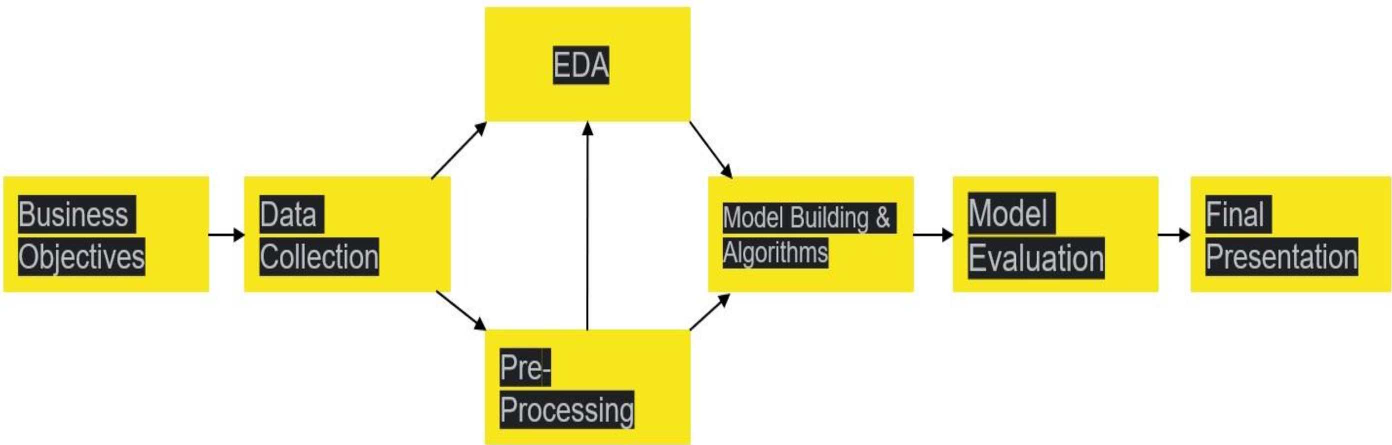
Topic:

Predictive Modelling for Customer Churn

Problem_Statement:

The objective of this assignment is to build a predictive model that can predict customer churn for a given company. The intern will use machine learning techniques to build the model and document the process, including feature selection, model evaluation, and performance metrics.

Approach to solve the problem:



Dataset and features:

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed.

Input variables:

 # bank client data:

1 - age (numeric)

2-job:type of job (categorical: unemployed", "management", "housemaid", "entrepreneur", "student", "bluecollar",etc)

3 - marital : marital status (categorical:"married","divorced","single"; note: "divorced" means divorced or widowed)

4 - education (categorical:"unknown","secondary","primary","tertiary")

5 - default: has credit in default? (binary: "yes","no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes","no")

8 - loan: has personal loan? (binary: "yes","no")

 # related with the last contact of the current campaign:

9 - contact: contact communication type (categorical:"unknown","telephone","cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec") *12*

- duration: last contact duration, in secs (numeric)

 # other attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical:"unknown","other","failure","success")

Output variable (desired target):

17 - y - has the client subscribed a term deposit? (binary: "yes", "no")

So, this dataset has been downloaded from Kaggle website:

https://www.kaggle.com/competitions/bank-marketing-uci/overview

	age;"job";"marital";"education";"default";"balance";"housing";"loan";"contact";"day";"month";"duration";"campaign";"pdays";"previous";"poutcome";"y"
0	30;"unemployed";"married";"primary";"no";1787;...
1	33;"services";"married";"secondary";"no";4789;...
2	35;"management";"single";"tertiary";"no";1350;...
3	30;"management";"married";"tertiary";"no";1476...
4	59;"blue-collar";"married";"secondary";"no";0;...
...	...
4516	33;"services";"married";"secondary";"no";-333;...
4517	57;"self-employed";"married";"tertiary";"yes";...
4518	57;"technician";"married";"secondary";"no";295...
4519	28;"blue-collar";"married";"secondary";"no";11...
4520	44;"entrepreneur";"single";"tertiary";"no";113...

