Business Problem

To ensure the safety and reliability of each and every unique car configuration before they hit the road, Daimler's engineers have developed a robust testing system. But, optimizing the speed of their testing system for so many possible feature combinations is complex and time-consuming. Hence, Daimler has challenged to reduce the time that cars spend on the test bench. The Objective of the Case Study is to optimize the testing process in a greener way i.e. reducing the testing time with lower carbon dioxide emissions without reducing Daimler's standards on safety and efficiency.

ML Formulation

We can pose this problem as a regression problem to predict the testing time by selecting some important features from the dataset to tackle the curse of dimensionality.

In order to know how our ML model is performing better, We will develop a baseline or random model and we will compare our models with the baseline model, to get a knowledge where our model stands.

Performance Metric

The metric we will use to evaluate our models is - R^2

Why R^2 as metric?

What is R^2

It is the amount of the variation in the output dependent attribute which is predictable from the input independent variable. It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model.

Interpretation -

Assume R2 = 0.68 It can be referred that 68% of the changeability of the dependent output attribute can be explained by the model while the remaining 32 % of the variability is still unaccounted for.

R-squared = 1 - (SSres / SStot)

SSres is the sum of squares of the residual errors.

SStot is the total sum of the errors.

which means we scale our simple MSE based on the difference of actual values from their mean.

R^2 is a convenient rescaling of MSE that is unit invariant.

It is also very interpretable as -

The best possible score is 1 which is obtained when the predicted values are the same as the actual values.

R^2 with value 0 means the model is same as simple mean model.

Negative value of R^2 mean that the model is worse than simple mean model.

Why?

MSE or MAE penalizes the large prediction errors hence the sum of errors can become very large and interpreting it won't be trivial.

Whereas, R^2 is a scale-free score i.e. irrespective of the values being small or large, the value of R square will be less than one or in worst cases just greater than 1.

EDA

In [1]:

```
'''importing dependencies'''
import pandas as pd
import warnings
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os

from xgboost import XGBRegressor
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
warnings.filterwarnings('ignore')
sns.set()
```

Reading the data

In [2]:

```
data = pd.read_csv('downloads/trainwa.csv')
data.head()
```

Out[2]:

	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

Knowing the data

Data Shape

In [3]:

```
print("Train Data Shape : = ", data.shape)
```

```
Train Data Shape : =  (4209, 378)
```

Detail View of Train Data

In [4]:

```
data.describe()
```

Out[4]:

	ID	y	X10	X11	X12	X13	X14	X15	X16	X17
count	4209.000000	4209.000000	4209.000000	4209.0	4209.000000	4209.000000	4209.000000	4209.000000	4209.000000	4209.000000
mean	4205.960798	100.669318	0.013305	0.0	0.075077	0.057971	0.428130	0.000475	0.002613	0.007603

	ID	y	X10	X11	X12	X13	X14	X15	X16	X17
std	2437.608688	12.679381	0.114590	0.0	0.263547	0.233716	0.494867	0.021796	0.051061	0.086872
min	0.000000	72.110000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	2095.000000	90.820000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	4220.000000	99.150000	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	6314.000000	109.010000	0.000000	0.0	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
max	8417.000000	265.320000	1.000000	0.0	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

8 rows × 370 columns

Checking for null values in the Data

In [5]:

```
print("Number of missing values in Train data: ",data.isnull().sum().sum())
```

```
Number of missing values in Train data:  0
```

Checking for duplicate values in the Data

In [6]:

```
print(len(data[data.duplicated()]))
```

```
0
```

Analyzing the Prediction Column 'y'

In [7]:

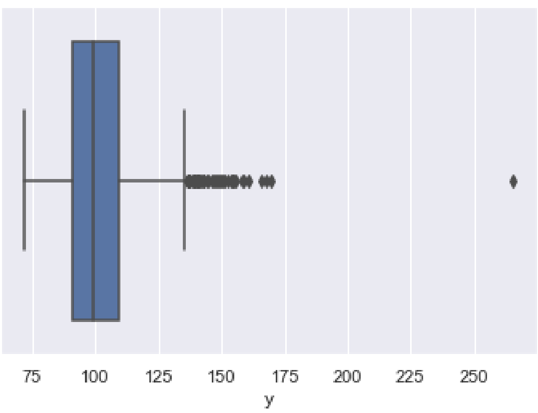
```
y = data['y']
y.describe()
```

Out[7]:

```
count    4209.000000
mean      100.669318
std        12.679381
min        72.110000
25%        90.820000
50%        99.150000
75%       109.010000
max       265.320000
Name: y, dtype: float64
```

In [8]:

```
ax = sns.boxplot(data['y'])
```



Observations

1. The dataset does not have null values.
2. The dataset does not have duplicate values.
3. The Prediction column y has some outlier points

We will first remove these outlier points then we will know the features

Knowing the percentiles to decide the threshold to remove the outliers

In [9]:

```
print("Listing all percentiles for training time: ")
print("100: ", np.percentile(data.y,100))
print("99.9: ", np.percentile(data.y,99.9))
print("99.8: ", np.percentile(data.y,99.8))
print("99.7: ", np.percentile(data.y,99.7))
print("99.6: ", np.percentile(data.y,99.6))
print("99.5: ", np.percentile(data.y,99.5))
print("99 : ", np.percentile(data.y,99))
```

```
Listing all percentiles for training time:
100:  265.32
99.9:  160.38328000000087
99.8:  154.68695999999994
99.7:  151.4276800000003
99.6:  149.0374399999998
99.5:  146.23040000000006
99 :  137.4304
```

Observations -

We can see from 99.7 percentile onwards the y value is increasing drastically. We decide the threshold as 99.7

Removing Noise

In [10]:

```
threshold = np.percentile(data.y,99.7)
outlliers = data[data['y']>=threshold]
data.drop(data[data['y']>=threshold].index, inplace = True)
```

Data Shape after removing noise

In [11]:

```
data.shape
```

Out[11]:

```
(4196, 378)
```

In [12]:

```
outlliers.head()
```

Out[12]:

	ID	y	X0	X1	X2	X3	X4	X5	X6	X8	...	X375	X376	X377	X378	X379	X380	X382	X383	X384	X385
342	681	169.91	aa	l	ak	f	d	i	c	d	...	0	0	0	0	0	0	0	0	0	0

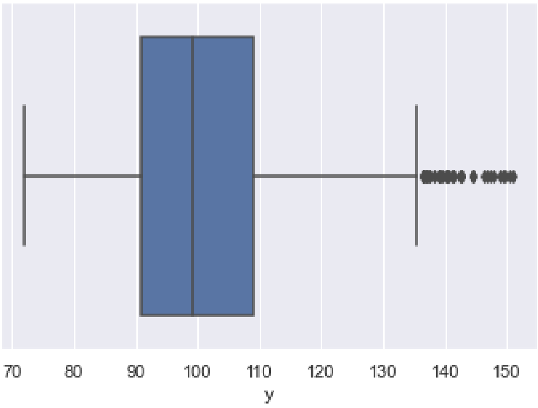
	ID	y	X0	X1	X2	X3	X4	X5	X6	X8	...	X375	X376	X377	X378	X379	X380	X382	X383	X384	X385
429	836	154.87	ak	l	ae	f	d	d	g	w	...	0	0	0	0	0	0	0	0	0	0
883	1770	265.32	y	r	ai	f	d	ag	l	t	...	0	0	0	0	0	0	0	0	0	0
889	1784	158.53	aj	l	as	f	d	ag	k	e	...	0	0	0	0	0	0	0	0	0	0
1060	2111	154.43	w	v	r	c	d	ag	d	q	...	1	0	0	0	0	0	0	0	0	0

5 rows × 378 columns

Box-Plot of y after removing noise

In [13]:

```
ax = sns.boxplot(data['y'])
```



Observations

1. The dataset looks more cleaner now.

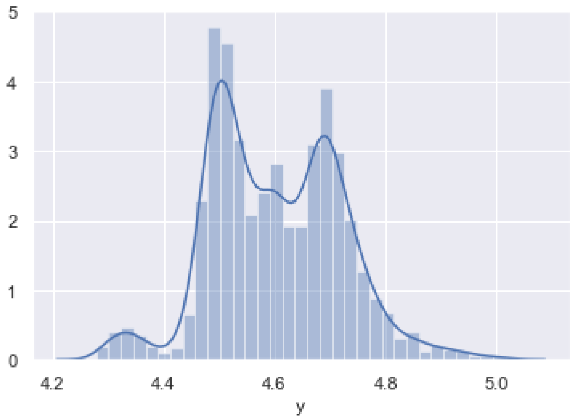
Log Transformation Distribution of target

In [14]:

```
sns.distplot(np.log(data['y']))
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa66abcb8e0>
```



Log Transformation Distribution of target variable in outlier data

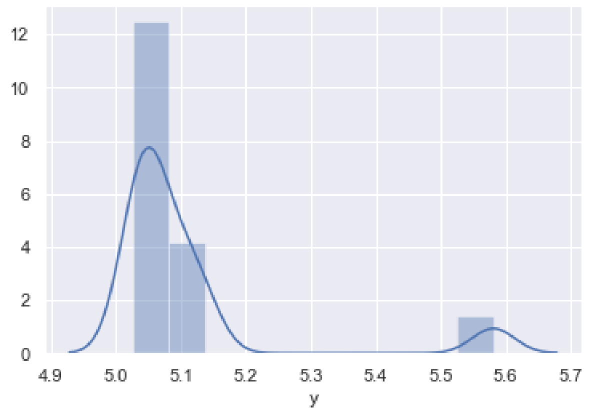
In [15]:

```
sns.distplot(np.log(outlliers['y']))
```

```
sns.distplot(np.log(outliers['y']))
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa66a203640>
```



Observations

1. Most of the y values lie between 80 and 140.
2. Only a bunch of values are greater than 150.
3. Few values are less than 80.

Selecting important features for EDA

Building a simple XGBoost Model to get Feature Importances

In [16]:

```
y = data['y']
x = data.drop(columns = ['ID','y'], axis = 1)
#x = x.drop('id')
x.shape, y.shape
cols = x.columns
```

In [17]:

```
x_cat = data.loc[:,'X0':'X8']
x_num = data.loc[:,'X10':]
```

In [18]:

```
from sklearn.preprocessing import LabelEncoder
enc = LabelEncoder()
for i in x_cat.columns:
    x_cat[i] = enc.fit_transform(x_cat[i])
```

In [19]:

```
x = pd.DataFrame(np.hstack((x_cat,x_num)), columns = cols)
```

In [20]:

```
x.head()
```

Out[20]:

	X0	X1	X2	X3	X4	X5	X6	X8	X10	X11	...	X375	X376	X377	X378	X379	X380	X382	X383	X384	X385
0	32	23	17	0	3	24	9	14	0	0	...	0	0	1	0	0	0	0	0	0	0

	X0	X1	X2	X3	X4	X5	X6	X8	X10	X11	...	X375	X376	X377	X378	X379	X380	X382	X383	X384	X385
1	32	21	19	4	3	28	11	14	0	0	...	1	0	0	0	0	0	0	0	0	0
2	20	24	34	2	3	27	9	23	0	0	...	0	0	0	0	0	0	1	0	0	0
3	20	21	34	5	3	27	11	4	0	0	...	0	0	0	0	0	0	0	0	0	0
4	20	23	34	5	3	12	3	13	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 376 columns

In [21]:

```
model = XGBRegressor(n_estimators=100, learning_rate = 0.1,n_jobs = -1)
model.fit(x,y)
```

Out[21]:

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.1, max_delta_step=0, max_depth=6,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=100, n_jobs=-1, num_parallel_tree=1, random_state=0,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

In [22]:

```
imp = pd.DataFrame()
imp['columns'] = x.columns
imp['importances'] = model.feature_importances_[0]
result = imp.sort_values(by = 'importances')[:10]
```

Top 10 important features with importances

In [23]:

```
result
```

Out[23]:

	columns	importances
0	X0	0.000737
255	X263	0.000737
254	X262	0.000737
253	X261	0.000737
252	X260	0.000737
251	X259	0.000737
250	X258	0.000737
249	X257	0.000737
248	X256	0.000737
247	X255	0.000737

In [24]:

```
print("The Top 10 important features are : ", list(result['columns']))
```

```
The Top 10 important features are :  ['X0', 'X263', 'X262', 'X261', 'X260', 'X259', 'X258',
'X257', 'X256', 'X255']
```

EDA for important Features

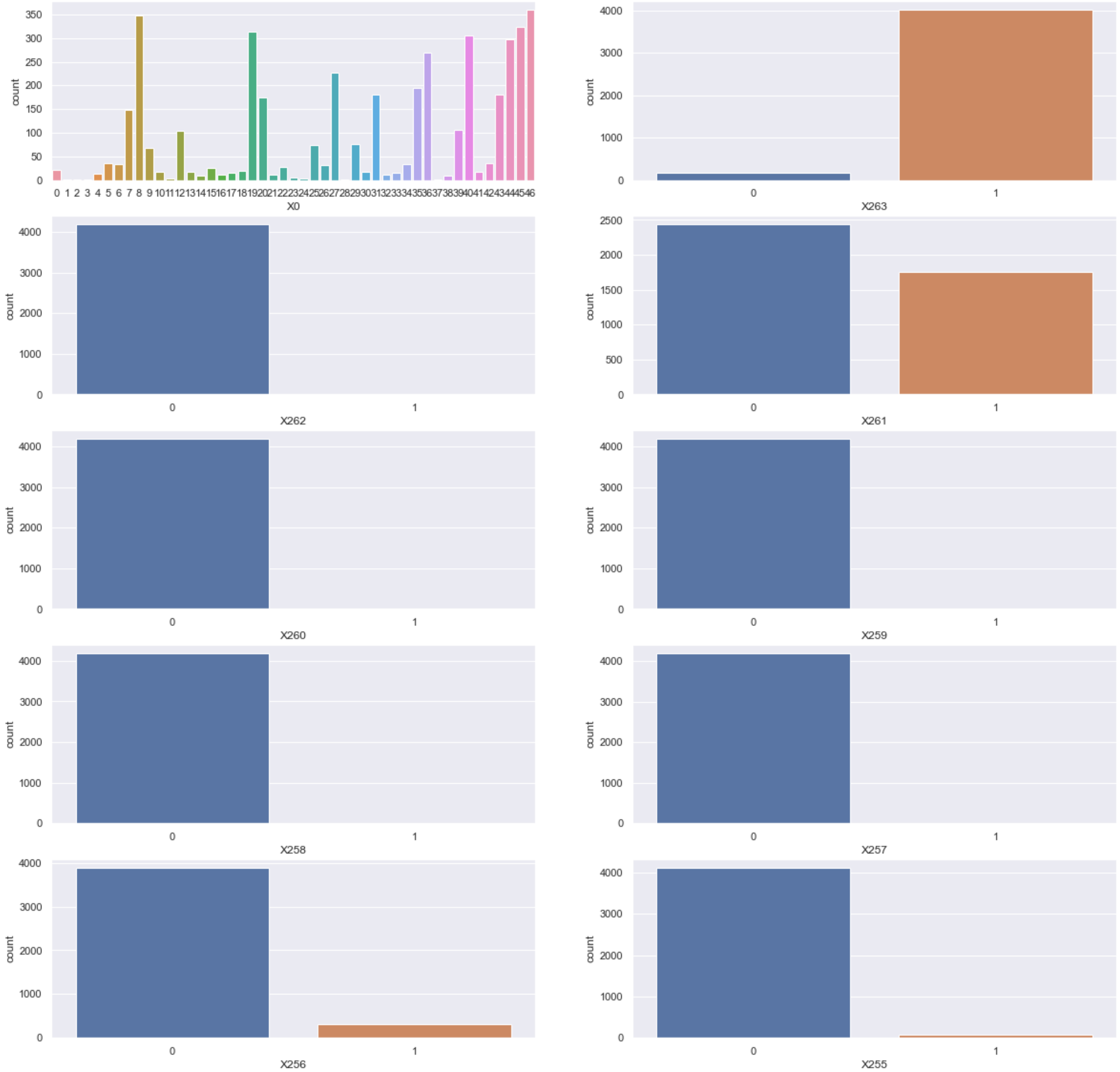
Knowing important features

In [25]:

```
imp_features = list(result['columns'])
imp_data = pd.DataFrame()
for i in imp_features:
    imp_data[i] = x[i]
```

In [26]:

```
#plotting categorical columns
fig, ax = plt.subplots(5, 2, figsize=(20, 20))
for variable, subplot in zip(imp_data.columns, ax.flatten()):
    sns.countplot(imp_data[variable], ax=subplot)
```



Observations -

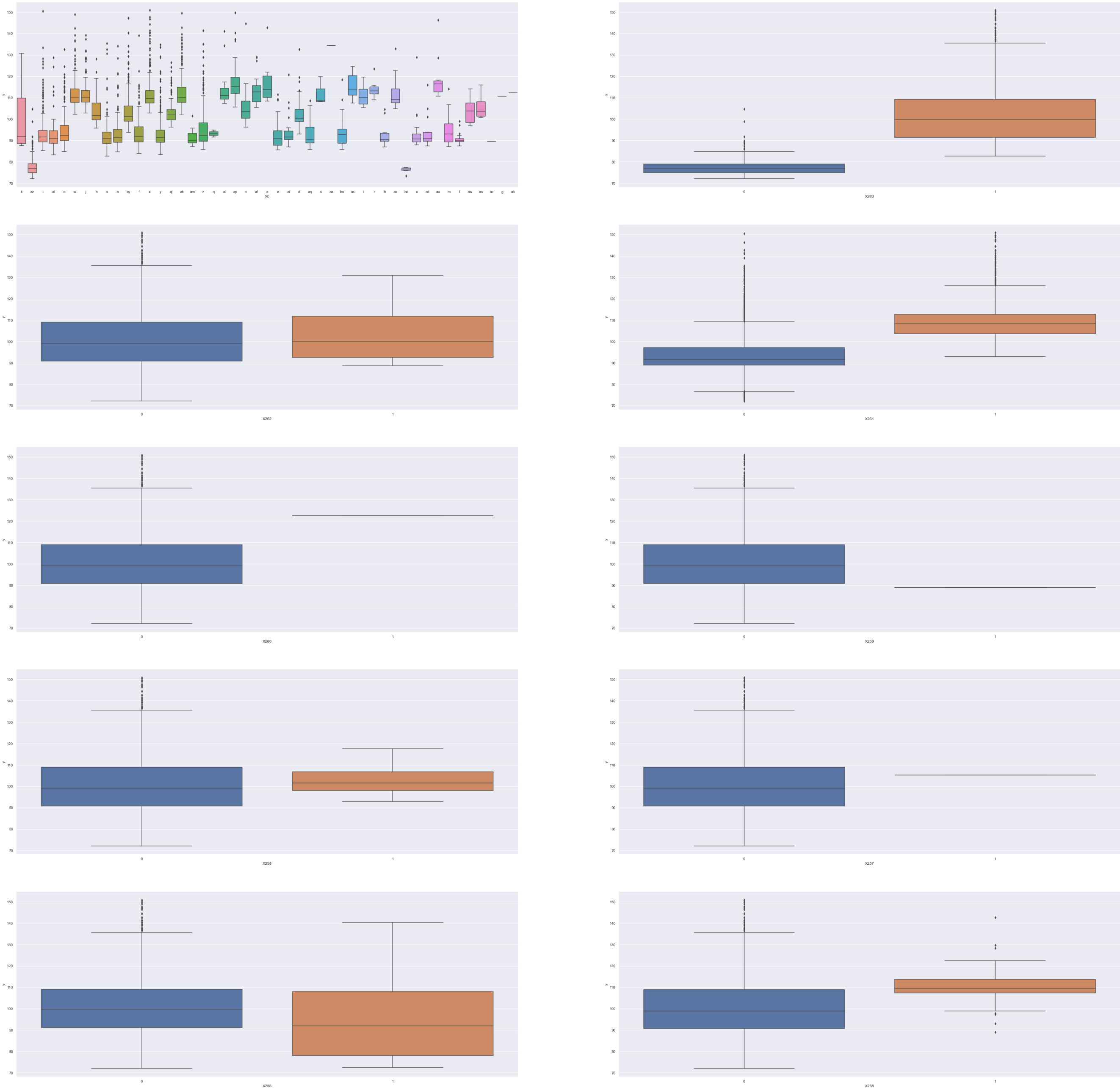
1. The Categorical Feature X0 has well distributed Categories.
2. For the binary features X261 and X263 have significant values of 1 while the rest have mostly 0 values, which means tey are sparse.

Univariate Analysis for Important features

Box - Plot

In [27]:

```
fig, ax = plt.subplots(5, 2, figsize=(60, 60))
for variable, subplot in zip(imp_data.columns, ax.flatten()):
    sns.boxplot(x = data[variable],y = data['y'], ax=subplot)
```



Observations -

1. For binary features X263, X261, X255, X260, X259 the testing time is distinguishable according to the 0 and 1 value.

1. For X0 the percentiles for all the categories is distinguishable so we can interpret the testing time acoording to category.

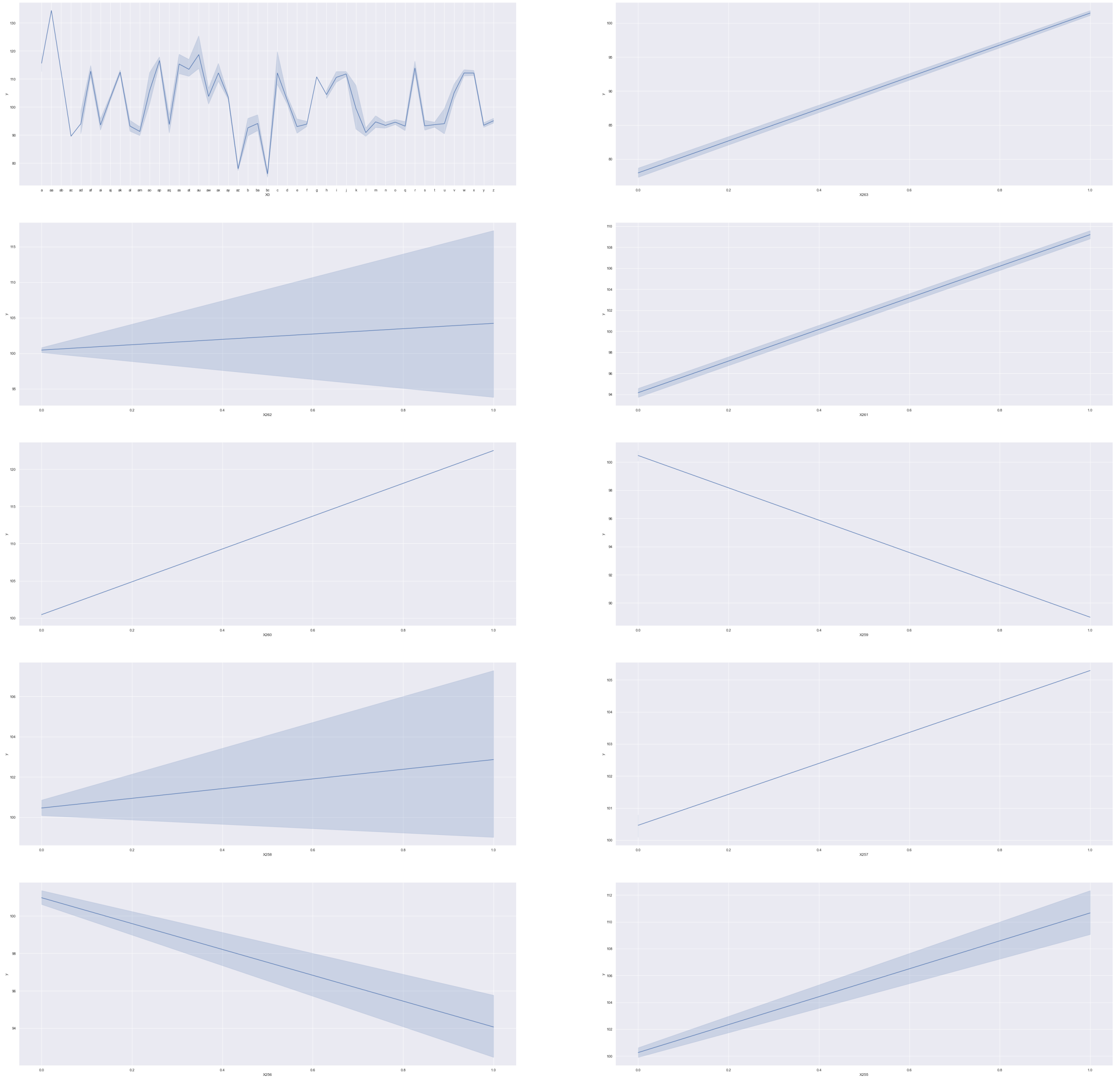
1. For X263, 0 value indicates testing time less than 85 and 1 value indicates testing time between 80 and 130.

1. For X261, 0 value indicates testing time between 75 and 110 and 1 value indicates testing time between 100 and 130.

Line Plot

In [28]:

```
fig, ax = plt.subplots(5, 2, figsize=(60, 60))
for variable, subplot in zip(imp_data.columns, ax.flatten()):
    sns.lineplot(x = data[variable],y = data['y'], ax=subplot)
```



Observations -

1. Line Plot is very interpretable for X0 which has many categories, while for binary features this does not give better information than Box-Plot.

1. For X0 we can conclude that, category 'aa' results in testing time greater than 130 while categories 'az' and 'bc' result in less than 80. The rest lie between 80 and 130.

