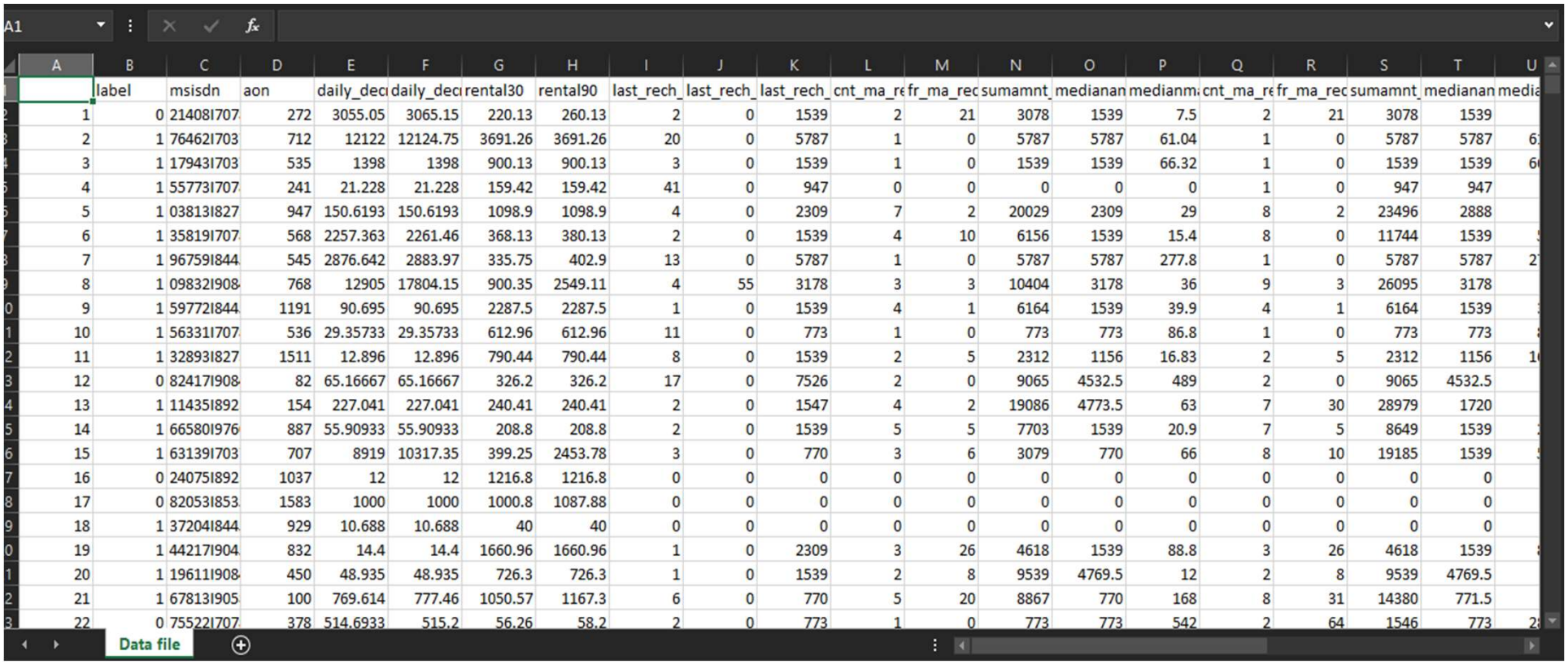
MICRO CREDIT DEFAULTER PROJECT

By
Vishnu vardhan reddy

Data Preparation

- With the help of Pandas Library We will upload our data to Jupyter Notebook.
- Once our data is uploaded with the help of predefined method (i.e. read_csv) we can read data for further processing.
