

Predicting Micro Credit Defaulter

**A Project Report by:**

**Archit Khanduja**

# Acknowledgement

The project would not have been built without the constant support from **DataTrained** and **Fliprobo** teams.

Following are the research papers, discussions and articles that helped me in completing the project:

<https://towardsdatascience.com/anomaly-detection-for-dummies-15f148e559c1>

<https://github.com/scikit-learn-contrib/imbalanced-learn/issues/534>

<https://www.researchgate.net/publication/220543125_SMOTE_Synthetic_Minority_Over-sampling_Technique>

<https://www.researchgate.net/post/how_does_smote_oversampling_technique_change_the_nature_of_the_data_that_affect_the_performance_of_the_classifiers>

<https://scikit-learn.org/stable/auto_examples/classification/plot_classification_probability.html>

# Introduction

## Business problem Framing

In this project, the main problem revolves around a Microfinance Institution (MFI) that offers financial services to low income populations. The concept of micro financing is being adopted here by a telecom company via mobile financing services, which many experts feel are more convenient and efficient, and cost saving, than the traditional high-touch model.

In this case, the telecom service provider will give a small sum to the customer as a loan which the customer has to return within 5 days with interest. The company needs to decide whether they should give loan to a particular customer based upon a few attributes of the customer and some historical data of about 2 lakh customers. The company needs to identify potential defaulters.

# Conceptual Background of the Domain Problem

Whenever a new customer comes for a loan to a financing institute, the customer’s details like income, age, number of dependents, education background and credit score are checked and based upon that, the customer is given a loan.

In our case, the customers are low income families asking for very minimal loans for mobile phone balance. They are bound to return the loan with interest within 5 days or they will be considered to be defaulters. In order to classify a customer as a potential defaulter, the MFI is going to use an advanced Machine Learning model which uses the historical data of almost 2 lakh customers and attributes of the new customer to predict whether the new customer will be a defaulter or not.

# Review of Literature

While researching more about the problem, it was figured out that in order to build a model to classify a new customer in a category of a defaulter or a non-defaulter, it would be more important to know what the data is telling us by visualizing it and knowing the correlations between various features. Also, it would be important to know which features will play the most important role in identification of a defaulter. While going through a few more credit risk defaulter projects, it was found out that Random Forest is the most popular method to identify such defaulters. Although, model building and gridsearch in large datasets is very time taking, the results with Random Forest are promising.

# Motivation for the Problem Undertaken

The Micro Finance Institutions in low income countries have a lot of opportunities to earn via interests. But these low income countries can be very risky markets because the probability of defaulters can be higher than middle income countries.

So, they indeed need an **accurate** solution to this problem of identifying whether a new customer will be a defaulter or not.

# Analytical Problem Framing

## Mathematical/ Analytical Modelling of the Problem

In this problem, we need to predict whether a new customer will be a defaulter or not. Clearly, it is a binary classification problem in which the target variables contains two types of entries, namely defaulter and non-defaulter. Defaulter will be denoted by 0 and non-defaulter will be denoted by 1.

Majority of the features in the dataset are continuous whereas a few of them are also binary or ordinal.

### Presence of Anomalies in the Data

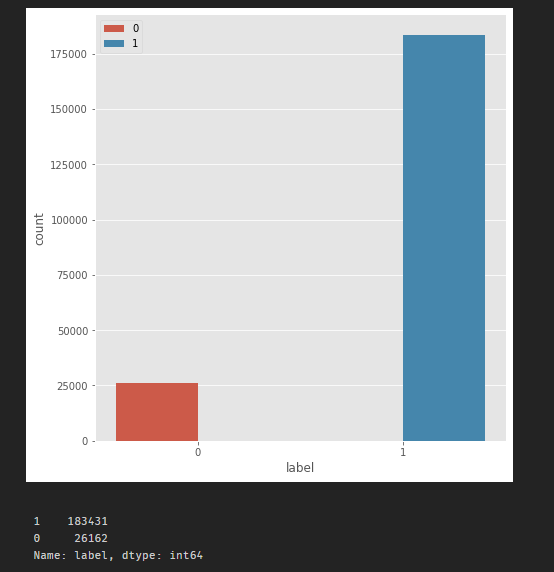
There are a few anomalies also present in the data; like, a few features are having negative values, which is not possible in these cases. It means that the data has been wrongly recorded in some places.

Also, there are some features whose maximum values are too large to be true. These incorrectly recorded values can gravitate our model towards them and impact it negatively.

### Data Redundancy and Correlation

In our dataset, we have the user data of 30 days and 90 days for multiple attributes. Firstly, former will be the subset of latter, i.e. 90 days data will have more information. Secondly, upon checking the correlation, these fields were found to be highly correlated with each other. Hence, the fields with 30 days data were dropped before training the model.

### Lack of Balance in the Target Variable

The target variable is not balanced as the number of occurrences of 0 is too much less than the number of occurrences of 1 as shown in the adjacent figure.