In [ ]:

```
```

In [ ]:

```
```

Bi - Variate Analysis of Important Features

Co - relation of features

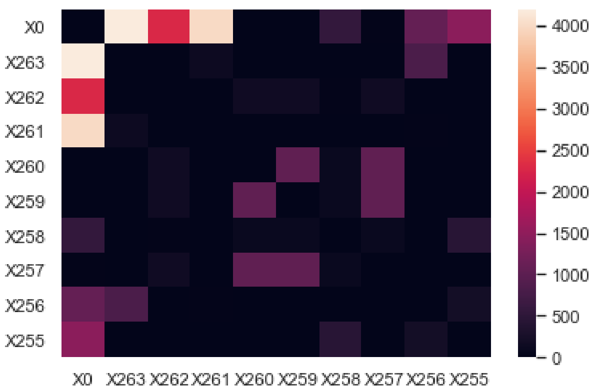
Chi Squared Test

In [29]:

```
import scipy.stats as stats
rows = imp_data.columns
col =  imp_data.columns
chi2_matrix = pd.DataFrame(columns = col, index = rows)
p_matrix = pd.DataFrame(columns = col, index = rows)
lesser_correlated_cols = []
for i in imp_features:
    for j in imp_features:
        if i != j:
            table = pd.crosstab(imp_data[i],imp_data[j])
            #Observed value
            obs_val = table.values
            chi2,p,dof,exp = stats.chi2_contingency(table)
            chi2_matrix[i][j] = chi2
            p_matrix[i][j] = p
            if p>=0.05:
                if (j,i) not in lesser_correlated_cols:
                    lesser_correlated_cols.append((i,j))

chi2_matrix = chi2_matrix.fillna(0)
print("The less realated column pairs are : ")
print(lesser_correlated_cols)
print("The heatmap for chi square values : ")
print(sns.heatmap(chi2_matrix))
```

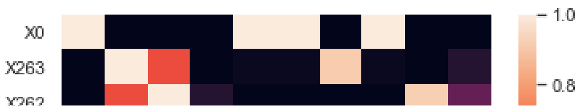
```
The less realated column pairs are :
[('X0', 'X260'), ('X0', 'X259'), ('X0', 'X257'), ('X263', 'X262'), ('X263', 'X258'), ('X263', 'X25
5'), ('X262', 'X261'), ('X262', 'X256'), ('X262', 'X255'), ('X261', 'X260'), ('X261', 'X259'), ('X
261', 'X257'), ('X261', 'X255'), ('X260', 'X256'), ('X259', 'X256'), ('X258', 'X256'), ('X257', 'X
256')]
The heatmap for chi square values :
AxesSubplot(0.125,0.125;0.62x0.755)
```

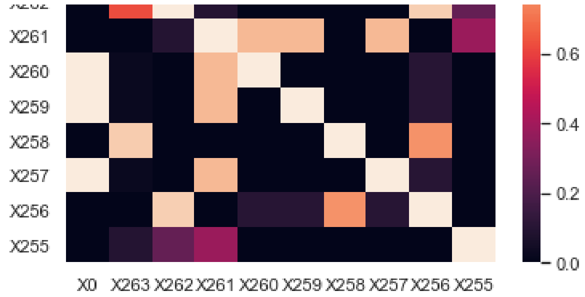


In [30]:

```
p_matrix = p_matrix.fillna(1)
print("The P-value matrix is :")
print(sns.heatmap(p_matrix))
```

```
The P-value matrix is :
AxesSubplot(0.125,0.125;0.62x0.755)
```





Observations -

1. The pair of less related features are :

   ('X0', 'X260'), ('X0', 'X259'), ('X0', 'X257'), ('X263', 'X262'), ('X263', 'X258'), ('X263', 'X255'), ('X262', 'X261'), ('X262', 'X256'), ('X262', 'X255'), ('X261', 'X260'), ('X261', 'X259'), ('X261', 'X257'), ('X261', 'X255'), ('X260', 'X256'), ('X259', 'X256'), ('X258', 'X256'), ('X257', 'X256')

1. The above pairs are decided on the basis of p-value, the null hypothesis that the features are not related is accepted.

Now lets see how these lesser corelated column pairs impact the target together

Feature Pairs with XO

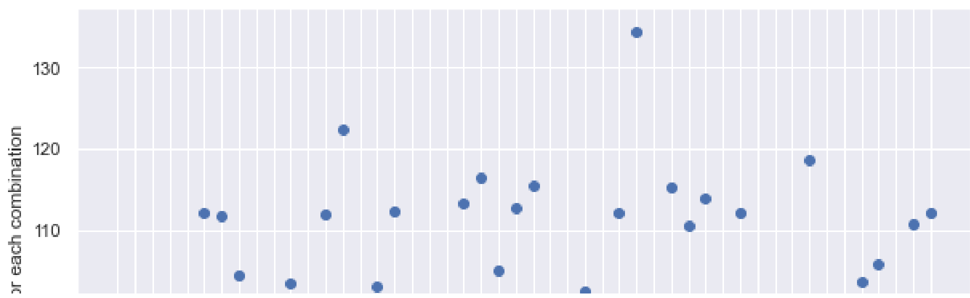
In [31]:

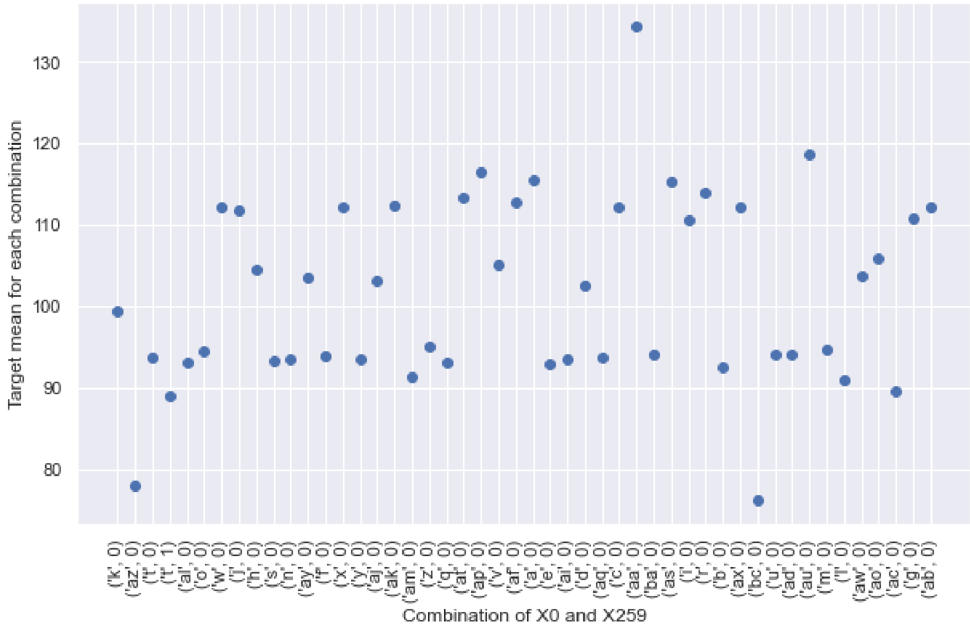
```
'''function to calculate mean of testing time for each pair of categories in the less co related f
eatures '''
def mean_with_category(i):
    col1 = i[0]
    col2 = i[1]
    means = {}
    for j in data[col1].unique():
        for k in data[col2].unique():
            temp = data[data[col1] == j]
            temp = temp[temp[col2] == k]
            if not temp.empty:
                target_mean = temp['y'].mean()
                means[(j,k)] = target_mean
    return means
```

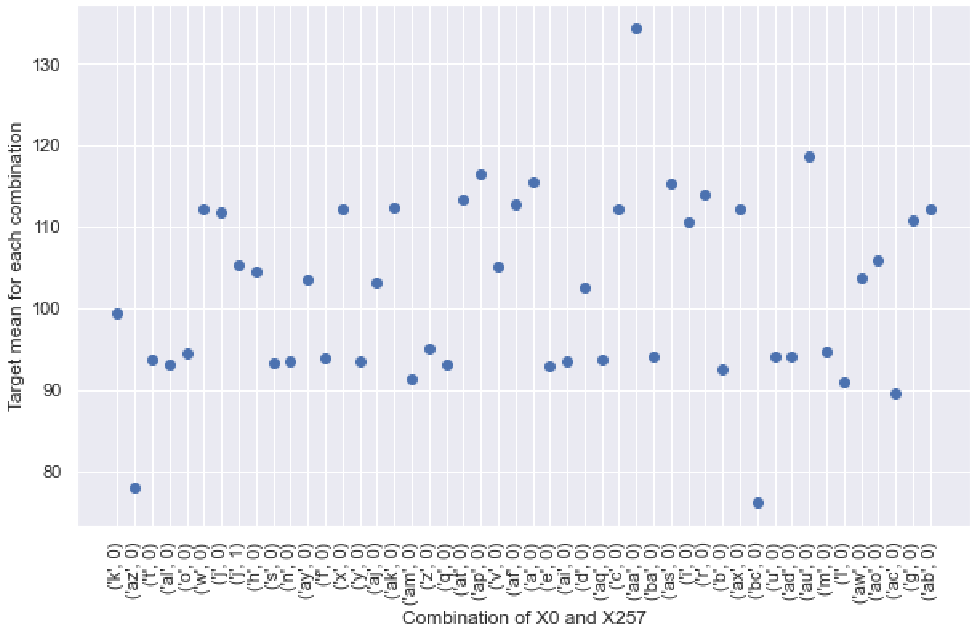
In [32]:

```
for i in lesser_correlated_cols[:3]:
    ans = mean_with_category(i)
    x_plot = ans.keys()
    y_plot = ans.values()

    plt.figure(figsize=(10, 6))
    plt.scatter([str(a) for a in x_plot], y_plot)
    plt.xticks(rotation='vertical')
    plt.xlabel("Combination of "+i[0]+ " and "+i[1])
    plt.ylabel("Target mean for each combination")
```







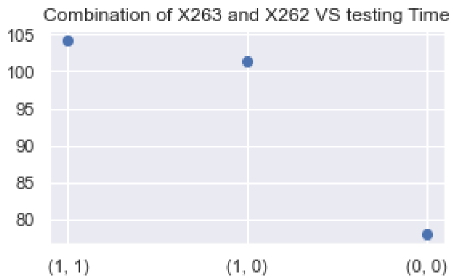
Observations -

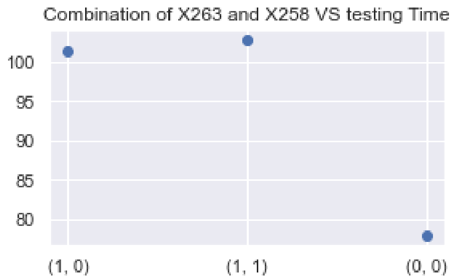
1. For the combination X0 with X260, X259, X257, the category 'aa' and value 0 has highest testing time mean and 'bc' and 0 has lowest.

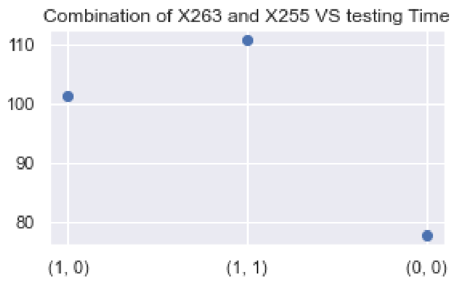
Binary-Binary features

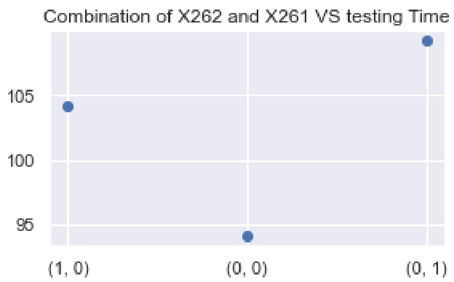
In [33]:

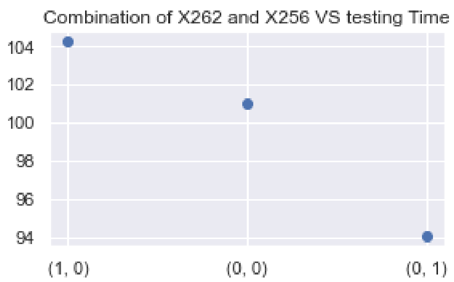
```
fig, ax = plt.subplots(14, 1, figsize=(5, 40))
for i ,subplot in zip(lesser_correlated_cols[3:], ax.flatten()):
    ans = mean_with_category(i)
    x_plot = ans.keys()
    y_plot = ans.values()
    subplot.scatter([str(a) for a in x_plot], y_plot)
    subplot.set_title("Combination of "+i[0]+ " and "+i[1]+" VS testing Time")
    fig.tight_layout(pad=3.0)
```

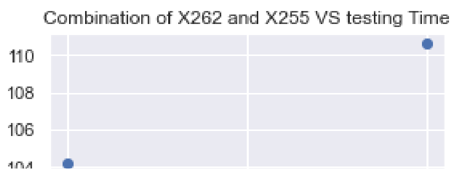


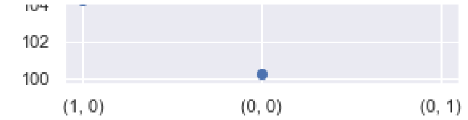




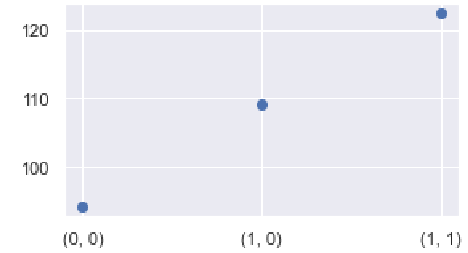




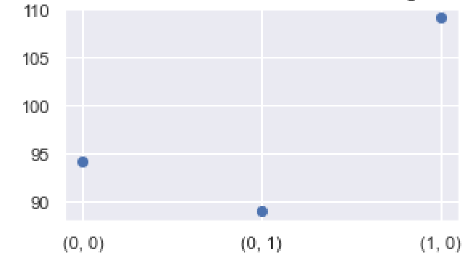




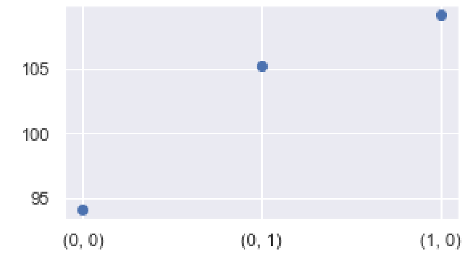
Combination of X261 and X260 VS testing Time



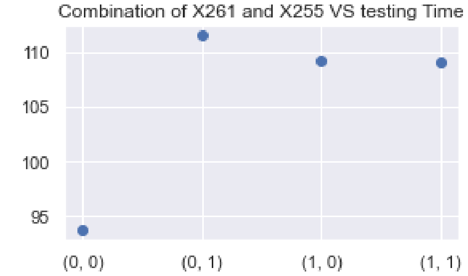
Combination of X261 and X259 VS testing Time



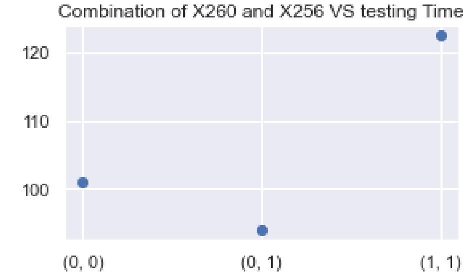
Combination of X261 and X257 VS testing Time



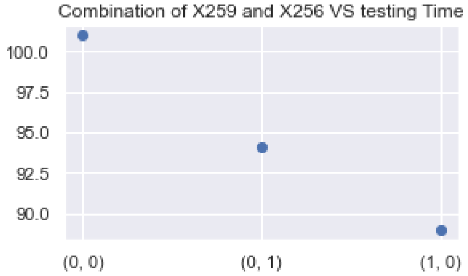
Combination of X261 and X255 VS testing Time

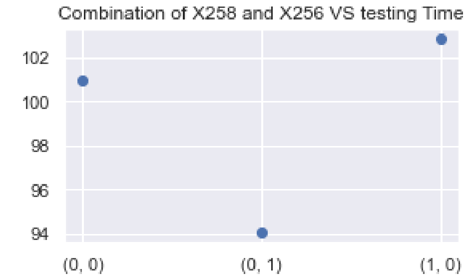


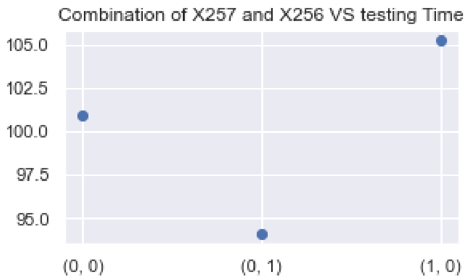
Combination of X260 and X256 VS testing Time



Combination of X259 and X256 VS testing Time







Observations -

1. For X263 and X262,

   if both have 0 values the average testing time is less than 80. if X263 has 1 and X262 has 0, the average testing time is more than 100. if both have value 1, the average testing time is more than 105.

1. For X263 and X258,

   if both have 0 values the average testing time is less than 80. if X263 has 1 and X258 has 0, the average testing time is close to 100. if both have value 1, the average testing time is more than 100.

1. For X263 and X255,

   if both have 0 values the average testing time is less than 80. if X263 has 1 and X255 has 0, the average testing time is close to 100. if both have value 1, the average testing time is more than 110.

1. For X262 and X261,

   if both have 0 values the average testing time is less than 95. if X262 has 1 and X261 has 0, the average testing time is close to 105. if X262 has 0 and X261 has 1, the average testing time is more than 105.

1. For X262 and X256,

   if both have 0 values the average testing time is close to 101. if X262 has 1 and X256 has 0, the average testing time is close to 104. if X262 has 0 and X261 has 1, the average testing time is more than 94.

1. For X262 and X255,

   if both have 0 values the average testing time is close to 100. if X262 has 1 and X255 has 0, the average testing time is close to 104. if X262 has 0 and X255 has 1, the average testing time is more than 110.

1. For X261 and X260,

   if both have 0 values the average testing time is less than 100. if X261 has 1 and X260 has 0, the average testing time is close to 110. if both have value 1, the average testing time is more than 120.

1. For X261 and X259,

   if both have 0 values the average testing time is close to 95. if X261 has 0 and X259 has 1, the average testing time is less than 90. if X261 has 1 and X259 has 0, the average testing time is close to 110.

1. For X261 and X257,

   if both have 0 values the average testing time is close to 95. if X261 has 0 and X257 has 1, the average testing time is less than 105. if X261 has 1 and X257 has 0, the average testing time is close to 110.

1. For X261 and X255,

if both have 0 values the average testing time is close to 95. if X261 has 0 and X255 has 1, the average testing time is more than 110. if X261 has 1 and X257 has 0, the average testing time is close to 110. if both have value 1, the average testing time is close to 110.

1. For X260 and X256,

   if both have 0 values the average testing time is close to 100. if X260 has 0 and X256 has 1, the average testing time is less than 100. if both have value 1, the average testing time is more than 120.

1. For X259 and X256,

   if both have 0 values the average testing time is close to 100. if X259 has 0 and X256 has 1, the average testing time is close to 94. if X259 has 1 and X256 has 0, the average testing time is more than 120.

1. For X258 and X256,

   if both have 0 values the average testing time is close to 101. if X258 has 0 and X256 has 1, the average testing time is close to 94. if X258 has 1 and X256 has 0, the average testing time is more than 102.

1. For X257 and X256,

   if both have 0 values the average testing time is close to 101. if X257 has 0 and X256 has 1, the average testing time is close to 94. if X257 has 1 and X256 has 0, the average testing time is more than 105.

EDA - Conclusion

1. There are no null and duplicate values in the data.

2. Categorical features seem to hold more information as they are more widely present and also account for testing time geater than 130 and less than 80. Also they give information about for testing time between 80 and 130

3. The numerical features are either 0 or 1 with most of the values being 0.

4. The most important features are 'X0', 'X263', 'X262', 'X261', 'X260', 'X259', 'X258', 'X257', 'X256', 'X255'

5. The prediction column y has most values between 80 and 150.

Feature Engineering

In [34]:

```
# split into train test sets
from sklearn.model_selection import train_test_split
y = data['y']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=1)
print("Done")
```

```
Done
```

Baseline model

model which outputs mean

In [35]:

```
y_pred_value  = y_train.mean()
y_pred = []
for i in range(0,len(y_test)):
    y_pred.append(y_pred_value)
```

In [36]:

```
from sklearn.metrics import r2_score
score = r2_score(y_test, y_pred)
```

```
score = r2_score(y_test, y_pred)
print(score)
```

```
-0.002042687819177935
```

In [ ]:

In [ ]:

In [37]:

```
x.head()
```

Out[37]:

	X0	X1	X2	X3	X4	X5	X6	X8	X10	X11	...	X375	X376	X377	X378	X379	X380	X382	X383	X384	X385
0	32	23	17	0	3	24	9	14	0	0	...	0	0	1	0	0	0	0	0	0	0
1	32	21	19	4	3	28	11	14	0	0	...	1	0	0	0	0	0	0	0	0	0
2	20	24	34	2	3	27	9	23	0	0	...	0	0	0	0	0	0	1	0	0	0
3	20	21	34	5	3	27	11	4	0	0	...	0	0	0	0	0	0	0	0	0	0
4	20	23	34	5	3	12	3	13	0	0	...	0	0	0	0	0	0	0	0	0	0

5 rows × 376 columns

In [38]:

```
# split into train test sets
y = data['y']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=1)
print("Done")
```

```
Done
```

In [39]:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

In [40]:

```
 #train autoencoder for regression with no compression in the bottleneck layer
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LeakyReLU
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.utils import plot_model
from matplotlib import pyplot
from keras import backend as K
```

In [41]:

```
n_inputs = x.shape[1]
```

In [42]:

```
# define encoder
input_data = Input(shape=(n_inputs,))
#encoder level 1
encoder = Dense(n_inputs*2)(input_data)
encoder = BatchNormalization()(encoder)
encoder = LeakyReLU()(encoder)

# define bottleneck
n_bottleneck = n_inputs
bottleneck = Dense(n_bottleneck)(encoder)

# decoder level 2
decoder = Dense(n_inputs*2)(bottleneck)
decoder = BatchNormalization()(decoder)
decoder = LeakyReLU()(decoder)

# output layer
output = Dense(n_inputs, activation='linear')(decoder)
# define autoencoder model
model = Model(inputs=input_data, outputs=output)
# compile autoencoder model
model.compile(optimizer='adam', loss='mse')
```

In [43]:

```
model.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 376)]             0         
_________________________________________________________________
dense (Dense)                (None, 752)               283504    
_________________________________________________________________
batch_normalization (BatchNo (None, 752)               3008      
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 752)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 376)               283128    
_________________________________________________________________
dense_2 (Dense)              (None, 752)               283504    
_________________________________________________________________
batch_normalization_1 (Batch (None, 752)               3008      
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 752)               0         
_________________________________________________________________
dense_3 (Dense)              (None, 376)               283128    
=================================================================
Total params: 1,139,280
Trainable params: 1,136,272
Non-trainable params: 3,008
_________________________________________________________________
```