- We have two type of variables in the data:-
    1. Dependent Variable
    2. Independent Variable



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		label	msisdn	aon	daily_dec	daily_dec	rental30	rental90	last_rech_	last_rech_	last_rech_	cnt_ma_re	fr_ma_rec	sumamnt_	medianan	medianm	cnt_ma_re	fr_ma_rec	sumamnt_	medianan	media
2	1	0	21408I707	272	3055.05	3065.15	220.13	260.13	2	0	1539	2	21	3078	1539	7.5	2	21	3078	1539	
3	2	1	76462I703	712	12122	12124.75	3691.26	3691.26	20	0	5787	1	0	5787	5787	61.04	1	0	5787	5787	6
4	3	1	17943I703	535	1398	1398	900.13	900.13	3	0	1539	1	0	1539	1539	66.32	1	0	1539	1539	6
5	4	1	55773I707	241	21.228	21.228	159.42	159.42	41	0	947	0	0	0	0	0	1	0	947	947	
6	5	1	03813I827	947	150.6193	150.6193	1098.9	1098.9	4	0	2309	7	2	20029	2309	29	8	2	23496	2888	
7	6	1	35819I707	568	2257.363	2261.46	368.13	380.13	2	0	1539	4	10	6156	1539	15.4	8	0	11744	1539	
8	7	1	96759I844	545	2876.642	2883.97	335.75	402.9	13	0	5787	1	0	5787	5787	277.8	1	0	5787	5787	2
9	8	1	09832I908	768	12905	17804.15	900.35	2549.11	4	55	3178	3	3	10404	3178	36	9	3	26095	3178	
10	9	1	59772I844	1191	90.695	90.695	2287.5	2287.5	1	0	1539	4	1	6164	1539	39.9	4	1	6164	1539	
11	10	1	56331I707	536	29.35733	29.35733	612.96	612.96	11	0	773	1	0	773	773	86.8	1	0	773	773	
12	11	1	32893I827	1511	12.896	12.896	790.44	790.44	8	0	1539	2	5	2312	1156	16.83	2	5	2312	1156	1
13	12	0	82417I908	82	65.16667	65.16667	326.2	326.2	17	0	7526	2	0	9065	4532.5	489	2	0	9065	4532.5	
14	13	1	11435I892	154	227.041	227.041	240.41	240.41	2	0	1547	4	2	19086	4773.5	63	7	30	28979	1720	
15	14	1	66580I976	887	55.90933	55.90933	208.8	208.8	2	0	1539	5	5	7703	1539	20.9	7	5	8649	1539	
16	15	1	63139I703	707	8919	10317.35	399.25	2453.78	3	0	770	3	6	3079	770	66	8	10	19185	1539	
17	16	0	24075I892	1037	12	12	1216.8	1216.8	0	0	0	0	0	0	0	0	0	0	0	0	
18	17	0	82053I853	1583	1000	1000	1000.8	1087.88	0	0	0	0	0	0	0	0	0	0	0	0	
19	18	1	37204I844	929	10.688	10.688	40	40	0	0	0	0	0	0	0	0	0	0	0	0	
20	19	1	44217I904	832	14.4	14.4	1660.96	1660.96	1	0	2309	3	26	4618	1539	88.8	3	26	4618	1539	
21	20	1	19611I908	450	48.935	48.935	726.3	726.3	1	0	1539	2	8	9539	4769.5	12	2	8	9539	4769.5	
22	21	1	67813I905	100	769.614	777.46	1050.57	1167.3	6	0	770	5	20	8867	770	168	8	31	14380	771.5	
23	22	0	75522I707	378	514.6933	515.2	56.26	58.2	2	0	773	1	0	773	773	542	2	64	1546	773	2

Data file

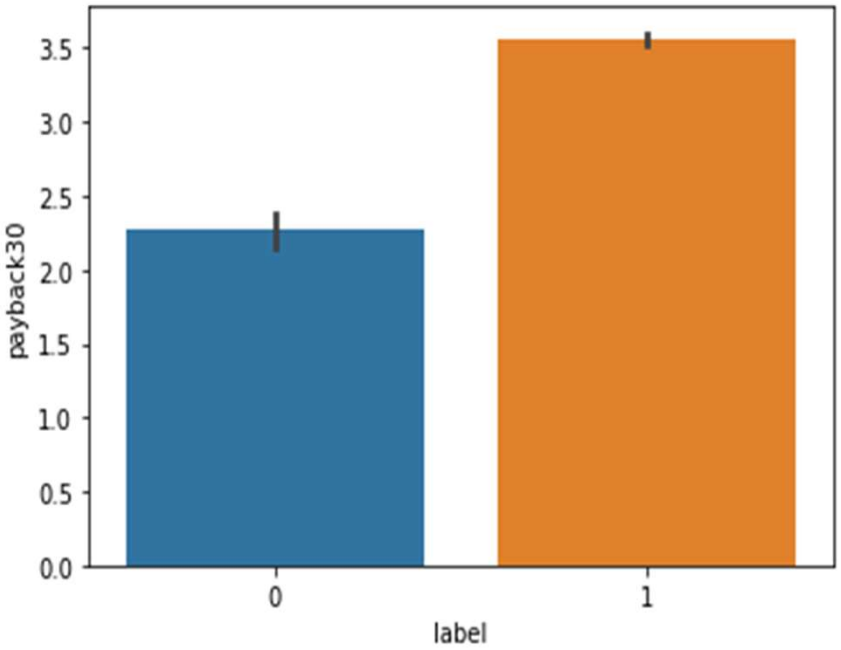
Dataset in CSV format

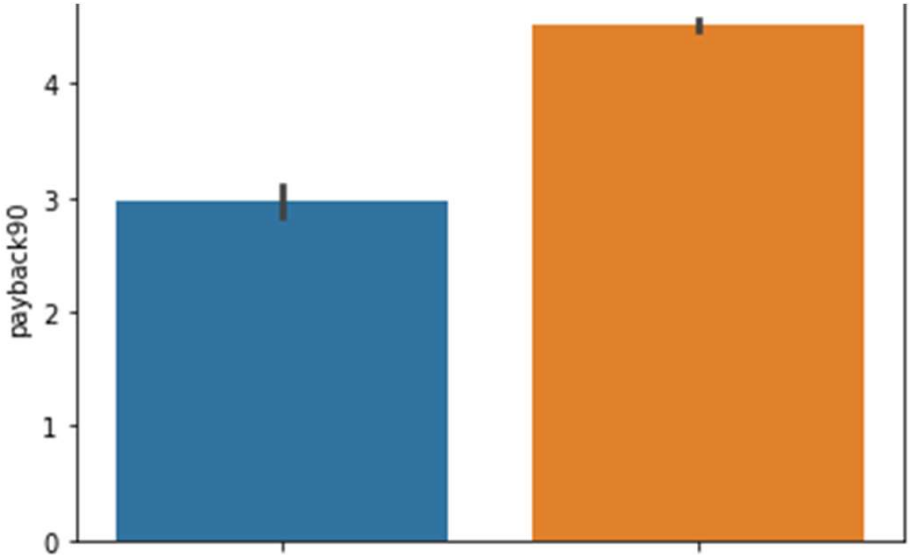
- Info shows the datatypes, count and not null values.
