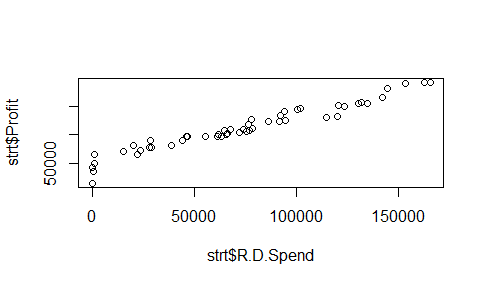
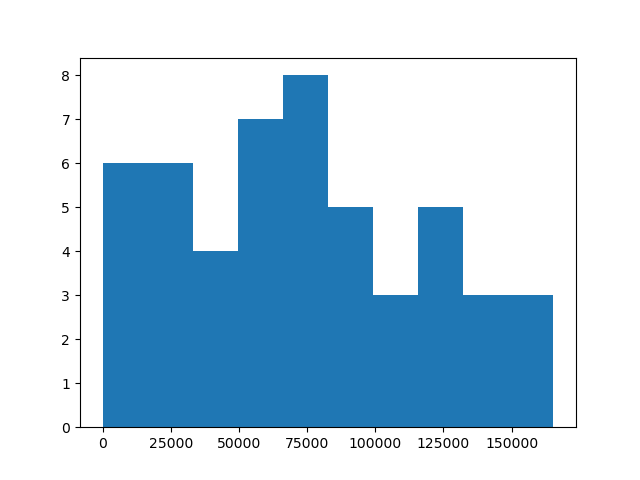
ANN Deep Learning with Python

1. **50 Start-up:**

****

The Linear Correlation between RnD revenue spend and Profit gained for it increased with same rate.



In the Marketing column there is significant increase in Marketing spending expenditure for startups in the range of 50000 to 100000 $ .

Using Sequential Model in python for ANN .

The following results:

4 neurons , Activation function =”Relu”

3 hidden layer , 1 output layer.

Loss = mse

Optimizer =Adam

Training Score : #[14042229760.0, 0.0]

Testing Data Score ; #[14363037696.0, 0.0]

mean\_squared\_error(y\_test,predictions)

# 14363409193.50244

#RMSE

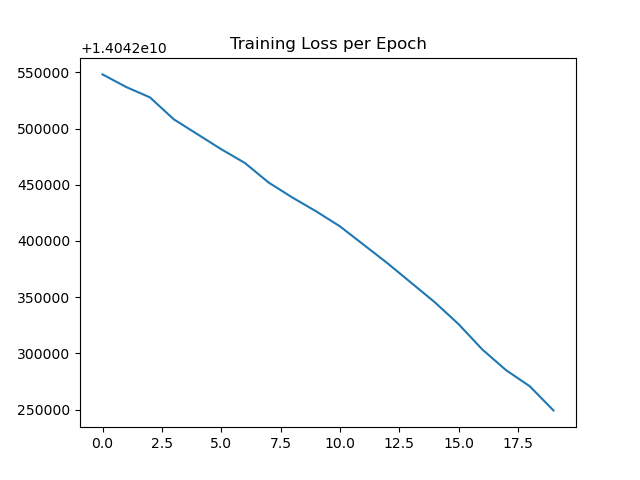
np.sqrt(mean\_squared\_error(y\_test,predictions))

#119847.44133064519

mean\_absolute\_error(y\_test,predictions)

#113826.2502284318

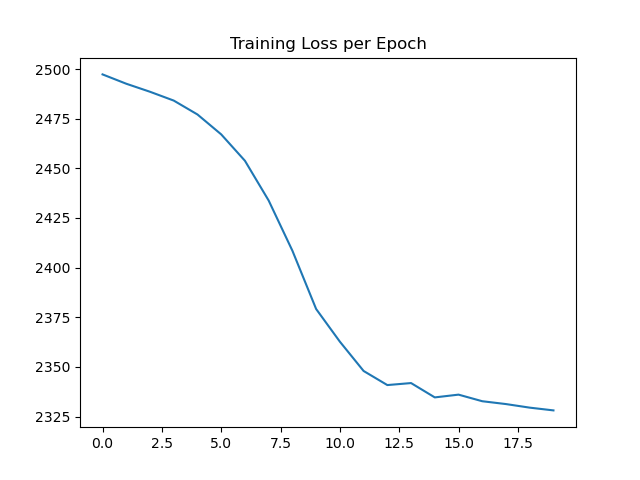
#mean is 112012.639200 almost closer to the MEAN ABS ERROR. So we conclude with this model is moderate for this dataset.