In [44]:

```
# plot the autoencoder
plot_model(model, 'autoencoder.png', show_shapes=True)

# fit the autoencoder model to reconstruct input
history = model.fit(X_train, y_train, epochs=400, batch_size=16, verbose=2, validation_data=(X_test
,y_test))
# plot loss
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
pyplot.show()
# define an encoder model (without the decoder)
encoder = Model(inputs=input_data, outputs=bottleneck)
plot_model(encoder, 'encoder.png', show_shapes=True)
# save the encoder to file
encoder.save('encoder.h5')
```

```
('Failed to import pydot. You must `pip install pydot` and install graphviz
```

```
('Failed to import pydot. You must `pip install pydot` and install graphviz
(https://graphviz.gitlab.io/download/), ', 'for `pydotprint` to work.')
Epoch 1/400
176/176 - 3s - loss: 5499.0928 - val_loss: 1826.5393
Epoch 2/400
176/176 - 1s - loss: 422.9340 - val_loss: 391.2757
Epoch 3/400
176/176 - 1s - loss: 133.7853 - val_loss: 194.6954
Epoch 4/400
176/176 - 1s - loss: 106.5347 - val_loss: 88.7789
Epoch 5/400
176/176 - 1s - loss: 106.5038 - val_loss: 99.2424
Epoch 6/400
176/176 - 1s - loss: 100.0545 - val_loss: 164.7752
Epoch 7/400
176/176 - 1s - loss: 90.8175 - val_loss: 103.7822
Epoch 8/400
176/176 - 1s - loss: 84.8220 - val_loss: 92.4677
Epoch 9/400
176/176 - 1s - loss: 82.8022 - val_loss: 128.8290
Epoch 10/400
176/176 - 1s - loss: 88.1136 - val_loss: 93.5977
Epoch 11/400
176/176 - 1s - loss: 82.5877 - val_loss: 81.1296
Epoch 12/400
176/176 - 1s - loss: 76.0784 - val_loss: 85.6597
Epoch 13/400
176/176 - 1s - loss: 78.2701 - val_loss: 72.4152
Epoch 14/400
176/176 - 1s - loss: 79.6855 - val_loss: 72.8835
Epoch 15/400
176/176 - 1s - loss: 74.8075 - val_loss: 74.6937
Epoch 16/400
176/176 - 1s - loss: 68.1959 - val_loss: 66.9120
Epoch 17/400
176/176 - 1s - loss: 69.8392 - val_loss: 79.9594
Epoch 18/400
176/176 - 1s - loss: 70.8375 - val_loss: 76.1121
Epoch 19/400
176/176 - 2s - loss: 70.6452 - val_loss: 76.6279
Epoch 20/400
176/176 - 2s - loss: 68.7051 - val_loss: 70.2009
Epoch 21/400
176/176 - 2s - loss: 65.8434 - val_loss: 75.0868
Epoch 22/400
176/176 - 2s - loss: 66.7507 - val_loss: 65.8429
Epoch 23/400
176/176 - 1s - loss: 65.5383 - val_loss: 70.0330
Epoch 24/400
176/176 - 2s - loss: 65.2324 - val_loss: 75.0619
Epoch 25/400
176/176 - 1s - loss: 65.7619 - val_loss: 75.8300
Epoch 26/400
176/176 - 1s - loss: 64.8737 - val_loss: 74.5101
Epoch 27/400
176/176 - 2s - loss: 66.2497 - val_loss: 66.1449
Epoch 28/400
176/176 - 1s - loss: 61.2127 - val_loss: 74.3116
Epoch 29/400
176/176 - 1s - loss: 63.1190 - val_loss: 73.6189
Epoch 30/400
176/176 - 2s - loss: 65.7285 - val_loss: 96.1582
Epoch 31/400
176/176 - 2s - loss: 62.6298 - val_loss: 72.1713
Epoch 32/400
176/176 - 2s - loss: 60.6657 - val_loss: 77.7556
Epoch 33/400
176/176 - 2s - loss: 62.9241 - val_loss: 68.8074
Epoch 34/400
176/176 - 2s - loss: 62.9711 - val_loss: 68.9775
Epoch 35/400
176/176 - 2s - loss: 59.9610 - val_loss: 74.4107
Epoch 36/400
176/176 - 2s - loss: 61.3350 - val_loss: 86.2266
Epoch 37/400
176/176 - 2s - loss: 58.2212 - val_loss: 76.8918
Epoch 38/400
176/176 - 1s - loss: 56.9443 - val_loss: 68.5377
```

```
Epoch 39/400
176/176 - 1s - loss: 60.5161 - val_loss: 84.7164
Epoch 40/400
176/176 - 1s - loss: 57.7570 - val_loss: 68.3400
Epoch 41/400
176/176 - 2s - loss: 58.8814 - val_loss: 81.0477
Epoch 42/400
176/176 - 2s - loss: 57.2878 - val_loss: 76.2670
Epoch 43/400
176/176 - 2s - loss: 54.8938 - val_loss: 69.1802
Epoch 44/400
176/176 - 2s - loss: 58.0282 - val_loss: 84.9829
Epoch 45/400
176/176 - 2s - loss: 56.5856 - val_loss: 80.8975
Epoch 46/400
176/176 - 2s - loss: 53.3907 - val_loss: 72.7264
Epoch 47/400
176/176 - 1s - loss: 54.5113 - val_loss: 86.7194
Epoch 48/400
176/176 - 2s - loss: 54.6124 - val_loss: 66.3194
Epoch 49/400
176/176 - 2s - loss: 54.5340 - val_loss: 67.7091
Epoch 50/400
176/176 - 2s - loss: 53.2115 - val_loss: 73.5927
Epoch 51/400
176/176 - 2s - loss: 52.1238 - val_loss: 71.7823
Epoch 52/400
176/176 - 2s - loss: 53.4635 - val_loss: 97.1081
Epoch 53/400
176/176 - 1s - loss: 54.1837 - val_loss: 68.5108
Epoch 54/400
176/176 - 1s - loss: 54.2739 - val_loss: 66.4236
Epoch 55/400
176/176 - 1s - loss: 55.0513 - val_loss: 69.6937
Epoch 56/400
176/176 - 1s - loss: 52.5711 - val_loss: 81.7859
Epoch 57/400
176/176 - 1s - loss: 54.8904 - val_loss: 75.9867
Epoch 58/400
176/176 - 2s - loss: 52.0961 - val_loss: 98.8686
Epoch 59/400
176/176 - 2s - loss: 53.4520 - val_loss: 74.7452
Epoch 60/400
176/176 - 2s - loss: 52.7307 - val_loss: 69.3547
Epoch 61/400
176/176 - 2s - loss: 51.4843 - val_loss: 76.7101
Epoch 62/400
176/176 - 1s - loss: 51.2800 - val_loss: 69.2934
Epoch 63/400
176/176 - 1s - loss: 51.7591 - val_loss: 73.5051
Epoch 64/400
176/176 - 2s - loss: 50.8940 - val_loss: 69.4393
Epoch 65/400
176/176 - 2s - loss: 52.2141 - val_loss: 69.5699
Epoch 66/400
176/176 - 2s - loss: 51.0712 - val_loss: 80.3294
Epoch 67/400
176/176 - 1s - loss: 52.2058 - val_loss: 70.1714
Epoch 68/400
176/176 - 1s - loss: 49.1044 - val_loss: 69.8486
Epoch 69/400
176/176 - 2s - loss: 50.4404 - val_loss: 96.3290
Epoch 70/400
176/176 - 2s - loss: 47.4721 - val_loss: 70.7383
Epoch 71/400
176/176 - 2s - loss: 47.6107 - val_loss: 70.6391
Epoch 72/400
176/176 - 1s - loss: 49.1079 - val_loss: 78.2389
Epoch 73/400
176/176 - 2s - loss: 49.7241 - val_loss: 69.7560
Epoch 74/400
176/176 - 2s - loss: 50.0472 - val_loss: 100.3777
Epoch 75/400
176/176 - 2s - loss: 48.2729 - val_loss: 76.0493
Epoch 76/400
176/176 - 2s - loss: 49.5903 - val_loss: 73.7908
Epoch 77/400
```

```
Epoch 77/400
176/176 - 2s - loss: 50.1444 - val_loss: 69.7101
Epoch 78/400
176/176 - 2s - loss: 47.6211 - val_loss: 69.6636
Epoch 79/400
176/176 - 2s - loss: 46.8562 - val_loss: 70.6914
Epoch 80/400
176/176 - 2s - loss: 49.3114 - val_loss: 75.9259
Epoch 81/400
176/176 - 2s - loss: 46.6632 - val_loss: 84.2512
Epoch 82/400
176/176 - 2s - loss: 48.5955 - val_loss: 81.8557
Epoch 83/400
176/176 - 2s - loss: 47.8074 - val_loss: 70.3463
Epoch 84/400
176/176 - 2s - loss: 47.3706 - val_loss: 74.1677
Epoch 85/400
176/176 - 2s - loss: 47.3961 - val_loss: 75.6681
Epoch 86/400
176/176 - 1s - loss: 46.1290 - val_loss: 71.4274
Epoch 87/400
176/176 - 2s - loss: 48.0312 - val_loss: 77.7676
Epoch 88/400
176/176 - 1s - loss: 47.7145 - val_loss: 71.8181
Epoch 89/400
176/176 - 1s - loss: 47.0475 - val_loss: 72.4865
Epoch 90/400
176/176 - 1s - loss: 46.4600 - val_loss: 70.9406
Epoch 91/400
176/176 - 1s - loss: 44.2532 - val_loss: 72.1382
Epoch 92/400
176/176 - 1s - loss: 45.9137 - val_loss: 69.8645
Epoch 93/400
176/176 - 1s - loss: 44.8460 - val_loss: 71.5068
Epoch 94/400
176/176 - 1s - loss: 46.4233 - val_loss: 89.4306
Epoch 95/400
176/176 - 2s - loss: 44.3616 - val_loss: 76.4762
Epoch 96/400
176/176 - 2s - loss: 47.1931 - val_loss: 78.4553
Epoch 97/400
176/176 - 2s - loss: 45.6001 - val_loss: 73.3102
Epoch 98/400
176/176 - 1s - loss: 44.1569 - val_loss: 70.1398
Epoch 99/400
176/176 - 2s - loss: 44.0157 - val_loss: 70.5667
Epoch 100/400
176/176 - 2s - loss: 44.1801 - val_loss: 72.3570
Epoch 101/400
176/176 - 1s - loss: 43.1739 - val_loss: 71.0491
Epoch 102/400
176/176 - 1s - loss: 43.5431 - val_loss: 71.3291
Epoch 103/400
176/176 - 1s - loss: 43.7190 - val_loss: 67.7769
Epoch 104/400
176/176 - 1s - loss: 42.9169 - val_loss: 73.9759
Epoch 105/400
176/176 - 1s - loss: 44.9072 - val_loss: 73.2137
Epoch 106/400
176/176 - 1s - loss: 44.4389 - val_loss: 72.0213
Epoch 107/400
176/176 - 2s - loss: 45.2041 - val_loss: 115.4611
Epoch 108/400
176/176 - 2s - loss: 44.4686 - val_loss: 79.0440
Epoch 109/400
176/176 - 1s - loss: 44.5816 - val_loss: 94.7577
Epoch 110/400
176/176 - 1s - loss: 42.6605 - val_loss: 71.8696
Epoch 111/400
176/176 - 2s - loss: 42.3586 - val_loss: 77.9566
Epoch 112/400
176/176 - 2s - loss: 42.8234 - val_loss: 70.3982
Epoch 113/400
176/176 - 2s - loss: 44.0534 - val_loss: 69.7192
Epoch 114/400
176/176 - 1s - loss: 42.7242 - val_loss: 79.7198
Epoch 115/400
176/176 - 1s - loss: 43.3492 - val_loss: 75.2929
```

```
Epoch 116/400
176/176 - 1s - loss: 44.7026 - val_loss: 70.6786
Epoch 117/400
176/176 - 1s - loss: 40.3295 - val_loss: 72.5420
Epoch 118/400
176/176 - 2s - loss: 41.7095 - val_loss: 76.0517
Epoch 119/400
176/176 - 1s - loss: 41.4533 - val_loss: 76.0074
Epoch 120/400
176/176 - 2s - loss: 42.7494 - val_loss: 81.4865
Epoch 121/400
176/176 - 2s - loss: 42.9156 - val_loss: 73.0981
Epoch 122/400
176/176 - 2s - loss: 43.4649 - val_loss: 73.1937
Epoch 123/400
176/176 - 2s - loss: 42.5243 - val_loss: 77.2516
Epoch 124/400
176/176 - 1s - loss: 41.3700 - val_loss: 78.2918
Epoch 125/400
176/176 - 1s - loss: 42.2446 - val_loss: 75.7171
Epoch 126/400
176/176 - 1s - loss: 40.5677 - val_loss: 74.2784
Epoch 127/400
176/176 - 1s - loss: 40.8233 - val_loss: 70.9316
Epoch 128/400
176/176 - 2s - loss: 41.8471 - val_loss: 84.5435
Epoch 129/400
176/176 - 2s - loss: 40.2313 - val_loss: 76.5672
Epoch 130/400
176/176 - 1s - loss: 41.4740 - val_loss: 74.1562
Epoch 131/400
176/176 - 2s - loss: 40.3366 - val_loss: 86.9290
Epoch 132/400
176/176 - 2s - loss: 42.7090 - val_loss: 72.3119
Epoch 133/400
176/176 - 2s - loss: 42.9966 - val_loss: 70.6961
Epoch 134/400
176/176 - 2s - loss: 41.2937 - val_loss: 79.9082
Epoch 135/400
176/176 - 2s - loss: 41.0696 - val_loss: 73.8363
Epoch 136/400
176/176 - 2s - loss: 42.9793 - val_loss: 73.7382
Epoch 137/400
176/176 - 1s - loss: 42.5368 - val_loss: 69.0523
Epoch 138/400
176/176 - 1s - loss: 40.1117 - val_loss: 73.8824
Epoch 139/400
176/176 - 2s - loss: 40.7774 - val_loss: 69.0042
Epoch 140/400
176/176 - 2s - loss: 41.0509 - val_loss: 69.2200
Epoch 141/400
176/176 - 2s - loss: 38.3438 - val_loss: 70.4052
Epoch 142/400
176/176 - 2s - loss: 40.0507 - val_loss: 71.5636
Epoch 143/400
176/176 - 2s - loss: 39.3088 - val_loss: 80.3238
Epoch 144/400
176/176 - 2s - loss: 40.9407 - val_loss: 72.5842
Epoch 145/400
176/176 - 2s - loss: 39.7627 - val_loss: 79.4307
Epoch 146/400
176/176 - 2s - loss: 39.5249 - val_loss: 73.1918
Epoch 147/400
176/176 - 2s - loss: 40.9066 - val_loss: 75.2320
Epoch 148/400
176/176 - 2s - loss: 39.5795 - val_loss: 71.3494
Epoch 149/400
176/176 - 2s - loss: 41.2543 - val_loss: 70.9091
Epoch 150/400
176/176 - 2s - loss: 39.8834 - val_loss: 79.1727
Epoch 151/400
176/176 - 1s - loss: 39.1067 - val_loss: 77.6904
Epoch 152/400
176/176 - 2s - loss: 39.0470 - val_loss: 73.6002
Epoch 153/400
176/176 - 1s - loss: 39.8822 - val_loss: 78.8301
Epoch 154/400
```

```
Epoch 154/400
176/176 - 1s - loss: 39.2833 - val_loss: 73.7295
Epoch 155/400
176/176 - 2s - loss: 40.4391 - val_loss: 71.4960
Epoch 156/400
176/176 - 1s - loss: 40.0136 - val_loss: 75.5244
Epoch 157/400
176/176 - 1s - loss: 38.7439 - val_loss: 83.7063
Epoch 158/400
176/176 - 1s - loss: 39.8912 - val_loss: 74.8859
Epoch 159/400
176/176 - 1s - loss: 37.6503 - val_loss: 82.7705
Epoch 160/400
176/176 - 2s - loss: 38.6661 - val_loss: 75.3175
Epoch 161/400
176/176 - 1s - loss: 38.5554 - val_loss: 74.6078
Epoch 162/400
176/176 - 1s - loss: 38.1169 - val_loss: 73.9857
Epoch 163/400
176/176 - 1s - loss: 41.2674 - val_loss: 73.9493
Epoch 164/400
176/176 - 2s - loss: 38.0883 - val_loss: 75.5907
Epoch 165/400
176/176 - 2s - loss: 39.0672 - val_loss: 72.2326
Epoch 166/400
176/176 - 2s - loss: 38.2700 - val_loss: 74.8087
Epoch 167/400
176/176 - 2s - loss: 39.9395 - val_loss: 76.7045
Epoch 168/400
176/176 - 2s - loss: 36.7781 - val_loss: 72.2839
Epoch 169/400
176/176 - 2s - loss: 37.7477 - val_loss: 73.0197
Epoch 170/400
176/176 - 1s - loss: 39.4674 - val_loss: 73.2275
Epoch 171/400
176/176 - 1s - loss: 37.6219 - val_loss: 73.5224
Epoch 172/400
176/176 - 1s - loss: 38.5473 - val_loss: 77.6256
Epoch 173/400
176/176 - 1s - loss: 38.4783 - val_loss: 73.6294
Epoch 174/400
176/176 - 1s - loss: 37.7623 - val_loss: 77.3915
Epoch 175/400
176/176 - 1s - loss: 39.4547 - val_loss: 72.7528
Epoch 176/400
176/176 - 2s - loss: 38.4913 - val_loss: 71.6550
Epoch 177/400
176/176 - 2s - loss: 37.5861 - val_loss: 73.1824
Epoch 178/400
176/176 - 2s - loss: 37.1741 - val_loss: 72.0115
Epoch 179/400
176/176 - 1s - loss: 36.5765 - val_loss: 74.4579
Epoch 180/400
176/176 - 1s - loss: 37.1536 - val_loss: 75.2929
Epoch 181/400
176/176 - 1s - loss: 38.8787 - val_loss: 73.1412
Epoch 182/400
176/176 - 1s - loss: 36.8764 - val_loss: 80.0059
Epoch 183/400
176/176 - 2s - loss: 37.2026 - val_loss: 99.5892
Epoch 184/400
176/176 - 2s - loss: 37.8607 - val_loss: 74.6229
Epoch 185/400
176/176 - 2s - loss: 35.9972 - val_loss: 75.9495
Epoch 186/400
176/176 - 2s - loss: 38.0483 - val_loss: 75.6715
Epoch 187/400
176/176 - 1s - loss: 37.7373 - val_loss: 78.9413
Epoch 188/400
176/176 - 2s - loss: 36.8910 - val_loss: 75.2199
Epoch 189/400
176/176 - 1s - loss: 36.5872 - val_loss: 74.4051
Epoch 190/400
176/176 - 1s - loss: 37.6495 - val_loss: 75.0209
Epoch 191/400
176/176 - 2s - loss: 38.1606 - val_loss: 77.1153
Epoch 192/400
```

```
176/176 - 1s - loss: 37.0850 - val_loss: 73.4537
Epoch 193/400
176/176 - 1s - loss: 37.9695 - val_loss: 79.2579
Epoch 194/400
176/176 - 1s - loss: 37.9605 - val_loss: 74.8661
Epoch 195/400
176/176 - 1s - loss: 36.6880 - val_loss: 79.2404
Epoch 196/400
176/176 - 1s - loss: 37.6131 - val_loss: 85.9251
Epoch 197/400
176/176 - 1s - loss: 36.7191 - val_loss: 73.4023
Epoch 198/400
176/176 - 1s - loss: 36.2469 - val_loss: 72.7060
Epoch 199/400
176/176 - 1s - loss: 35.5827 - val_loss: 79.9430
Epoch 200/400
176/176 - 1s - loss: 36.2927 - val_loss: 78.2766
Epoch 201/400
176/176 - 1s - loss: 35.9062 - val_loss: 76.6085
Epoch 202/400
176/176 - 1s - loss: 35.0304 - val_loss: 79.1186
Epoch 203/400
176/176 - 2s - loss: 36.6377 - val_loss: 72.0735
Epoch 204/400
176/176 - 1s - loss: 37.4170 - val_loss: 72.2780
Epoch 205/400
176/176 - 2s - loss: 36.9788 - val_loss: 87.4624
Epoch 206/400
176/176 - 1s - loss: 35.9530 - val_loss: 75.5042
Epoch 207/400
176/176 - 2s - loss: 35.2597 - val_loss: 76.0645
Epoch 208/400
176/176 - 1s - loss: 36.6005 - val_loss: 73.0535
Epoch 209/400
176/176 - 2s - loss: 34.9531 - val_loss: 72.1466
Epoch 210/400
176/176 - 1s - loss: 34.4998 - val_loss: 88.3679
Epoch 211/400
176/176 - 1s - loss: 34.9375 - val_loss: 71.7683
Epoch 212/400
176/176 - 1s - loss: 36.7127 - val_loss: 90.3684
Epoch 213/400
176/176 - 1s - loss: 36.4936 - val_loss: 70.9737
Epoch 214/400
176/176 - 1s - loss: 35.5509 - val_loss: 78.3897
Epoch 215/400
176/176 - 1s - loss: 35.0383 - val_loss: 73.7384
Epoch 216/400
176/176 - 1s - loss: 35.4438 - val_loss: 71.8860
Epoch 217/400
176/176 - 2s - loss: 34.8358 - val_loss: 72.6873
Epoch 218/400
176/176 - 2s - loss: 37.7358 - val_loss: 85.1001
Epoch 219/400
176/176 - 2s - loss: 34.5646 - val_loss: 72.8228
Epoch 220/400
176/176 - 2s - loss: 35.7835 - val_loss: 79.1234
Epoch 221/400
176/176 - 2s - loss: 36.3326 - val_loss: 80.5708
Epoch 222/400
176/176 - 1s - loss: 36.7242 - val_loss: 76.8412
Epoch 223/400
176/176 - 2s - loss: 35.8849 - val_loss: 76.0899
Epoch 224/400
176/176 - 2s - loss: 36.7013 - val_loss: 72.3019
Epoch 225/400
176/176 - 2s - loss: 35.1019 - val_loss: 74.7658
Epoch 226/400
176/176 - 2s - loss: 33.3039 - val_loss: 70.9419
Epoch 227/400
176/176 - 2s - loss: 36.4513 - val_loss: 77.6647
Epoch 228/400
176/176 - 2s - loss: 33.1491 - val_loss: 71.9657
Epoch 229/400
176/176 - 2s - loss: 34.9525 - val_loss: 75.7952
Epoch 230/400
176/176 - 2s - loss: 34.1068 - val_loss: 75.2301
```

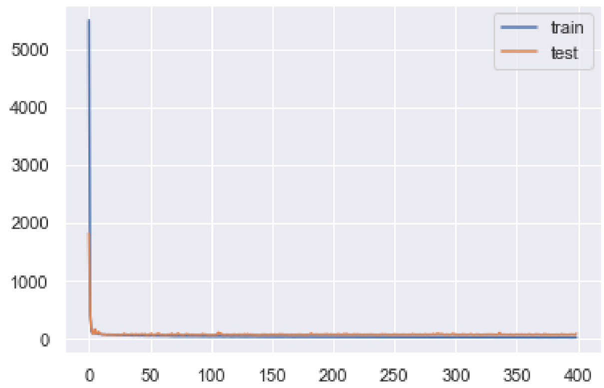
```
Epoch 231/400
176/176 - 1s - loss: 33.1462 - val_loss: 76.3310
Epoch 232/400
176/176 - 1s - loss: 35.9665 - val_loss: 78.5925
Epoch 233/400
176/176 - 1s - loss: 35.2663 - val_loss: 76.5972
Epoch 234/400
176/176 - 1s - loss: 34.4851 - val_loss: 74.7185
Epoch 235/400
176/176 - 1s - loss: 37.1037 - val_loss: 72.6755
Epoch 236/400
176/176 - 2s - loss: 35.0229 - val_loss: 74.4079
Epoch 237/400
176/176 - 2s - loss: 33.2192 - val_loss: 70.5984
Epoch 238/400
176/176 - 2s - loss: 34.8407 - val_loss: 72.2998
Epoch 239/400
176/176 - 2s - loss: 33.0130 - val_loss: 73.6929
Epoch 240/400
176/176 - 1s - loss: 34.9152 - val_loss: 76.1458
Epoch 241/400
176/176 - 1s - loss: 36.7966 - val_loss: 82.6680
Epoch 242/400
176/176 - 1s - loss: 35.2473 - val_loss: 76.0237
Epoch 243/400
176/176 - 1s - loss: 32.1315 - val_loss: 79.9937
Epoch 244/400
176/176 - 2s - loss: 34.1305 - val_loss: 78.1944
Epoch 245/400
176/176 - 2s - loss: 34.9625 - val_loss: 71.9832
Epoch 246/400
176/176 - 2s - loss: 34.0084 - val_loss: 77.7942
Epoch 247/400
176/176 - 2s - loss: 33.2884 - val_loss: 73.2565
Epoch 248/400
176/176 - 2s - loss: 33.8622 - val_loss: 72.2278
Epoch 249/400
176/176 - 2s - loss: 34.4497 - val_loss: 74.2774
Epoch 250/400
176/176 - 2s - loss: 33.3307 - val_loss: 72.0307
Epoch 251/400
176/176 - 2s - loss: 32.8144 - val_loss: 85.4721
Epoch 252/400
176/176 - 1s - loss: 33.4107 - val_loss: 71.5661
Epoch 253/400
176/176 - 1s - loss: 34.1140 - val_loss: 78.3422
Epoch 254/400
176/176 - 1s - loss: 33.4270 - val_loss: 75.4869
Epoch 255/400
176/176 - 1s - loss: 34.1697 - val_loss: 72.0176
Epoch 256/400
176/176 - 1s - loss: 33.3299 - val_loss: 76.4766
Epoch 257/400
176/176 - 2s - loss: 33.7922 - val_loss: 73.4611
Epoch 258/400
176/176 - 2s - loss: 32.8450 - val_loss: 77.0736
Epoch 259/400
176/176 - 2s - loss: 31.7463 - val_loss: 76.3922
Epoch 260/400
176/176 - 2s - loss: 32.4095 - val_loss: 76.6776
Epoch 261/400
176/176 - 2s - loss: 34.0251 - val_loss: 74.7458
Epoch 262/400
176/176 - 2s - loss: 33.6038 - val_loss: 73.0118
Epoch 263/400
176/176 - 2s - loss: 30.4197 - val_loss: 77.0520
Epoch 264/400
176/176 - 2s - loss: 33.4008 - val_loss: 73.2791
Epoch 265/400
176/176 - 2s - loss: 33.7199 - val_loss: 73.9358
Epoch 266/400
176/176 - 1s - loss: 33.2749 - val_loss: 87.7678
Epoch 267/400
176/176 - 1s - loss: 34.3428 - val_loss: 76.8553
Epoch 268/400
176/176 - 2s - loss: 33.5765 - val_loss: 77.1879
Epoch 269/400
```

```
176/176 - 2s - loss: 32.9146 - val_loss: 75.0721
Epoch 270/400
176/176 - 2s - loss: 34.2702 - val_loss: 75.6055
Epoch 271/400
176/176 - 2s - loss: 33.6392 - val_loss: 73.0568
Epoch 272/400
176/176 - 2s - loss: 32.6657 - val_loss: 80.1733
Epoch 273/400
176/176 - 2s - loss: 33.3614 - val_loss: 74.5138
Epoch 274/400
176/176 - 2s - loss: 33.0479 - val_loss: 73.6188
Epoch 275/400
176/176 - 1s - loss: 33.7294 - val_loss: 71.8412
Epoch 276/400
176/176 - 2s - loss: 31.4378 - val_loss: 83.2375
Epoch 277/400
176/176 - 2s - loss: 32.7230 - val_loss: 77.3573
Epoch 278/400
176/176 - 1s - loss: 34.4467 - val_loss: 74.4060
Epoch 279/400
176/176 - 2s - loss: 31.7374 - val_loss: 76.6179
Epoch 280/400
176/176 - 2s - loss: 32.0174 - val_loss: 77.4170
Epoch 281/400
176/176 - 2s - loss: 32.1911 - val_loss: 78.0881
Epoch 282/400
176/176 - 2s - loss: 32.2001 - val_loss: 77.5577
Epoch 283/400
176/176 - 2s - loss: 32.9292 - val_loss: 90.0914
Epoch 284/400
176/176 - 2s - loss: 34.2033 - val_loss: 70.1018
Epoch 285/400
176/176 - 2s - loss: 31.6988 - val_loss: 75.9812
Epoch 286/400
176/176 - 2s - loss: 33.8524 - val_loss: 95.5078
Epoch 287/400
176/176 - 2s - loss: 33.5633 - val_loss: 95.2012
Epoch 288/400
176/176 - 2s - loss: 31.8574 - val_loss: 71.6039
Epoch 289/400
176/176 - 2s - loss: 32.8924 - val_loss: 86.2127
Epoch 290/400
176/176 - 1s - loss: 32.3214 - val_loss: 87.0471
Epoch 291/400
176/176 - 1s - loss: 32.8245 - val_loss: 78.1744
Epoch 292/400
176/176 - 1s - loss: 32.1466 - val_loss: 73.3320
Epoch 293/400
176/176 - 2s - loss: 31.9461 - val_loss: 76.3347
Epoch 294/400
176/176 - 1s - loss: 33.3875 - val_loss: 80.0905
Epoch 295/400
176/176 - 1s - loss: 32.0325 - val_loss: 71.7802
Epoch 296/400
176/176 - 1s - loss: 29.8512 - val_loss: 75.3915
Epoch 297/400
176/176 - 1s - loss: 30.3108 - val_loss: 76.4246
Epoch 298/400
176/176 - 1s - loss: 30.1642 - val_loss: 75.7130
Epoch 299/400
176/176 - 1s - loss: 30.3259 - val_loss: 93.3852
Epoch 300/400
176/176 - 1s - loss: 33.0983 - val_loss: 73.9966
Epoch 301/400
176/176 - 1s - loss: 33.8358 - val_loss: 84.4772
Epoch 302/400
176/176 - 1s - loss: 32.6685 - val_loss: 70.6843
Epoch 303/400
176/176 - 2s - loss: 30.9637 - val_loss: 75.9548
Epoch 304/400
176/176 - 1s - loss: 30.3543 - val_loss: 72.6787
Epoch 305/400
176/176 - 1s - loss: 31.6203 - val_loss: 74.8412
Epoch 306/400
176/176 - 1s - loss: 31.9061 - val_loss: 75.1914
Epoch 307/400
176/176 - 1s - loss: 30.3028 - val_loss: 83.8828
```

```
Epoch 308/400
176/176 - 2s - loss: 31.1076 - val_loss: 74.2560
Epoch 309/400
176/176 - 2s - loss: 32.0953 - val_loss: 77.1800
Epoch 310/400
176/176 - 2s - loss: 30.7334 - val_loss: 74.3671
Epoch 311/400
176/176 - 1s - loss: 31.7907 - val_loss: 71.4913
Epoch 312/400
176/176 - 1s - loss: 30.5630 - val_loss: 77.6507
Epoch 313/400
176/176 - 1s - loss: 31.0626 - val_loss: 81.4490
Epoch 314/400
176/176 - 1s - loss: 30.7659 - val_loss: 85.2107
Epoch 315/400
176/176 - 1s - loss: 31.8239 - val_loss: 71.9903
Epoch 316/400
176/176 - 1s - loss: 31.5599 - val_loss: 69.8480
Epoch 317/400
176/176 - 2s - loss: 32.5166 - val_loss: 70.6072
Epoch 318/400
176/176 - 2s - loss: 30.2911 - val_loss: 75.8833
Epoch 319/400
176/176 - 1s - loss: 30.8746 - val_loss: 81.3253
Epoch 320/400
176/176 - 2s - loss: 31.3071 - val_loss: 74.7825
Epoch 321/400
176/176 - 2s - loss: 30.4705 - val_loss: 74.4348
Epoch 322/400
176/176 - 2s - loss: 30.8644 - val_loss: 70.3873
Epoch 323/400
176/176 - 2s - loss: 29.9974 - val_loss: 72.8805
Epoch 324/400
176/176 - 2s - loss: 30.7602 - val_loss: 75.8659
Epoch 325/400
176/176 - 2s - loss: 31.2430 - val_loss: 74.5988
Epoch 326/400
176/176 - 1s - loss: 30.5551 - val_loss: 75.5245
Epoch 327/400
176/176 - 2s - loss: 30.3881 - val_loss: 78.9752
Epoch 328/400
176/176 - 2s - loss: 31.5809 - val_loss: 78.6183
Epoch 329/400
176/176 - 2s - loss: 30.4139 - val_loss: 78.7944
Epoch 330/400
176/176 - 1s - loss: 30.5994 - val_loss: 76.9351
Epoch 331/400
176/176 - 2s - loss: 31.3348 - val_loss: 81.2462
Epoch 332/400
176/176 - 2s - loss: 29.3209 - val_loss: 72.5476
Epoch 333/400
176/176 - 1s - loss: 29.5907 - val_loss: 75.4488
Epoch 334/400
176/176 - 1s - loss: 29.6908 - val_loss: 75.0651
Epoch 335/400
176/176 - 2s - loss: 30.6860 - val_loss: 73.2232
Epoch 336/400
176/176 - 2s - loss: 29.2031 - val_loss: 73.7189
Epoch 337/400
176/176 - 2s - loss: 29.0096 - val_loss: 108.5419
Epoch 338/400
176/176 - 2s - loss: 29.6752 - val_loss: 76.9124
Epoch 339/400
176/176 - 1s - loss: 30.0419 - val_loss: 76.1593
Epoch 340/400
176/176 - 2s - loss: 30.7494 - val_loss: 75.1866
Epoch 341/400
176/176 - 2s - loss: 30.0602 - val_loss: 78.7228
Epoch 342/400
176/176 - 2s - loss: 29.0066 - val_loss: 75.8863
Epoch 343/400
176/176 - 1s - loss: 29.4396 - val_loss: 73.4764
Epoch 344/400
176/176 - 1s - loss: 30.6552 - val_loss: 74.2472
Epoch 345/400
176/176 - 1s - loss: 29.3349 - val_loss: 73.6597
Epoch 346/400
```

```
176/176 - 1s - loss: 31.0770 - val_loss: 78.9724
Epoch 347/400
176/176 - 1s - loss: 30.2400 - val_loss: 76.6946
Epoch 348/400
176/176 - 1s - loss: 30.5301 - val_loss: 68.9275
Epoch 349/400
176/176 - 1s - loss: 28.5551 - val_loss: 72.0549
Epoch 350/400
176/176 - 1s - loss: 30.6016 - val_loss: 76.7853
Epoch 351/400
176/176 - 1s - loss: 29.4519 - val_loss: 80.3212
Epoch 352/400
176/176 - 1s - loss: 29.9399 - val_loss: 76.6044
Epoch 353/400
176/176 - 2s - loss: 28.0644 - val_loss: 75.3617
Epoch 354/400
176/176 - 1s - loss: 31.4846 - val_loss: 75.2439
Epoch 355/400
176/176 - 1s - loss: 28.6634 - val_loss: 80.0869
Epoch 356/400
176/176 - 1s - loss: 29.8296 - val_loss: 74.4235
Epoch 357/400
176/176 - 1s - loss: 30.5101 - val_loss: 76.2021
Epoch 358/400
176/176 - 1s - loss: 28.7483 - val_loss: 77.9865
Epoch 359/400
176/176 - 1s - loss: 30.1102 - val_loss: 71.4483
Epoch 360/400
176/176 - 1s - loss: 29.0026 - val_loss: 74.4442
Epoch 361/400
176/176 - 1s - loss: 29.6333 - val_loss: 76.2279
Epoch 362/400
176/176 - 2s - loss: 28.7602 - val_loss: 82.9859
Epoch 363/400
176/176 - 2s - loss: 27.8475 - val_loss: 70.7980
Epoch 364/400
176/176 - 1s - loss: 30.1193 - val_loss: 72.7172
Epoch 365/400
176/176 - 1s - loss: 30.2034 - val_loss: 72.0500
Epoch 366/400
176/176 - 1s - loss: 29.0643 - val_loss: 74.8982
Epoch 367/400
176/176 - 1s - loss: 29.8525 - val_loss: 83.8622
Epoch 368/400
176/176 - 1s - loss: 27.9630 - val_loss: 77.9593
Epoch 369/400
176/176 - 1s - loss: 28.0144 - val_loss: 76.9533
Epoch 370/400
176/176 - 1s - loss: 28.5126 - val_loss: 76.9640
Epoch 371/400
176/176 - 1s - loss: 31.4024 - val_loss: 68.2840
Epoch 372/400
176/176 - 2s - loss: 29.6269 - val_loss: 73.6603
Epoch 373/400
176/176 - 1s - loss: 28.9449 - val_loss: 72.1174
Epoch 374/400
176/176 - 1s - loss: 29.3474 - val_loss: 88.4905
Epoch 375/400
176/176 - 1s - loss: 29.5241 - val_loss: 74.6628
Epoch 376/400
176/176 - 1s - loss: 28.1063 - val_loss: 69.8975
Epoch 377/400
176/176 - 1s - loss: 28.2611 - val_loss: 79.4127
Epoch 378/400
176/176 - 1s - loss: 28.3519 - val_loss: 71.3558
Epoch 379/400
176/176 - 1s - loss: 28.9158 - val_loss: 74.1024
Epoch 380/400
176/176 - 1s - loss: 29.2656 - val_loss: 73.8958
Epoch 381/400
176/176 - 1s - loss: 28.4899 - val_loss: 74.9737
Epoch 382/400
176/176 - 1s - loss: 28.5299 - val_loss: 71.8360
Epoch 383/400
176/176 - 1s - loss: 28.6185 - val_loss: 75.7301
Epoch 384/400
176/176 - 1s - loss: 27.7710 - val_loss: 76.3501
```

```
Epoch 385/400
176/176 - 2s - loss: 29.0874 - val_loss: 72.7840
Epoch 386/400
176/176 - 1s - loss: 29.3795 - val_loss: 70.3512
Epoch 387/400
176/176 - 1s - loss: 28.2418 - val_loss: 72.5651
Epoch 388/400
176/176 - 1s - loss: 28.6041 - val_loss: 85.1789
Epoch 389/400
176/176 - 1s - loss: 27.6295 - val_loss: 73.5796
Epoch 390/400
176/176 - 2s - loss: 29.7335 - val_loss: 75.8557
Epoch 391/400
176/176 - 2s - loss: 27.8086 - val_loss: 77.2627
Epoch 392/400
176/176 - 2s - loss: 28.9726 - val_loss: 76.2319
Epoch 393/400
176/176 - 2s - loss: 28.2413 - val_loss: 69.9003
Epoch 394/400
176/176 - 1s - loss: 27.4743 - val_loss: 71.7077
Epoch 395/400
176/176 - 1s - loss: 28.5333 - val_loss: 81.6068
Epoch 396/400
176/176 - 1s - loss: 27.3425 - val_loss: 78.5002
Epoch 397/400
176/176 - 3s - loss: 28.5365 - val_loss: 75.9045
Epoch 398/400
176/176 - 2s - loss: 28.7745 - val_loss: 75.7964
Epoch 399/400
176/176 - 2s - loss: 28.4715 - val_loss: 71.8274
Epoch 400/400
176/176 - 2s - loss: 29.0769 - val_loss: 93.3401
```




```
('Failed to import pydot. You must `pip install pydot` and install graphviz
(https://graphviz.gitlab.io/download/), ', 'for `pydotprint` to work.')
```

