

Micro Credit Defaulter project



Submitted by:

Kirti Ohlan

**ACKNOWLEDGMENT**

I would like to express my deepest gratitude to my SME (Subject Matter Expert) Shwetank Mishra as well as Flip Robo Technologies who gave me the opportunity to do this project on Micro Credit Defaulter, which also helped me in doing lots of research wherein I came to know about so many new things.

Also, I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement. All the external resources that were used in creating this project are listed below:

1. <https://www.google.com/>
2. <https://www.youtube.com/>
3. <https://scikit-learn.org/stable/user_guide.html>
4. <https://github.com/>
5. <https://www.kaggle.com/>
6. <https://medium.com/>
7. <https://towardsdatascience.com/>
8. <https://www.analyticsvidhya.com/>

**INTRODUCTION**

* Business Problem Framing

This is a classic Business problem which helps Micro Financing Institutions and other Lending companies reduce Credit risks by recognizing potential Defaulters.

* Conceptual Background of the Domain Problem

Before advancement of Data Science, loan lending companies used to risk a high rate of defaulting. Many a times a perfect candidate would display erratic financial and repayment behavior after being approved for loan. Machine Learning can help lenders predict potential defaulters before approving their candidature using their past data. The candidates’ income, past debt and repayment behavior can be important metrics for the same.

* Review of Literature

In Indonesia, the Government has emphasized increasing access to financial services as a priority component of its financial sector reform agenda. As described in the World Bank‘s report on ―Making the New Indonesia Work for the Poor‖, significant barriers in accessing loan and saving instruments remain for households and enterprises of all sizes, making them vulnerable to economic shocks. Extending the reach of financial services to the poor could have a major impact upon employment generation and poverty reduction

Prediction of defaulters utilizing machine learning approach, A dataset of micro credit defaulter was gathered and utilized to develop the machine learning technique for the study effort. Various figures of variables have been used to train the classifiers to demonstrate how feature extraction might affect validity of the model.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

We are building a model in Machine Learning to predict the actual value of the prospective properties and decide whether to invest in them or not. So, this model will help us to determine which variables are important to predict the of defaulters This will help to determine the defaulter with the available independent variables.

Classification analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The most common form of classification analysis is logistic regression, in which one finds the line that most closely fits the data according to a specific mathematical criterion. For specific mathematical reasons this allows the researcher to estimate the

conditional expectation of the dependent variable when the independent variables take on a given set of values.

Classification analysis is also a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

* Data Sources and their formats
* Data set provided by Flip Robo was in the format of CSV (Comma Separated Values). The dimension of data is 209593 rows and 37 columns.

.

* Data Preprocessing Done

Data pre-processing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre- process our data before feeding it into our model. Therefore, it is the first and crucial step while creating a machine learning model. I have used some following pre-processing steps:

1. Creating the dataset as a dataframe
2. Used pandas to set display I ensuring we do not see any truncated information
3. Checked the number of rows and columns present in our dataset
4. Checked for missing data and the number of rows with null values
5. There is no null value presents
6. Dropped all the unwanted columns and duplicate data present in our dataframe
7. Checked the unique values information in each column to get a list for categorical data
8. Used Pandas Profiling during the visualization phase along with cat plot, count plot, scatter plot and the others
9. Thoroughly checked for outliers and skewness information
10. With the help of heatmap, correlation bar graph was able to understand the Feature vs Label relativity and insights on multicollinearity amongst the feature columns
11. Separated feature and label data to ensure feature scaling is performed avoiding any kind of biasness
12. Checked for the best random state to be used on our classification Machine Learning model pertaining to the feature importance details
13. Finally created a classification model function along with evaluation metrics to pass through various model formats

* Data Inputs- Logic- Output Relationships
* When we loaded the dataset, we had to go through various data pre processing steps to understand what was given to us and what we were expected to predict for the project. When it comes to logical part the domain expertise of understanding. In Data Science community there is a saying “Garbage In Garbage Out” therefore we had to be very cautious and spent almost 80% of our project building time in understanding each and every aspect of the data how they were related to each other as well as our target label.
* With the objective of predicting defaulters accurately we had to make sure that a model was built that understood the customer priorities trending in the market imposing those norms . I tried my best to retain as much data possible that was collected. I did not want to impute data and then cause a biasness in the machine learning model from values that did not come from real people.
* State the set of assumptions (if any) related to the problem under consideration

The assumption part for me was relying strictly on the data provided to me and taking into consideration that datasets were obtained from real people surveyed for their preferences

* Hardware and Software Requirements and Tools Used

Hardware Used:

1. RAM: 8 GB
2. CPU: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz
3. GPU: AMD Radeon ™ Vega 8 Graphics andNVIDIA GeForce GTX 1650 Ti

Software Used:

1. Programming language: Python
2. Distribution: Anaconda Navigator
3. Browser based language shell: Jupyter Notebook

Libraries/Packages Used:

Pandas, NumPy, matplotlib, seaborn, scikit-learn and pandas\_profiling

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data and EDA to check the correlation of independent and dependent features. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models.

