 'e' 'n' 's' 'a' 'h' 'p' 'm' 'k' 'd' 'i' 'v' 'j' 'b' 'q' 'w' 'g'
 'y' 'l' 'f' 'u' 'r' 't' 'c']
X10 [0 1]
X11 [0]
X12 [0 1]
X13 [1 0]
X14 [0 1]
X15 [0 1]
X16 [0 1]
X17 [0 1]
X18 [1 0]
X19 [0 1]
X20 [0 1]
X21 [1 0]
X22 [0 1]
X23 [0 1]
X24 [0 1]
X26 [0 1]
X27 [0 1]
X28 [0 1]
X29 [0 1]
X30 [0 1]
X31 [1 0]
X32 [0 1]
```

```
X33 [0 1]
X34 [0 1]
X35 [1 0]
X36 [0 1]
X37 [1 0]
X38 [0 1]
X39 [0 1]
X40 [0 1]
X41 [0 1]
X42 [0 1]
X43 [0 1]
X44 [0 1]
X45 [0 1]
X46 [1 0]
X47 [0 1]
X48 [0 1]
X49 [0 1]
X50 [0 1]
X51 [0 1]
X52 [0 1]
X53 [0 1]
X54 [0 1]
X55 [0 1]
X56 [0 1]
X57 [0 1]
X58 [1 0]
X59 [0 1]
X60 [0 1]
X61 [0 1]
X62 [0 1]
X63 [0 1]
X64 [0 1]
X65 [0 1]
X66 [0 1]
X67 [0 1]
X68 [1 0]
X69 [0 1]
X70 [1 0]
X71 [0 1]
X73 [0 1]
X74 [1 0]
X75 [0 1]
X76 [0 1]
X77 [0 1]
X78 [0 1]
```

```
X79 [0 1]
X80 [0 1]
X81 [0 1]
X82 [0 1]
X83 [0 1]
X84 [0 1]
X85 [1 0]
X86 [0 1]
X87 [0 1]
X88 [0 1]
X89 [0 1]
X90 [0 1]
X91 [0 1]
X92 [0 1]
X93 [0]
X94 [0 1]
X95 [0 1]
X96 [0 1]
X97 [0 1]
X98 [0 1]
X99 [0 1]
X100 [0 1]
X101 [0 1]
X102 [0 1]
X103 [0 1]
X104 [0 1]
X105 [0 1]
X106 [0 1]
X107 [0]
X108 [0 1]
X109 [0 1]
X110 [0 1]
X111 [1 0]
X112 [0 1]
X113 [0 1]
X114 [1 0]
X115 [0 1]
X116 [1 0]
X117 [0 1]
X118 [1 0]
X119 [1 0]
X120 [1 0]
X122 [0 1]
X123 [0 1]
X124 [0 1]
```

```
X125 [0 1]
X126 [0 1]
X127 [0 1]
X128 [1 0]
X129 [0 1]
X130 [0 1]
X131 [1 0]
X132 [0 1]
X133 [0 1]
X134 [0 1]
X135 [0 1]
X136 [1 0]
X137 [1 0]
X138 [0 1]
X139 [0 1]
X140 [0 1]
X141 [0 1]
X142 [1 0]
X143 [0 1]
X144 [1 0]
X145 [0 1]
X146 [0 1]
X147 [0 1]
X148 [0 1]
X150 [1 0]
X151 [0 1]
X152 [0 1]
X153 [0 1]
X154 [0 1]
X155 [0 1]
X156 [1 0]
X157 [0 1]
X158 [0 1]
X159 [0 1]
X160 [0 1]
X161 [0 1]
X162 [0 1]
X163 [0 1]
X164 [0 1]
X165 [0 1]
X166 [0 1]
X167 [0 1]
X168 [0 1]
X169 [0 1]
X170 [1 0]
```