  Label is an independent variable where as all of the other element are dependent variable.

```
df.info()
```

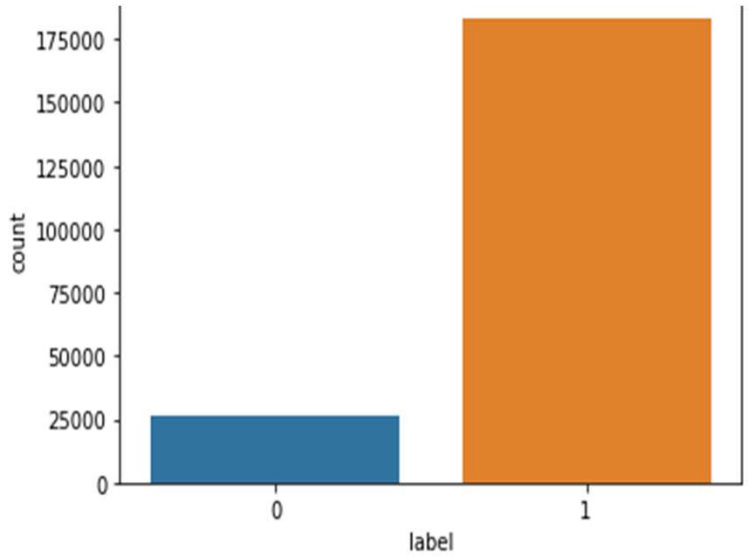
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 36 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   label                 209593 non-null  int64
 1   msisdn                209593 non-null  object
 2   aon                   209593 non-null  float64
 3   daily_decr30          209593 non-null  float64
 4   daily_decr90          209593 non-null  float64
 5   rental30              209593 non-null  float64
 6   rental90              209593 non-null  float64
 7   last_rech_date_ma     209593 non-null  float64
 8   last_rech_date_da     209593 non-null  float64
 9   last_rech_amt_ma      209593 non-null  int64
 10  cnt_ma_rech30         209593 non-null  int64
 11  fr_ma_rech30          209593 non-null  float64
 12  sumamnt_ma_rech30     209593 non-null  float64
 13  medianamnt_ma_rech30  209593 non-null  float64
 14  medianmarechprebal30  209593 non-null  float64
 15  cnt_ma_rech90         209593 non-null  int64
 16  fr_ma_rech90          209593 non-null  int64
 17  sumamnt_ma_rech90     209593 non-null  int64
 18  medianamnt_ma_rech90  209593 non-null  float64
 19  medianmarechprebal90  209593 non-null  float64
 20  cnt_da_rech30         209593 non-null  float64