In [45]:

```python
from tensorflow.keras.models import load_model
# load the model from file
encoder = load_model('encoder.h5')
```

```
WARNING:tensorflow:No training configuration found in the save file, so the model was *not*
compiled. Compile it manually.
```

In [46]:

```python
X_train_encode = encoder.predict(X_train)
# encode the test data
X_test_encode = encoder.predict(X_test)
```

In [47]:

```python
X_train_encode.shape
```

Out[47]:

```
(2811, 376)
```

Random XGBoost Model with encoded Features

In [48]:

```
reg = XGBRegressor(n_estimators=100, learning_rate = 0.1)
reg.fit(X_train_encode,y_train)
y_pred = reg.predict(X_test_encode)
score = r2_score(y_test, y_pred)
print(score)
```

```
0.44974943013321533
```

Observation -

The Random XGBoost Model with encoded features produces better r2 than Simple mean model

Some Other Feature Engg Techniques

PCA Features

In [49]:

```
from sklearn.decomposition import PCA
#taking top 10 components
components = 10
pca = PCA(n_components=components, random_state=420)

x_pca = pd.DataFrame(pca.fit_transform(x_num))

print(x_pca.shape)
print(x_pca.head())
```

```
(4196, 10)
          0         1         2         3         4         5         6  \
0  0.749079  2.255215  1.016952  0.929835  1.388466  0.044720  0.608854
1 -0.216276  1.103237 -0.832779 -0.670249  0.243599  0.037451  1.199850
2 -0.888967  2.978889  0.269069  2.570786 -0.994356  3.279242 -0.877726
3 -0.509223  2.445044 -0.645017  2.985013 -1.728502  3.132381  0.108157
4 -0.488367  2.236544 -0.787629  3.193679 -2.052340  3.167224 -0.091210

          7         8         9
0 -0.929240  0.200321 -0.730352
1 -0.563451 -0.083035  0.459006
2  0.533689 -0.939597 -0.102745
3  0.017575 -1.028170  0.260647
4  0.125130 -1.729599 -0.344936
```

SVD

In [50]:

```
# get the matrix factors
U, S, VT = np.linalg.svd(x_num,full_matrices=1)
# calculating the aspect ratio b
m = x_num.shape[1]
n = x_num.shape[0]
b = m/n

#taking w_b from table correspondng to b
w_b = 1.6089

# getting the median singular value
```

```
ymed = np.median(S)

# finding the  Hard threshold
cutoff = w_b * ymed
print("The Hard Threshold for Truncation = ",cutoff)
# get the number of components
r = np.max(np.where(S > cutoff))
print("Number of total components to be selected = ",r)
```

```
The Hard Threshold for Truncation =  3.8431396016312185
Number of total components to be selected =  152
```

In [51]:

```
from sklearn.decomposition import TruncatedSVD
n_comp = r

tsvd = TruncatedSVD(n_components=r, random_state=420)

x_svd= tsvd.fit_transform(x_num)


print(x_svd.shape)
```

```
(4196, 152)
```

Different Models

XGBoost

In [52]:

```
results = []
```

Feature Set - 1

Auto - Encoded Features + XGBoost

In [53]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=1)
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.33)
print("Done")
```

```
Done
```

In [54]:

```
X_train_encode = encoder.predict(X_train)
# encode the test data
X_test_encode = encoder.predict(X_test)
#
X_cv_encode = encoder.predict(X_cv)
```

In [55]:

```
X_train_encode.shape
```

Out[55]:

```
(1883, 376)
```

In [56]:

In [56]:

```
from sklearn.metrics import r2_score
learning_rate = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
n_estimators = [5,10,50,75,100,200]
auc_train = []
auc_cv = []
plot_rate,plot_estim = [],[]
for i in learning_rate:
    for j in n_estimators:
        clf = XGBRegressor(learning_rate = i, n_estimators = j,verbosity = 0,n_jobs = -1)
        clf.fit(X_train_encode ,y_train)
        y_train_pred = clf.predict(X_train_encode)
        y_cv_pred = clf.predict(X_cv_encode)
        auc_train.append(r2_score(y_train,y_train_pred))
        auc_cv.append(r2_score(y_cv,y_cv_pred))
        plot_rate.append(i)
        plot_estim.append(j)
```

In [57]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_rate,z=auc_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_rate,z=auc_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='learning_rate'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [58]:

```
model = XGBRegressor(n_estimators=50, learning_rate =0.1)
model.fit(X_train_encode,y_train)
y_te = model.predict(X_test_encode)
score1 = r2_score(y_test, y_te)
results.append(score1)
print("Test Score for 1st feature set : ", score1)
```

```
Test Score for 1st feature set :  0.4783789326440554
```

In [ ]:

```

```

In [ ]:

```

```

Feature Set - 2

Auto - Encoded Features + PCA + XGBoost

In [59]:

```
X_train_Set2, X_test_Set2, y_train, y_test = train_test_split(x_pca, y, test_size=0.33, random_stat
e=1)
X_train_Set2, X_cv_Set2, y_train, y_cv = train_test_split(X_train_Set2, y_train, test_size=0.33)
print("Done")
```

```
Done
```

In [60]:

```
X_train_Set2.shape
```

Out[60]:

```
(1883, 10)
```

In [61]:

```
X_train_Set2 = pd.DataFrame(np.hstack((X_train_encode,X_train_Set2)))
X_cv_Set2 = pd.DataFrame(np.hstack((X_cv_encode,X_cv_Set2)))
X_test_Set2 = pd.DataFrame(np.hstack((X_test_encode,X_test_Set2)))
print(X_train_Set2.shape,X_test_Set2.shape,X_cv_Set2.shape)
```

```
(1883, 386) (1385, 386) (928, 386)
```

In [62]:

```
from sklearn.metrics import r2_score
learning_rate = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_rate,plot_estim = [],[]
for i in learning_rate:
    for j in n_estimators:
        #scaling the positive weight to tackle imbalanced data
        clf = XGBRegressor(learning_rate = i, n_estimators = j,verbosity = 0,n_jobs = -1)
        clf.fit(X_train_Set2 ,y_train)
        y_train_pred = clf.predict(X_train_Set2)
        y_cv_pred = clf.predict(X_cv_Set2)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_rate.append(i)
        plot_estim.append(j)
```

In [63]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]
```

```
layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='learning_rate'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [64]:

```
model = XGBRegressor(n_estimators=75, learning_rate =0.1)
model.fit(X_train_Set2,y_train)
y_te = model.predict(X_test_Set2)
score2 = r2_score(y_test, y_te)
results.append(score2)
print("Test Score for 2nd feature set : ", score2)
```

```
Test Score for 2nd feature set :  0.49980931503390214
```

In [ ]:

```

```

In [ ]:

```

```

Feature Set - 3

PCA + SVD + XGBoost

In [65]:

```
X_Set3 = pd.DataFrame(np.hstack((x_pca,x_svd)))
print(X_Set3.shape)
```

```
(4196, 162)
```

In [66]:

```
X_train_Set3, X_test_Set3, y_train, y_test = train_test_split(X_Set3, y, test_size=0.33, random_sta
te=1)
X_train_Set3, X_cv_Set3, y_train, y_cv = train_test_split(X_train_Set3, y_train, test_size=0.33)
print("Done")
```

Done

In [67]:

```
print(X_train_Set3.shape,X_test_Set3.shape,X_cv_Set3.shape)
```

(1883, 162) (1385, 162) (928, 162)

In [68]:

```
from sklearn.metrics import r2_score
learning_rate = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_rate,plot_estim = [],[]
for i in learning_rate:
    for j in n_estimators:
        #scaling the positive weight to tackle imbalanced data
        clf = XGBRegressor(learning_rate = i, n_estimators = j,verbosity = 0,n_jobs = -1)
        clf.fit(X_train_Set3 ,y_train)
        y_train_pred = clf.predict(X_train_Set3)
        y_cv_pred = clf.predict(X_cv_Set3)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_rate.append(i)
        plot_estim.append(j)
```

In [69]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='learning_rate'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [70]:

```
model = XGBRegressor(n_estimators=50, learning_rate =0.1)
model.fit(X_train_Set3,y_train)
y_te = model.predict(X_test_Set3)
score3 = r2_score(y_test, y_te)
results.append(score3)
print("Test Score for 3rd feature set : ", score3)
```

```
Test Score for 3rd feature set :  0.5168185624963257
```

Feature Set - 4

Label Encoded Categorical features + original Binary Features + PCA + SVD + XGBoost

In [71]:

```
X_Set4 = pd.DataFrame(np.hstack((x,x_pca,x_svd)))
print(X_Set4.shape)
```

```
(4196, 538)
```

In [72]:

```
X_train_Set4, X_test_Set4, y_train, y_test = train_test_split(X_Set4, y, test_size=0.33, random_sta
te=1)
X_train_Set4, X_cv_Set4, y_train, y_cv = train_test_split(X_train_Set4, y_train, test_size=0.33)
print("Done")
```

```
Done
```

In [73]:

```
print(X_train_Set4.shape,X_test_Set4.shape,X_cv_Set4.shape)
```

```
(1883, 538) (1385, 538) (928, 538)
```

In [74]:

```
from sklearn.metrics import r2_score
learning_rate = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_rate,plot_estim = [],[]
for i in learning_rate:
    for j in n_estimators:
        clf = XGBRegressor(learning_rate = i, n_estimators = j,verbosity = 0,n_jobs = -1)
        clf.fit(X_train_Set4 ,y_train)
        y_train_pred = clf.predict(X_train_Set4)
        y_cv_pred = clf.predict(X_cv_Set4)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_rate.append(i)
        plot_estim.append(j)
```

In [75]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='learning_rate'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [76]:

```
model = XGBRegressor(n_estimators=50, learning_rate =0.1)
model.fit(X_train_Set4,y_train)
y_te = model.predict(X_test_Set4)
score4 = r2_score(y_test, y_te)
results.append(score4)
print("Test Score for 4th feature set : ", score4)
```

```
Test Score for 3rd feature set :  0.5854359536270364
```

In [ ]:

```

```

Feature Set - 5

Label Encoded Categorical features + original Binary Features + SVD + XGBoost

In [77]:

```
X_Set5 = pd.DataFrame(np.hstack((x,x_svd)))
print(X_Set5.shape)
```

```
(4196, 528)
```

In [78]:

```
X_train_Set5, X_test_Set5, y_train, y_test = train_test_split(X_Set5, y, test_size=0.33, random_sta
te=1)
X_train_Set5, X_cv_Set5, y_train, y_cv = train_test_split(X_train_Set5, y_train, test_size=0.33)
print("Done")
```

```
Done
```

In [79]:

```
print(X_train_Set5.shape,X_test_Set5.shape,X_cv_Set5.shape)
```

```
(1883, 528) (1385, 528) (928, 528)
```

In [80]:

```
from sklearn.metrics import r2_score
learning_rate = [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_rate,plot_estim = [],[]
for i in learning_rate:
    for j in n_estimators:
        clf = XGBRegressor(learning_rate = i, n_estimators = j,verbosity = 0,n_jobs = -1)
        clf.fit(X_train_Set5 ,y_train)
        y_train_pred = clf.predict(X_train_Set5)
        y_cv_pred = clf.predict(X_cv_Set5)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_rate.append(i)
        plot_estim.append(j)
```

In [81]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_rate,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='learning_rate'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [82]:

```
model = XGBRegressor(n_estimators=50, learning_rate =0.1)
model.fit(X_train_Set5,y_train)
y_te = model.predict(X_test_Set5)
score5 = r2_score(y_test, y_te)
results.append(score5)
print("Test Score for 5th feature set : ", score5)
```

```
Test Score for 3rd feature set :  0.5858048821103229
```

Linear Regression

With Feature Set - 4

In [84]:

```
from sklearn.linear_model import LinearRegression
X_Set4 = pd.DataFrame(np.hstack((x,x_pca,x_svd)))
print(X_Set4.shape)
```

```
(4196, 538)
```

In [85]:

```
X_train_LR_Set4, X_test_LR_Set4, y_train, y_test = train_test_split(X_Set4, y, test_size=0.33, rand
om_state=1)
print("Done")
```

```
Done
```

In [88]:

```
lr = LinearRegression()
lr.fit(X_train_LR_Set4,y_train)
y_pred = lr.predict(X_test_LR_Set4)
score6 = r2_score(y_test,y_pred)
print('R_2 Error on test : ', score6)
```

```
R_2 Error on test :  -6.05298153727229e+19
```

In [ ]:

```

```

With Feature Set - 5

In [89]:

```
X_Set5 = pd.DataFrame(np.hstack((x,x_svd)))
print(X_Set5.shape)
```

```
(4196, 528)
```

In [90]:

```
X_train_LR_Set5, X_test_LR_Set5, y_train, y_test = train_test_split(X_Set5, y, test_size=0.33, rand
om_state=1)
print("Done")
```

```
Done
```

In [91]:

```
lr = LinearRegression()
lr.fit(X_train_LR_Set5,y_train)
y_pred = lr.predict(X_test_LR_Set5)
score7 = r2_score(y_test,y_pred)
print('R_2 Error on test : ', score7)
```

```
R_2 Error on test :  -9.760415548410465e+19
```

In [ ]:

```

```

Random Forest

With Feature Set - 4

In [227]:

```
X_Set4 = pd.DataFrame(np.hstack((x,x_pca,x_svd)))
print(X_Set4.shape)
```

```
(4196, 538)
```

In [228]:

```
X_train_RF_Set4, X_test_RF_Set4, y_train, y_test = train_test_split(X_Set4, y, test_size=0.33, rand
om_state=1)
X_train_RF_Set4, X_cv_RF_Set4, y_train, y_cv = train_test_split(X_train_RF_Set4, y_train,
test_size=0.33)
print("Done")
```

```
Done
```

In [229]:

```
from sklearn.ensemble import RandomForestRegressor
max_depth = [5, 10, 15,20, 25, 40]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_dep,plot_estim = [],[]
for i in max_depth:
    for j in n_estimators:
        clf = RandomForestRegressor(max_depth = i, n_estimators = j, verbose = 0,n_jobs = -1)
        clf.fit(X_train_RF_Set4 ,y_train)
        y_train_pred = clf.predict(X_train_RF_Set4)
        y_cv_pred = clf.predict(X_cv_RF_Set4)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_dep.append(i)
        plot_estim.append(j)
```

In [230]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_dep,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_dep,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='max_depth'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [231]:

```
model = RandomForestRegressor(n_estimators=200, max_depth =5)
model.fit(X_train_RF_Set4,y_train)
y_te = model.predict(X_test_RF_Set4)
score8 = r2_score(y_test, y_te)
print("Test Score for 4th feature set : ", score8)
```

```
Test Score for 4th feature set :  0.5919103348411419
```

With Feature Set - 5

In [206]:

```
X_Set5 = pd.DataFrame(np.hstack((x,x_svd)))
print(X_Set5.shape)
```

```
(4196, 528)
```

In [207]:

```
X_train_RF_Set5, X_test_RF_Set5, y_train, y_test = train_test_split(X_Set5, y, test_size=0.33, rand
om_state=1)
X_train_RF_Set5, X_cv_RF_Set5, y_train, y_cv = train_test_split(X_train_RF_Set5, y_train,
test_size=0.33)
print("Done")
```

```
print('Done')
```

Done

In [208]:

```
max_depth = [5, 10, 15,20, 25, 40]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_dep,plot_estim = [],[]
for i in max_depth:
    for j in n_estimators:
        clf = RandomForestRegressor(max_depth = i, n_estimators = j, verbose = 0,n_jobs = -1)
        clf.fit(X_train_RF_Set5 ,y_train)
        y_train_pred = clf.predict(X_train_RF_Set5)
        y_cv_pred = clf.predict(X_cv_RF_Set5)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_dep.append(i)
        plot_estim.append(j)
```

In [209]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_dep,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_dep,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='max_depth'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [213]:

```
model = RandomForestRegressor(n_estimators=200, max_depth =5)
model.fit(X_train_RF_Set5,y_train)
y_te = model.predict(X_test_RF_Set5)
```

```
score9 = r2_score(y_test, y_te)
print("Test Score for 4th feature set : ", score9)
```

```
Test Score for 4th feature set :  0.5977776730961734
```

MLP

With Feature Set - 4

In [138]:

```
from keras.models import Sequential
from keras.utils import np_utils
from keras.layers.core import Dense, Activation, Dropout
from keras.layers import BatchNormalization
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.callbacks import ModelCheckpoint
import tensorflow as tf
import datetime
```

In [153]:

```
from keras import backend as K
"""Custom R2 Score"""
def rsquared(y_true, y_pred):
    from keras import backend as K
    SS_res =  K.sum(K.square( y_true-y_pred ))
    SS_tot = K.sum(K.square( y_true - K.mean(y_true) ) )
    return ( 1 - SS_res/(SS_tot + K.epsilon()) )
```

In [154]:

```
X_Set4 = pd.DataFrame(np.hstack((x,x_pca,x_svd)))
print(X_Set4.shape)
```

```
(4196, 538)
```

In [155]:

```
X_train_MLP_Set4, X_test_MLP_Set4, y_train, y_test = train_test_split(X_Set4, y, test_size=0.33, ra
ndom_state=1)
print("Done")
```

```
Done
```

In [170]:

```
input_dim = X_train_MLP_Set4.shape[1]

# The Input Layer :
model = Sequential()
model.add(Dense(128,kernel_initializer='normal', input_dim=input_dim, activation='relu'))

# The Hidden Layers :
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dropout(0.15))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
# The Output Layer :
model.add(Dense(1, kernel_initializer='normal',activation='linear'))


model.compile(loss='mean_squared_error', optimizer='adam', metrics=[rsquared])
model.summary()
```

```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_124 (Dense)            (None, 128)               68992     
_________________________________________________________________
dense_125 (Dense)            (None, 256)               33024     
_________________________________________________________________
dense_126 (Dense)            (None, 256)               65792     
_________________________________________________________________
dense_127 (Dense)            (None, 256)               65792     
_________________________________________________________________
dropout_11 (Dropout)         (None, 256)               0         
_________________________________________________________________
dense_128 (Dense)            (None, 256)               65792     
_________________________________________________________________
dense_129 (Dense)            (None, 256)               65792     
_________________________________________________________________
dense_130 (Dense)            (None, 1)                 257       
=================================================================
Total params: 365,441
Trainable params: 365,441
Non-trainable params: 0
_________________________________________________________________
```

In [171]:

```
filepath="/tmp/checkpoint"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_rsquared', verbose=1, save_best_only=True, mode='max')

optimizer = tf.keras.optimizers.Adam(0.01)


#time = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
log_dir= "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,histogram_freq=1, write_graph=True,write_grads=True)

callbacks_list = [checkpoint,tensorboard_callback]

model.fit(X_train_MLP_Set4,y_train,epochs=200, validation_data=(X_test_MLP_Set4,y_test), batch_size=1000, callbacks=callbacks_list)
```

```
WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the `TensorBoard` Callback.
Epoch 1/200
3/3 [==============================] - 1s 199ms/step - loss: 10279.8257 - rsquared: -70.8920 - val_loss: 9970.0342 - val_rsquared: -71.6742

Epoch 00001: val_rsquared improved from -inf to -71.67418, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 2/200
3/3 [==============================] - 0s 74ms/step - loss: 9987.7981 - rsquared: -66.4005 - val_loss: 8989.9873 - val_rsquared: -64.5589

Epoch 00002: val_rsquared improved from -71.67418 to -64.55894, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 3/200
3/3 [==============================] - 0s 79ms/step - loss: 8536.6604 - rsquared: -58.2436 - val_loss: 5360.4272 - val_rsquared: -38.1855

Epoch 00003: val_rsquared improved from -64.55894 to -38.18546, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 4/200
3/3 [==============================] - 0s 62ms/step - loss: 4149.1461 - rsquared: -26.8366 - val_loss: 842.4279 - val_rsquared: -5.0031

Epoch 00004: val_rsquared improved from -38.18546 to -5.00307, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 5/200
3/3 [==============================] - 0s 64ms/step - loss: 1834.2585 - rsquared: -11.6816 - val_loss: 852.6474 - val_rsquared: -5.0748

Epoch 00005: val_rsquared did not improve from -5.00307
Epoch 6/200
```

```
3/3 [==============================] - 0s 75ms/step - loss: 710.5410 - rsquared: -3.9172 -
val_loss: 1131.0057 - val_rsquared: -7.3391