```
X171 [0 1]
X172 [0 1]
X173 [0 1]
X174 [0 1]
X175 [0 1]
X176 [0 1]
X177 [0 1]
X178 [0 1]
X179 [1 0]
X180 [0 1]
X181 [0 1]
X182 [0 1]
X183 [0 1]
X184 [1 0]
X185 [0 1]
X186 [0 1]
X187 [1 0]
X189 [1 0]
X190 [0 1]
X191 [0 1]
X192 [0 1]
X194 [1 0]
X195 [0 1]
X196 [0 1]
X197 [0 1]
X198 [0 1]
X199 [0 1]
X200 [0 1]
X201 [0 1]
X202 [0 1]
X203 [0 1]
X204 [1 0]
X205 [0 1]
X206 [0 1]
X207 [0 1]
X208 [0 1]
X209 [1 0]
X210 [0 1]
X211 [0 1]
X212 [0 1]
X213 [0 1]
X214 [0 1]
X215 [0 1]
X216 [0 1]
X217 [0 1]
```

```
X218 [0 1]
X219 [0 1]
X220 [1 0]
X221 [0 1]
X222 [0 1]
X223 [0 1]
X224 [0 1]
X225 [0 1]
X226 [0 1]
X227 [0 1]
X228 [0 1]
X229 [0 1]
X230 [0 1]
X231 [0 1]
X232 [0 1]
X233 [0]
X234 [1 0]
X235 [0]
X236 [0 1]
X237 [1 0]
X238 [0 1]
X239 [0 1]
X240 [0 1]
X241 [0 1]
X242 [0 1]
X243 [0 1]
X244 [0 1]
X245 [0 1]
X246 [0 1]
X247 [0 1]
X248 [0 1]
X249 [0 1]
X250 [0 1]
X251 [0 1]
X252 [0 1]
X253 [0 1]
X254 [0 1]
X255 [0 1]
X256 [0 1]
X257 [0 1]
X258 [0 1]
X259 [0 1]
X260 [0 1]
X261 [0 1]
X262 [1 0]
```

```
X263 [1 0]
X264 [0 1]
X265 [0 1]
X266 [1 0]
X267 [0 1]
X268 [0]
X269 [0 1]
X270 [0 1]
X271 [0 1]
X272 [0 1]
X273 [1 0]
X274 [0 1]
X275 [1 0]
X276 [0 1]
X277 [0 1]
X278 [0 1]
X279 [0 1]
X280 [0 1]
X281 [0 1]
X282 [0 1]
X283 [0 1]
X284 [0 1]
X285 [1 0]
X286 [0 1]
X287 [0 1]
X288 [0 1]
X289 [0]
X290 [0]
X291 [0 1]
X292 [0 1]
X293 [0]
X294 [0 1]
X295 [0 1]
X296 [0 1]
X297 [0]
X298 [0 1]
X299 [0 1]
X300 [0 1]
X301 [0 1]
X302 [0 1]
X304 [0 1]
X305 [0 1]
X306 [1 0]
X307 [0 1]
X308 [0 1]
```

```
X309 [0 1]
X310 [0 1]
X311 [0 1]
X312 [0 1]
X313 [0 1]
X314 [0 1]
X315 [0 1]
X316 [1 0]
X317 [0 1]
X318 [0 1]
X319 [0 1]
X320 [0 1]
X321 [0 1]
X322 [0 1]
X323 [0 1]
X324 [1 0]
X325 [0 1]
X326 [0 1]
X327 [1 0]
X328 [0 1]
X329 [1 0]
X330 [0]
X331 [0 1]
X332 [0 1]
X333 [0 1]
X334 [1 0]
X335 [0 1]
X336 [0 1]
X337 [0 1]
X338 [0 1]
X339 [0 1]
X340 [0 1]
X341 [0 1]
X342 [0 1]
X343 [0 1]
X344 [0 1]
X345 [0 1]
X346 [0 1]
X347 [0]
X348 [0 1]
X349 [0 1]
X350 [0 1]
X351 [0 1]
X352 [0 1]
X353 [0 1]
```