For this project we need to predict the defaulter, means our target column is categorical so this is a classification problem. I have used various classification algorithms and tested for the prediction. By doing various evaluations I have selected Extra Trees Regressor as best suitable algorithm for our final model as it is giving good r2-score and least difference in r2-score and CV-score among all the algorithms used.Other regression algorithms are also giving me good accuracy but some are over-fitting and some are with under-fitting the results which may be because of less amount of data.

* In order to get good performance as well as accuracy and to check my model from over-fitting and under-fitting I have made use of the K-Fold cross validation and then hyper parameter tuned the final model.

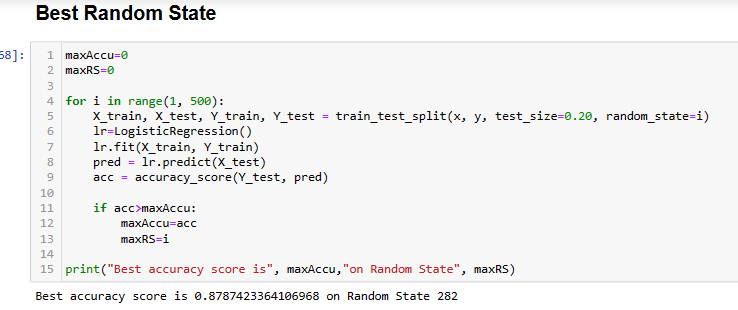
* Testing of Identified Approaches (Algorithms)

The algorithms used on training and test data are as follows:

1. Logistic Classification Model
2. Decision Tree Classification Model
3. Random Forest Classification Model
4. K Nearest Neighbours Classification Model
5. Ada Boost Classification Model

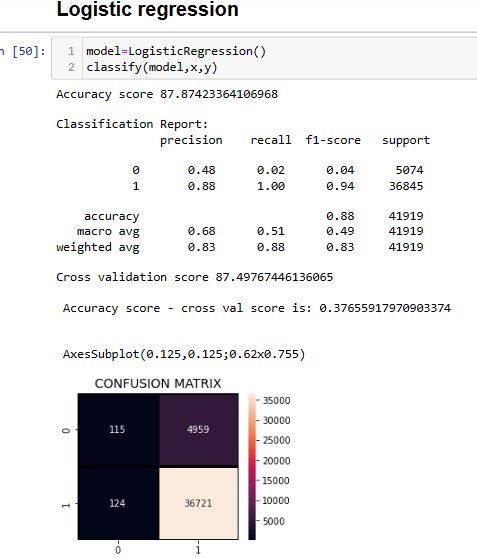
* Run and Evaluate selected models

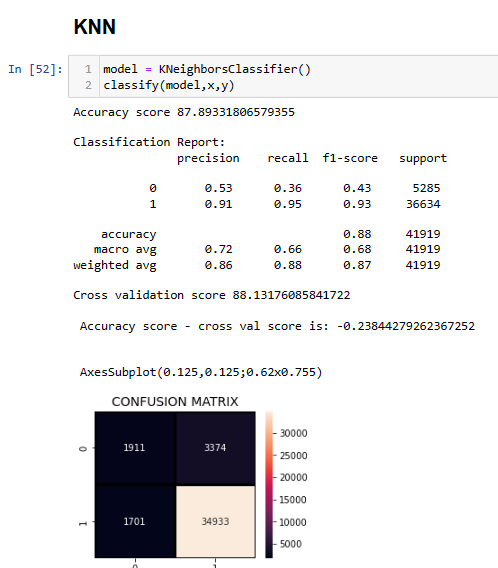
I used a total of 5 Classification Models after choosing the random state amongst 1-500 number. Then I even defined a function for getting the classification model trained and evaluated. The code for the models is listed below.