## Data Sources and their Formats

|  |  |
| --- | --- |
| Variable | Definition |
| **label** | Flag indicating whether the user paid back the credit amount within 5 days of  issuing the loan{1:success, 0:failure} |
| **msisdn** | mobile number of user |
| **aon** | age on cellular network in days |
| **daily\_decr30** | Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah) |
| **daily\_decr90** | Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah) |
| **rental30** | Average main account balance over last 30 days |
| **rental90** | Average main account balance over last 90 days |
| **last\_rech\_date\_ma** | Number of days till last recharge of main account |
| **last\_rech\_date\_da** | Number of days till last recharge of data account |
| **last\_rech\_amt\_ma** | Amount of last recharge of main account (in Indonesian Rupiah) |
| **cnt\_ma\_rech30** | Number of times main account got recharged in last 30 days |
| **fr\_ma\_rech30** | Frequency of main account recharged in last 30 days |
| **sumamnt\_ma\_rech30** | Total amount of recharge in main account over last 30 days (in Indonesian Rupiah) |
| **medianamnt\_ma\_rech30** | Median of amount of recharges done in main account over last 30 days at user  level (in Indonesian Rupiah) |
| **medianmarechprebal30** | Median of main account balance just before recharge in last 30 days at user  level (in Indonesian Rupiah) |
| **cnt\_ma\_rech90** | Number of times main account got recharged in last 90 days |
| **fr\_ma\_rech90** | Frequency of main account recharged in last 90 days |
| **sumamnt\_ma\_rech90** | Total amount of recharge in main account over last 90 days (in Indonasian Rupiah) |
| **medianamnt\_ma\_rech90** | Median of amount of recharges done in main account over last 90 days at user  level (in Indonasian Rupiah) |
| **medianmarechprebal90** | Median of main account balance just before recharge in last 90 days at user  level (in Indonasian Rupiah) |
| **cnt\_da\_rech30** | Number of times data account got recharged in last 30 days |
| **fr\_da\_rech30** | Frequency of data account recharged in last 30 days |
| **cnt\_da\_rech90** | Number of times data account got recharged in last 90 days |
| **fr\_da\_rech90** | Frequency of data account recharged in last 90 days |
| **cnt\_loans30** | Number of loans taken by user in last 30 days |
| **amnt\_loans30** | Total amount of loans taken by user in last 30 days |
| **maxamnt\_loans30** | maximum amount of loan taken by the user in last 30 days |
| **medianamnt\_loans30** | Median of amounts of loan taken by the user in last 30 days |
| **cnt\_loans90** | Number of loans taken by user in last 90 days |
| **amnt\_loans90** | Total amount of loans taken by user in last 90 days |
| **maxamnt\_loans90** | maximum amount of loan taken by the user in last 90 days |
| **medianamnt\_loans90** | Median of amounts of loan taken by the user in last 90 days |
| **payback30** | Average payback time in days over last 30 days |
| **payback90** | Average payback time in days over last 90 days |
| **pcircle** | telecom circle |
| **pdate** | date |

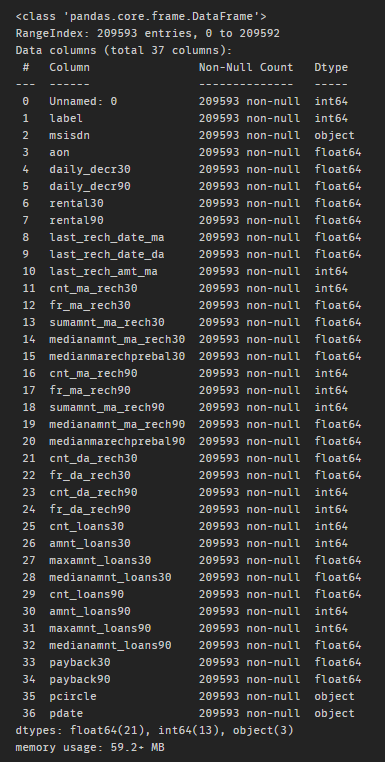


Table 1: Attributes and their data types

## Data Preprocessing Done

* A few columns were found irrelevant after going through the data dictionary. Following are those columns along with the reason they were dropped:

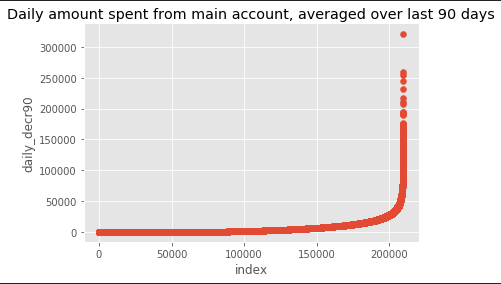
Columns to be dropped after going through data dictionary:

1. unnamed: Serial Number(unique)
2. msisdn: Mobile number(unique)
3. pcircle: All entries are the same
4. pdate: Date

* After checking the correlation, it was observed that features with 30 and 90 days of similar attributes were highly correlated. All the features with 30 days were dropped as the data contained in them would be lesser than the data contained in 90 days.
* The entire dataset is full of outliers. If we build a model without treating them, the entire model will gravitate towards them and the accuracy score will be impacted. Given the cost of our data, we could not really drop the entire rows filled with outliers because that would have resulted in a data loss. The best way could be to impute the outliers with median values of the columns. As we had a lot of samples, there will be multiple occurrences of various values and hence, median will be our preferred method of imputation. Even if we impute the data with median, the data will not be normally distributed. We need to apply cube root to the data as well to achieve a normal distribution.
* Since the dataset was imbalanced, SMOTETomek was used to balance it.
* Finally, the data was split into X and Y wherein Y is the target variable and X are the feature variables.