Epoch 00006: val_rsquared did not improve from -5.00307
Epoch 7/200
3/3 [==============================] - 0s 76ms/step - loss: 1322.4523 - rsquared: -8.2082 -
val_loss: 1376.0593 - val_rsquared: -9.1267

Epoch 00007: val_rsquared did not improve from -5.00307
Epoch 8/200
3/3 [==============================] - 0s 84ms/step - loss: 1213.2979 - rsquared: -7.4892 -
val_loss: 375.1458 - val_rsquared: -1.7568

Epoch 00008: val_rsquared improved from -5.00307 to -1.75680, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 9/200
3/3 [==============================] - 0s 67ms/step - loss: 406.5538 - rsquared: -1.8256 -
val_loss: 805.6639 - val_rsquared: -4.7317

Epoch 00009: val_rsquared did not improve from -1.75680
Epoch 10/200
3/3 [==============================] - 0s 67ms/step - loss: 797.7793 - rsquared: -4.3488 -
val_loss: 312.0695 - val_rsquared: -1.2193

Epoch 00010: val_rsquared improved from -1.75680 to -1.21927, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 11/200
3/3 [==============================] - 0s 64ms/step - loss: 302.5273 - rsquared: -1.0764 -
val_loss: 397.1273 - val_rsquared: -1.9160

Epoch 00011: val_rsquared did not improve from -1.21927
Epoch 12/200
3/3 [==============================] - 0s 66ms/step - loss: 461.1630 - rsquared: -2.1790 -
val_loss: 362.2863 - val_rsquared: -1.6574

Epoch 00012: val_rsquared did not improve from -1.21927
Epoch 13/200
3/3 [==============================] - 0s 63ms/step - loss: 340.4578 - rsquared: -1.3116 -
val_loss: 182.8946 - val_rsquared: -0.2953

Epoch 00013: val_rsquared improved from -1.21927 to -0.29532, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 14/200
3/3 [==============================] - 0s 61ms/step - loss: 246.1349 - rsquared: -0.7045 -
val_loss: 269.5939 - val_rsquared: -0.9046

Epoch 00014: val_rsquared did not improve from -0.29532
Epoch 15/200
3/3 [==============================] - 0s 64ms/step - loss: 252.0918 - rsquared: -0.7132 -
val_loss: 147.0453 - val_rsquared: -0.0523

Epoch 00015: val_rsquared improved from -0.29532 to -0.05230, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 16/200
3/3 [==============================] - 0s 69ms/step - loss: 187.0069 - rsquared: -0.2790 -
val_loss: 189.0066 - val_rsquared: -0.3669

Epoch 00016: val_rsquared did not improve from -0.05230
Epoch 17/200
3/3 [==============================] - 0s 60ms/step - loss: 202.0992 - rsquared: -0.3893 -
val_loss: 116.7072 - val_rsquared: 0.1796

Epoch 00017: val_rsquared improved from -0.05230 to 0.17963, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 18/200
3/3 [==============================] - 0s 63ms/step - loss: 143.3302 - rsquared: 0.0283 -
val_loss: 149.3207 - val_rsquared: -0.0453

Epoch 00018: val_rsquared did not improve from 0.17963
Epoch 19/200
3/3 [==============================] - 0s 104ms/step - loss: 160.5569 - rsquared: -0.1027 -
val_loss: 102.9286 - val_rsquared: 0.2799

Epoch 00019: val_rsquared improved from 0.17963 to 0.27992, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 20/200
3/3 [==============================] - 0s 61ms/step - loss: 131.1961 - rsquared: 0.0973 -
```

```
val_loss: 113.8689 - val_rsquared: 0.1970

Epoch 00020: val_rsquared did not improve from 0.27992
Epoch 21/200
3/3 [==============================] - 0s 60ms/step - loss: 128.9985 - rsquared: 0.1057 -
val_loss: 95.4064 - val_rsquared: 0.3405

Epoch 00021: val_rsquared improved from 0.27992 to 0.34051, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 22/200
3/3 [==============================] - 0s 67ms/step - loss: 113.0565 - rsquared: 0.2238 -
val_loss: 93.4564 - val_rsquared: 0.3549

Epoch 00022: val_rsquared improved from 0.34051 to 0.35492, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 23/200
3/3 [==============================] - 0s 84ms/step - loss: 104.0866 - rsquared: 0.2847 -
val_loss: 89.4747 - val_rsquared: 0.3800

Epoch 00023: val_rsquared improved from 0.35492 to 0.37998, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 24/200
3/3 [==============================] - 0s 66ms/step - loss: 103.9080 - rsquared: 0.2893 -
val_loss: 81.4669 - val_rsquared: 0.4394

Epoch 00024: val_rsquared improved from 0.37998 to 0.43936, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 25/200
3/3 [==============================] - 0s 78ms/step - loss: 90.9099 - rsquared: 0.3770 - val_loss:
83.3247 - val_rsquared: 0.4274

Epoch 00025: val_rsquared did not improve from 0.43936
Epoch 26/200
3/3 [==============================] - 0s 79ms/step - loss: 96.1800 - rsquared: 0.3530 - val_loss:
76.9323 - val_rsquared: 0.4727

Epoch 00026: val_rsquared improved from 0.43936 to 0.47265, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 27/200
3/3 [==============================] - 0s 80ms/step - loss: 87.3149 - rsquared: 0.3939 - val_loss:
77.8089 - val_rsquared: 0.4662

Epoch 00027: val_rsquared did not improve from 0.47265
Epoch 28/200
3/3 [==============================] - 0s 92ms/step - loss: 90.1345 - rsquared: 0.3893 - val_loss:
76.4617 - val_rsquared: 0.4778

Epoch 00028: val_rsquared improved from 0.47265 to 0.47776, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 29/200
3/3 [==============================] - 0s 74ms/step - loss: 84.8664 - rsquared: 0.4068 - val_loss:
74.2161 - val_rsquared: 0.4936

Epoch 00029: val_rsquared improved from 0.47776 to 0.49362, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 30/200
3/3 [==============================] - 0s 64ms/step - loss: 83.8918 - rsquared: 0.4263 - val_loss:
75.4038 - val_rsquared: 0.4843

Epoch 00030: val_rsquared did not improve from 0.49362
Epoch 31/200
3/3 [==============================] - 0s 93ms/step - loss: 86.5582 - rsquared: 0.4091 - val_loss:
73.5344 - val_rsquared: 0.4990

Epoch 00031: val_rsquared improved from 0.49362 to 0.49899, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 32/200
3/3 [==============================] - 0s 74ms/step - loss: 82.5278 - rsquared: 0.4307 - val_loss:
72.3219 - val_rsquared: 0.5071

Epoch 00032: val_rsquared improved from 0.49899 to 0.50707, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 33/200
3/3 [==============================] - 0s 86ms/step - loss: 80.6965 - rsquared: 0.4545 - val_loss:
71.9376 - val_rsquared: 0.5097

Epoch 00033: val_rsquared improved from 0.50707 to 0.50970, saving model to /tmp/checkpoint
```

```
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 34/200
3/3 [==============================] - 0s 63ms/step - loss: 80.4052 - rsquared: 0.4378 - val_loss:
71.0064 - val_rsquared: 0.5169

Epoch 00034: val_rsquared improved from 0.50970 to 0.51689, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 35/200
3/3 [==============================] - 0s 70ms/step - loss: 77.1847 - rsquared: 0.4735 - val_loss:
70.4507 - val_rsquared: 0.5208

Epoch 00035: val_rsquared improved from 0.51689 to 0.52078, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 36/200
3/3 [==============================] - 0s 65ms/step - loss: 81.4129 - rsquared: 0.4467 - val_loss:
70.1320 - val_rsquared: 0.5227

Epoch 00036: val_rsquared improved from 0.52078 to 0.52275, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 37/200
3/3 [==============================] - 0s 82ms/step - loss: 77.2593 - rsquared: 0.4629 - val_loss:
69.5968 - val_rsquared: 0.5266

Epoch 00037: val_rsquared improved from 0.52275 to 0.52661, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 38/200
3/3 [==============================] - 0s 67ms/step - loss: 76.9979 - rsquared: 0.4671 - val_loss:
69.6498 - val_rsquared: 0.5259

Epoch 00038: val_rsquared did not improve from 0.52661
Epoch 39/200
3/3 [==============================] - 0s 66ms/step - loss: 75.0152 - rsquared: 0.4917 - val_loss:
68.7835 - val_rsquared: 0.5324

Epoch 00039: val_rsquared improved from 0.52661 to 0.53238, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 40/200
3/3 [==============================] - 0s 72ms/step - loss: 75.0896 - rsquared: 0.4817 - val_loss:
69.1991 - val_rsquared: 0.5291

Epoch 00040: val_rsquared did not improve from 0.53238
Epoch 41/200
3/3 [==============================] - 0s 65ms/step - loss: 71.1458 - rsquared: 0.5093 - val_loss:
68.5306 - val_rsquared: 0.5339

Epoch 00041: val_rsquared improved from 0.53238 to 0.53387, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 42/200
3/3 [==============================] - 0s 71ms/step - loss: 75.1079 - rsquared: 0.4968 - val_loss:
68.3718 - val_rsquared: 0.5350

Epoch 00042: val_rsquared improved from 0.53387 to 0.53499, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 43/200
3/3 [==============================] - 0s 65ms/step - loss: 71.1604 - rsquared: 0.5071 - val_loss:
68.6292 - val_rsquared: 0.5330

Epoch 00043: val_rsquared did not improve from 0.53499
Epoch 44/200
3/3 [==============================] - 0s 82ms/step - loss: 71.1817 - rsquared: 0.5167 - val_loss:
67.8660 - val_rsquared: 0.5386

Epoch 00044: val_rsquared improved from 0.53499 to 0.53857, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 45/200
3/3 [==============================] - 0s 57ms/step - loss: 70.8315 - rsquared: 0.5201 - val_loss:
68.9349 - val_rsquared: 0.5309

Epoch 00045: val_rsquared did not improve from 0.53857
Epoch 46/200
3/3 [==============================] - 0s 74ms/step - loss: 71.2136 - rsquared: 0.5083 - val_loss:
67.8376 - val_rsquared: 0.5390

Epoch 00046: val_rsquared improved from 0.53857 to 0.53900, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 47/200
3/3 [==============================] - 0s 72ms/step - loss: 67.7787 - rsquared: 0.5407 - val_loss:
```

```
69.1060 - val_rsquared: 0.5297

Epoch 00047: val_rsquared did not improve from 0.53900
Epoch 48/200
3/3 [==============================] - 0s 74ms/step - loss: 66.1576 - rsquared: 0.5380 - val_loss: 
67.0180 - val_rsquared: 0.5450

Epoch 00048: val_rsquared improved from 0.53900 to 0.54501, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 49/200
3/3 [==============================] - 0s 64ms/step - loss: 66.3617 - rsquared: 0.5461 - val_loss: 
68.2759 - val_rsquared: 0.5358

Epoch 00049: val_rsquared did not improve from 0.54501
Epoch 50/200
3/3 [==============================] - 0s 58ms/step - loss: 65.8169 - rsquared: 0.5507 - val_loss: 
67.7525 - val_rsquared: 0.5399

Epoch 00050: val_rsquared did not improve from 0.54501
Epoch 51/200
3/3 [==============================] - 0s 57ms/step - loss: 66.7395 - rsquared: 0.5444 - val_loss: 
67.9638 - val_rsquared: 0.5383

Epoch 00051: val_rsquared did not improve from 0.54501
Epoch 52/200
3/3 [==============================] - 0s 60ms/step - loss: 66.3995 - rsquared: 0.5447 - val_loss: 
67.4911 - val_rsquared: 0.5417

Epoch 00052: val_rsquared did not improve from 0.54501
Epoch 53/200
3/3 [==============================] - 0s 85ms/step - loss: 62.1014 - rsquared: 0.5655 - val_loss: 
69.0038 - val_rsquared: 0.5309

Epoch 00053: val_rsquared did not improve from 0.54501
Epoch 54/200
3/3 [==============================] - 0s 63ms/step - loss: 67.5456 - rsquared: 0.5408 - val_loss: 
70.7501 - val_rsquared: 0.5183

Epoch 00054: val_rsquared did not improve from 0.54501
Epoch 55/200
3/3 [==============================] - 0s 65ms/step - loss: 65.8908 - rsquared: 0.5541 - val_loss: 
66.5192 - val_rsquared: 0.5491

Epoch 00055: val_rsquared improved from 0.54501 to 0.54913, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 56/200
3/3 [==============================] - 0s 66ms/step - loss: 63.7396 - rsquared: 0.5554 - val_loss: 
69.8151 - val_rsquared: 0.5247

Epoch 00056: val_rsquared did not improve from 0.54913
Epoch 57/200
3/3 [==============================] - 0s 64ms/step - loss: 64.8433 - rsquared: 0.5527 - val_loss: 
68.7004 - val_rsquared: 0.5327

Epoch 00057: val_rsquared did not improve from 0.54913
Epoch 58/200
3/3 [==============================] - 0s 64ms/step - loss: 61.5315 - rsquared: 0.5778 - val_loss: 
67.0625 - val_rsquared: 0.5447

Epoch 00058: val_rsquared did not improve from 0.54913
Epoch 59/200
3/3 [==============================] - 0s 59ms/step - loss: 64.2548 - rsquared: 0.5595 - val_loss: 
70.2925 - val_rsquared: 0.5210

Epoch 00059: val_rsquared did not improve from 0.54913
Epoch 60/200
3/3 [==============================] - 0s 61ms/step - loss: 62.9717 - rsquared: 0.5714 - val_loss: 
66.7742 - val_rsquared: 0.5467

Epoch 00060: val_rsquared did not improve from 0.54913
Epoch 61/200
3/3 [==============================] - 0s 90ms/step - loss: 59.9568 - rsquared: 0.5878 - val_loss: 
67.1131 - val_rsquared: 0.5444

Epoch 00061: val_rsquared did not improve from 0.54913
Epoch 62/200
3/3 [==============================] - 0s 74ms/step - loss: 61.8591 - rsquared: 0.5710 - val_loss: 
```

```
67.0458 - val_rsquared: 0.5448

Epoch 00062: val_rsquared did not improve from 0.54913
Epoch 63/200
3/3 [==============================] - 0s 67ms/step - loss: 60.0151 - rsquared: 0.5902 - val_loss:
68.1583 - val_rsquared: 0.5362

Epoch 00063: val_rsquared did not improve from 0.54913
Epoch 64/200
3/3 [==============================] - 0s 64ms/step - loss: 63.4159 - rsquared: 0.5696 - val_loss:
77.1320 - val_rsquared: 0.4703

Epoch 00064: val_rsquared did not improve from 0.54913
Epoch 65/200
3/3 [==============================] - 0s 67ms/step - loss: 61.4031 - rsquared: 0.5722 - val_loss:
65.1900 - val_rsquared: 0.5575

Epoch 00065: val_rsquared improved from 0.54913 to 0.55745, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 66/200
3/3 [==============================] - 0s 58ms/step - loss: 61.7343 - rsquared: 0.5748 - val_loss:
74.8183 - val_rsquared: 0.4872

Epoch 00066: val_rsquared did not improve from 0.55745
Epoch 67/200
3/3 [==============================] - 0s 90ms/step - loss: 61.7245 - rsquared: 0.5877 - val_loss:
71.4546 - val_rsquared: 0.5117

Epoch 00067: val_rsquared did not improve from 0.55745
Epoch 68/200
3/3 [==============================] - 0s 77ms/step - loss: 60.3507 - rsquared: 0.5799 - val_loss:
65.8219 - val_rsquared: 0.5531

Epoch 00068: val_rsquared did not improve from 0.55745
Epoch 69/200
3/3 [==============================] - 0s 73ms/step - loss: 62.9887 - rsquared: 0.5765 - val_loss:
80.1174 - val_rsquared: 0.4490

Epoch 00069: val_rsquared did not improve from 0.55745
Epoch 70/200
3/3 [==============================] - 0s 67ms/step - loss: 59.8719 - rsquared: 0.5828 - val_loss:
65.0544 - val_rsquared: 0.5586

Epoch 00070: val_rsquared improved from 0.55745 to 0.55857, saving model to /tmp/checkpoint
INFO:tensorflow:Assets written to: /tmp/checkpoint/assets
Epoch 71/200
3/3 [==============================] - 0s 86ms/step - loss: 61.0214 - rsquared: 0.5747 - val_loss:
79.1400 - val_rsquared: 0.4560

Epoch 00071: val_rsquared did not improve from 0.55857
Epoch 72/200
3/3 [==============================] - 0s 77ms/step - loss: 59.4106 - rsquared: 0.5874 - val_loss:
67.7525 - val_rsquared: 0.5391

Epoch 00072: val_rsquared did not improve from 0.55857
Epoch 73/200
3/3 [==============================] - 0s 81ms/step - loss: 57.6848 - rsquared: 0.5966 - val_loss:
69.5335 - val_rsquared: 0.5261

Epoch 00073: val_rsquared did not improve from 0.55857
Epoch 74/200
3/3 [==============================] - 0s 111ms/step - loss: 59.7862 - rsquared: 0.5909 -
val_loss: 78.5204 - val_rsquared: 0.4604

Epoch 00074: val_rsquared did not improve from 0.55857
Epoch 75/200
3/3 [==============================] - 0s 77ms/step - loss: 59.2446 - rsquared: 0.5908 - val_loss:
65.1455 - val_rsquared: 0.5580

Epoch 00075: val_rsquared did not improve from 0.55857
Epoch 76/200
3/3 [==============================] - 0s 89ms/step - loss: 59.1641 - rsquared: 0.5894 - val_loss:
77.8923 - val_rsquared: 0.4655

Epoch 00076: val_rsquared did not improve from 0.55857
Epoch 77/200
3/3 [==============================] - 0s 76ms/step - loss: 57.1644 - rsquared: 0.6101 - val_loss:
```

```
67.4149 - val_rsquared: 0.5415

Epoch 00077: val_rsquared did not improve from 0.55857
Epoch 78/200
3/3 [==============================] - 0s 66ms/step - loss: 57.3681 - rsquared: 0.6084 - val_loss:
74.3124 - val_rsquared: 0.4915

Epoch 00078: val_rsquared did not improve from 0.55857
Epoch 79/200
3/3 [==============================] - 0s 63ms/step - loss: 59.2638 - rsquared: 0.5956 - val_loss:
72.5151 - val_rsquared: 0.5046

Epoch 00079: val_rsquared did not improve from 0.55857
Epoch 80/200
3/3 [==============================] - 0s 112ms/step - loss: 58.0894 - rsquared: 0.6013 -
val_loss: 67.6984 - val_rsquared: 0.5395

Epoch 00080: val_rsquared did not improve from 0.55857
Epoch 81/200
3/3 [==============================] - 0s 76ms/step - loss: 56.1943 - rsquared: 0.6053 - val_loss:
75.2134 - val_rsquared: 0.4851

Epoch 00081: val_rsquared did not improve from 0.55857
Epoch 82/200
3/3 [==============================] - 0s 65ms/step - loss: 55.9839 - rsquared: 0.6140 - val_loss:
70.6613 - val_rsquared: 0.5180

Epoch 00082: val_rsquared did not improve from 0.55857
Epoch 83/200
3/3 [==============================] - 0s 62ms/step - loss: 53.9874 - rsquared: 0.6284 - val_loss:
72.7045 - val_rsquared: 0.5033

Epoch 00083: val_rsquared did not improve from 0.55857
Epoch 84/200
3/3 [==============================] - 0s 64ms/step - loss: 55.6942 - rsquared: 0.6202 - val_loss:
76.4739 - val_rsquared: 0.4759

Epoch 00084: val_rsquared did not improve from 0.55857
Epoch 85/200
3/3 [==============================] - 0s 59ms/step - loss: 55.9679 - rsquared: 0.6151 - val_loss:
68.3054 - val_rsquared: 0.5350

Epoch 00085: val_rsquared did not improve from 0.55857
Epoch 86/200
3/3 [==============================] - 0s 58ms/step - loss: 53.9984 - rsquared: 0.6273 - val_loss:
78.0263 - val_rsquared: 0.4650

Epoch 00086: val_rsquared did not improve from 0.55857
Epoch 87/200
3/3 [==============================] - 0s 59ms/step - loss: 55.6996 - rsquared: 0.6173 - val_loss:
72.6334 - val_rsquared: 0.5043

Epoch 00087: val_rsquared did not improve from 0.55857
Epoch 88/200
3/3 [==============================] - 0s 59ms/step - loss: 55.5067 - rsquared: 0.6247 - val_loss:
72.4031 - val_rsquared: 0.5060

Epoch 00088: val_rsquared did not improve from 0.55857
Epoch 89/200
3/3 [==============================] - 0s 60ms/step - loss: 56.3525 - rsquared: 0.6154 - val_loss:
75.6621 - val_rsquared: 0.4822

Epoch 00089: val_rsquared did not improve from 0.55857
Epoch 90/200
3/3 [==============================] - 0s 80ms/step - loss: 52.2207 - rsquared: 0.6274 - val_loss:
67.8651 - val_rsquared: 0.5388

Epoch 00090: val_rsquared did not improve from 0.55857
Epoch 91/200
3/3 [==============================] - 0s 89ms/step - loss: 54.7120 - rsquared: 0.6181 - val_loss:
82.4200 - val_rsquared: 0.4342

Epoch 00091: val_rsquared did not improve from 0.55857
Epoch 92/200
3/3 [==============================] - 0s 59ms/step - loss: 55.9617 - rsquared: 0.6256 - val_loss:
72.3376 - val_rsquared: 0.5072
```

```
Epoch 00092: val_rsquared did not improve from 0.55857
Epoch 93/200
3/3 [==============================] - 0s 59ms/step - loss: 53.7121 - rsquared: 0.6320 - val_loss: 71.8829 - val_rsquared: 0.5105

Epoch 00093: val_rsquared did not improve from 0.55857
Epoch 94/200
3/3 [==============================] - 0s 62ms/step - loss: 53.9575 - rsquared: 0.6327 - val_loss: 78.1009 - val_rsquared: 0.4655

Epoch 00094: val_rsquared did not improve from 0.55857
Epoch 95/200
3/3 [==============================] - 0s 64ms/step - loss: 54.2684 - rsquared: 0.6210 - val_loss: 70.7946 - val_rsquared: 0.5187

Epoch 00095: val_rsquared did not improve from 0.55857
Epoch 96/200
3/3 [==============================] - 0s 89ms/step - loss: 55.6678 - rsquared: 0.6200 - val_loss: 71.0826 - val_rsquared: 0.5161

Epoch 00096: val_rsquared did not improve from 0.55857
Epoch 97/200
3/3 [==============================] - 0s 67ms/step - loss: 53.3867 - rsquared: 0.6311 - val_loss: 76.7842 - val_rsquared: 0.4748

Epoch 00097: val_rsquared did not improve from 0.55857
Epoch 98/200
3/3 [==============================] - 0s 66ms/step - loss: 52.6063 - rsquared: 0.6413 - val_loss: 72.4915 - val_rsquared: 0.5056

Epoch 00098: val_rsquared did not improve from 0.55857
Epoch 99/200
3/3 [==============================] - 0s 64ms/step - loss: 55.2208 - rsquared: 0.6286 - val_loss: 79.3060 - val_rsquared: 0.4563

Epoch 00099: val_rsquared did not improve from 0.55857
Epoch 100/200
3/3 [==============================] - 0s 91ms/step - loss: 55.4752 - rsquared: 0.6312 - val_loss: 79.8149 - val_rsquared: 0.4526

Epoch 00100: val_rsquared did not improve from 0.55857
Epoch 101/200
3/3 [==============================] - 0s 65ms/step - loss: 53.1928 - rsquared: 0.6322 - val_loss: 69.8191 - val_rsquared: 0.5248

Epoch 00101: val_rsquared did not improve from 0.55857
Epoch 102/200
3/3 [==============================] - 0s 64ms/step - loss: 54.6802 - rsquared: 0.6204 - val_loss: 74.1511 - val_rsquared: 0.4938

Epoch 00102: val_rsquared did not improve from 0.55857
Epoch 103/200
3/3 [==============================] - 0s 65ms/step - loss: 51.1717 - rsquared: 0.6428 - val_loss: 77.2919 - val_rsquared: 0.4718

Epoch 00103: val_rsquared did not improve from 0.55857
Epoch 104/200
3/3 [==============================] - 0s 77ms/step - loss: 51.0362 - rsquared: 0.6504 - val_loss: 74.8077 - val_rsquared: 0.4902

Epoch 00104: val_rsquared did not improve from 0.55857
Epoch 105/200
3/3 [==============================] - 0s 72ms/step - loss: 53.0733 - rsquared: 0.6368 - val_loss: 77.8782 - val_rsquared: 0.4682

Epoch 00105: val_rsquared did not improve from 0.55857
Epoch 106/200
3/3 [==============================] - 0s 65ms/step - loss: 53.4517 - rsquared: 0.6416 - val_loss: 81.8952 - val_rsquared: 0.4390

Epoch 00106: val_rsquared did not improve from 0.55857
Epoch 107/200
3/3 [==============================] - 0s 66ms/step - loss: 49.3913 - rsquared: 0.6515 - val_loss: 73.0498 - val_rsquared: 0.5032

Epoch 00107: val_rsquared did not improve from 0.55857
Epoch 108/200
```

```
3/3 [==============================] - 0s 64ms/step - loss: 51.9417 - rsquared: 0.6446 - val_loss: 76.2555 - val_rsquared: 0.4804

Epoch 00108: val_rsquared did not improve from 0.55857
Epoch 109/200
3/3 [==============================] - 0s 63ms/step - loss: 50.6972 - rsquared: 0.6530 - val_loss: 74.8916 - val_rsquared: 0.4907

Epoch 00109: val_rsquared did not improve from 0.55857
Epoch 110/200
3/3 [==============================] - 0s 73ms/step - loss: 50.5062 - rsquared: 0.6602 - val_loss: 79.2534 - val_rsquared: 0.4593

Epoch 00110: val_rsquared did not improve from 0.55857
Epoch 111/200
3/3 [==============================] - 0s 60ms/step - loss: 50.5370 - rsquared: 0.6522 - val_loss: 74.7931 - val_rsquared: 0.4916

Epoch 00111: val_rsquared did not improve from 0.55857
Epoch 112/200
3/3 [==============================] - 0s 59ms/step - loss: 53.0009 - rsquared: 0.6444 - val_loss: 73.8821 - val_rsquared: 0.4989

Epoch 00112: val_rsquared did not improve from 0.55857
Epoch 113/200
3/3 [==============================] - 0s 60ms/step - loss: 49.7991 - rsquared: 0.6565 - val_loss: 77.2740 - val_rsquared: 0.4742

Epoch 00113: val_rsquared did not improve from 0.55857
Epoch 114/200
3/3 [==============================] - 0s 58ms/step - loss: 52.7547 - rsquared: 0.6414 - val_loss: 78.1639 - val_rsquared: 0.4676

Epoch 00114: val_rsquared did not improve from 0.55857
Epoch 115/200
3/3 [==============================] - 0s 66ms/step - loss: 49.3940 - rsquared: 0.6581 - val_loss: 71.8377 - val_rsquared: 0.5126

Epoch 00115: val_rsquared did not improve from 0.55857
Epoch 116/200
3/3 [==============================] - 0s 60ms/step - loss: 52.2651 - rsquared: 0.6497 - val_loss: 84.5959 - val_rsquared: 0.4206