 21  fr_da_rech30          209593 non-null  float64
 22  cnt_da_rech90         209593 non-null  int64
 23  fr_da_rech90          209593 non-null  int64
 24  cnt_loans30           209593 non-null  int64
 25  amnt_loans30          209593 non-null  int64
 26  maxamnt_loans30       209593 non-null  float64
 27  medianamnt_loans30    209593 non-null  float64
 28  cnt_loans90           209593 non-null  float64
 29  amnt_loans90          209593 non-null  int64
 30  maxamnt_loans90       209593 non-null  int64
 31  medianamnt_loans90    209593 non-null  float64
 32  payback30             209593 non-null  float64
 33  payback90             209593 non-null  float64
 34  pcircle               209593 non-null  object
 35  pdate                 209593 non-null  object
dtypes: float64(21), int64(12), object(3)
```

- In the label value 0 represent the failure of the payment and 1 Represents the successor who had paid credit successfully. Based on the given data following are the our initial findings:-

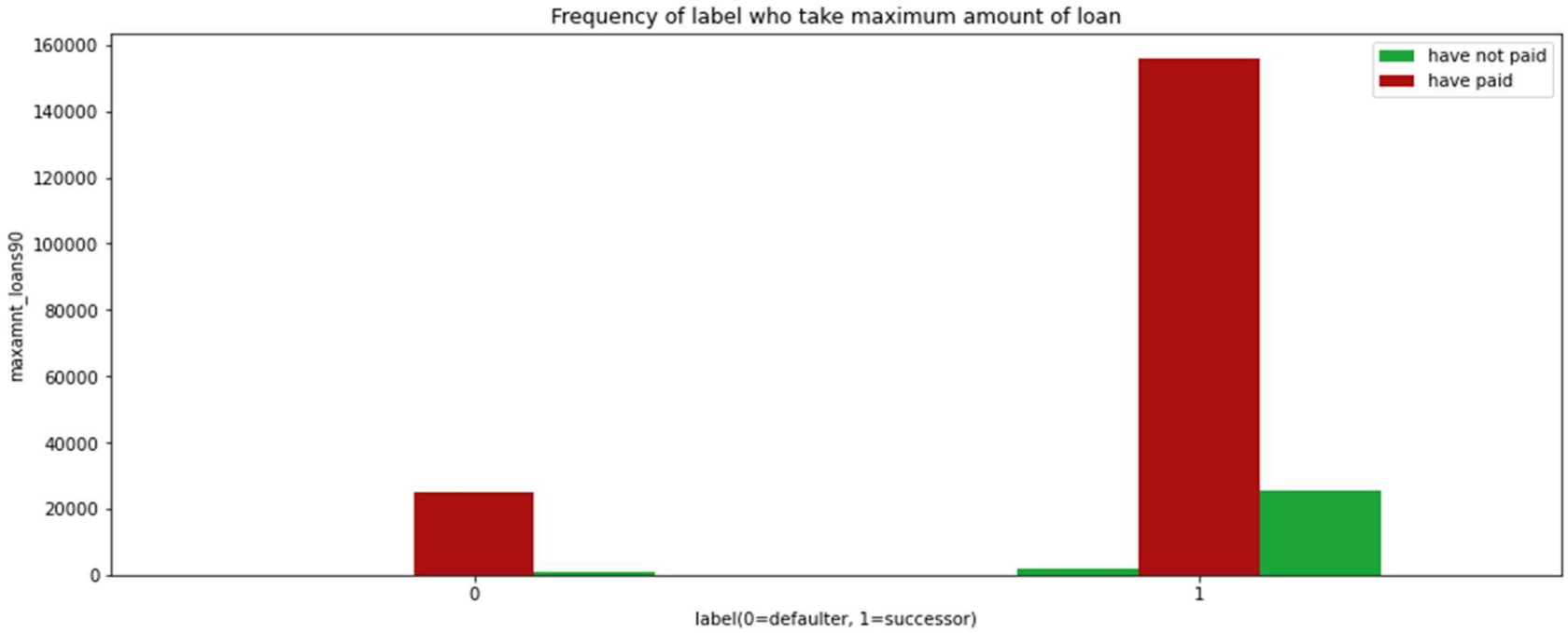




- The average payback time is less for the defaulter cases as compare to other category in both of the scenarios.

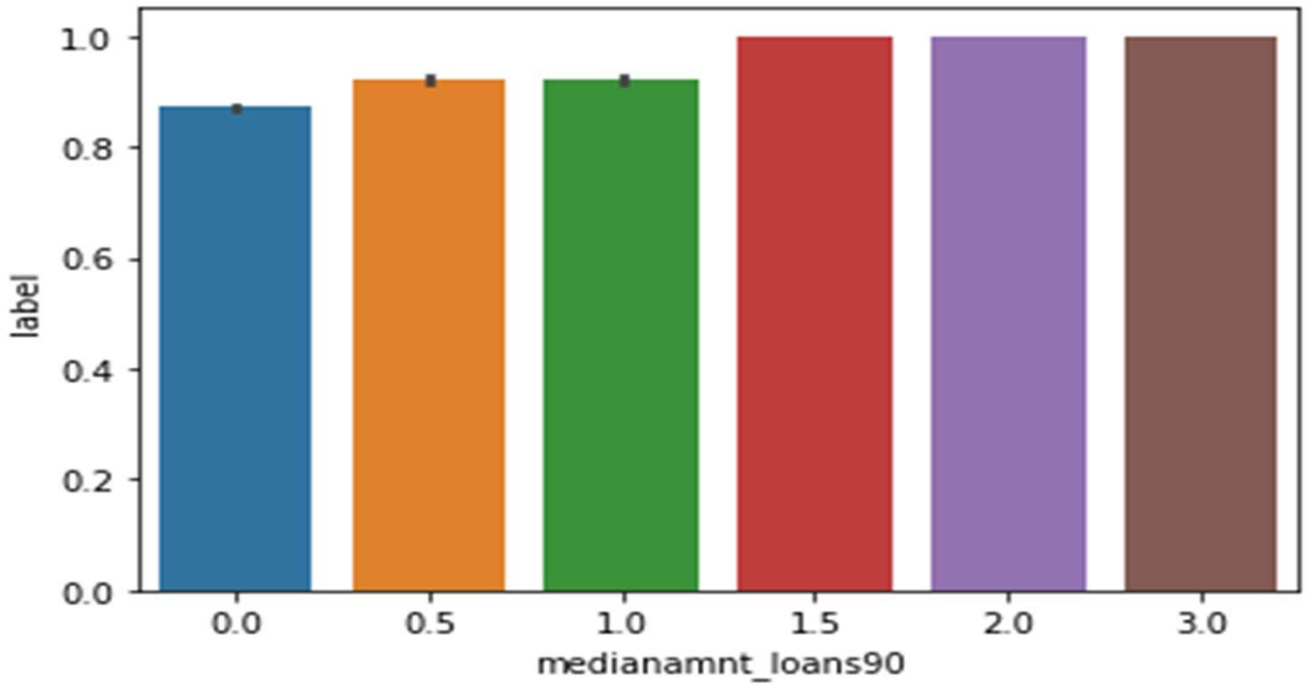


- From the above graphical chart we can easily understand that out of 200K subscribers on 25K did not paid the credit back.