## Data Inputs- Logic- Output Relationships

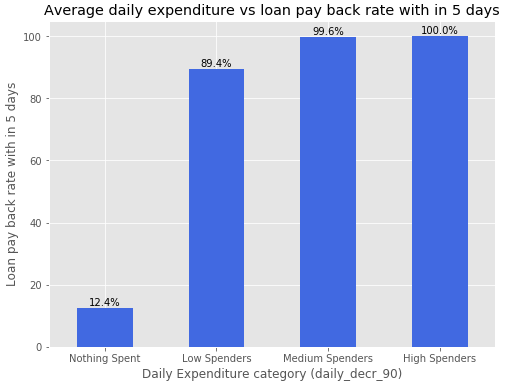
Here, we will see the univariate and bivariate analysis done during our study of the data



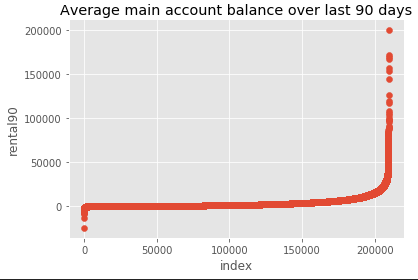
The above graph depicts ‘Daily amount spent from main account, averaged over last 90 days’. As we can see, there are a few values rising above 250000, which is quite unrealistic for a single person to spend in a day.

Now, let's divide our dataset into different brackets of spenders based upon how much they have spent. We will have 4 categories here:

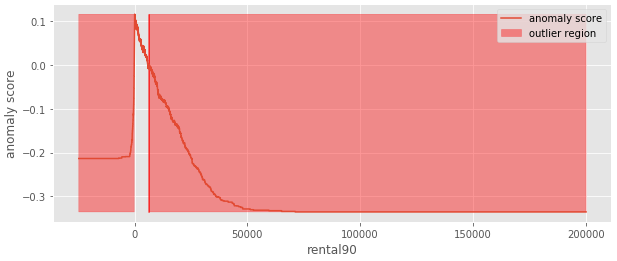
1. Spent 0 or less: Nothing spent
2. Spent between 0 and 26592(1% of max): Low Spenders
3. Spent between 26592 and 159555(60% of max): Medium Spenders
4. Spent more than 159555: High Spenders



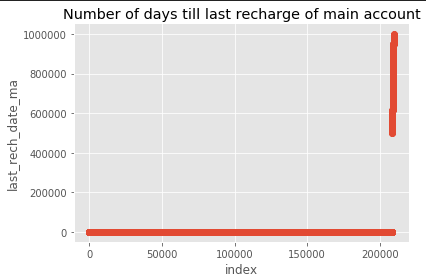
The above plot shows that if a customer spend more daily, it’s likely that she will payback the amount on time.



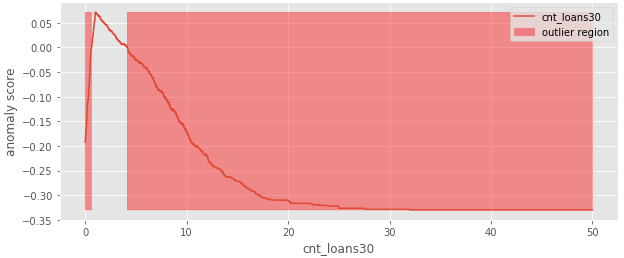
When we had a look at average main account balance over the last 90 days, it was found that a few values are going below 0 and some are going above 150000. The extremes are touching the unrealistic values.



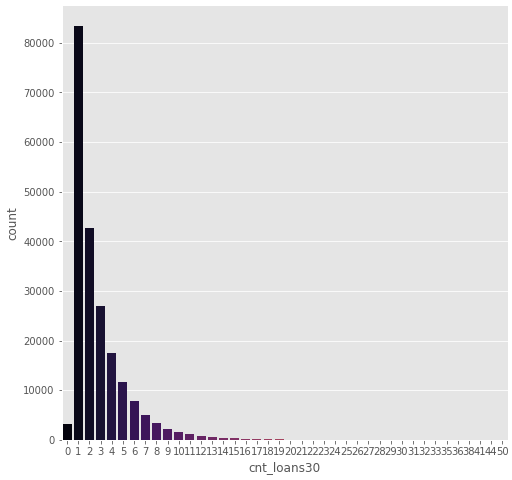
Since there were extreme values present in rental90, we were interested in knowing how much region is surrounded by the outliers. From the above plot, it is evident that majority of our data lies in a very small region and rest of it lies in the region of outliers. This can impact our model negatively and the model can gravitate towards outliers which will impact our accuracy score.



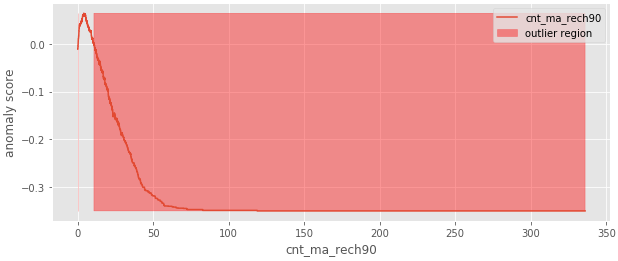
From the above graph, we can see that there are a few anomalies in the dataset. These are the number of days since last recharge. Some of the days are even more than 400000, which means more than 1000 years, when the mobile phones were not even invented. Hence a lot of data is incorrect here and these values will be removed while treating the outliers.



In the above graph, we can see the distribution of number of loans taken by the user in last 30 days. A significant amount of our data belongs to the outlier region. Most of the users have taken only 1 or 2 loans in the last 30 days and rest of the occurrences are fairly less, which makes their probability of occurrence significantly low.



Above graph reiterates the number of loans taken by users in last 30 days in a different kind of plot.