Epoch 00116: val_rsquared did not improve from 0.55857
Epoch 117/200
3/3 [==============================] - 0s 58ms/step - loss: 49.6199 - rsquared: 0.6471 - val_loss: 71.0078 - val_rsquared: 0.5192

Epoch 00117: val_rsquared did not improve from 0.55857
Epoch 118/200
3/3 [==============================] - 0s 59ms/step - loss: 49.7052 - rsquared: 0.6603 - val_loss: 74.2935 - val_rsquared: 0.4963

Epoch 00118: val_rsquared did not improve from 0.55857
Epoch 119/200
3/3 [==============================] - 0s 111ms/step - loss: 50.8790 - rsquared: 0.6546 - val_loss: 83.7192 - val_rsquared: 0.4288

Epoch 00119: val_rsquared did not improve from 0.55857
Epoch 120/200
3/3 [==============================] - 0s 88ms/step - loss: 49.9076 - rsquared: 0.6645 - val_loss: 85.2839 - val_rsquared: 0.4176

Epoch 00120: val_rsquared did not improve from 0.55857
Epoch 121/200
3/3 [==============================] - 0s 64ms/step - loss: 51.4885 - rsquared: 0.6548 - val_loss: 76.4137 - val_rsquared: 0.4815

Epoch 00121: val_rsquared did not improve from 0.55857
Epoch 122/200
3/3 [==============================] - 0s 82ms/step - loss: 50.5179 - rsquared: 0.6584 - val_loss: 85.8734 - val_rsquared: 0.4134

Epoch 00122: val_rsquared did not improve from 0.55857
Epoch 123/200
3/3 [==============================] - 0s 63ms/step - loss: 49.7220 - rsquared: 0.6630 - val_loss: 79.0272 - val_rsquared: 0.4627
```

```
Epoch 00123: val_rsquared did not improve from 0.55857
Epoch 124/200
3/3 [==============================] - 0s 69ms/step - loss: 46.8006 - rsquared: 0.6733 - val_loss: 
69.7140 - val_rsquared: 0.5297

Epoch 00124: val_rsquared did not improve from 0.55857
Epoch 125/200
3/3 [==============================] - 0s 87ms/step - loss: 48.3959 - rsquared: 0.6648 - val_loss: 
80.4410 - val_rsquared: 0.4527

Epoch 00125: val_rsquared did not improve from 0.55857
Epoch 126/200
3/3 [==============================] - 0s 76ms/step - loss: 46.2929 - rsquared: 0.6814 - val_loss: 
87.4756 - val_rsquared: 0.4013

Epoch 00126: val_rsquared did not improve from 0.55857
Epoch 127/200
3/3 [==============================] - 0s 64ms/step - loss: 47.2261 - rsquared: 0.6737 - val_loss: 
73.6424 - val_rsquared: 0.5004

Epoch 00127: val_rsquared did not improve from 0.55857
Epoch 128/200
3/3 [==============================] - 0s 66ms/step - loss: 46.3031 - rsquared: 0.6761 - val_loss: 
75.9006 - val_rsquared: 0.4841

Epoch 00128: val_rsquared did not improve from 0.55857
Epoch 129/200
3/3 [==============================] - 0s 73ms/step - loss: 47.4016 - rsquared: 0.6816 - val_loss: 
91.3664 - val_rsquared: 0.3730

Epoch 00129: val_rsquared did not improve from 0.55857
Epoch 130/200
3/3 [==============================] - 0s 65ms/step - loss: 48.8505 - rsquared: 0.6609 - val_loss: 
84.9563 - val_rsquared: 0.4204

Epoch 00130: val_rsquared did not improve from 0.55857
Epoch 131/200
3/3 [==============================] - 0s 65ms/step - loss: 45.6900 - rsquared: 0.6858 - val_loss: 
75.0424 - val_rsquared: 0.4919

Epoch 00131: val_rsquared did not improve from 0.55857
Epoch 132/200
3/3 [==============================] - 0s 66ms/step - loss: 47.9710 - rsquared: 0.6762 - val_loss: 
86.7497 - val_rsquared: 0.4079

Epoch 00132: val_rsquared did not improve from 0.55857
Epoch 133/200
3/3 [==============================] - 0s 79ms/step - loss: 45.3910 - rsquared: 0.6852 - val_loss: 
80.3070 - val_rsquared: 0.4542

Epoch 00133: val_rsquared did not improve from 0.55857
Epoch 134/200
3/3 [==============================] - 0s 67ms/step - loss: 47.1064 - rsquared: 0.6780 - val_loss: 
82.8414 - val_rsquared: 0.4361

Epoch 00134: val_rsquared did not improve from 0.55857
Epoch 135/200
3/3 [==============================] - 0s 65ms/step - loss: 48.6565 - rsquared: 0.6665 - val_loss: 
95.2016 - val_rsquared: 0.3466

Epoch 00135: val_rsquared did not improve from 0.55857
Epoch 136/200
3/3 [==============================] - 0s 66ms/step - loss: 46.1335 - rsquared: 0.6852 - val_loss: 
76.2901 - val_rsquared: 0.4823

Epoch 00136: val_rsquared did not improve from 0.55857
Epoch 137/200
3/3 [==============================] - 0s 83ms/step - loss: 46.0979 - rsquared: 0.6932 - val_loss: 
82.9736 - val_rsquared: 0.4337

Epoch 00137: val_rsquared did not improve from 0.55857
Epoch 138/200
3/3 [==============================] - 0s 65ms/step - loss: 46.1814 - rsquared: 0.6858 - val_loss: 
89.2286 - val_rsquared: 0.3892

Epoch 00138: val_rsquared did not improve from 0.55857
```

```
Epoch 139/200
3/3 [==============================] - 0s 82ms/step - loss: 45.7228 - rsquared: 0.6821 - val_loss:
74.0043 - val_rsquared: 0.4989

Epoch 00139: val_rsquared did not improve from 0.55857
Epoch 140/200
3/3 [==============================] - 0s 66ms/step - loss: 47.3584 - rsquared: 0.6728 - val_loss:
80.5739 - val_rsquared: 0.4524

Epoch 00140: val_rsquared did not improve from 0.55857
Epoch 141/200
3/3 [==============================] - 0s 65ms/step - loss: 46.2628 - rsquared: 0.6851 - val_loss:
101.1057 - val_rsquared: 0.3047

Epoch 00141: val_rsquared did not improve from 0.55857
Epoch 142/200
3/3 [==============================] - 0s 87ms/step - loss: 49.3752 - rsquared: 0.6640 - val_loss:
82.7865 - val_rsquared: 0.4376

Epoch 00142: val_rsquared did not improve from 0.55857
Epoch 143/200
3/3 [==============================] - 0s 64ms/step - loss: 46.2444 - rsquared: 0.6863 - val_loss:
78.4658 - val_rsquared: 0.4683

Epoch 00143: val_rsquared did not improve from 0.55857
Epoch 144/200
3/3 [==============================] - 0s 64ms/step - loss: 46.5149 - rsquared: 0.6810 - val_loss:
82.8297 - val_rsquared: 0.4366

Epoch 00144: val_rsquared did not improve from 0.55857
Epoch 145/200
3/3 [==============================] - 0s 102ms/step - loss: 45.2890 - rsquared: 0.6879 -
val_loss: 89.3104 - val_rsquared: 0.3900

Epoch 00145: val_rsquared did not improve from 0.55857
Epoch 146/200
3/3 [==============================] - 0s 67ms/step - loss: 46.0237 - rsquared: 0.6922 - val_loss:
96.1804 - val_rsquared: 0.3407

Epoch 00146: val_rsquared did not improve from 0.55857
Epoch 147/200
3/3 [==============================] - 0s 70ms/step - loss: 46.9504 - rsquared: 0.6815 - val_loss:
92.5646 - val_rsquared: 0.3672

Epoch 00147: val_rsquared did not improve from 0.55857
Epoch 148/200
3/3 [==============================] - 0s 68ms/step - loss: 46.7145 - rsquared: 0.6778 - val_loss:
74.8287 - val_rsquared: 0.4953

Epoch 00148: val_rsquared did not improve from 0.55857
Epoch 149/200
3/3 [==============================] - 0s 64ms/step - loss: 47.2984 - rsquared: 0.6803 - val_loss:
76.5572 - val_rsquared: 0.4833

Epoch 00149: val_rsquared did not improve from 0.55857
Epoch 150/200
3/3 [==============================] - 0s 86ms/step - loss: 45.5978 - rsquared: 0.6867 - val_loss:
99.8910 - val_rsquared: 0.3149

Epoch 00150: val_rsquared did not improve from 0.55857
Epoch 151/200
3/3 [==============================] - 0s 74ms/step - loss: 47.6372 - rsquared: 0.6698 - val_loss:
82.1617 - val_rsquared: 0.4419

Epoch 00151: val_rsquared did not improve from 0.55857
Epoch 152/200
3/3 [==============================] - 0s 84ms/step - loss: 43.8164 - rsquared: 0.7010 - val_loss:
76.4029 - val_rsquared: 0.4826

Epoch 00152: val_rsquared did not improve from 0.55857
Epoch 153/200
3/3 [==============================] - 0s 74ms/step - loss: 43.7410 - rsquared: 0.7001 - val_loss:
98.3576 - val_rsquared: 0.3251

Epoch 00153: val_rsquared did not improve from 0.55857
Epoch 154/200
3/3 [==============================] - 0s 66ms/step - loss: 43.4161 - rsquared: 0.7016 - val_loss:
```

```
91.5270 - val_rsquared: 0.3756

Epoch 00154: val_rsquared did not improve from 0.55857
Epoch 155/200
3/3 [==============================] - 0s 77ms/step - loss: 43.7051 - rsquared: 0.6976 - val_loss:
80.9335 - val_rsquared: 0.4528

Epoch 00155: val_rsquared did not improve from 0.55857
Epoch 156/200
3/3 [==============================] - 0s 63ms/step - loss: 43.7908 - rsquared: 0.6998 - val_loss:
90.0024 - val_rsquared: 0.3872

Epoch 00156: val_rsquared did not improve from 0.55857
Epoch 157/200
3/3 [==============================] - 0s 90ms/step - loss: 42.0049 - rsquared: 0.7118 - val_loss:
91.6177 - val_rsquared: 0.3751

Epoch 00157: val_rsquared did not improve from 0.55857
Epoch 158/200
3/3 [==============================] - 0s 66ms/step - loss: 44.5883 - rsquared: 0.6884 - val_loss:
86.7724 - val_rsquared: 0.4105

Epoch 00158: val_rsquared did not improve from 0.55857
Epoch 159/200
3/3 [==============================] - 0s 97ms/step - loss: 42.7587 - rsquared: 0.7068 - val_loss:
81.9646 - val_rsquared: 0.4454

Epoch 00159: val_rsquared did not improve from 0.55857
Epoch 160/200
3/3 [==============================] - 0s 88ms/step - loss: 42.2717 - rsquared: 0.7044 - val_loss:
81.5969 - val_rsquared: 0.4479

Epoch 00160: val_rsquared did not improve from 0.55857
Epoch 161/200
3/3 [==============================] - 0s 66ms/step - loss: 42.6781 - rsquared: 0.7054 - val_loss:
88.7645 - val_rsquared: 0.3961

Epoch 00161: val_rsquared did not improve from 0.55857
Epoch 162/200
3/3 [==============================] - 0s 94ms/step - loss: 41.6320 - rsquared: 0.7150 - val_loss:
91.8355 - val_rsquared: 0.3737

Epoch 00162: val_rsquared did not improve from 0.55857
Epoch 163/200
3/3 [==============================] - 0s 75ms/step - loss: 41.0678 - rsquared: 0.7125 - val_loss:
85.0970 - val_rsquared: 0.4229

Epoch 00163: val_rsquared did not improve from 0.55857
Epoch 164/200
3/3 [==============================] - 0s 66ms/step - loss: 40.0109 - rsquared: 0.7244 - val_loss:
87.7767 - val_rsquared: 0.4045

Epoch 00164: val_rsquared did not improve from 0.55857
Epoch 165/200
3/3 [==============================] - 0s 65ms/step - loss: 42.3820 - rsquared: 0.7106 - val_loss:
89.0937 - val_rsquared: 0.3955

Epoch 00165: val_rsquared did not improve from 0.55857
Epoch 166/200
3/3 [==============================] - 0s 80ms/step - loss: 42.1718 - rsquared: 0.7105 - val_loss:
96.5374 - val_rsquared: 0.3419

Epoch 00166: val_rsquared did not improve from 0.55857
Epoch 167/200
3/3 [==============================] - 0s 67ms/step - loss: 40.2974 - rsquared: 0.7175 - val_loss:
85.8999 - val_rsquared: 0.4187

Epoch 00167: val_rsquared did not improve from 0.55857
Epoch 168/200
3/3 [==============================] - 0s 58ms/step - loss: 41.5300 - rsquared: 0.7139 - val_loss:
86.9287 - val_rsquared: 0.4110

Epoch 00168: val_rsquared did not improve from 0.55857
Epoch 169/200
3/3 [==============================] - 0s 60ms/step - loss: 40.3036 - rsquared: 0.7216 - val_loss:
96.2006 - val_rsquared: 0.3434
```

```
Epoch 00169: val_rsquared did not improve from 0.55857
Epoch 170/200
3/3 [==============================] - 0s 90ms/step - loss: 40.0654 - rsquared: 0.7231 - val_loss:
91.8136 - val_rsquared: 0.3762

Epoch 00170: val_rsquared did not improve from 0.55857
Epoch 171/200
3/3 [==============================] - 0s 65ms/step - loss: 39.0594 - rsquared: 0.7331 - val_loss:
98.4182 - val_rsquared: 0.3293

Epoch 00171: val_rsquared did not improve from 0.55857
Epoch 172/200
3/3 [==============================] - 0s 74ms/step - loss: 38.8313 - rsquared: 0.7325 - val_loss:
96.8621 - val_rsquared: 0.3404

Epoch 00172: val_rsquared did not improve from 0.55857
Epoch 173/200
3/3 [==============================] - 0s 78ms/step - loss: 41.1149 - rsquared: 0.7188 - val_loss:
94.7509 - val_rsquared: 0.3546

Epoch 00173: val_rsquared did not improve from 0.55857
Epoch 174/200
3/3 [==============================] - 0s 73ms/step - loss: 37.8668 - rsquared: 0.7351 - val_loss:
80.9984 - val_rsquared: 0.4535

Epoch 00174: val_rsquared did not improve from 0.55857
Epoch 175/200
3/3 [==============================] - 0s 86ms/step - loss: 42.4518 - rsquared: 0.7100 - val_loss:
88.2593 - val_rsquared: 0.4014

Epoch 00175: val_rsquared did not improve from 0.55857
Epoch 176/200
3/3 [==============================] - 0s 86ms/step - loss: 38.9921 - rsquared: 0.7333 - val_loss:
97.8620 - val_rsquared: 0.3329

Epoch 00176: val_rsquared did not improve from 0.55857
Epoch 177/200
3/3 [==============================] - 0s 82ms/step - loss: 41.7886 - rsquared: 0.7165 - val_loss:
103.0694 - val_rsquared: 0.2950

Epoch 00177: val_rsquared did not improve from 0.55857
Epoch 178/200
3/3 [==============================] - 0s 72ms/step - loss: 40.9358 - rsquared: 0.7193 - val_loss:
97.3295 - val_rsquared: 0.3356

Epoch 00178: val_rsquared did not improve from 0.55857
Epoch 179/200
3/3 [==============================] - 0s 87ms/step - loss: 40.3479 - rsquared: 0.7226 - val_loss:
89.5533 - val_rsquared: 0.3927

Epoch 00179: val_rsquared did not improve from 0.55857
Epoch 180/200
3/3 [==============================] - 0s 64ms/step - loss: 37.8009 - rsquared: 0.7349 - val_loss:
88.4306 - val_rsquared: 0.4016

Epoch 00180: val_rsquared did not improve from 0.55857
Epoch 181/200
3/3 [==============================] - 0s 76ms/step - loss: 39.3603 - rsquared: 0.7308 - val_loss:
92.2603 - val_rsquared: 0.3746

Epoch 00181: val_rsquared did not improve from 0.55857
Epoch 182/200
3/3 [==============================] - 0s 79ms/step - loss: 39.4529 - rsquared: 0.7296 - val_loss:
100.1925 - val_rsquared: 0.3169

Epoch 00182: val_rsquared did not improve from 0.55857
Epoch 183/200
3/3 [==============================] - 0s 64ms/step - loss: 37.5067 - rsquared: 0.7413 - val_loss:
96.7032 - val_rsquared: 0.3417

Epoch 00183: val_rsquared did not improve from 0.55857
Epoch 184/200
3/3 [==============================] - 0s 90ms/step - loss: 38.7727 - rsquared: 0.7324 - val_loss:
89.5015 - val_rsquared: 0.3946

Epoch 00184: val_rsquared did not improve from 0.55857
Epoch 185/200
```

```
Epoch 185/200
3/3 [==============================] - 0s 106ms/step - loss: 39.4371 - rsquared: 0.7238 -
val_loss: 94.6274 - val_rsquared: 0.3591

Epoch 00185: val_rsquared did not improve from 0.55857
Epoch 186/200
3/3 [==============================] - 0s 69ms/step - loss: 39.0207 - rsquared: 0.7343 - val_loss:
102.6150 - val_rsquared: 0.3020

Epoch 00186: val_rsquared did not improve from 0.55857
Epoch 187/200
3/3 [==============================] - 0s 81ms/step - loss: 38.3621 - rsquared: 0.7339 - val_loss:
99.3175 - val_rsquared: 0.3254

Epoch 00187: val_rsquared did not improve from 0.55857
Epoch 188/200
3/3 [==============================] - 0s 90ms/step - loss: 37.6014 - rsquared: 0.7422 - val_loss:
103.5989 - val_rsquared: 0.2947

Epoch 00188: val_rsquared did not improve from 0.55857
Epoch 189/200
3/3 [==============================] - 0s 80ms/step - loss: 40.8111 - rsquared: 0.7237 - val_loss:
103.3889 - val_rsquared: 0.2966

Epoch 00189: val_rsquared did not improve from 0.55857
Epoch 190/200
3/3 [==============================] - 0s 66ms/step - loss: 38.2734 - rsquared: 0.7388 - val_loss:
101.8172 - val_rsquared: 0.3075

Epoch 00190: val_rsquared did not improve from 0.55857
Epoch 191/200
3/3 [==============================] - 0s 84ms/step - loss: 37.9822 - rsquared: 0.7368 - val_loss:
95.0816 - val_rsquared: 0.3555

Epoch 00191: val_rsquared did not improve from 0.55857
Epoch 192/200
3/3 [==============================] - 0s 91ms/step - loss: 38.5661 - rsquared: 0.7373 - val_loss:
103.3661 - val_rsquared: 0.2968

Epoch 00192: val_rsquared did not improve from 0.55857
Epoch 193/200
3/3 [==============================] - 0s 64ms/step - loss: 36.5188 - rsquared: 0.7469 - val_loss:
97.9688 - val_rsquared: 0.3355

Epoch 00193: val_rsquared did not improve from 0.55857
Epoch 194/200
3/3 [==============================] - 0s 69ms/step - loss: 37.4633 - rsquared: 0.7393 - val_loss:
98.8919 - val_rsquared: 0.3280

Epoch 00194: val_rsquared did not improve from 0.55857
Epoch 195/200
3/3 [==============================] - 0s 100ms/step - loss: 38.0235 - rsquared: 0.7395 -
val_loss: 114.5364 - val_rsquared: 0.2159

Epoch 00195: val_rsquared did not improve from 0.55857
Epoch 196/200
3/3 [==============================] - 0s 65ms/step - loss: 38.9999 - rsquared: 0.7289 - val_loss:
112.1199 - val_rsquared: 0.2340

Epoch 00196: val_rsquared did not improve from 0.55857
Epoch 197/200
3/3 [==============================] - 0s 71ms/step - loss: 38.6595 - rsquared: 0.7335 - val_loss:
92.2749 - val_rsquared: 0.3765

Epoch 00197: val_rsquared did not improve from 0.55857
Epoch 198/200
3/3 [==============================] - 0s 94ms/step - loss: 39.8894 - rsquared: 0.7260 - val_loss:
84.5297 - val_rsquared: 0.4332

Epoch 00198: val_rsquared did not improve from 0.55857
Epoch 199/200
3/3 [==============================] - 0s 71ms/step - loss: 40.4082 - rsquared: 0.7174 - val_loss:
91.2670 - val_rsquared: 0.3845

Epoch 00199: val_rsquared did not improve from 0.55857
Epoch 200/200
3/3 [==============================] - 0s 67ms/step - loss: 36.5855 - rsquared: 0.7471 - val_loss:
101.1644 - val_rsquared: 0.3139
```

```
101.1644 - val_rsquared: 0.3139

Epoch 00200: val_rsquared did not improve from 0.55857
```

Out[171]:

```
<tensorflow.python.keras.callbacks.History at 0x7fa63dcbcdf0>
```

In [214]:

```
score10 = 0.5587
```

With Feature Set - 5

In [177]:

```
X_Set5 = pd.DataFrame(np.hstack((x,x_svd)))
print(X_Set5.shape)
```

```
(4196, 528)
```

In [178]:

```
X_train_MLP_Set5, X_test_MLP_Set5, y_train, y_test = train_test_split(X_Set5, y, test_size=0.33, ra
ndom_state=1)
print("Done")
```

```
Done
```

In [179]:

```
input_dim = X_train_MLP_Set5.shape[1]

# The Input Layer :
model = Sequential()
model.add(Dense(128,kernel_initializer='normal', input_dim=input_dim, activation='relu'))

# The Hidden Layers :
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dropout(0.15))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
model.add(Dense(256, kernel_initializer='normal',activation='relu'))
# The Output Layer :
model.add(Dense(1, kernel_initializer='normal',activation='linear'))


model.compile(loss='mean_squared_error', optimizer='adam', metrics=[rsquared])
model.summary()
```

```
Model: "sequential_17"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_131 (Dense)            (None, 128)               67712
_________________________________________________________________
dense_132 (Dense)            (None, 256)               33024
_________________________________________________________________
dense_133 (Dense)            (None, 256)               65792
_________________________________________________________________
dense_134 (Dense)            (None, 256)               65792
_________________________________________________________________
dropout_12 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_135 (Dense)            (None, 256)               65792
_________________________________________________________________
dense_136 (Dense)            (None, 256)               65792
_________________________________________________________________
dense_137 (Dense)            (None, 1)                 257
```

```
=================================================================
Total params: 364,161
Trainable params: 364,161
Non-trainable params: 0
_________________________________________________________________
```

In [180]:

```
filepath="/tmp/checkpoint2"
checkpoint = ModelCheckpoint(filepath=filepath, monitor='val_rsquared', verbose=1, save_best_only=True, mode='max')

optimizer = tf.keras.optimizers.Adam(0.01)


#time = datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
log_dir= "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=log_dir,histogram_freq=1, write_graph=True,write_grads=True)

callbacks_list = [checkpoint,tensorboard_callback]

model.fit(X_train_MLP_Set5,y_train,epochs=200, validation_data=(X_test_MLP_Set5,y_test), batch_size=1000, callbacks=callbacks_list)
```

```
WARNING:tensorflow:`write_grads` will be ignored in TensorFlow 2.0 for the `TensorBoard` Callback.
Epoch 1/200
3/3 [==============================] - 1s 201ms/step - loss: 10214.4258 - rsquared: -68.6927 - val_loss: 9756.2100 - val_rsquared: -70.1243

Epoch 00001: val_rsquared improved from -inf to -70.12425, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 2/200
3/3 [==============================] - 0s 62ms/step - loss: 9636.3743 - rsquared: -63.5476 - val_loss: 8004.1074 - val_rsquared: -57.4111

Epoch 00002: val_rsquared improved from -70.12425 to -57.41114, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 3/200
3/3 [==============================] - 0s 64ms/step - loss: 7195.3334 - rsquared: -48.1186 - val_loss: 2582.0291 - val_rsquared: -17.9900

Epoch 00003: val_rsquared improved from -57.41114 to -17.98996, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 4/200
3/3 [==============================] - 0s 59ms/step - loss: 1941.3082 - rsquared: -12.4506 - val_loss: 2828.3259 - val_rsquared: -19.1378

Epoch 00004: val_rsquared did not improve from -17.98996
Epoch 5/200
3/3 [==============================] - 0s 65ms/step - loss: 2172.7600 - rsquared: -13.9567 - val_loss: 498.8972 - val_rsquared: -2.6868

Epoch 00005: val_rsquared improved from -17.98996 to -2.68678, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 6/200
3/3 [==============================] - 0s 72ms/step - loss: 754.6315 - rsquared: -4.3003 - val_loss: 1433.2096 - val_rsquared: -9.5800

Epoch 00006: val_rsquared did not improve from -2.68678
Epoch 7/200
3/3 [==============================] - 0s 62ms/step - loss: 1412.8954 - rsquared: -8.4879 - val_loss: 785.8000 - val_rsquared: -4.8199

Epoch 00007: val_rsquared did not improve from -2.68678
Epoch 8/200
3/3 [==============================] - 0s 68ms/step - loss: 665.3932 - rsquared: -3.4842 - val_loss: 498.0602 - val_rsquared: -2.5393

Epoch 00008: val_rsquared improved from -2.68678 to -2.53931, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 9/200
3/3 [==============================] - 0s 63ms/step - loss: 649.3281 - rsquared: -3.4635 - val_loss: 617.5739 - val_rsquared: -3.3761

Epoch 00009: val_rsquared did not improve from -2.53931
```

```
Epoch 00009: val_rsquared did not improve from -2.53931
Epoch 10/200
3/3 [==============================] - 0s 63ms/step - loss: 550.5984 - rsquared: -2.7879 -
val_loss: 335.7969 - val_rsquared: -1.4705

Epoch 00010: val_rsquared improved from -2.53931 to -1.47054, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 11/200
3/3 [==============================] - 0s 61ms/step - loss: 422.2456 - rsquared: -1.9180 -
val_loss: 459.4618 - val_rsquared: -2.3916

Epoch 00011: val_rsquared did not improve from -1.47054
Epoch 12/200
3/3 [==============================] - 0s 64ms/step - loss: 453.0011 - rsquared: -1.9873 -
val_loss: 209.4948 - val_rsquared: -0.5093

Epoch 00012: val_rsquared improved from -1.47054 to -0.50927, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 13/200
3/3 [==============================] - 0s 61ms/step - loss: 263.0817 - rsquared: -0.7956 -
val_loss: 314.5125 - val_rsquared: -1.2218

Epoch 00013: val_rsquared did not improve from -0.50927
Epoch 14/200
3/3 [==============================] - 0s 63ms/step - loss: 304.4309 - rsquared: -1.0552 -
val_loss: 164.8149 - val_rsquared: -0.1794

Epoch 00014: val_rsquared improved from -0.50927 to -0.17938, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 15/200
3/3 [==============================] - 0s 60ms/step - loss: 203.4211 - rsquared: -0.4153 -
val_loss: 208.0563 - val_rsquared: -0.5108