```
X354 [1 0]
X355 [0 1]
X356 [0 1]
X357 [0 1]
X358 [0 1]
X359 [0 1]
X360 [0 1]
X361 [1 0]
X362 [0 1]
X363 [0 1]
X364 [0 1]
X365 [0 1]
X366 [0 1]
X367 [0 1]
X368 [0 1]
X369 [0 1]
X370 [0 1]
X371 [0 1]
X372 [0 1]
X373 [0 1]
X374 [0 1]
X375 [0 1]
X376 [0 1]
X377 [1 0]
X378 [0 1]
X379 [0 1]
X380 [0 1]
X382 [0 1]
X383 [0 1]
X384 [0 1]
X385 [0 1]
```

Performing Label encoder

Removing the columns where variance is zero

In [35]:

```
for c in columns_usable:
    cardinality = len(np.unique(X_train[c]))
    if cardinality == 1:
        X_train.drop(c, axis=1)
        X_test.drop(c, axis=1)
    if cardinality > 2:
        func = lambda x: sum([ord(digit) for digit in x])
        X_train[c] = X_train[c].apply(func)
        X_test[c] = X_test[c].apply(func)
X_train.head()
```

Out[35]:

	X76	X361	X376	X12	X263	X71	X88	X156	X379	X262	X135	X290	X75	X184	X366	X143	X146	X131	X91	X226	X261	X20	X189
0	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	1
1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

In [37]:

```
X_train.dtypes.value_counts()
```

Out[37]:

```
int64    376
dtype: int64
```

Performing Dimentionality Reduction- PCA

In [38]:

```
from sklearn.decomposition import PCA
```

In [39]:

```
pca = PCA(n_components=12, random_state=200)
pca1_results_train = pca.fit_transform(X_train)
pca1_results_test = pca.transform(X_test)
```

Training with XGboost

In [40]:

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

In [42]:

```
x_train, x_valid, y_train, y_valid = train_test_split(pca1_results_train, y_train, test_size=0.2, random_state=500)
```

In [43]:

```
d_train = xgb.DMatrix(x_train, label=y_train)
d_valid = xgb.DMatrix(x_valid, label=y_valid)
d_test = xgb.DMatrix(pca1_results_test)
```

In [44]:

```
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)
```

```
[10:05:51] WARNING: C:/Users/Administrator/workspace/xgboost-win64_release_1.4.0/src/objective/regression_obj.cu:171:
reg:linear is now deprecated in favor of reg:squarederror.
[0]	train-rmse:98.87196	train-r2:-61.82338	valid-rmse:99.38930	valid-r2:-53.54857
[10]	train-rmse:81.02709	train-r2:-41.19254	valid-rmse:81.55784	valid-r2:-35.73123
[20]	train-rmse:66.48245	train-r2:-27.40463	valid-rmse:67.02181	valid-r2:-23.80483
[30]	train-rmse:54.64143	train-r2:-18.18753	valid-rmse:55.21193	valid-r2:-15.83332
[40]	train-rmse:45.01747	train-r2:-12.02378	valid-rmse:45.64390	valid-r2:-10.50455
[50]	train-rmse:37.21109	train-r2:-7.89856	valid-rmse:37.90889	valid-r2:-6.93572
[60]	train-rmse:30.90405	train-r2:-5.13770	valid-rmse:31.68104	valid-r2:-4.54247
[70]	train-rmse:25.81951	train-r2:-3.28421	valid-rmse:26.71688	valid-r2:-2.94163
[80]	train-rmse:21.75307	train-r2:-2.04100	valid-rmse:22.79634	valid-r2:-1.86969
[90]	train-rmse:18.52706	train-r2:-1.20591	valid-rmse:19.73264	valid-r2:-1.15018
[100]	train-rmse:15.99703	train-r2:-0.64458	valid-rmse:17.37044	valid-r2:-0.66619
[110]	train-rmse:14.03769	train-r2:-0.26639	valid-rmse:15.58379	valid-r2:-0.34107
[120]	train-rmse:12.53541	train-r2:-0.00984	valid-rmse:14.24185	valid-r2:-0.12005
[130]	train-rmse:11.41196	train-r2:0.16306	valid-rmse:13.27024	valid-r2:0.02756
[140]	train-rmse:10.57714	train-r2:0.28103	valid-rmse:12.57575	valid-r2:0.12668
[150]	train-rmse:9.96344	train-r2:0.36204	valid-rmse:12.08267	valid-r2:0.19382
[160]	train-rmse:9.51869	train-r2:0.41772	valid-rmse:11.73733	valid-r2:0.23925
```