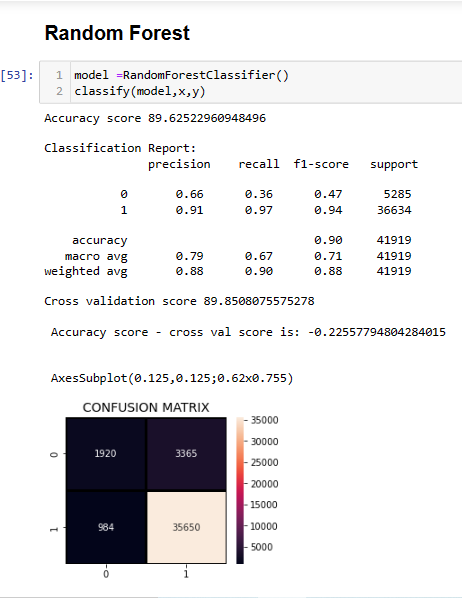


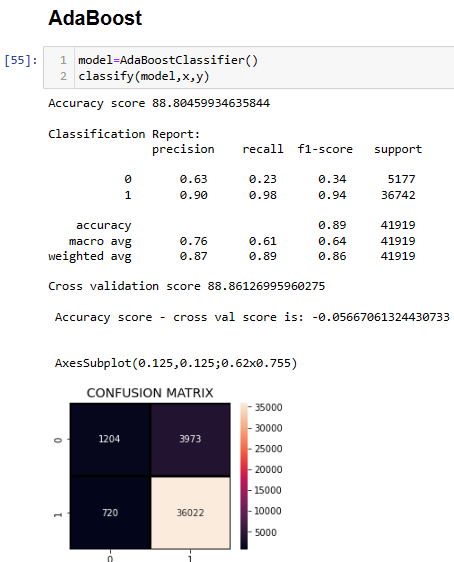
**Classification Model function:**

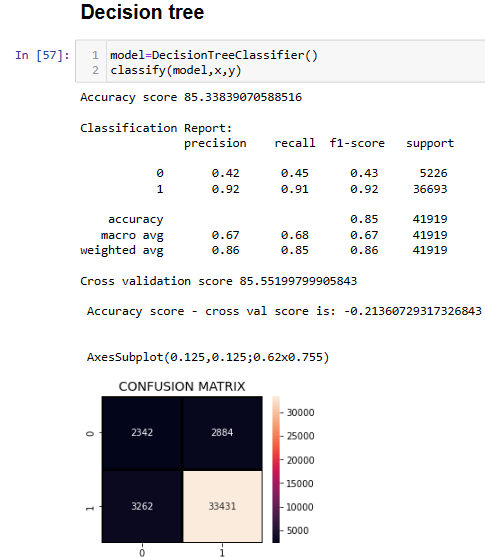


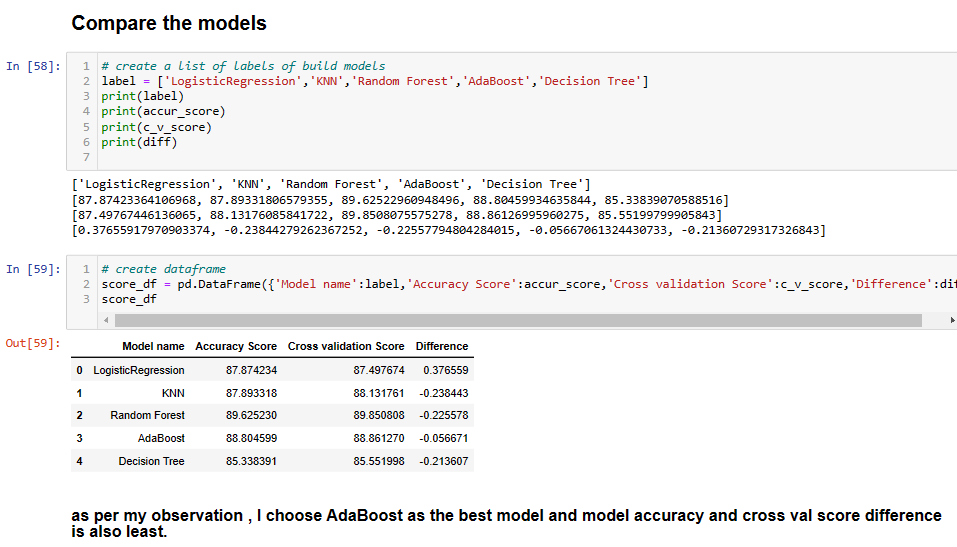












* Key Metrics for success in solving problem under consideration

The key metrics used here were Accuracy score, Confusion matrix, classification report, cross validation score, roc auc curve. We tried to find out the best parameters and also to increase our scores by using Hyperparameter Tuning and we will be using GridSearchCV method.

**1. Cross Validation:**

Cross-validation helps to find out the over fitting and under fitting of the model. In the cross validation the model is made to run on different subsets of the dataset which will get multiple measures of the model. If we take 5 folds, the data will be divided into 5 pieces

where each part being 20% of full dataset. While running the Cross-validation the 1st part (20%) of the 5 parts will be kept out as a holdout set for validation and everything else is used for training data. This way we will get the first estimate of the model quality of the dataset.