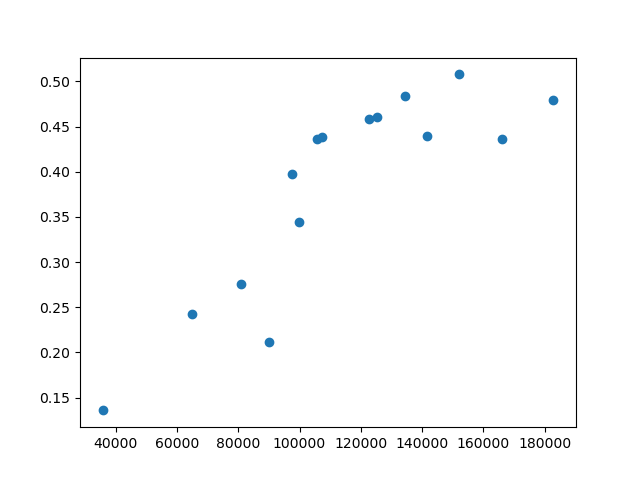
The X is the Len of Range of loss calculated with decreases with wrt to epochs.

At first, we have random assumption hence loss is high at start and slowly decreases.

1. **Forest Fires**



In the Forest Fire Dataset, the training loss comes at stable rate at around 12.5.



The Scatter plot of ytest, prediction also follows a straight line showing that prediction is acceptable.

mean\_squared\_error(y\_test,predictions)

# 7941.9169801851764

#RMSE

np.sqrt(mean\_squared\_error(y\_test,predictions))

#89.11743364900705

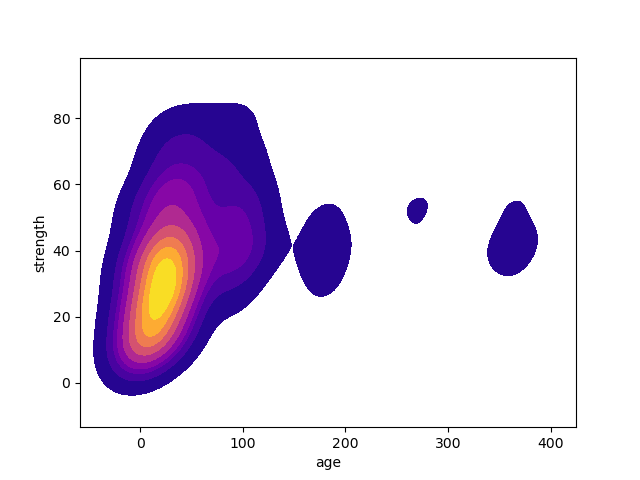
mean\_absolute\_error(y\_test,predictions)

#20.253505704464057

forest['area'].describe()

#mean is 12.847292 almost closer to the MEAN ABS ERROR 20.253505704464057

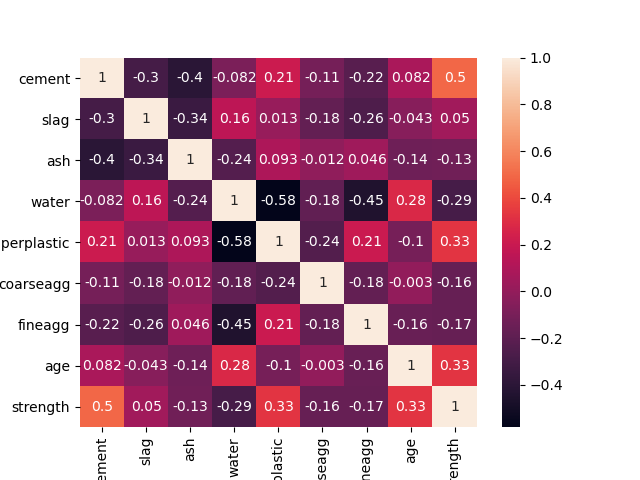
1. **Cement Data:**



Objective : To find the strength of cement based on other parameters.

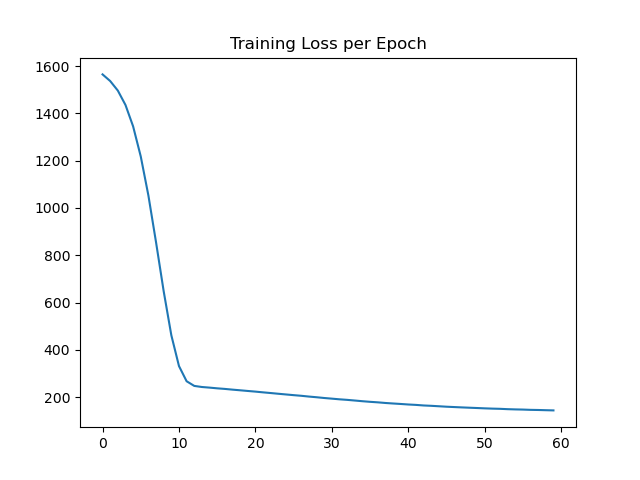
In the above kdeplot of Cement Age v Strength to see If it looses its strength after decades.

And the majority of strength is during the initial phase as shown towards -0-100.



The Strength and Cement has correlation of 0.50 .

The training loss with its range per epoch.



The training loss takes a dip after 10 Epoch and that’s where the model started to perform well and reduce training error.

mean\_squared\_error(y\_test,predictions)

# 7941.9169801851764

#RMSE

np.sqrt(mean\_squared\_error(y\_test,predictions))

#89.11743364900705

mean\_absolute\_error(y\_test,predictions)

#20.253505704464057

forest['area'].describe()

#mean is 12.847292 almost closer to the MEAN ABS ERROR 20.253505704464057