```
[170]	train-rmse:9.19556	train-r2:0.45659	valid-rmse:11.49227	valid-r2:0.27068
[180]	train-rmse:8.95083	train-r2:0.48512	valid-rmse:11.31579	valid-r2:0.29291
[190]	train-rmse:8.77279	train-r2:0.50540	valid-rmse:11.19468	valid-r2:0.30797
[200]	train-rmse:8.65518	train-r2:0.51858	valid-rmse:11.10982	valid-r2:0.31842
[210]	train-rmse:8.56034	train-r2:0.52907	valid-rmse:11.05489	valid-r2:0.32514
[220]	train-rmse:8.49525	train-r2:0.53620	valid-rmse:11.01617	valid-r2:0.32986
[230]	train-rmse:8.43849	train-r2:0.54238	valid-rmse:10.98880	valid-r2:0.33319
[240]	train-rmse:8.39388	train-r2:0.54721	valid-rmse:10.96834	valid-r2:0.33567
[250]	train-rmse:8.35440	train-r2:0.55146	valid-rmse:10.95422	valid-r2:0.33738
[260]	train-rmse:8.31935	train-r2:0.55521	valid-rmse:10.94559	valid-r2:0.33842
[270]	train-rmse:8.28738	train-r2:0.55862	valid-rmse:10.93702	valid-r2:0.33946
[280]	train-rmse:8.25398	train-r2:0.56217	valid-rmse:10.93402	valid-r2:0.33982
[290]	train-rmse:8.22442	train-r2:0.56530	valid-rmse:10.93141	valid-r2:0.34013
[300]	train-rmse:8.19395	train-r2:0.56852	valid-rmse:10.92644	valid-r2:0.34073
[310]	train-rmse:8.16746	train-r2:0.57130	valid-rmse:10.91937	valid-r2:0.34159
[320]	train-rmse:8.13445	train-r2:0.57476	valid-rmse:10.91470	valid-r2:0.34215
[330]	train-rmse:8.10646	train-r2:0.57768	valid-rmse:10.91103	valid-r2:0.34259
[340]	train-rmse:8.07377	train-r2:0.58108	valid-rmse:10.90767	valid-r2:0.34299
[350]	train-rmse:8.05197	train-r2:0.58334	valid-rmse:10.91010	valid-r2:0.34270
[360]	train-rmse:8.02356	train-r2:0.58628	valid-rmse:10.91021	valid-r2:0.34269
[370]	train-rmse:7.99718	train-r2:0.58899	valid-rmse:10.90985	valid-r2:0.34273
[380]	train-rmse:7.96847	train-r2:0.59194	valid-rmse:10.90977	valid-r2:0.34274
[390]	train-rmse:7.93935	train-r2:0.59492	valid-rmse:10.90791	valid-r2:0.34297
[394]	train-rmse:7.92760	train-r2:0.59611	valid-rmse:10.90833	valid-r2:0.34292
```

In [46]:

```
p_test = clf.predict(d_test)

sub = pd.DataFrame()
sub['ID'] = id_test
sub['y'] = p_test


sub.head()
```

Out[46]:

	ID	y
0	1	80.068291
1	2	94.189964
2	3	81.952644
3	4	77.850136

```
   ID           y
4   5  111.262535
```

```
In [ ]:
```