The above graph shows the distribution of ‘Number of times the main account got recharged in last 90 days’. As we can see, majority of the values belong to the area which depicts that number of recharges in the last 90 days which are more than ~8 will be treated as outliers.

## Assumptions

1. Attributes are independent of each other (low or no multicollinearity).
2. Objects in the target variable are identically distributed.
3. A linear classifier will assume that the decision boundaries are linear.
4. In case of logistic regression, we assume that there is a linear relationship between the logit of the outcome and each predictor variables.

## Hardware and Software Requirements and Tools Used

For the building of this model, an MSI Machine with Intel CORE i7 7th Gen processor and an 8 GB RAM was used.

The programming language used was Python. The compiler that was used was Anaconda Navigator. The programs were run in Jupyter Notebook environments.

The libraries used were as follows: numpy, pandas, matplotlib, seaborn, sklearn, scipy, imblearn.

# Model/s Development and Evaluation

## Identification of possible problem-solving approaches

Upon doing some research on credit default models, it was found that tree based models tend to outperform the linear models as they map non-linear relationships quite well. They work good for both categorical and continuous features. Even though we know that tree based models work best, we have tested multiple models and have selected one final model with the best accuracy score.

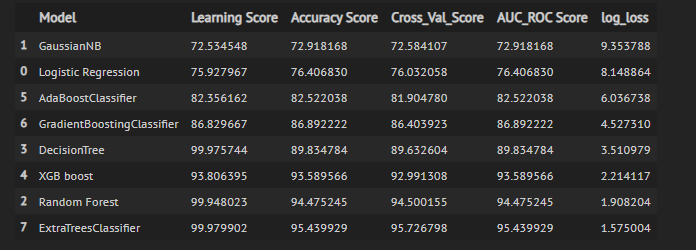
Following are the techniques that were used and compared for machine learning:

1. Logistic Regression
2. Gaussian NB
3. Random Forest
4. Decision Tree
5. XG Boost
6. Ada Boost
7. Gradient Boosting
8. Extra Trees

## Testing of Identified Approaches

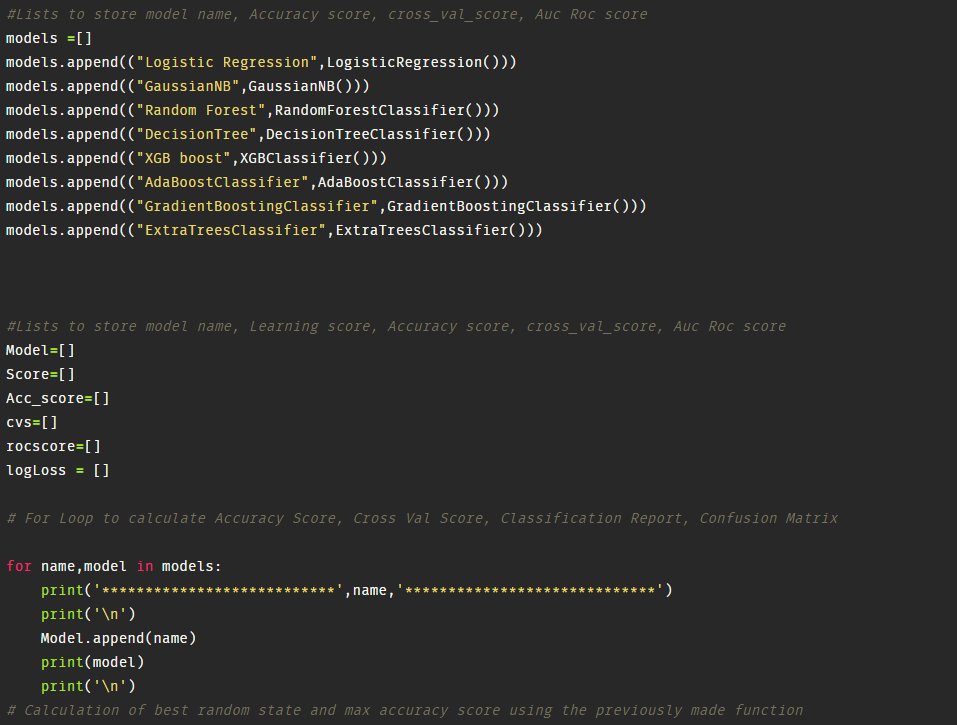
After the models were trained, they were cross validated and their cross validation score was checked and the corresponding accuracies were stored in a dataframe.