Epoch 00015: val_rsquared did not improve from -0.17938
Epoch 16/200
3/3 [==============================] - 0s 63ms/step - loss: 224.0285 - rsquared: -0.5184 -
val_loss: 129.4624 - val_rsquared: 0.0857

Epoch 00016: val_rsquared improved from -0.17938 to 0.08568, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 17/200
3/3 [==============================] - 0s 76ms/step - loss: 162.4594 - rsquared: -0.1271 -
val_loss: 154.3812 - val_rsquared: -0.0846

Epoch 00017: val_rsquared did not improve from 0.08568
Epoch 18/200
3/3 [==============================] - 0s 64ms/step - loss: 160.9762 - rsquared: -0.1201 -
val_loss: 115.4702 - val_rsquared: 0.1845

Epoch 00018: val_rsquared improved from 0.08568 to 0.18454, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 19/200
3/3 [==============================] - 0s 61ms/step - loss: 144.8060 - rsquared: 0.0071 -
val_loss: 115.1973 - val_rsquared: 0.1866

Epoch 00019: val_rsquared improved from 0.18454 to 0.18661, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 20/200
3/3 [==============================] - 0s 78ms/step - loss: 133.6477 - rsquared: 0.0735 -
val_loss: 110.0284 - val_rsquared: 0.2310

Epoch 00020: val_rsquared improved from 0.18661 to 0.23098, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 21/200
3/3 [==============================] - 0s 82ms/step - loss: 125.9330 - rsquared: 0.1306 -
val_loss: 96.5357 - val_rsquared: 0.3271

Epoch 00021: val_rsquared improved from 0.23098 to 0.32712, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 22/200
3/3 [==============================] - 0s 68ms/step - loss: 111.2948 - rsquared: 0.2363 -
val_loss: 98.7842 - val_rsquared: 0.3093

Epoch 00022: val_rsquared did not improve from 0.32712
Epoch 23/200
3/3 [==============================] - 0s 66ms/step - loss: 111.5426 - rsquared: 0.2299 -
val_loss: 89.7993 - val_rsquared: 0.3761
```

```
Epoch 00023: val_rsquared improved from 0.32712 to 0.37615, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 24/200
3/3 [==============================] - 0s 58ms/step - loss: 105.0301 - rsquared: 0.2922 -
val_loss: 88.0242 - val_rsquared: 0.3892

Epoch 00024: val_rsquared improved from 0.37615 to 0.38923, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 25/200
3/3 [==============================] - 0s 59ms/step - loss: 100.3643 - rsquared: 0.3210 -
val_loss: 85.9408 - val_rsquared: 0.4041

Epoch 00025: val_rsquared improved from 0.38923 to 0.40413, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 26/200
3/3 [==============================] - 0s 80ms/step - loss: 97.7162 - rsquared: 0.3383 - val_loss:
82.7151 - val_rsquared: 0.4290

Epoch 00026: val_rsquared improved from 0.40413 to 0.42899, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 27/200
3/3 [==============================] - 0s 67ms/step - loss: 94.4690 - rsquared: 0.3362 - val_loss:
80.5978 - val_rsquared: 0.4447

Epoch 00027: val_rsquared improved from 0.42899 to 0.44467, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 28/200
3/3 [==============================] - 0s 80ms/step - loss: 91.7069 - rsquared: 0.3689 - val_loss:
79.6083 - val_rsquared: 0.4522

Epoch 00028: val_rsquared improved from 0.44467 to 0.45220, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 29/200
3/3 [==============================] - 0s 72ms/step - loss: 86.3275 - rsquared: 0.4145 - val_loss:
78.2499 - val_rsquared: 0.4632

Epoch 00029: val_rsquared improved from 0.45220 to 0.46316, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 30/200
3/3 [==============================] - 0s 77ms/step - loss: 85.6729 - rsquared: 0.4085 - val_loss:
77.2031 - val_rsquared: 0.4708

Epoch 00030: val_rsquared improved from 0.46316 to 0.47084, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 31/200
3/3 [==============================] - 0s 62ms/step - loss: 83.7433 - rsquared: 0.4191 - val_loss:
76.3970 - val_rsquared: 0.4770

Epoch 00031: val_rsquared improved from 0.47084 to 0.47699, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 32/200
3/3 [==============================] - 0s 79ms/step - loss: 81.8454 - rsquared: 0.4354 - val_loss:
75.7652 - val_rsquared: 0.4814

Epoch 00032: val_rsquared improved from 0.47699 to 0.48145, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 33/200
3/3 [==============================] - 0s 74ms/step - loss: 83.9043 - rsquared: 0.4237 - val_loss:
75.0990 - val_rsquared: 0.4863

Epoch 00033: val_rsquared improved from 0.48145 to 0.48629, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 34/200
3/3 [==============================] - 0s 75ms/step - loss: 83.5861 - rsquared: 0.4309 - val_loss:
74.6045 - val_rsquared: 0.4900

Epoch 00034: val_rsquared improved from 0.48629 to 0.49004, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 35/200
3/3 [==============================] - 0s 60ms/step - loss: 79.8860 - rsquared: 0.4509 - val_loss:
73.8529 - val_rsquared: 0.4958

Epoch 00035: val_rsquared improved from 0.49004 to 0.49577, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 36/200
3/3 [==============================] - 0s 59ms/step - loss: 79.0235 - rsquared: 0.4650 - val_loss:
```

```
73.8401 - val_rsquared: 0.4952

Epoch 00036: val_rsquared did not improve from 0.49577
Epoch 37/200
3/3 [==============================] - 0s 58ms/step - loss: 81.5210 - rsquared: 0.4384 - val_loss: 
72.8288 - val_rsquared: 0.5023

Epoch 00037: val_rsquared improved from 0.49577 to 0.50225, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 38/200
3/3 [==============================] - 0s 90ms/step - loss: 79.1069 - rsquared: 0.4600 - val_loss: 
72.6703 - val_rsquared: 0.5026

Epoch 00038: val_rsquared improved from 0.50225 to 0.50257, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 39/200
3/3 [==============================] - 0s 84ms/step - loss: 80.7671 - rsquared: 0.4459 - val_loss: 
72.1722 - val_rsquared: 0.5060

Epoch 00039: val_rsquared improved from 0.50257 to 0.50603, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 40/200
3/3 [==============================] - 0s 58ms/step - loss: 77.0643 - rsquared: 0.4697 - val_loss: 
72.3239 - val_rsquared: 0.5049

Epoch 00040: val_rsquared did not improve from 0.50603
Epoch 41/200
3/3 [==============================] - 0s 60ms/step - loss: 77.6791 - rsquared: 0.4724 - val_loss: 
71.4928 - val_rsquared: 0.5113

Epoch 00041: val_rsquared improved from 0.50603 to 0.51130, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 42/200
3/3 [==============================] - 0s 57ms/step - loss: 73.1313 - rsquared: 0.4988 - val_loss: 
71.4189 - val_rsquared: 0.5113

Epoch 00042: val_rsquared did not improve from 0.51130
Epoch 43/200
3/3 [==============================] - 0s 58ms/step - loss: 73.6332 - rsquared: 0.4813 - val_loss: 
70.7068 - val_rsquared: 0.5159

Epoch 00043: val_rsquared improved from 0.51130 to 0.51585, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 44/200
3/3 [==============================] - 0s 57ms/step - loss: 73.4616 - rsquared: 0.4946 - val_loss: 
70.4935 - val_rsquared: 0.5168

Epoch 00044: val_rsquared improved from 0.51585 to 0.51675, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 45/200
3/3 [==============================] - 0s 60ms/step - loss: 70.3921 - rsquared: 0.5198 - val_loss: 
70.5366 - val_rsquared: 0.5161

Epoch 00045: val_rsquared did not improve from 0.51675
Epoch 46/200
3/3 [==============================] - 0s 77ms/step - loss: 73.2983 - rsquared: 0.4952 - val_loss: 
70.0920 - val_rsquared: 0.5194

Epoch 00046: val_rsquared improved from 0.51675 to 0.51941, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 47/200
3/3 [==============================] - 0s 58ms/step - loss: 71.4334 - rsquared: 0.5126 - val_loss: 
69.0915 - val_rsquared: 0.5270

Epoch 00047: val_rsquared improved from 0.51941 to 0.52698, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 48/200
3/3 [==============================] - 0s 58ms/step - loss: 70.4326 - rsquared: 0.5105 - val_loss: 
71.4922 - val_rsquared: 0.5090

Epoch 00048: val_rsquared did not improve from 0.52698
Epoch 49/200
3/3 [==============================] - 0s 61ms/step - loss: 71.3697 - rsquared: 0.5122 - val_loss: 
68.8644 - val_rsquared: 0.5287

Epoch 00049: val_rsquared improved from 0.52698 to 0.52871, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
```

```
Epoch 50/200
3/3 [==============================] - 0s 63ms/step - loss: 69.2081 - rsquared: 0.5164 - val_loss: 74.7543 - val_rsquared: 0.4844

Epoch 00050: val_rsquared did not improve from 0.52871
Epoch 51/200
3/3 [==============================] - 0s 58ms/step - loss: 69.2409 - rsquared: 0.5187 - val_loss: 69.0619 - val_rsquared: 0.5270

Epoch 00051: val_rsquared did not improve from 0.52871
Epoch 52/200
3/3 [==============================] - 0s 60ms/step - loss: 72.6894 - rsquared: 0.5023 - val_loss: 75.3992 - val_rsquared: 0.4790

Epoch 00052: val_rsquared did not improve from 0.52871
Epoch 53/200
3/3 [==============================] - 0s 62ms/step - loss: 67.1307 - rsquared: 0.5372 - val_loss: 68.2077 - val_rsquared: 0.5320

Epoch 00053: val_rsquared improved from 0.52871 to 0.53202, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 54/200
3/3 [==============================] - 0s 61ms/step - loss: 65.9909 - rsquared: 0.5436 - val_loss: 76.0041 - val_rsquared: 0.4737

Epoch 00054: val_rsquared did not improve from 0.53202
Epoch 55/200
3/3 [==============================] - 0s 60ms/step - loss: 67.5918 - rsquared: 0.5394 - val_loss: 67.8792 - val_rsquared: 0.5341

Epoch 00055: val_rsquared improved from 0.53202 to 0.53405, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 56/200
3/3 [==============================] - 0s 58ms/step - loss: 68.5057 - rsquared: 0.5316 - val_loss: 73.9734 - val_rsquared: 0.4890

Epoch 00056: val_rsquared did not improve from 0.53405
Epoch 57/200
3/3 [==============================] - 0s 58ms/step - loss: 68.3138 - rsquared: 0.5332 - val_loss: 67.6415 - val_rsquared: 0.5363

Epoch 00057: val_rsquared improved from 0.53405 to 0.53627, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 58/200
3/3 [==============================] - 0s 74ms/step - loss: 67.6280 - rsquared: 0.5414 - val_loss: 71.8457 - val_rsquared: 0.5054

Epoch 00058: val_rsquared did not improve from 0.53627
Epoch 59/200
3/3 [==============================] - 0s 65ms/step - loss: 64.8092 - rsquared: 0.5489 - val_loss: 67.4676 - val_rsquared: 0.5378

Epoch 00059: val_rsquared improved from 0.53627 to 0.53778, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 60/200
3/3 [==============================] - 0s 80ms/step - loss: 64.1897 - rsquared: 0.5498 - val_loss: 70.8118 - val_rsquared: 0.5130

Epoch 00060: val_rsquared did not improve from 0.53778
Epoch 61/200
3/3 [==============================] - 0s 67ms/step - loss: 62.7475 - rsquared: 0.5696 - val_loss: 69.3310 - val_rsquared: 0.5235

Epoch 00061: val_rsquared did not improve from 0.53778
Epoch 62/200
3/3 [==============================] - 0s 64ms/step - loss: 63.6893 - rsquared: 0.5641 - val_loss: 69.5181 - val_rsquared: 0.5221

Epoch 00062: val_rsquared did not improve from 0.53778
Epoch 63/200
3/3 [==============================] - 0s 102ms/step - loss: 64.7873 - rsquared: 0.5569 - val_loss: 71.2141 - val_rsquared: 0.5099

Epoch 00063: val_rsquared did not improve from 0.53778
Epoch 64/200
3/3 [==============================] - 0s 65ms/step - loss: 65.7008 - rsquared: 0.5542 - val_loss: 68.7566 - val_rsquared: 0.5286
```

```
Epoch 00064: val_rsquared did not improve from 0.53778
Epoch 65/200
3/3 [==============================] - 0s 76ms/step - loss: 63.0212 - rsquared: 0.5706 - val_loss: 
71.6643 - val_rsquared: 0.5074

Epoch 00065: val_rsquared did not improve from 0.53778
Epoch 66/200
3/3 [==============================] - 0s 71ms/step - loss: 62.9322 - rsquared: 0.5752 - val_loss: 
67.8015 - val_rsquared: 0.5354

Epoch 00066: val_rsquared did not improve from 0.53778
Epoch 67/200
3/3 [==============================] - 0s 67ms/step - loss: 62.6915 - rsquared: 0.5587 - val_loss: 
71.9556 - val_rsquared: 0.5043

Epoch 00067: val_rsquared did not improve from 0.53778
Epoch 68/200
3/3 [==============================] - 0s 60ms/step - loss: 64.6640 - rsquared: 0.5509 - val_loss: 
67.9391 - val_rsquared: 0.5331

Epoch 00068: val_rsquared did not improve from 0.53778
Epoch 69/200
3/3 [==============================] - 0s 59ms/step - loss: 59.9447 - rsquared: 0.5900 - val_loss: 
71.8466 - val_rsquared: 0.5045

Epoch 00069: val_rsquared did not improve from 0.53778
Epoch 70/200
3/3 [==============================] - 0s 61ms/step - loss: 59.0770 - rsquared: 0.6063 - val_loss: 
69.3957 - val_rsquared: 0.5226

Epoch 00070: val_rsquared did not improve from 0.53778
Epoch 71/200
3/3 [==============================] - 0s 59ms/step - loss: 60.8957 - rsquared: 0.5857 - val_loss: 
70.2096 - val_rsquared: 0.5167

Epoch 00071: val_rsquared did not improve from 0.53778
Epoch 72/200
3/3 [==============================] - 0s 97ms/step - loss: 58.5075 - rsquared: 0.5882 - val_loss: 
67.2373 - val_rsquared: 0.5386

Epoch 00072: val_rsquared improved from 0.53778 to 0.53856, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 73/200
3/3 [==============================] - 0s 60ms/step - loss: 62.8202 - rsquared: 0.5730 - val_loss: 
72.3569 - val_rsquared: 0.5007

Epoch 00073: val_rsquared did not improve from 0.53856
Epoch 74/200
3/3 [==============================] - 0s 65ms/step - loss: 61.5040 - rsquared: 0.5841 - val_loss: 
71.3760 - val_rsquared: 0.5083

Epoch 00074: val_rsquared did not improve from 0.53856
Epoch 75/200
3/3 [==============================] - 0s 71ms/step - loss: 59.4301 - rsquared: 0.5943 - val_loss: 
68.9405 - val_rsquared: 0.5265

Epoch 00075: val_rsquared did not improve from 0.53856
Epoch 76/200
3/3 [==============================] - 0s 66ms/step - loss: 58.3503 - rsquared: 0.6008 - val_loss: 
69.9777 - val_rsquared: 0.5194

Epoch 00076: val_rsquared did not improve from 0.53856
Epoch 77/200
3/3 [==============================] - 0s 61ms/step - loss: 59.2923 - rsquared: 0.5866 - val_loss: 
69.6021 - val_rsquared: 0.5225

Epoch 00077: val_rsquared did not improve from 0.53856
Epoch 78/200
3/3 [==============================] - 0s 59ms/step - loss: 59.4800 - rsquared: 0.5926 - val_loss: 
71.9532 - val_rsquared: 0.5050

Epoch 00078: val_rsquared did not improve from 0.53856
Epoch 79/200
3/3 [==============================] - 0s 58ms/step - loss: 59.3174 - rsquared: 0.6023 - val_loss: 
71.8070 - val_rsquared: 0.5055
```

```
Epoch 00079: val_rsquared did not improve from 0.53856
Epoch 80/200
3/3 [==============================] - 0s 63ms/step - loss: 55.9470 - rsquared: 0.5984 - val_loss:
68.3922 - val_rsquared: 0.5306

Epoch 00080: val_rsquared did not improve from 0.53856
Epoch 81/200
3/3 [==============================] - 0s 100ms/step - loss: 55.8564 - rsquared: 0.6139 -
val_loss: 73.2566 - val_rsquared: 0.4955

Epoch 00081: val_rsquared did not improve from 0.53856
Epoch 82/200
3/3 [==============================] - 0s 65ms/step - loss: 58.0794 - rsquared: 0.6067 - val_loss:
72.7973 - val_rsquared: 0.4996

Epoch 00082: val_rsquared did not improve from 0.53856
Epoch 83/200
3/3 [==============================] - 0s 66ms/step - loss: 56.9063 - rsquared: 0.6082 - val_loss:
68.8117 - val_rsquared: 0.5288

Epoch 00083: val_rsquared did not improve from 0.53856
Epoch 84/200
3/3 [==============================] - 0s 66ms/step - loss: 56.2570 - rsquared: 0.6067 - val_loss:
72.1400 - val_rsquared: 0.5039

Epoch 00084: val_rsquared did not improve from 0.53856
Epoch 85/200
3/3 [==============================] - 0s 65ms/step - loss: 55.8098 - rsquared: 0.6164 - val_loss:
77.3433 - val_rsquared: 0.4667

Epoch 00085: val_rsquared did not improve from 0.53856
Epoch 86/200
3/3 [==============================] - 0s 64ms/step - loss: 55.4111 - rsquared: 0.6203 - val_loss:
70.8468 - val_rsquared: 0.5145

Epoch 00086: val_rsquared did not improve from 0.53856
Epoch 87/200
3/3 [==============================] - 0s 59ms/step - loss: 57.4141 - rsquared: 0.6169 - val_loss:
76.9284 - val_rsquared: 0.4696

Epoch 00087: val_rsquared did not improve from 0.53856
Epoch 88/200
3/3 [==============================] - 0s 58ms/step - loss: 56.4445 - rsquared: 0.6117 - val_loss:
76.0167 - val_rsquared: 0.4766

Epoch 00088: val_rsquared did not improve from 0.53856
Epoch 89/200
3/3 [==============================] - 0s 68ms/step - loss: 54.8637 - rsquared: 0.6262 - val_loss:
70.3056 - val_rsquared: 0.5184

Epoch 00089: val_rsquared did not improve from 0.53856
Epoch 90/200
3/3 [==============================] - 0s 75ms/step - loss: 52.4261 - rsquared: 0.6329 - val_loss:
73.1342 - val_rsquared: 0.4983

Epoch 00090: val_rsquared did not improve from 0.53856
Epoch 91/200
3/3 [==============================] - 0s 67ms/step - loss: 56.2477 - rsquared: 0.6165 - val_loss:
81.6685 - val_rsquared: 0.4359

Epoch 00091: val_rsquared did not improve from 0.53856
Epoch 92/200
3/3 [==============================] - 0s 65ms/step - loss: 57.2474 - rsquared: 0.6212 - val_loss:
78.9322 - val_rsquared: 0.4552

Epoch 00092: val_rsquared did not improve from 0.53856
Epoch 93/200
3/3 [==============================] - 0s 66ms/step - loss: 56.0190 - rsquared: 0.6146 - val_loss:
67.7202 - val_rsquared: 0.5366

Epoch 00093: val_rsquared did not improve from 0.53856
Epoch 94/200
3/3 [==============================] - 0s 67ms/step - loss: 55.3028 - rsquared: 0.6205 - val_loss:
78.2793 - val_rsquared: 0.4605

Epoch 00094: val_rsquared did not improve from 0.53856
Epoch 95/200
```

```
3/3 [==============================] - 0s 60ms/step - loss: 56.0653 - rsquared: 0.6206 - val_loss:
88.9948 - val_rsquared: 0.3824

Epoch 00095: val_rsquared did not improve from 0.53856
Epoch 96/200
3/3 [==============================] - 0s 61ms/step - loss: 57.4594 - rsquared: 0.6116 - val_loss:
67.6363 - val_rsquared: 0.5374

Epoch 00096: val_rsquared did not improve from 0.53856
Epoch 97/200
3/3 [==============================] - 0s 58ms/step - loss: 56.5504 - rsquared: 0.6053 - val_loss:
72.5770 - val_rsquared: 0.5013

Epoch 00097: val_rsquared did not improve from 0.53856
Epoch 98/200
3/3 [==============================] - 0s 62ms/step - loss: 55.6222 - rsquared: 0.6143 - val_loss:
89.4280 - val_rsquared: 0.3788

Epoch 00098: val_rsquared did not improve from 0.53856
Epoch 99/200
3/3 [==============================] - 0s 67ms/step - loss: 55.8601 - rsquared: 0.6139 - val_loss:
67.2236 - val_rsquared: 0.5397

Epoch 00099: val_rsquared improved from 0.53856 to 0.53969, saving model to /tmp/checkpoint2
INFO:tensorflow:Assets written to: /tmp/checkpoint2/assets
Epoch 100/200
3/3 [==============================] - 0s 88ms/step - loss: 55.8586 - rsquared: 0.6244 - val_loss:
84.5866 - val_rsquared: 0.4143

Epoch 00100: val_rsquared did not improve from 0.53969
Epoch 101/200
3/3 [==============================] - 0s 65ms/step - loss: 53.7479 - rsquared: 0.6284 - val_loss:
83.0296 - val_rsquared: 0.4260

Epoch 00101: val_rsquared did not improve from 0.53969
Epoch 102/200
3/3 [==============================] - 0s 63ms/step - loss: 53.4769 - rsquared: 0.6364 - val_loss:
72.7112 - val_rsquared: 0.5010

Epoch 00102: val_rsquared did not improve from 0.53969
Epoch 103/200
3/3 [==============================] - 0s 77ms/step - loss: 55.1748 - rsquared: 0.6286 - val_loss:
86.5635 - val_rsquared: 0.4005

Epoch 00103: val_rsquared did not improve from 0.53969
Epoch 104/200
3/3 [==============================] - 0s 64ms/step - loss: 55.6567 - rsquared: 0.6256 - val_loss:
86.8937 - val_rsquared: 0.3990

Epoch 00104: val_rsquared did not improve from 0.53969
Epoch 105/200
3/3 [==============================] - 0s 63ms/step - loss: 53.2668 - rsquared: 0.6295 - val_loss:
69.3262 - val_rsquared: 0.5264

Epoch 00105: val_rsquared did not improve from 0.53969
Epoch 106/200
3/3 [==============================] - 0s 113ms/step - loss: 52.0305 - rsquared: 0.6424 -
val_loss: 82.1705 - val_rsquared: 0.4334

Epoch 00106: val_rsquared did not improve from 0.53969
Epoch 107/200
3/3 [==============================] - 0s 68ms/step - loss: 49.3686 - rsquared: 0.6526 - val_loss:
79.4275 - val_rsquared: 0.4533

Epoch 00107: val_rsquared did not improve from 0.53969
Epoch 108/200
3/3 [==============================] - 0s 68ms/step - loss: 51.7111 - rsquared: 0.6519 - val_loss:
73.5621 - val_rsquared: 0.4965

Epoch 00108: val_rsquared did not improve from 0.53969
Epoch 109/200
3/3 [==============================] - 0s 67ms/step - loss: 48.7434 - rsquared: 0.6562 - val_loss:
76.8162 - val_rsquared: 0.4729

Epoch 00109: val_rsquared did not improve from 0.53969
Epoch 110/200
3/3 [==============================] - 0s 102ms/step - loss: 50.9564 - rsquared: 0.6447 -
```

```
val_loss: 80.0590 - val_rsquared: 0.4487

Epoch 00110: val_rsquared did not improve from 0.53969
Epoch 111/200
3/3 [==============================] - 0s 67ms/step - loss: 52.0874 - rsquared: 0.6461 - val_loss: 
87.2218 - val_rsquared: 0.3966

Epoch 00111: val_rsquared did not improve from 0.53969
Epoch 112/200
3/3 [==============================] - 0s 64ms/step - loss: 47.8052 - rsquared: 0.6701 - val_loss: 
71.2139 - val_rsquared: 0.5131

Epoch 00112: val_rsquared did not improve from 0.53969
Epoch 113/200
3/3 [==============================] - 0s 65ms/step - loss: 51.2432 - rsquared: 0.6544 - val_loss: 
80.2733 - val_rsquared: 0.4476

Epoch 00113: val_rsquared did not improve from 0.53969
Epoch 114/200
3/3 [==============================] - 0s 70ms/step - loss: 51.4870 - rsquared: 0.6482 - val_loss: 
87.4570 - val_rsquared: 0.3960

Epoch 00114: val_rsquared did not improve from 0.53969
Epoch 115/200
3/3 [==============================] - 0s 95ms/step - loss: 53.7331 - rsquared: 0.6365 - val_loss: 
91.0872 - val_rsquared: 0.3699

Epoch 00115: val_rsquared did not improve from 0.53969
Epoch 116/200
3/3 [==============================] - 0s 66ms/step - loss: 51.2065 - rsquared: 0.6393 - val_loss: 
72.5323 - val_rsquared: 0.5042

Epoch 00116: val_rsquared did not improve from 0.53969
Epoch 117/200
3/3 [==============================] - 0s 68ms/step - loss: 50.2324 - rsquared: 0.6541 - val_loss: 
86.6257 - val_rsquared: 0.4028

Epoch 00117: val_rsquared did not improve from 0.53969
Epoch 118/200
3/3 [==============================] - 0s 62ms/step - loss: 49.6989 - rsquared: 0.6558 - val_loss: 
82.4244 - val_rsquared: 0.4331

Epoch 00118: val_rsquared did not improve from 0.53969
Epoch 119/200
3/3 [==============================] - 0s 66ms/step - loss: 48.1473 - rsquared: 0.6679 - val_loss: 
71.8875 - val_rsquared: 0.5084

Epoch 00119: val_rsquared did not improve from 0.53969
Epoch 120/200
3/3 [==============================] - 0s 79ms/step - loss: 49.5121 - rsquared: 0.6565 - val_loss: 
83.3800 - val_rsquared: 0.4247

Epoch 00120: val_rsquared did not improve from 0.53969
Epoch 121/200
3/3 [==============================] - 0s 68ms/step - loss: 50.1299 - rsquared: 0.6532 - val_loss: 
82.9033 - val_rsquared: 0.4282

Epoch 00121: val_rsquared did not improve from 0.53969
Epoch 122/200
3/3 [==============================] - 0s 65ms/step - loss: 49.8462 - rsquared: 0.6562 - val_loss: 
73.1227 - val_rsquared: 0.4986

Epoch 00122: val_rsquared did not improve from 0.53969
Epoch 123/200
3/3 [==============================] - 0s 64ms/step - loss: 48.5529 - rsquared: 0.6624 - val_loss: 
76.0466 - val_rsquared: 0.4784

Epoch 00123: val_rsquared did not improve from 0.53969
Epoch 124/200
3/3 [==============================] - 0s 64ms/step - loss: 48.0227 - rsquared: 0.6698 - val_loss: 
90.1360 - val_rsquared: 0.3781