In the similar way further iterations are made for the second 20% of the dataset is held as a holdout set and remaining 4 parts are used for training data during process. This way we will get the second estimate of the model quality of the dataset. These steps are repeated during the cross-validation process to get the remaining estimate of the model quality.

**2. Accuracy Score:**

It is a statistical measure that represents the goodness of fit of a classification model. Accuracy score in machine learning is an evaluation matrics that measure the no of correct predictions made by a model in relation to the total no of predictions.

**3. Confusion matrix:**

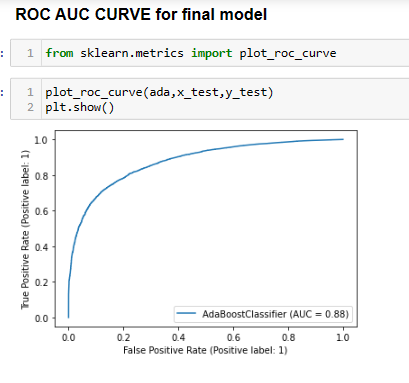
Confusion matrix a table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcomes

**4. Classification report:**

It is used to show the precision, recall,f1 score and support of your trained classification model.

**5. ROC AUC Curve:**

It plots the true positive rate versus the false positive rate at different classification thresholds. The thresholds are different probability cutoffs that separate the two classes in binary classification

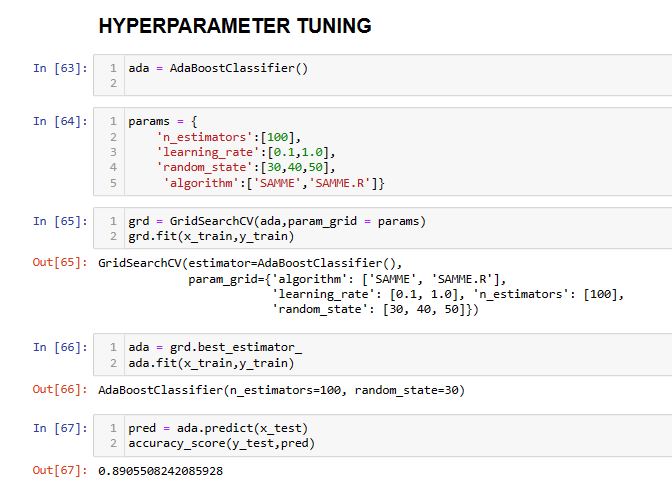


**5. Hyperparameter Tuning:**

There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. You must select from a specific list of hyperparameters for a given model as it varies from model to model.

We are not aware of optimal values for hyperparameters which would generate the best model output. So, what we tell the model is to explore and select the optimal model architecture automatically. This selection procedure for hyperparameter is known as Hyperparameter Tuning. We can do tuning by using GridSearchCV.

GridSearchCV is a function that comes in Scikit-learn (or SK-learn) model selection package. An important point here to note is that we need to have Scikit-learn library installed on the computer. This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, we can select the best parameters from the listed hyperparameters.



* **Visualizations**

I used pandas profiling to get the over viewed visualization on the pre-processed data.pandas-profiling is an open-source Python module with which we can quickly do an exploratory data analysis with just a few lines of code. It generates interactive reports in web format that can be presented to any person, even if they don’t know programming. It also offers report generation for the dataset with lots of features and customizations for the report generated. In short, what pandas-profiling does is save us all the work of visualizing and understanding the distribution of each variable. It generates a report with all the information easily available.

I created cat plots, count plots and dist plots to get further visual insights on our training dataset feature values.

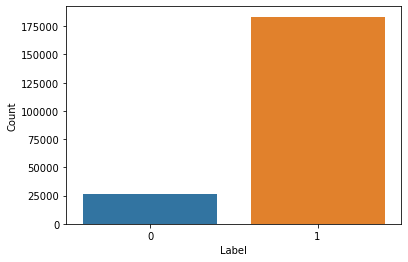
Code:

g=sns.countplot(x="label",data=data)

g.set(xlabel="Label",ylabel="Count")

plt.show()

Output:



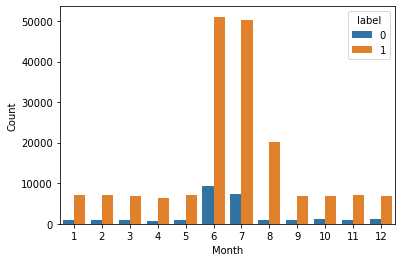
Code:

g=sns.countplot(x="Month",hue='label',data=data)

g.set(xlabel="Month",ylabel="Count")

plt.show()

Output:



Code:

data.describe()

plt.figure(figsize=(20,18))

sns.heatmap(round(data.describe()[1:].transpose(),2),annot=True,fmt='f',cmap='coolwarm')

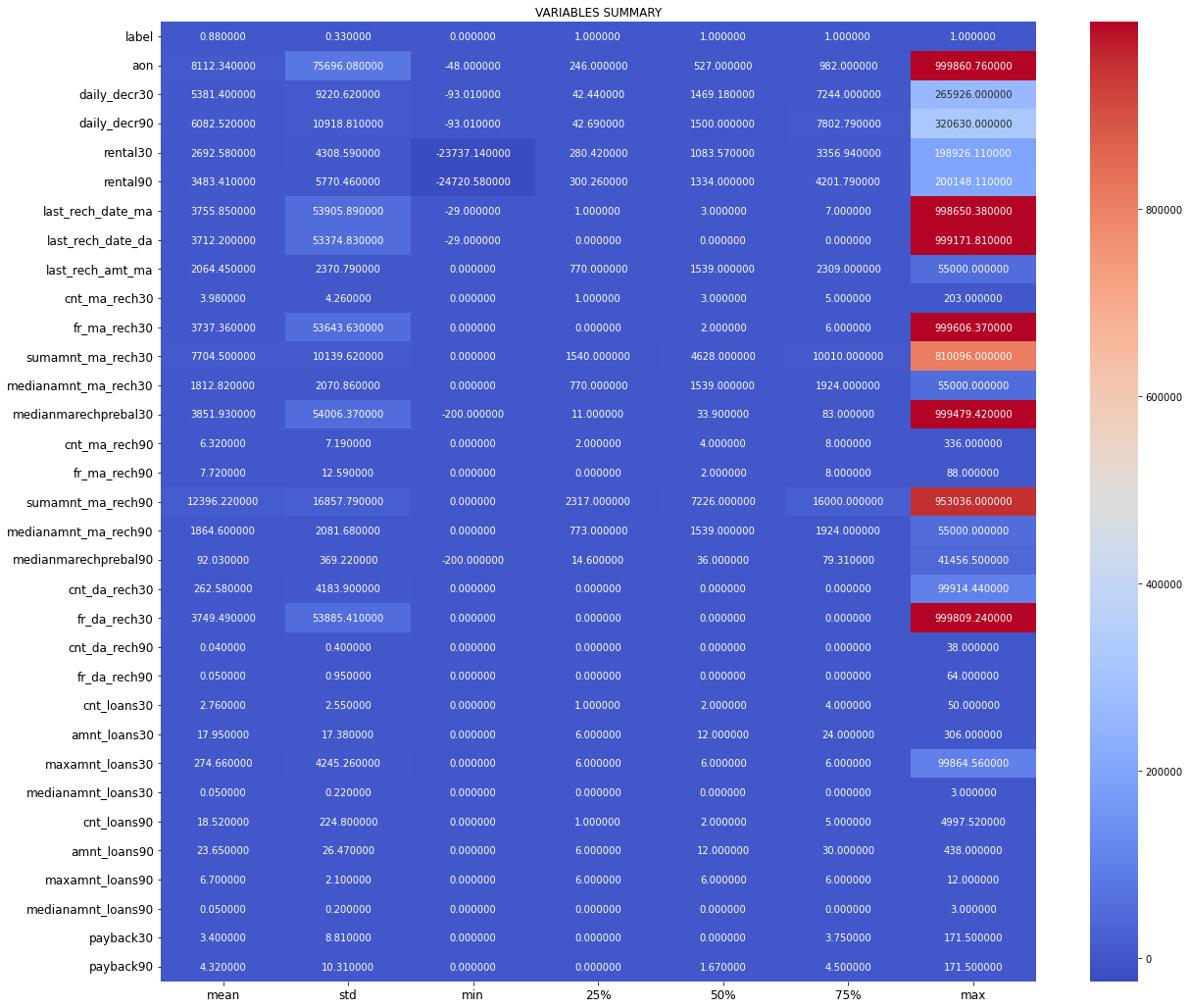
plt.xticks(fontsize=12)

plt.yticks(fontsize=12)

plt.title("VARIABLES SUMMARY")

plt.show

Output:



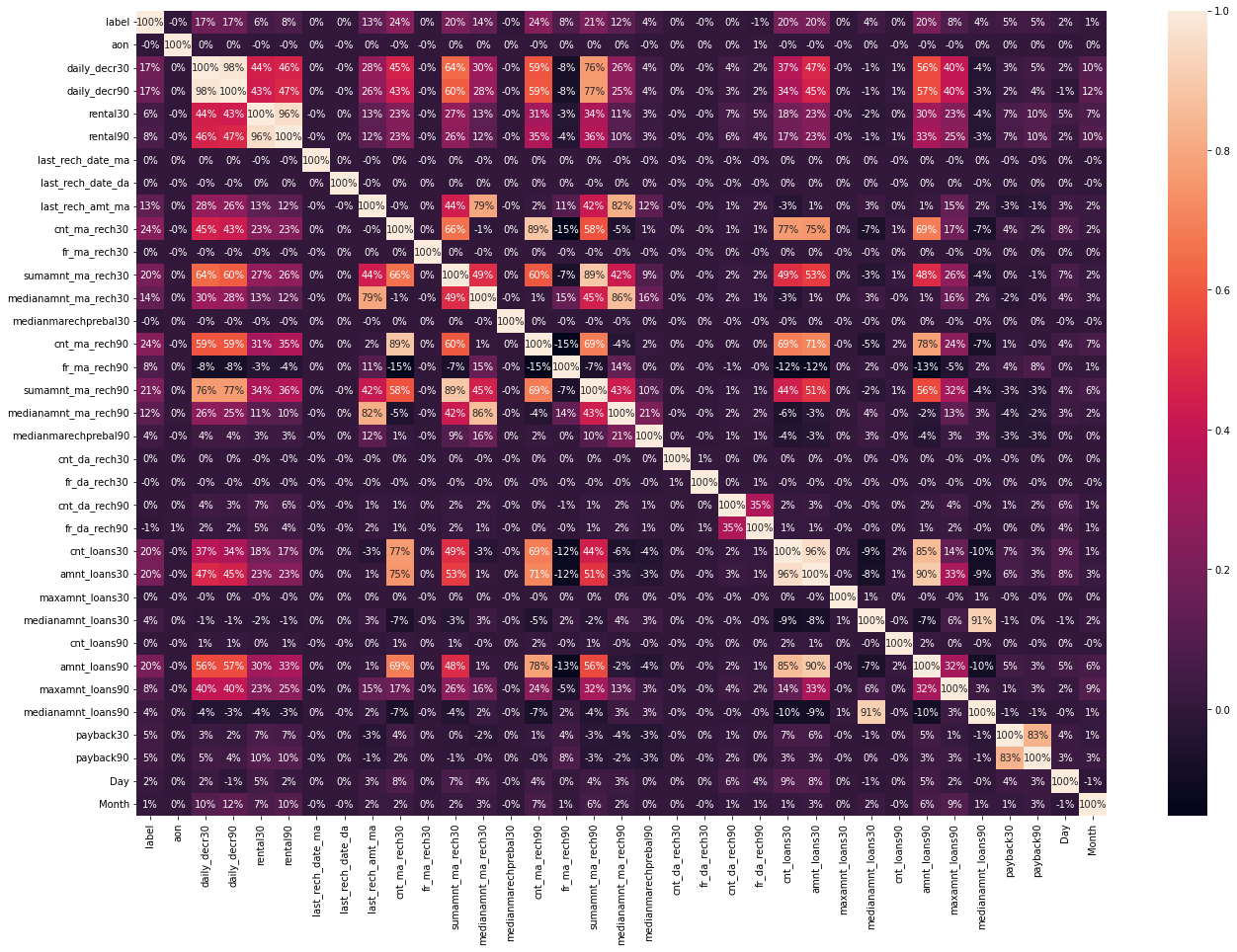
* Interpretation of the Results
* Visualizations: It helped me to understand the correlation between independent and dependent features. Also, helped me with feature importanceand to check for multi collinearity issues. Detected outliers/skewness with the help of boxplot and distribution plot. I got to know the count of a particular category for each feature by using count plot and most importantly with predicted target value distribution as well asscatter plot helped me to select the best model.
* Pre-processing: Basically, before building the model the dataset should be cleaned and scaled by performing few steps. As I mentioned above in the pre-processing steps where all the important features are present in the dataset and ready for model building.
* Model Creation: Now, after performing the train test split, I have x\_train, x\_test, y\_train&y\_test, which are required to build Machine learning models. I have built multiple classification models to get the best Accuracy\_score, confusion matrix, classification report, roc auc curve out of all the models.