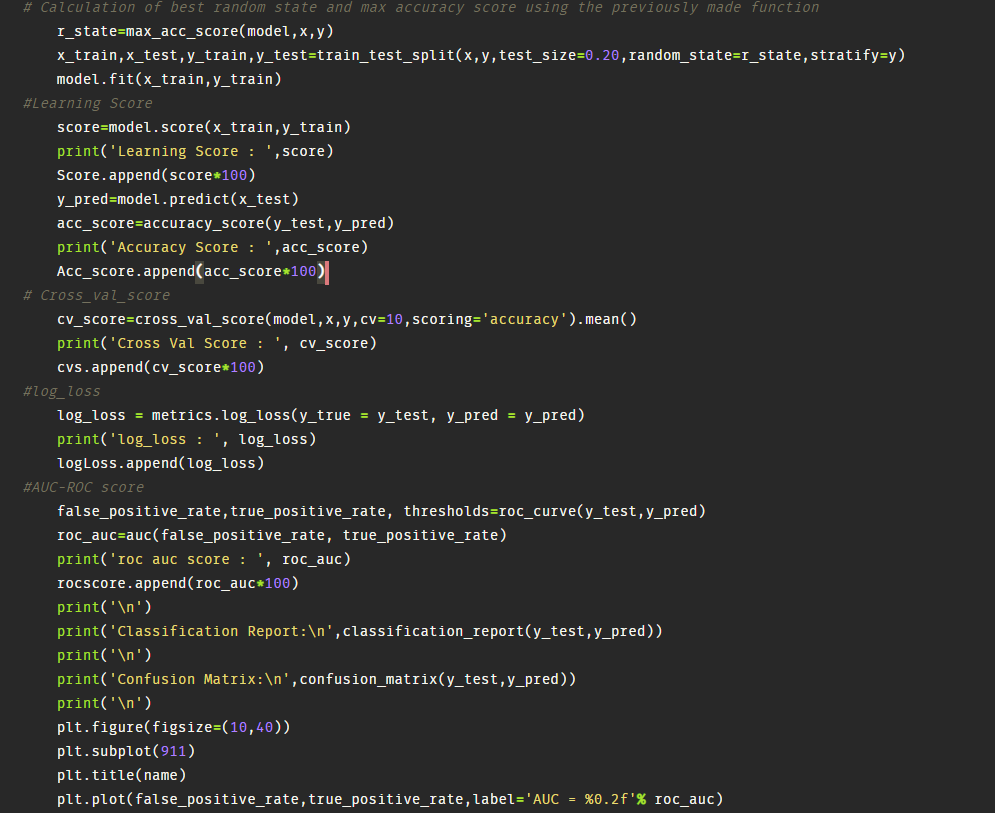
Following is the resultant dataframe in an increasing order of accuracy scores.

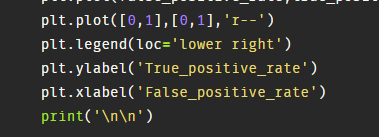


## Run and Evaluate selected models

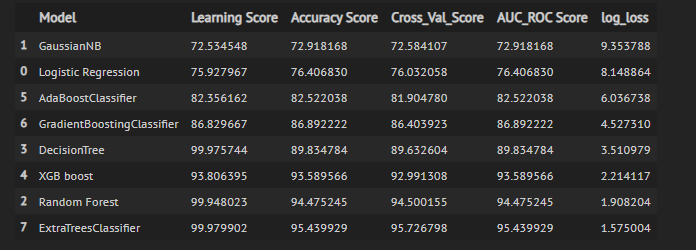
We have used the aforementioned algorithms by first storing all the models in a list. Then lists were defined to store model name, Learning score, Accuracy score, cross\_val\_score and Auc Roc score . Then in the same cell, a for loop was made to calculate Accuracy Score, Cross Val Score, Classification Report and Confusion Matrix. Also, the best random state was calculated in the same loop.