- From the above graphical chart following is our finding:-
- The subscriber who have not paid the credit is less the 10% of the total who have taken credit.

- Median of amounts of loan taken by the user in last 90 days.



Data Preprocessing

- The Complete data is divided in the ration of 70:30 for train and test respectively

- I have dropped the Pcircle column since for the current processing it was not used

- There is null value in the dataset and there are some outliers present in the dataset which has been removed with the help of predefine methods

- Once our data is ready categorical variables are converted into the other form, which we can apply further on algorithms

Evaluation Process

Evaluation Matrices:

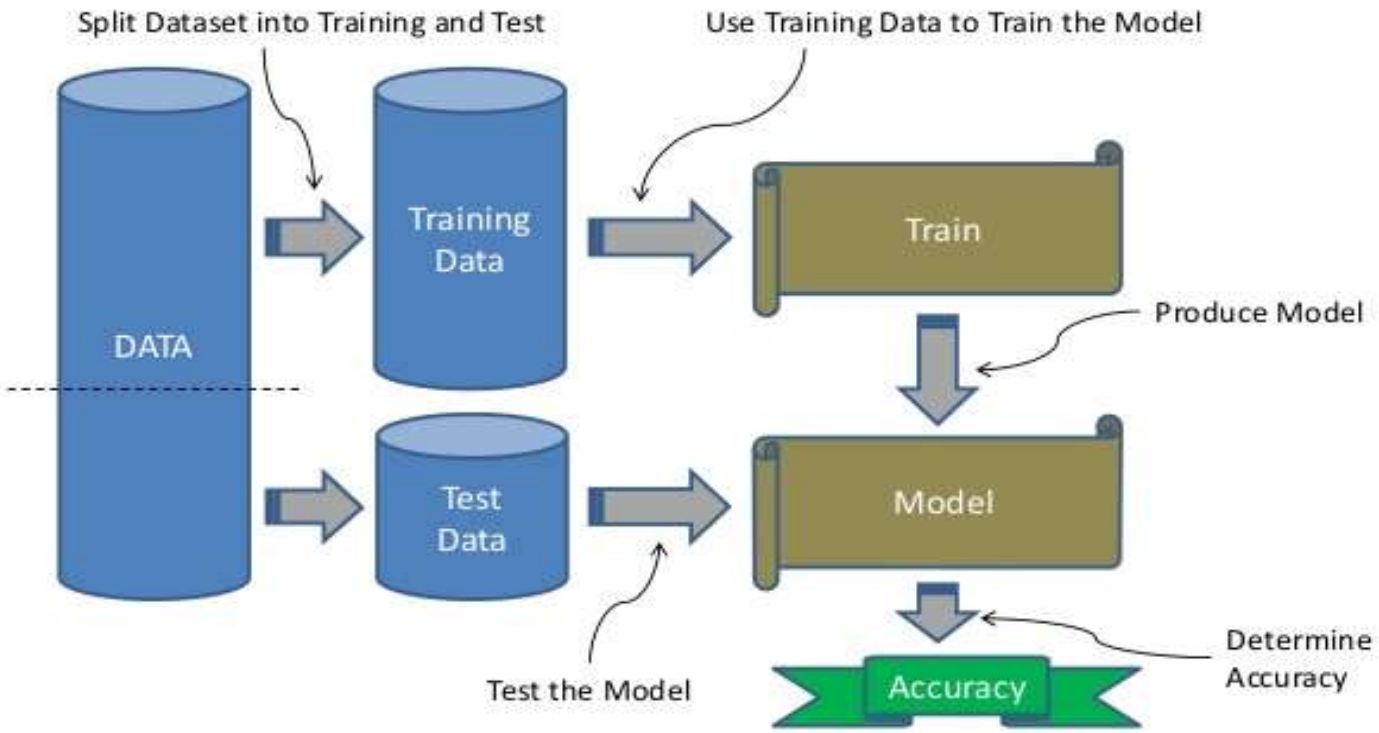
- **Accuracy** - it determines how often a model predicts default and non default correctly.
- **Precision**-it calculates whenever our models predicts it is default how often it is correct.
- **Recall**- Recall regulate the actual default that the model is actually predict.
- **Precision Recall Curve** - PRC will display the tradeoff between Precision and Recall threshold.

Cross Validations:

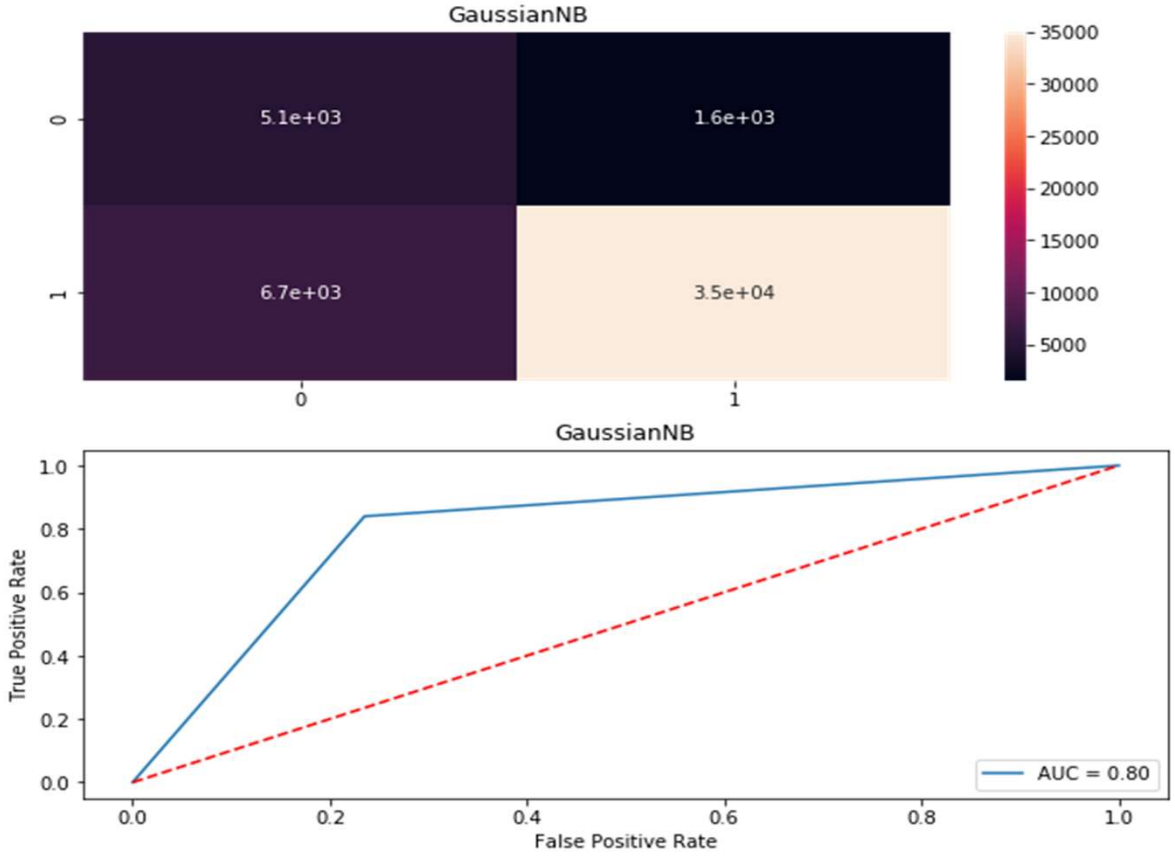
- K Fold cross validations , K = 5

It's About Training

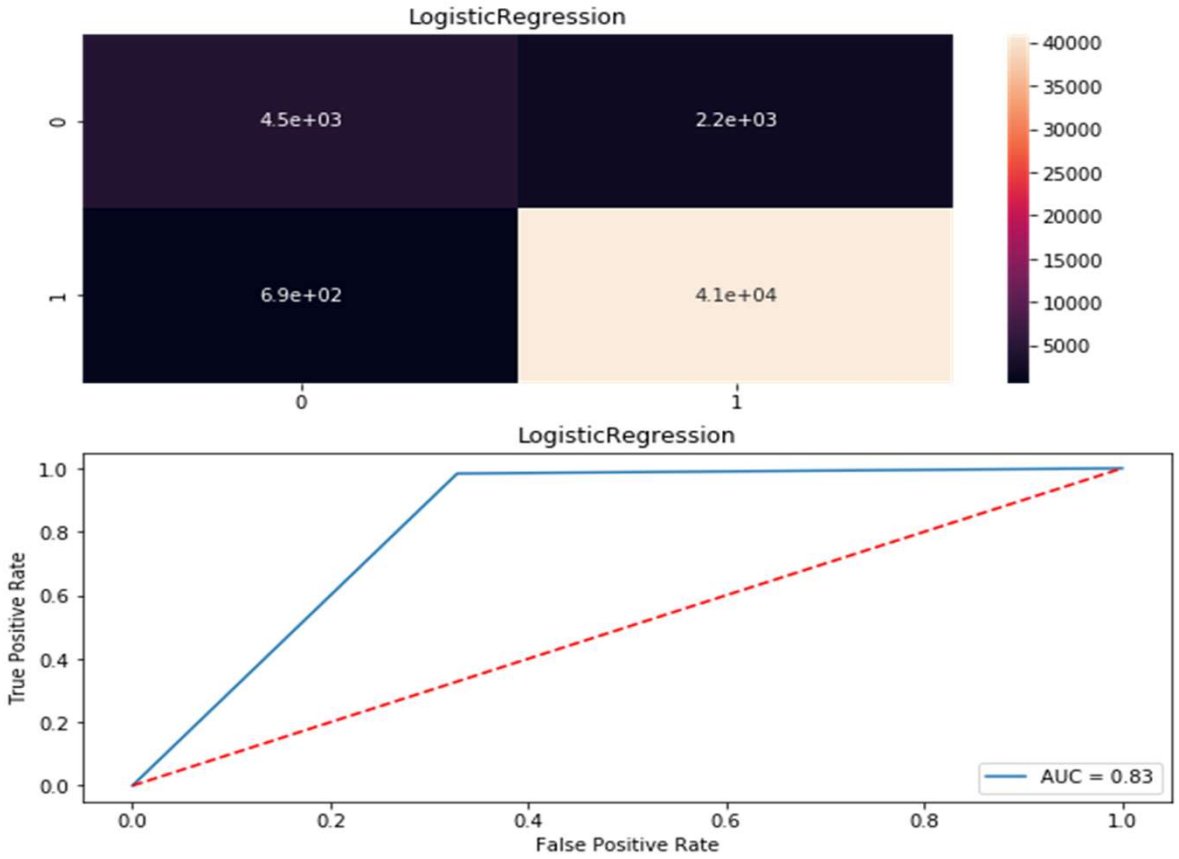
Machine Learning is about using data to train a model



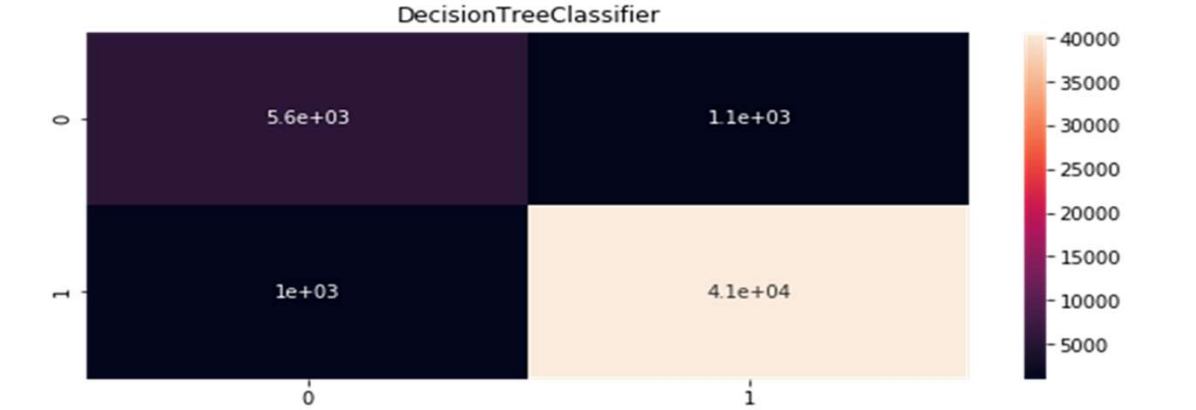
Gaussian NB AUC

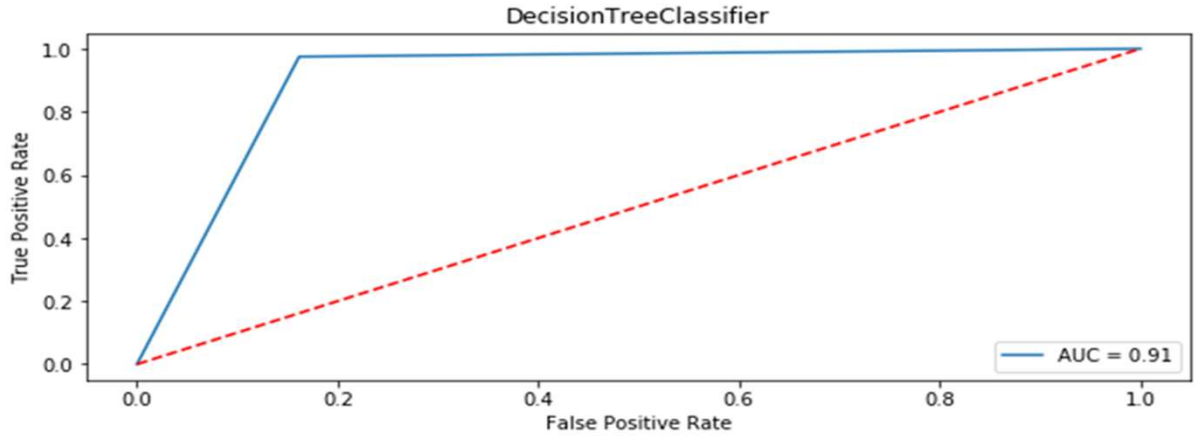


Logistic Regression AUC

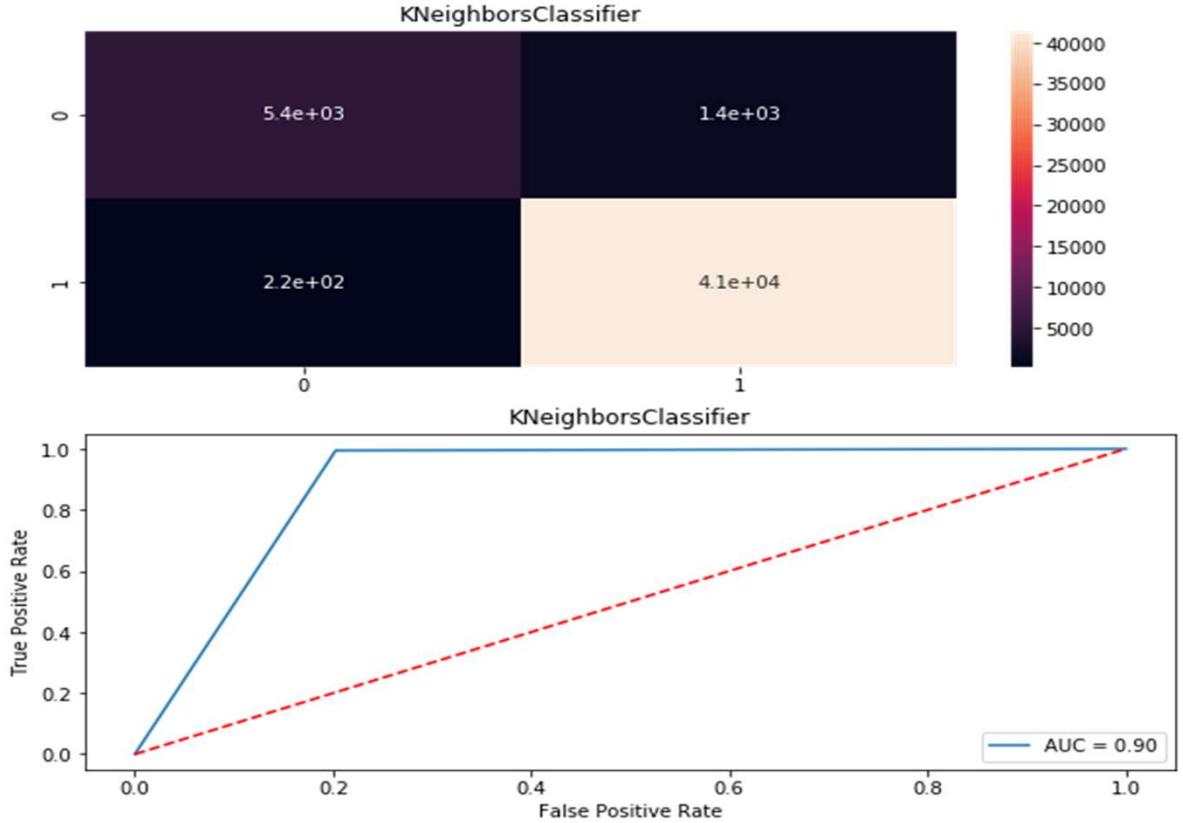


Decision Tree AUC





KNN Neighbours AUC



Result

- From the details in the below table it is clearly understandable that we are getting best result with the help of KN**eighborsClassifier** so we save this model with the help of **joblib** Library.

Out[90]:

	Model	Accuracy_score	Cross_val_score	Roc_auc_curve
0	KNeighborsClassifier	96.763006	96.904593	89.675366
1	LogisticRegression	94.031792	94.011706	82.759118
2	DecisionTreeClassifier	95.656482	95.776174	90.830687
3	GaussianNB	82.947977	83.213085	80.238295

Conclusion

- I checked the data first and uploaded the data in jupyter notebook
- I visualized the features, Performed the preprocessing in the data and understood the relationship between different features
- I used both train-validation split and the cross validation to evaluate the model effectiveness to predict the target values
- At the end I applied four predictive models in the data
- I started with KNeighboursClassifier, Logistic Regression, Decision Tree Classifier, GaussianNB and based on the best result I decided to go ahead with KNeighboursClassifier