**CONCLUSION**

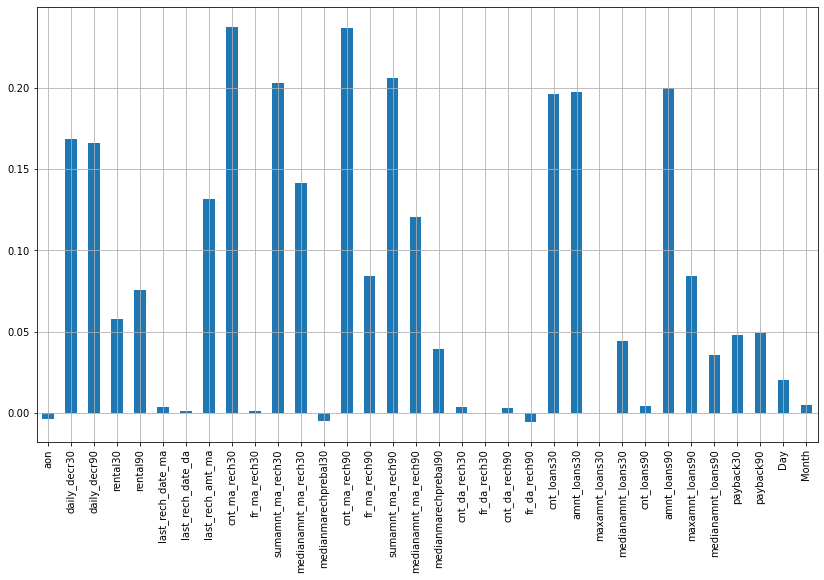
* Key Findings and Conclusions of the Study

I observed all the encoded dataset information by plotting various graphs and visualised further insights.