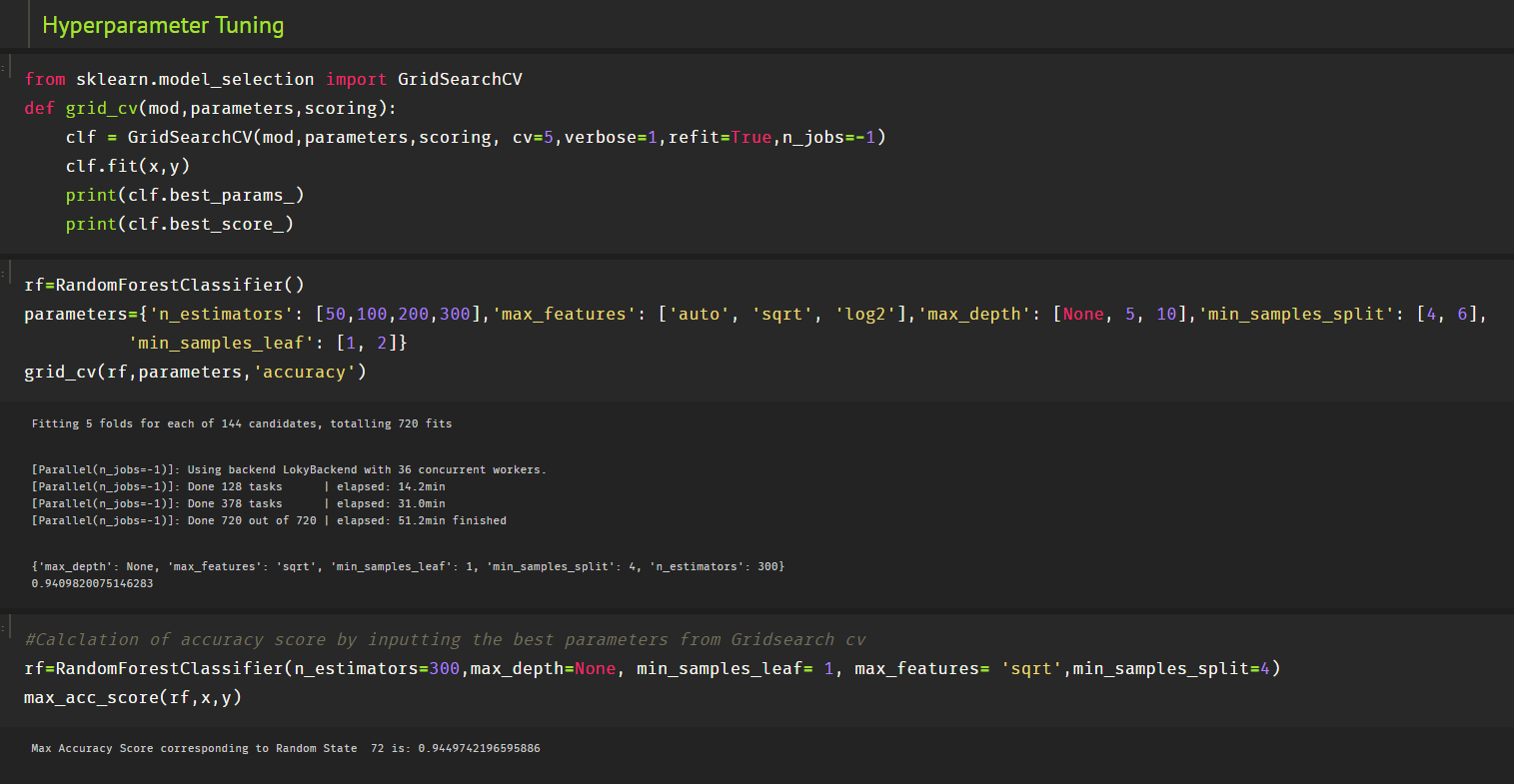


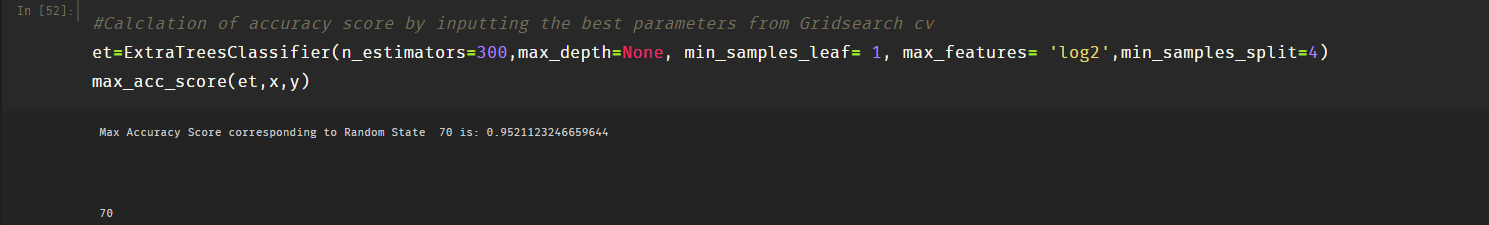


The following dataframe summarizes the result in a tabular form:



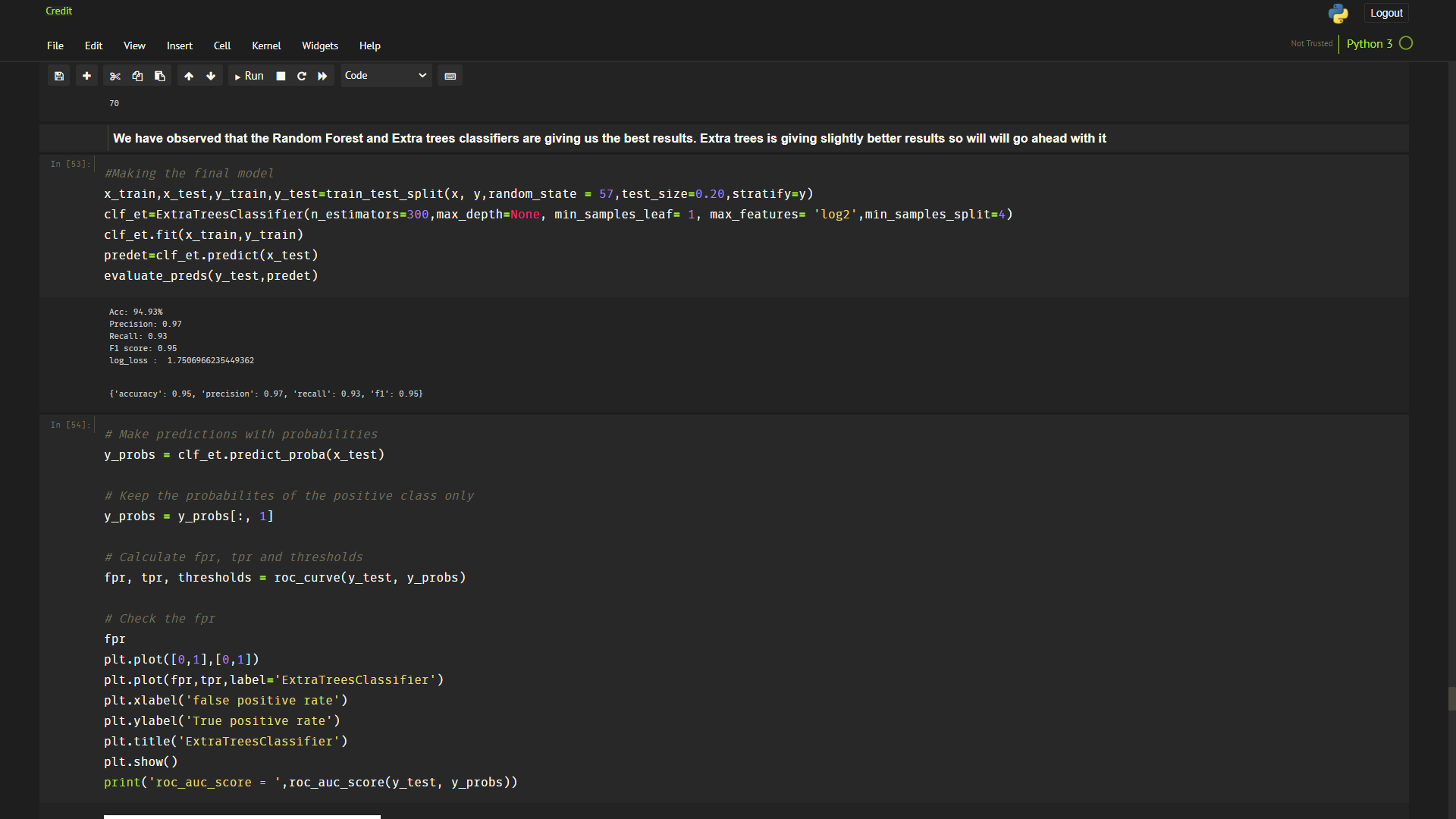
It was observed that Random Forest and Extra Trees performed the best. Hence, we went ahead and used gridsearch cv to tune the hyperparameters for the same. After finding the best parameters,