4521 rows × 1 columns

Analysis of the project:

First task was to make dataset in structured form and clean the data.

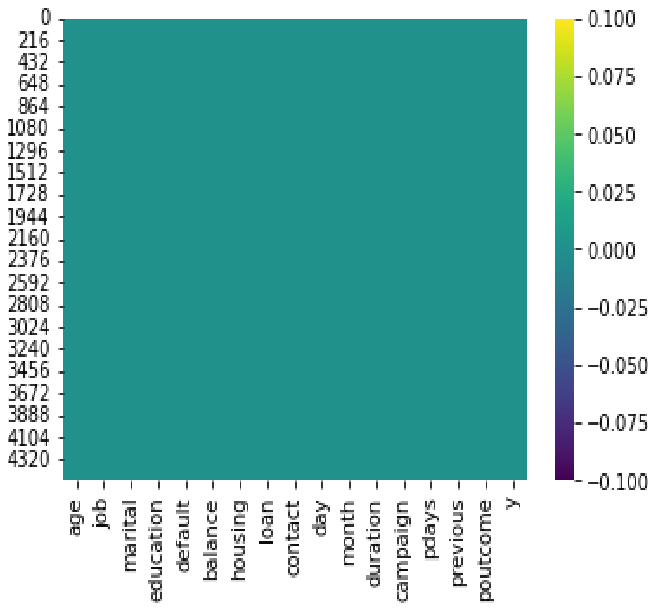
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular	30	jul	329	5	-1	0	unknown	no
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	9	may	153	1	-1	0	unknown	no
4518	57	technician	married	secondary	no	295	no	no	cellular	19	aug	151	11	-1	0	unknown	no
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	6	feb	129	4	211	3	other	no
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	3	apr	345	2	249	7	other	no

4521 rows × 17 columns

Next, as we can see every column’s data types are object, so converting some column’s data type to integer.

```
age           int32
job          object
marital      object
education    object
default      object
balance       int32
housing      object
loan         object
contact      object
day           int32
month        object
duration      int32
campaign      int32
pdays         int32
previous      int32
poutcome     object
y            object
dtype: object
```

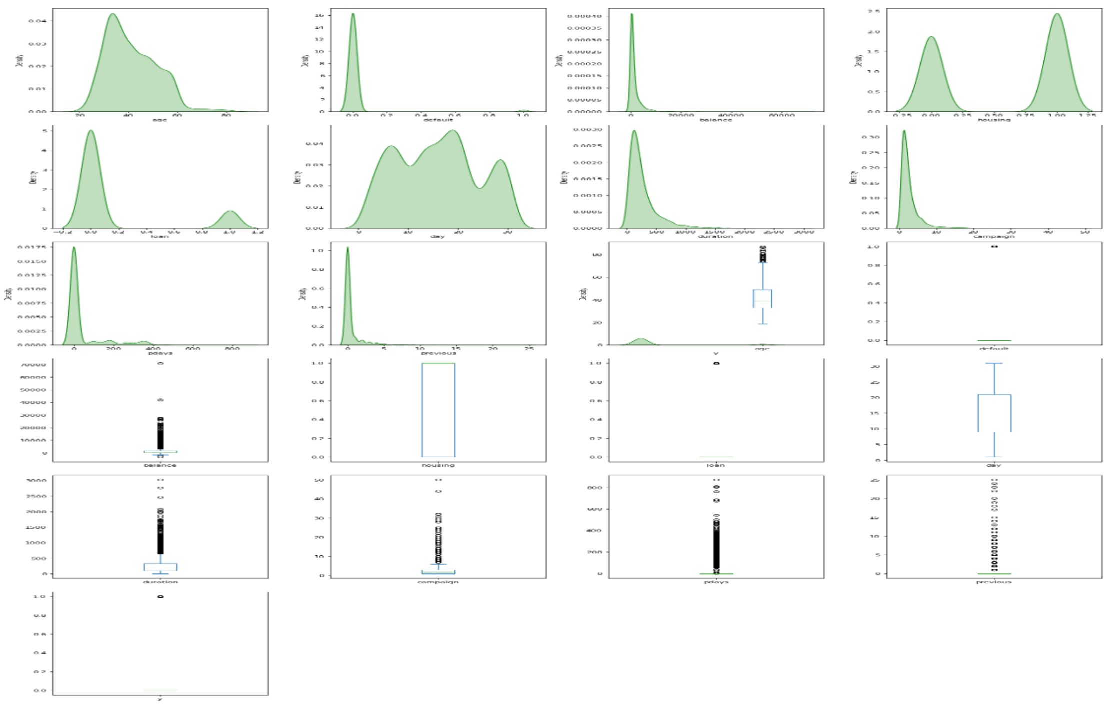
There were no as such missing values in dataset to remove or fill.




```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

Outliers are the data points that differs significantly from other observations, i.e source where model accuracy can fluctuate. Analysing and removing unnecessary outliers are mandatory.

So, there we less outliers so we removed all the outliers.



Next, we implemented label_encoding and count_encoding to transform the categorical values of the relevant features into numerical ones as machine leaning model only works with numerical values.

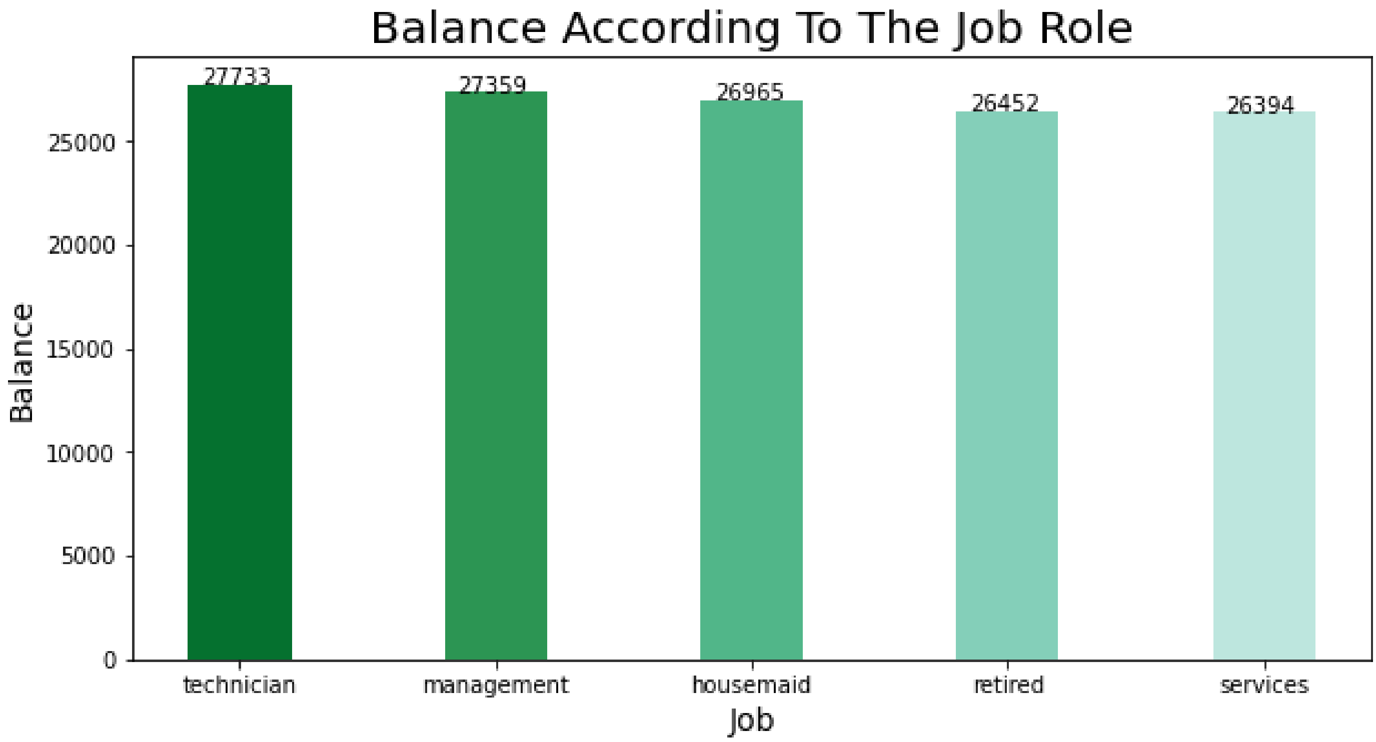
Label_encdoing columns: default, housing, loan, y

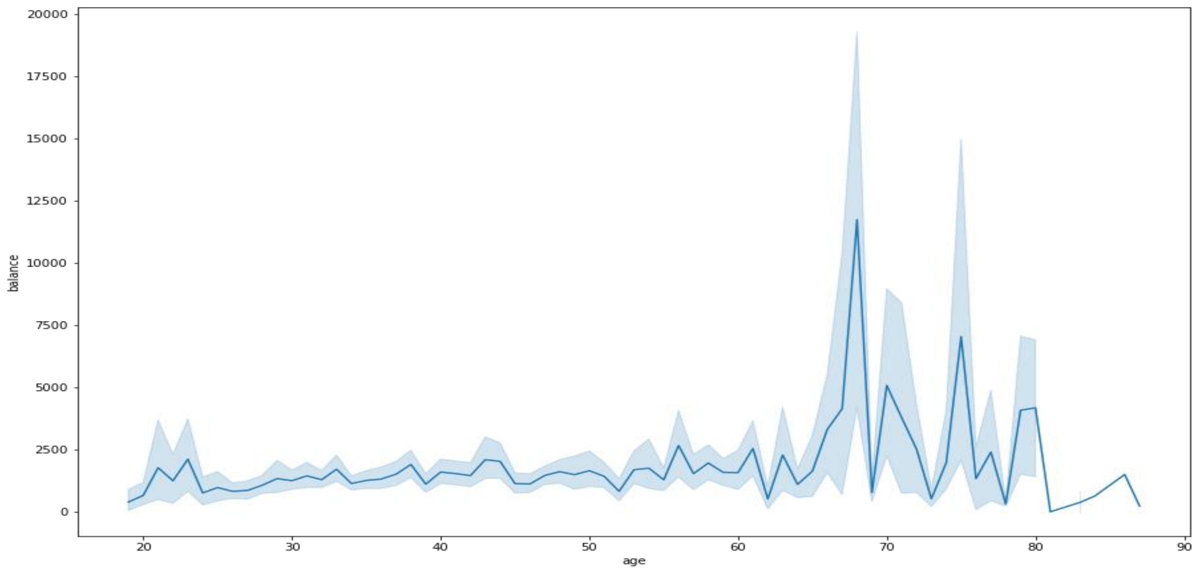
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	0	1787	0	0	cellular	19	oct	79	1	-1	0	unknown	0
2	35	management	single	tertiary	0	1350	1	0	cellular	16	apr	185	1	330	1	failure	0
4	59	blue-collar	married	secondary	0	0	1	0	unknown	5	may	226	1	-1	0	unknown	0
5	35	management	single	tertiary	0	747	0	0	cellular	23	feb	141	2	176	3	failure	0
6	36	self-employed	married	tertiary	0	307	1	0	cellular	14	may	341	1	330	2	other	0

Count_encoding columns: except column y, all other columns

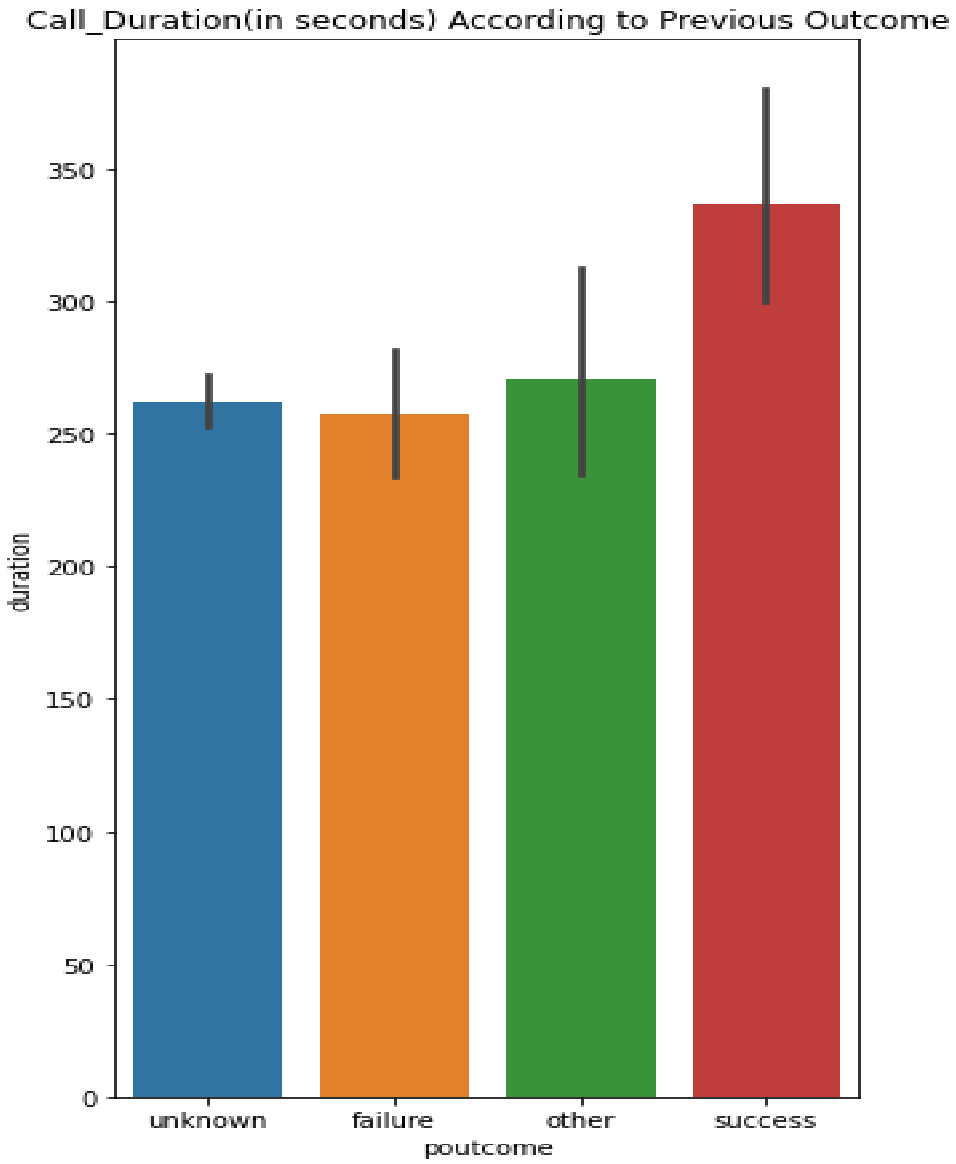
age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
.033752	0.029827	0.611722	0.152276	0.986656	0.000523	0.437729	1.0	0.638148	0.045003	0.018577	0.002355	0.380429	0.813710	0.813710	0.813710	0
.039246	0.221612	0.273940	0.306384	0.986656	0.000262	0.562271	1.0	0.638148	0.034275	0.068289	0.003925	0.380429	0.001047	0.066196	0.109105	0
.017792	0.206436	0.611722	0.494505	0.986656	0.079540	0.562271	1.0	0.294610	0.041340	0.316327	0.002355	0.380429	0.813710	0.813710	0.813710	0
.039246	0.221612	0.273940	0.306384	0.986656	0.000785	0.437729	1.0	0.638148	0.021193	0.049974	0.002878	0.284406	0.001308	0.025641	0.109105	0
.043171	0.040031	0.611722	0.306384	0.986656	0.000262	0.562271	1.0	0.638148	0.043694	0.316327	0.001308	0.380429	0.001047	0.044218	0.045003	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
.036107	0.206436	0.611722	0.494505	0.986656	0.000523	0.562271	1.0	0.638148	0.060963	0.068289	0.002355	0.071952	0.000523	0.066196	0.109105	0
.048404	0.089482	0.273940	0.494505	0.986656	0.001308	0.562271	1.0	0.638148	0.041340	0.129252	0.000262	0.036892	0.813710	0.813710	0.813710	0
.040293	0.089482	0.611722	0.494505	0.986656	0.000262	0.562271	1.0	0.638148	0.039246	0.129252	0.001047	0.036892	0.813710	0.813710	0.813710	0
.018315	0.169806	0.611722	0.494505	0.986656	0.001047	0.437729	1.0	0.638148	0.045003	0.147567	0.003663	0.004971	0.813710	0.813710	0.813710	0
.022240	0.206436	0.611722	0.494505	0.986656	0.000262	0.437729	1.0	0.638148	0.041601	0.049974	0.003401	0.071952	0.000523	0.025641	0.045003	0

 # Some basic visualizations for better understanding:





20000
17500
15000
12500
10000
7500
5000
2500
0
balance
20
30
40
50
60
70
80
90
age



Call_Duration(in seconds) According to Previous Outcome
350
300
250
200
150
100
50
0
duration
unknown
failure
other
success
poutcome

Working with models:

The most significant difference between regression vs classification is that while regression helps predict a continuous quantity, classification predicts discrete class labels.

Here, classification models have been used as there are two label of classes.

Models used are:- Logistic_Regression, SVM, Random_Forest and AdaBoost.

```
print ("Accuracy For Logistic Regression          : ", accuracy_score(y_pred, y_test))
print("Accuracy For SVM                          : ", metrics.accuracy_score(y_pred1, y_test))
print("Accuracy For Random Forest                : ", metrics.accuracy_score(y_pred2, y_test))
print("Accuracy For Adaboost                     : ", metrics.accuracy_score(y_pred3, y_test))
```

```
Accuracy For Logistic Regression          :  0.8928104575163399
Accuracy For SVM                          :  0.8928104575163399
Accuracy For Random Forest                :  0.8875816993464052
Accuracy For Adaboost                     :  0.8954248366013072
```

From above image we can observe that accuracy_scores lies between 88% to 90% which is acceptable.

Accuracy for AdaBoost is 0.8954 i.e 89.54% which is top from all other models.

As we can see AdaBoost is giving the best result.

```
print('Logistic Regression_RMSE                :', np.sqrt(metrics.mean_squared_error(y_pred, y_test)))
print('SVM_RMSE                                :', np.sqrt(metrics.mean_squared_error(y_pred1, y_test)))
print('Random Forest_RMSE                      :', np.sqrt(metrics.mean_squared_error(y_pred2, y_test)))
print('AdaBoost_RMSE                           :', np.sqrt(metrics.mean_squared_error(y_pred3, y_test)))
```

```
Logistic Regression_RMSE                : 0.3273981406234008
SVM_RMSE                                : 0.3273981406234008
Random Forest_RMSE                      : 0.3352883843105734
AdaBoost_RMSE                           : 0.3233808333817773
```

Similarly, from another image, we can observe RMSE values. RMSE values between 0.3to 0.5 are acceptable.

The lower the RMSE, the better the model and its predictions (RMSE values should lie between 0 to 1).