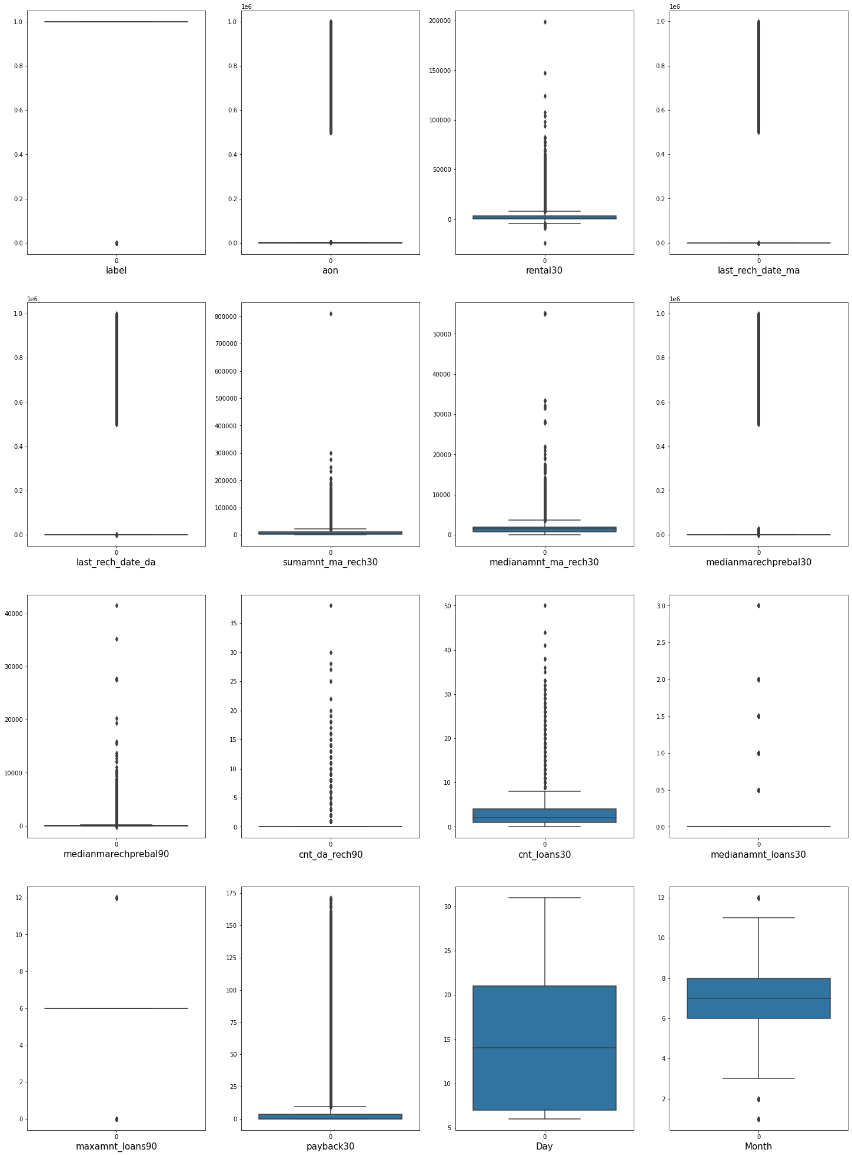
**Heat map:**



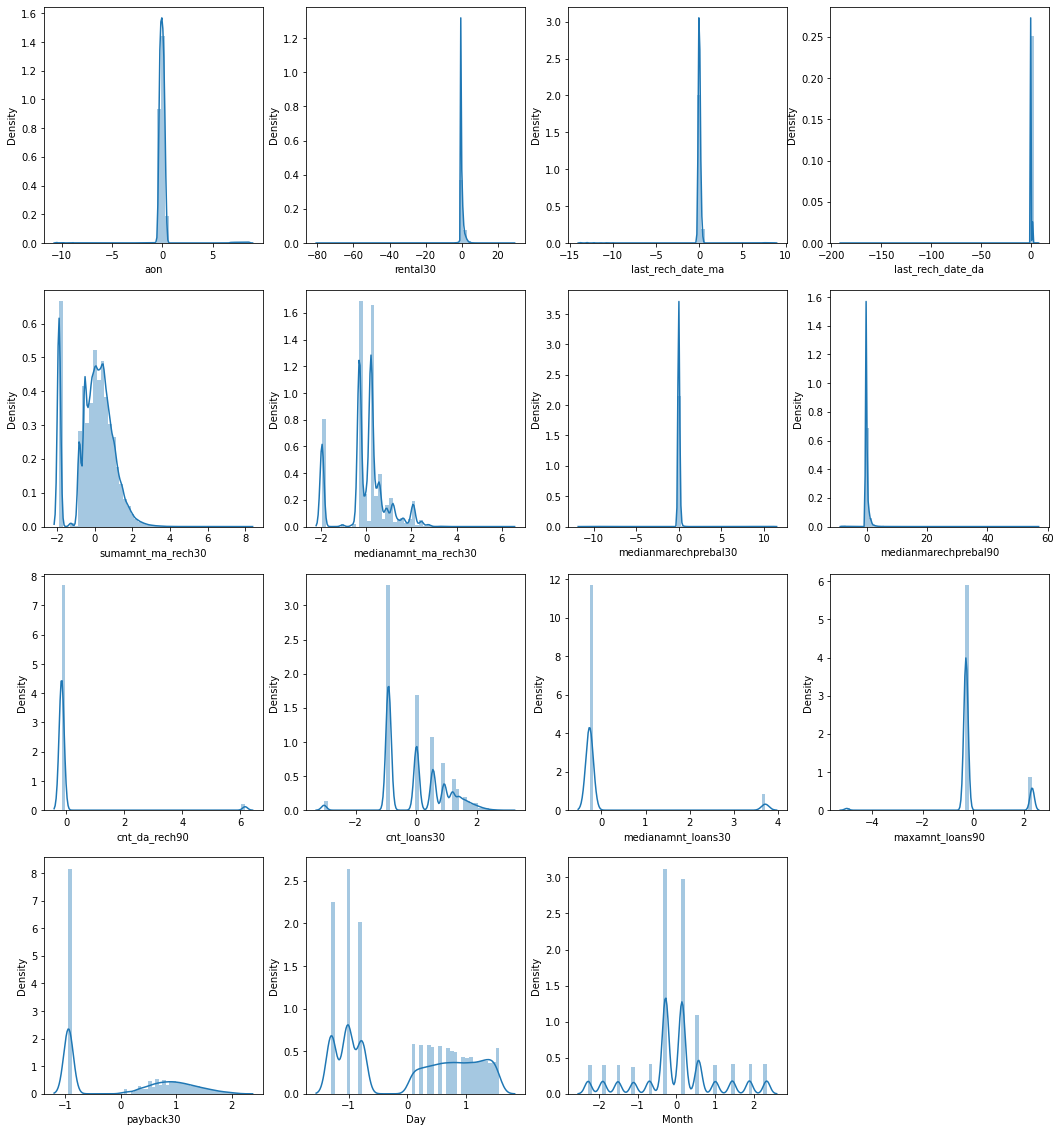
**Correlation:**



**Box plot:**



**Distribution plot:**



Post model building and choosing the appropriate model I went ahead and loaded the testing dataset. After applying all the data pre processing steps as the training dataset I was then able to get the predicted sale price results. Since the values were in array format, I converted them into a dataframe and merged it with the original testing dataframe that consisted only our feature columns. Once the testing dataset with feature columns and predicted label was formed, I exported the values in a comma separated values file to be accessed as needed.

* Learning Outcomes of the Study in respect of Data Science

Data Exploration and Cleaning On data exploration, I found that the dataset was imbalanced for the target feature(87.5% for Non-defaulters and 12.5% for Defaulters). Also, I found that the data had some very unrealitic values such as 999860 days which is not possible. Also, there were negative values for variables which must not have one (example:frequency,amount of recharge etc

Data Visualization On visualizing data, there were two important insights I gathered. a. Imbalance of data b. Distribution was not normal