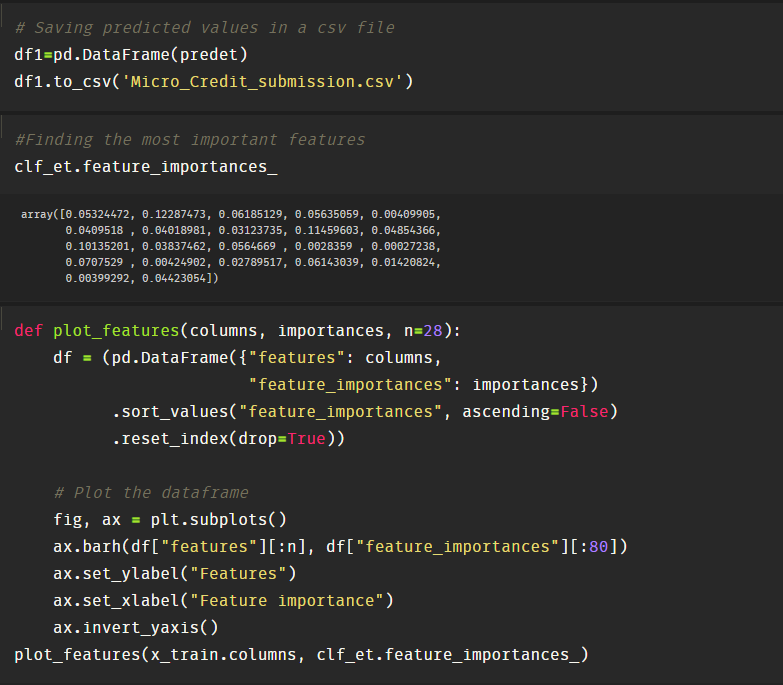


After tuning the hyperparameters, the final model was made using Extra Trees Classifier.

Also, predictions were made with probabilities and only the probabilities of the positive class were kept.



After calculating the probabilities, most important features were found out and finally, the model was saved as a pickle.





# Key Metrics for success in solving problem under consideration

The key metrics used in solving the problem were applied in the same code of model building. Those metrics were Accuracy Score, Cross validation Score, AUC ROC Score, log\_loss and learning score.

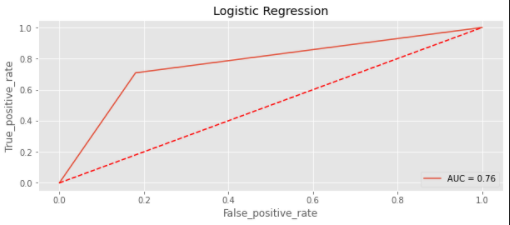
Cross validation score was used to create a more generic model so that it performs well under different circumstances and in various permutations and combinations of data.

Also, as we can see, log loss score is also inversely proportional to accuracy score and it is closer to zero in case of RF and Extra Trees. It has helped us in strengthening our conclusion of cross val scores.

In addition to this, AUC ROC score is one of the key metric for evaluation as it tells us how capable the model is in distinguishing between the positive and negative classes. It means that it observes the True Positive Rate and False Positive Rate for users who paid the loan and are falsely marked as defaulters.

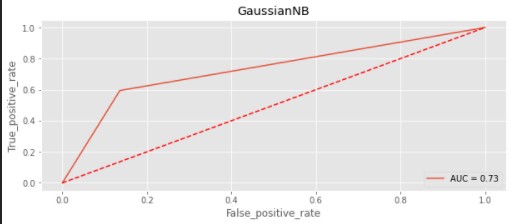
## Visualizations and Interpretations

1. AUC ROC curve
2. Logistic Regression



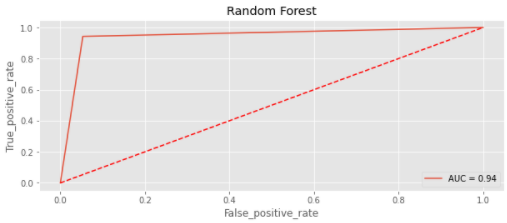
The area under the curve is 0.76, which means that 76% of the predictions by the model are correct. Closer the line is towards the Y axis, more is the area under the curve.

1. Gaussian NB



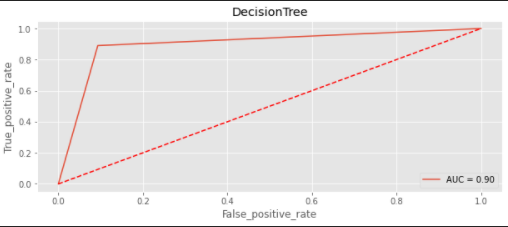
The area under the curve is 0.73, which means that 73% of the predictions by the model are correct.

1. Random Forest



The area under the curve is 0.94, which means that 94% of the predictions by the model are correct.

1. Decision Tree



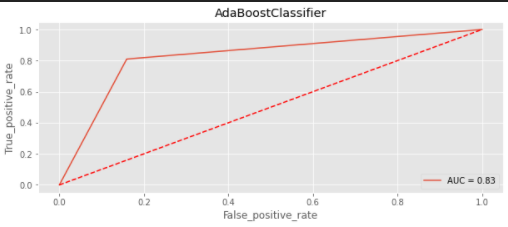
The area under the curve is 0.90, which means that 90% of the predictions by the model are correct.

1. XG Boost



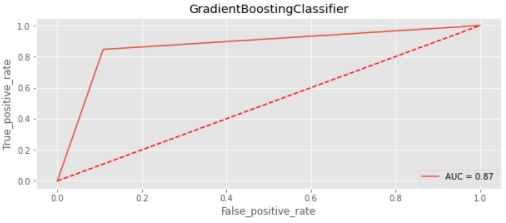
The area under the curve is 0.94, which means that 94% of the predictions by the model are correct.

1. ADA Boost



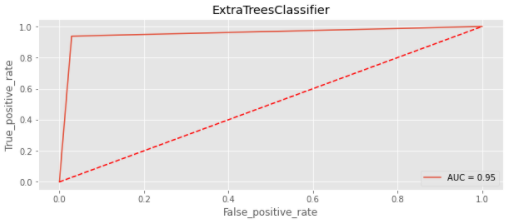
The area under the curve is 0.83, which means that 83% of the predictions by the model are correct.

1. Gradient Boosting



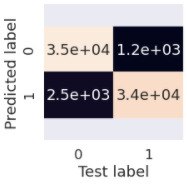
The area under the curve is 0.87, which means that 87% of the predictions by the model are correct.

1. Extra Trees Classifier



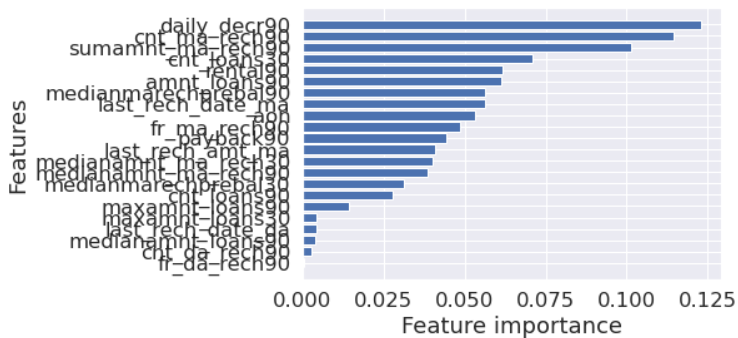
The area under the curve is 0.95, which means that 95% of the predictions by the model are correct.

1. Confusion Matrix



From the above confusion matrix, we can interpret that the majority of the values are in the positive diagonal which indicates a high accuracy score(95%) for our model.

1. Most Important Features



Here, we can see that daily\_decr90, cnt\_ma\_rech90 and sumamnt\_ma\_rech90 are the most important features. Changing them will impact the result drastically.

# Conclusion

## Key Findings and Conclusions of the Study

After doing the whole study, following were the key findings:

1. If the daily amount spent from the main account in the last 90 days is 0, then the probability of person being defaulter will be (100-12.4)= 87.6%, which is very high. The company should not give loans to those who did not spend anything in the last 90 days.
2. Higher the expenditure of the person, lower is the person’s probability of being a defaulter.
3. More the number of loans taken in the past, less is the probability of a person to be a defaulter.
4. Tree based algorithms heavily outperformed the Euclidian distance based algorithms and linear algorithms in this case as expected.
5. Even if we do not drop the entire row of outliers and impute them with mean or median, we can achieve a high accuracy score without losing much data.
6. daily\_decr90, cnt\_ma\_rech90 and sumamnt\_ma\_rech90 are the most important features in the dataset and the company should focus their marketing strategies around these features. Company can use SMS alerts whenever the balance is low to boost the amount of recharges done in 90 days in order to boost the loans and earn more interest from them.

## Learning Outcomes of the Study in respect of Data Science

There were various learning outcomes out of this project.

1. Anomaly Detection

In a large dataset, when we are interested in visualising the region of outliers, we can use the Isolation Forest algorithm to return the anomaly score of each sample and plot the distribution on a graph.

1. Outlier Imputation

Whenever we are not ready to lose the data but our dataset is filled with outliers, we can impute the outliers with mean or median and the using the cube roots of the values to normalize the data, hereby preserving the precious data without jeopardising the accuracy score of the model.

1. Handling an Imbalanced Dataset

In our dataset, we had an imbalance in the classes of the target variable, which would have resulted in a low cross validation score. We used SMOTETomek to balance the dataset.

## Limitations of this work and Scope for Future Work

* While scaling the dataset, there were a lot of null values formed in a few features, due to which scaling was not done. Scaling is important because the model considers features with high values to be more important than features with low values. Although, there is another point to note that feature scaling is not necessary for tree based models and hence, our tree based models performed the best.
* The dataset was very bulky, which led to a lot of time in building the models. There were more techniques that could have been proven even better and show better results had time not been a constraint.