Epoch 00124: val_rsquared did not improve from 0.53969
Epoch 125/200
3/3 [==============================] - 0s 79ms/step - loss: 46.6503 - rsquared: 0.6716 - val_loss: 
83.5692 - val_rsquared: 0.4260
```

```
Epoch 00125: val_rsquared did not improve from 0.53969
Epoch 126/200
3/3 [==============================] - 0s 64ms/step - loss: 48.8686 - rsquared: 0.6660 - val_loss:
82.0059 - val_rsquared: 0.4376

Epoch 00126: val_rsquared did not improve from 0.53969
Epoch 127/200
3/3 [==============================] - 0s 64ms/step - loss: 49.5645 - rsquared: 0.6573 - val_loss:
84.6544 - val_rsquared: 0.4181

Epoch 00127: val_rsquared did not improve from 0.53969
Epoch 128/200
3/3 [==============================] - 0s 69ms/step - loss: 49.0734 - rsquared: 0.6586 - val_loss:
88.7015 - val_rsquared: 0.3883

Epoch 00128: val_rsquared did not improve from 0.53969
Epoch 129/200
3/3 [==============================] - 0s 65ms/step - loss: 49.0830 - rsquared: 0.6614 - val_loss:
83.9302 - val_rsquared: 0.4228

Epoch 00129: val_rsquared did not improve from 0.53969
Epoch 130/200
3/3 [==============================] - ETA: 0s - loss: 49.7452 - rsquared: 0.666 - 0s 65ms/step -
loss: 49.3301 - rsquared: 0.6678 - val_loss: 82.9012 - val_rsquared: 0.4301

Epoch 00130: val_rsquared did not improve from 0.53969
Epoch 131/200
3/3 [==============================] - 0s 68ms/step - loss: 46.6329 - rsquared: 0.6780 - val_loss:
84.6411 - val_rsquared: 0.4180

Epoch 00131: val_rsquared did not improve from 0.53969
Epoch 132/200
3/3 [==============================] - 0s 65ms/step - loss: 45.8757 - rsquared: 0.6822 - val_loss:
90.1682 - val_rsquared: 0.3785

Epoch 00132: val_rsquared did not improve from 0.53969
Epoch 133/200
3/3 [==============================] - 0s 86ms/step - loss: 49.6857 - rsquared: 0.6674 - val_loss:
92.1388 - val_rsquared: 0.3637

Epoch 00133: val_rsquared did not improve from 0.53969
Epoch 134/200
3/3 [==============================] - 0s 68ms/step - loss: 46.3166 - rsquared: 0.6831 - val_loss:
80.8116 - val_rsquared: 0.4453

Epoch 00134: val_rsquared did not improve from 0.53969
Epoch 135/200
3/3 [==============================] - 0s 67ms/step - loss: 49.7061 - rsquared: 0.6653 - val_loss:
87.9854 - val_rsquared: 0.3934

Epoch 00135: val_rsquared did not improve from 0.53969
Epoch 136/200
3/3 [==============================] - 0s 65ms/step - loss: 49.4780 - rsquared: 0.6680 - val_loss:
96.9019 - val_rsquared: 0.3281

Epoch 00136: val_rsquared did not improve from 0.53969
Epoch 137/200
3/3 [==============================] - 0s 64ms/step - loss: 47.0181 - rsquared: 0.6732 - val_loss:
83.8929 - val_rsquared: 0.4228

Epoch 00137: val_rsquared did not improve from 0.53969
Epoch 138/200
3/3 [==============================] - 0s 65ms/step - loss: 45.9871 - rsquared: 0.6751 - val_loss:
74.7960 - val_rsquared: 0.4887

Epoch 00138: val_rsquared did not improve from 0.53969
Epoch 139/200
3/3 [==============================] - 0s 68ms/step - loss: 49.4193 - rsquared: 0.6578 - val_loss:
83.8910 - val_rsquared: 0.4214

Epoch 00139: val_rsquared did not improve from 0.53969
Epoch 140/200
3/3 [==============================] - 0s 63ms/step - loss: 48.1778 - rsquared: 0.6714 - val_loss:
100.4094 - val_rsquared: 0.3003

Epoch 00140: val_rsquared did not improve from 0.53969
Epoch 141/200
```

```
3/3 [==============================] - 0s 80ms/step - loss: 45.6307 - rsquared: 0.6785 - val_loss:
83.3491 - val_rsquared: 0.4261

Epoch 00141: val_rsquared did not improve from 0.53969
Epoch 142/200
3/3 [==============================] - 0s 66ms/step - loss: 46.0428 - rsquared: 0.6889 - val_loss:
84.1785 - val_rsquared: 0.4214

Epoch 00142: val_rsquared did not improve from 0.53969
Epoch 143/200
3/3 [==============================] - 0s 64ms/step - loss: 47.2884 - rsquared: 0.6722 - val_loss:
88.4691 - val_rsquared: 0.3906

Epoch 00143: val_rsquared did not improve from 0.53969
Epoch 144/200
3/3 [==============================] - 0s 66ms/step - loss: 45.9549 - rsquared: 0.6882 - val_loss:
101.5430 - val_rsquared: 0.2964

Epoch 00144: val_rsquared did not improve from 0.53969
Epoch 145/200
3/3 [==============================] - 0s 76ms/step - loss: 47.1522 - rsquared: 0.6734 - val_loss:
90.8139 - val_rsquared: 0.3756

Epoch 00145: val_rsquared did not improve from 0.53969
Epoch 146/200
3/3 [==============================] - 0s 82ms/step - loss: 46.9178 - rsquared: 0.6810 - val_loss:
89.2050 - val_rsquared: 0.3859

Epoch 00146: val_rsquared did not improve from 0.53969
Epoch 147/200
3/3 [==============================] - 0s 63ms/step - loss: 45.9789 - rsquared: 0.6848 - val_loss:
98.1981 - val_rsquared: 0.3187

Epoch 00147: val_rsquared did not improve from 0.53969
Epoch 148/200
3/3 [==============================] - 0s 63ms/step - loss: 46.6770 - rsquared: 0.6789 - val_loss:
91.5471 - val_rsquared: 0.3675

Epoch 00148: val_rsquared did not improve from 0.53969
Epoch 149/200
3/3 [==============================] - 0s 74ms/step - loss: 45.0151 - rsquared: 0.6898 - val_loss:
87.2931 - val_rsquared: 0.3988

Epoch 00149: val_rsquared did not improve from 0.53969
Epoch 150/200
3/3 [==============================] - 0s 64ms/step - loss: 44.8067 - rsquared: 0.6888 - val_loss:
83.6271 - val_rsquared: 0.4258

Epoch 00150: val_rsquared did not improve from 0.53969
Epoch 151/200
3/3 [==============================] - 0s 99ms/step - loss: 48.9313 - rsquared: 0.6668 - val_loss:
108.4963 - val_rsquared: 0.2444

Epoch 00151: val_rsquared did not improve from 0.53969
Epoch 152/200
3/3 [==============================] - 0s 66ms/step - loss: 46.2641 - rsquared: 0.6756 - val_loss:
102.9463 - val_rsquared: 0.2861

Epoch 00152: val_rsquared did not improve from 0.53969
Epoch 153/200
3/3 [==============================] - 0s 76ms/step - loss: 45.5970 - rsquared: 0.6848 - val_loss:
81.0736 - val_rsquared: 0.4465

Epoch 00153: val_rsquared did not improve from 0.53969
Epoch 154/200
3/3 [==============================] - 0s 69ms/step - loss: 48.8023 - rsquared: 0.6606 - val_loss:
80.3773 - val_rsquared: 0.4494

Epoch 00154: val_rsquared did not improve from 0.53969
Epoch 155/200
3/3 [==============================] - 0s 69ms/step - loss: 47.0500 - rsquared: 0.6808 - val_loss:
100.8352 - val_rsquared: 0.2985

Epoch 00155: val_rsquared did not improve from 0.53969
Epoch 156/200
3/3 [==============================] - 0s 83ms/step - loss: 45.0259 - rsquared: 0.6910 - val_loss:
101.1413 - val_rsquared: 0.2975
```

```
Epoch 00156: val_rsquared did not improve from 0.53969
Epoch 157/200
3/3 [==============================] - 0s 84ms/step - loss: 43.5085 - rsquared: 0.6996 - val_loss: 
102.5588 - val_rsquared: 0.2881

Epoch 00157: val_rsquared did not improve from 0.53969
Epoch 158/200
3/3 [==============================] - 0s 68ms/step - loss: 45.1255 - rsquared: 0.6867 - val_loss: 
83.4762 - val_rsquared: 0.4269

Epoch 00158: val_rsquared did not improve from 0.53969
Epoch 159/200
3/3 [==============================] - 0s 65ms/step - loss: 43.4023 - rsquared: 0.6990 - val_loss: 
85.7464 - val_rsquared: 0.4098

Epoch 00159: val_rsquared did not improve from 0.53969
Epoch 160/200
3/3 [==============================] - 0s 82ms/step - loss: 44.4027 - rsquared: 0.7000 - val_loss: 
100.2592 - val_rsquared: 0.3057

Epoch 00160: val_rsquared did not improve from 0.53969
Epoch 161/200
3/3 [==============================] - 0s 80ms/step - loss: 42.5047 - rsquared: 0.7087 - val_loss: 
104.2024 - val_rsquared: 0.2773

Epoch 00161: val_rsquared did not improve from 0.53969
Epoch 162/200
3/3 [==============================] - 0s 63ms/step - loss: 42.5162 - rsquared: 0.7081 - val_loss: 
94.6722 - val_rsquared: 0.3464

Epoch 00162: val_rsquared did not improve from 0.53969
Epoch 163/200
3/3 [==============================] - 0s 66ms/step - loss: 42.7066 - rsquared: 0.7081 - val_loss: 
89.1959 - val_rsquared: 0.3852

Epoch 00163: val_rsquared did not improve from 0.53969
Epoch 164/200
3/3 [==============================] - 0s 99ms/step - loss: 45.2654 - rsquared: 0.6873 - val_loss: 
95.4777 - val_rsquared: 0.3391

Epoch 00164: val_rsquared did not improve from 0.53969
Epoch 165/200
3/3 [==============================] - 0s 112ms/step - loss: 44.0883 - rsquared: 0.6947 - 
val_loss: 104.5227 - val_rsquared: 0.2734

Epoch 00165: val_rsquared did not improve from 0.53969
Epoch 166/200
3/3 [==============================] - 0s 71ms/step - loss: 42.6483 - rsquared: 0.7049 - val_loss: 
106.6308 - val_rsquared: 0.2589

Epoch 00166: val_rsquared did not improve from 0.53969
Epoch 167/200
3/3 [==============================] - 0s 67ms/step - loss: 43.1972 - rsquared: 0.7022 - val_loss: 
98.9396 - val_rsquared: 0.3146

Epoch 00167: val_rsquared did not improve from 0.53969
Epoch 168/200
3/3 [==============================] - 0s 79ms/step - loss: 40.6389 - rsquared: 0.7181 - val_loss: 
90.2194 - val_rsquared: 0.3772

Epoch 00168: val_rsquared did not improve from 0.53969
Epoch 169/200
3/3 [==============================] - 0s 81ms/step - loss: 42.2517 - rsquared: 0.7072 - val_loss: 
85.0897 - val_rsquared: 0.4150

Epoch 00169: val_rsquared did not improve from 0.53969
Epoch 170/200
3/3 [==============================] - 0s 65ms/step - loss: 47.2784 - rsquared: 0.6777 - val_loss: 
85.1683 - val_rsquared: 0.4149

Epoch 00170: val_rsquared did not improve from 0.53969
Epoch 171/200
3/3 [==============================] - 0s 67ms/step - loss: 45.2556 - rsquared: 0.6904 - val_loss: 
113.5731 - val_rsquared: 0.2070

Epoch 00171: val_rsquared did not improve from 0.53969
```

```
Epoch 00171: val_rsquared did not improve from 0.53969
Epoch 172/200
3/3 [==============================] - 0s 95ms/step - loss: 45.9267 - rsquared: 0.6883 - val_loss:
116.7531 - val_rsquared: 0.1835

Epoch 00172: val_rsquared did not improve from 0.53969
Epoch 173/200
3/3 [==============================] - 0s 80ms/step - loss: 46.7662 - rsquared: 0.6830 - val_loss:
116.8704 - val_rsquared: 0.1823

Epoch 00173: val_rsquared did not improve from 0.53969
Epoch 174/200
3/3 [==============================] - 0s 81ms/step - loss: 41.9794 - rsquared: 0.7100 - val_loss:
103.4597 - val_rsquared: 0.2816

Epoch 00174: val_rsquared did not improve from 0.53969
Epoch 175/200
3/3 [==============================] - 0s 74ms/step - loss: 42.6374 - rsquared: 0.7027 - val_loss:
89.9147 - val_rsquared: 0.3810

Epoch 00175: val_rsquared did not improve from 0.53969
Epoch 176/200
3/3 [==============================] - 0s 86ms/step - loss: 43.8012 - rsquared: 0.6982 - val_loss:
93.2107 - val_rsquared: 0.3562

Epoch 00176: val_rsquared did not improve from 0.53969
Epoch 177/200
3/3 [==============================] - 0s 80ms/step - loss: 44.3605 - rsquared: 0.6974 - val_loss:
95.4100 - val_rsquared: 0.3394

Epoch 00177: val_rsquared did not improve from 0.53969
Epoch 178/200
3/3 [==============================] - 0s 66ms/step - loss: 42.8901 - rsquared: 0.7024 - val_loss:
130.0516 - val_rsquared: 0.0880

Epoch 00178: val_rsquared did not improve from 0.53969
Epoch 179/200
3/3 [==============================] - 0s 66ms/step - loss: 46.6886 - rsquared: 0.6873 - val_loss:
125.9213 - val_rsquared: 0.1192

Epoch 00179: val_rsquared did not improve from 0.53969
Epoch 180/200
3/3 [==============================] - 0s 84ms/step - loss: 44.0371 - rsquared: 0.6940 - val_loss:
102.8483 - val_rsquared: 0.2863

Epoch 00180: val_rsquared did not improve from 0.53969
Epoch 181/200
3/3 [==============================] - 0s 68ms/step - loss: 44.0383 - rsquared: 0.6972 - val_loss:
92.3802 - val_rsquared: 0.3622

Epoch 00181: val_rsquared did not improve from 0.53969
Epoch 182/200
3/3 [==============================] - 0s 72ms/step - loss: 41.1700 - rsquared: 0.7180 - val_loss:
95.1402 - val_rsquared: 0.3421

Epoch 00182: val_rsquared did not improve from 0.53969
Epoch 183/200
3/3 [==============================] - 0s 73ms/step - loss: 40.3699 - rsquared: 0.7166 - val_loss:
100.4846 - val_rsquared: 0.3034

Epoch 00183: val_rsquared did not improve from 0.53969
Epoch 184/200
3/3 [==============================] - 0s 64ms/step - loss: 42.3769 - rsquared: 0.7131 - val_loss:
106.7722 - val_rsquared: 0.2577

Epoch 00184: val_rsquared did not improve from 0.53969
Epoch 185/200
3/3 [==============================] - 0s 87ms/step - loss: 39.9216 - rsquared: 0.7278 - val_loss:
132.3289 - val_rsquared: 0.0704

Epoch 00185: val_rsquared did not improve from 0.53969
Epoch 186/200
3/3 [==============================] - 0s 62ms/step - loss: 44.2179 - rsquared: 0.6963 - val_loss:
127.9638 - val_rsquared: 0.1026

Epoch 00186: val_rsquared did not improve from 0.53969
Epoch 187/200
3/3 [==============================] - 0s 89ms/step - loss: 42.7678 - rsquared: 0.7061 - val_loss:
```

```
118.0099 - val_rsquared: 0.1762

Epoch 00187: val_rsquared did not improve from 0.53969
Epoch 188/200
3/3 [==============================] - 0s 66ms/step - loss: 38.9799 - rsquared: 0.7316 - val_loss:
101.3739 - val_rsquared: 0.2974

Epoch 00188: val_rsquared did not improve from 0.53969
Epoch 189/200
3/3 [==============================] - 0s 78ms/step - loss: 40.8055 - rsquared: 0.7239 - val_loss:
108.7451 - val_rsquared: 0.2427

Epoch 00189: val_rsquared did not improve from 0.53969
Epoch 190/200
3/3 [==============================] - 0s 63ms/step - loss: 40.2510 - rsquared: 0.7257 - val_loss:
121.6437 - val_rsquared: 0.1481

Epoch 00190: val_rsquared did not improve from 0.53969
Epoch 191/200
3/3 [==============================] - 0s 98ms/step - loss: 41.9841 - rsquared: 0.7187 - val_loss:
108.2384 - val_rsquared: 0.2461

Epoch 00191: val_rsquared did not improve from 0.53969
Epoch 192/200
3/3 [==============================] - 0s 65ms/step - loss: 39.9053 - rsquared: 0.7237 - val_loss:
94.6782 - val_rsquared: 0.3452

Epoch 00192: val_rsquared did not improve from 0.53969
Epoch 193/200
3/3 [==============================] - 0s 114ms/step - loss: 40.7619 - rsquared: 0.7224 -
val_loss: 97.3004 - val_rsquared: 0.3257

Epoch 00193: val_rsquared did not improve from 0.53969
Epoch 194/200
3/3 [==============================] - 0s 69ms/step - loss: 40.5964 - rsquared: 0.7223 - val_loss:
111.7724 - val_rsquared: 0.2202

Epoch 00194: val_rsquared did not improve from 0.53969
Epoch 195/200
3/3 [==============================] - 0s 81ms/step - loss: 40.3564 - rsquared: 0.7211 - val_loss:
112.8108 - val_rsquared: 0.2129

Epoch 00195: val_rsquared did not improve from 0.53969
Epoch 196/200
3/3 [==============================] - 0s 64ms/step - loss: 40.5591 - rsquared: 0.7236 - val_loss:
125.7706 - val_rsquared: 0.1184

Epoch 00196: val_rsquared did not improve from 0.53969
Epoch 197/200
3/3 [==============================] - 0s 62ms/step - loss: 40.7545 - rsquared: 0.7187 - val_loss:
132.7331 - val_rsquared: 0.0680

Epoch 00197: val_rsquared did not improve from 0.53969
Epoch 198/200
3/3 [==============================] - 0s 60ms/step - loss: 42.9593 - rsquared: 0.7064 - val_loss:
134.7598 - val_rsquared: 0.0537

Epoch 00198: val_rsquared did not improve from 0.53969
Epoch 199/200
3/3 [==============================] - 0s 67ms/step - loss: 40.2247 - rsquared: 0.7304 - val_loss:
122.4880 - val_rsquared: 0.1436

Epoch 00199: val_rsquared did not improve from 0.53969
Epoch 200/200
3/3 [==============================] - 0s 62ms/step - loss: 38.2202 - rsquared: 0.7347 - val_loss:
101.6148 - val_rsquared: 0.2959

Epoch 00200: val_rsquared did not improve from 0.53969
```

Out[180]:

```
<tensorflow.python.keras.callbacks.History at 0x7fa6284a9c40>
```

In [215]:

```
score11 = 0.53969
```

Decision Tree

In [216]:

```
from sklearn.tree import DecisionTreeRegressor
```

With Feature Set - 4

In [217]:

```
X_Set4 = pd.DataFrame(np.hstack((x,x_pca,x_svd)))
print(X_Set4.shape)
```

```
(4196, 538)
```

In [218]:

```
X_train_DT_Set4, X_test_DT_Set4, y_train, y_test = train_test_split(X_Set4, y, test_size=0.33, rand
om_state=1)
X_train_DT_Set4, X_cv_DT_Set4, y_train, y_cv = train_test_split(X_train_DT_Set4, y_train,
test_size=0.33)
print("Done")
```

```
Done
```

In [219]:

```
max_depth = [1,5,10,50]
samples_split = [5,10,100,500]
score_train = []
score_cv = []
plot_dep,plot_sample = [],[]
for i in max_depth:
    for j in samples_split:
        clf = DecisionTreeRegressor(max_depth = i, min_samples_split = j)
        clf.fit(X_train_DT_Set4 ,y_train)
        y_train_pred = clf.predict(X_train_DT_Set4)
        y_cv_pred = clf.predict(X_cv_DT_Set4)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_dep.append(i)
        plot_sample.append(j)
```

In [220]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_sample,y=plot_dep,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_sample,y=plot_dep,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='min_samples_split'),
        yaxis = dict(title='max_depth'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [221]:

```
model = DecisionTreeRegressor(max_depth =5, min_samples_split =500 )
model.fit(X_train_DT_Set4,y_train)
y_te = model.predict(X_test_DT_Set4)
score12 = r2_score(y_test, y_te)
print("Test Score for 4th feature set : ", score12)
```

```
Test Score for 4th feature set :  0.5833543142557482
```

With Feature Set - 5

In [222]:

```
X_Set5 = pd.DataFrame(np.hstack((x,x_svd)))
print(X_Set5.shape)
```

```
(4196, 528)
```

In [223]:

```
X_train_DT_Set5, X_test_DT_Set5, y_train, y_test = train_test_split(X_Set5, y, test_size=0.33, rand
om_state=1)
X_train_DT_Set5, X_cv_DT_Set5, y_train, y_cv = train_test_split(X_train_DT_Set5, y_train,
test_size=0.33)
print("Done")
```

```
Done
```

In [224]:

```
max_depth = [1,5,10,50]
samples_split = [5,10,100,500]
score_train = []
score_cv = []
plot_dep,plot_sample = [],[]
for i in max_depth:
    for j in samples_split:
        clf = DecisionTreeRegressor(max_depth = i, min_samples_split = j)
        clf.fit(X_train_DT_Set5 ,y_train)
        y_train_pred = clf.predict(X_train_DT_Set5)
        y_cv_pred = clf.predict(X_cv_DT_Set5)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_dep.append(i)
        plot_sample.append(j)
```

In [225]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_sample,y=plot_dep,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_sample,y=plot_dep,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='min_samples_split'),
        yaxis = dict(title='max_depth'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [226]:

```
model = DecisionTreeRegressor(max_depth =10, min_samples_split =500 )
model.fit(X_train_DT_Set5,y_train)
y_te = model.predict(X_test_DT_Set5)
score13 = r2_score(y_test, y_te)
print("Test Score for 4th feature set : ", score13)
```

```
Test Score for 4th feature set :  0.6021492704025679
```

Concluding Results of different Models

In [232]:

```
from tabulate import tabulate
print(tabulate([['Auto - Encoded Features','XGBoost', score1],
                 ['Auto - Encoded Features + PCA','XGBoost', score2],
                 ['PCA + SVD','XGBoost', score3],
                 ['Label - Encoded Features + PCA + SVD','XGBoost', score4],
                 ['Label - Encoded Features + SVD','XGBoost', score5],
                ['Label - Encoded Features + PCA + SVD','Linear Regression', score6],
                ['Label - Encoded Features + SVD','Linear Regression', score7],
               ['Label - Encoded Features + PCA + SVD','Random Forest', score8],
                ['Label - Encoded Features + SVD','Random Forest', score9],
                ['Label - Encoded Features + PCA + SVD','MLP', score10],
```

```
                ['Label - Encoded Features + SVD','MLP', score11],
                ['Label - Encoded Features + PCA + SVD','Decision Tree', score12],
                ['Label - Encoded Features + SVD','Decision Tree', score13]],
                headers=['Features', 'Model','R2_Score'], tablefmt='orgtbl'))
```

```
| Features                             | Model             |      R2_Score |
|--------------------------------------+-------------------+---------------|
| Auto - Encoded Features              | XGBoost           |   0.478379    |
| Auto - Encoded Features + PCA        | XGBoost           |   0.499809    |
| PCA + SVD                            | XGBoost           |   0.516819    |
| Label - Encoded Features + PCA + SVD | XGBoost           |   0.585436    |
| Label - Encoded Features + SVD       | XGBoost           |   0.585805    |
| Label - Encoded Features + PCA + SVD | Linear Regression | -6.05298e+19  |
| Label - Encoded Features + SVD       | Linear Regression | -9.76042e+19  |
| Label - Encoded Features + PCA + SVD | Random Forest     |   0.59191     |
| Label - Encoded Features + SVD       | Random Forest     |   0.597778    |
| Label - Encoded Features + PCA + SVD | MLP               |   0.5587      |
| Label - Encoded Features + SVD       | MLP               |   0.53969     |
| Label - Encoded Features + PCA + SVD | Decision Tree     |   0.583354    |
| Label - Encoded Features + SVD       | Decision Tree     |   0.602149    |
```

Building Final Model

In [233]:

```
enc = LabelEncoder()
for i in x_cat.columns:
    x_cat[i] = enc.fit_transform(x_cat[i])
```

In [234]:

```
test_data = pd.read_csv('downloads/testwa.csv')
```

In [235]:

```
test_cat = test_data.loc[:,'X0':'X8']
test_num = test_data.loc[:,'X10':]
```

In [236]:

```
for i in x_cat.columns:
    test_cat[i] = enc.fit_transform(test_cat[i])
```

In [237]:

```
test_svd = tsvd.transform(test_num)
```

In [238]:

```
test = pd.DataFrame(np.hstack((test_cat,test_num,test_svd)))
print(test.shape)
```

```
(4209, 528)
```

In [239]:

```
X_final = pd.DataFrame(np.hstack((x,x_svd)))
print(X_final.shape)
```

```
(4196, 528)
```

In [240]:

```
X_train_final, X_cv_final, y_train, y_cv = train_test_split(X_final, y, test_size=0.33)
print("Done")
```

```
```

```
Done
```

In [241]:

```
max_depth = [5, 10, 15,20, 25, 40]
n_estimators = [5,10,50,75,100,200]
score_train = []
score_cv = []
plot_dep,plot_estim = [],[]
for i in max_depth:
    for j in n_estimators:
        clf = RandomForestRegressor(max_depth = i, n_estimators = j, verbose = 0,n_jobs = -1)
        clf.fit(X_train_final ,y_train)
        y_train_pred = clf.predict(X_train_final)
        y_cv_pred = clf.predict(X_cv_final)
        score_train.append(r2_score(y_train,y_train_pred))
        score_cv.append(r2_score(y_cv,y_cv_pred))
        plot_dep.append(i)
        plot_estim.append(j)
```

In [243]:

```
#plotting the auc corresponding to different hyper parameter permutations to understand
trace1 = go.Scatter3d(x=plot_estim,y=plot_dep,z=score_train, name = 'train')
trace2 = go.Scatter3d(x=plot_estim,y=plot_dep,z=score_cv, name = 'Cross validation')
data = [trace1, trace2]

layout = go.Layout(scene = dict(
        xaxis = dict(title='n_estimators'),
        yaxis = dict(title='learning_rate'),
        zaxis = dict(title='R2'),))

fig = go.Figure(data=data, layout=layout)
fig.show()
```

In [244]:

```
model = RandomForestRegressor(n_estimators=200, max_depth =5)
model.fit(X_train_final,y_train)
y_pred = model.predict(test)
```

In [245]:

```
y_pred
```

Out[245]:

```
array([ 77.55105402,  94.20096004,  77.39609305, ...,  93.95882902,
       112.08235747,  93.89945284])
```

In [ ]: