**Credit Card Default Prediction**

**Kota Lakshmana Rao,**

**Data science trainee,**

**AlmaBetter , Bangalore**

**Abstract:**

**In today’s world credit cards have become a lifeline to a lot of people so banks provide us with credit cards..**

**Our experiment can help understand what could be the reason for the classification of such labels by feature selection, data analysis and prediction with machine learning algorithms taking into account previous trends to determine the correct classification**. **For finding defaulters**

***Keywords: machine learning ,Credit card default prediction, classified labels***

**1.Problem Statement**

### **This project is aimed at predicting the case of customers' default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the**[**K-S chart**](https://www.listendata.com/2019/07/KS-Statistics-Python.html)**to evaluate which customers will default on their credit card payments**

**A Credit Card is a type of payment card in which charges are made against a line of credit instead of the account holder’s cash deposit. When someone uses a credit card to make purchase, that person’s account accrues a balance that must be paid off each month**

**The main objective is to build a predictive model, which could help them in predicting the surge pricing type proactively. This would in turn help them in matching the right cabs with the right customers quickly and efficiently**.

## **2.Data Description**

### Attribute Information:

### **This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:**

### **X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.**

### **X2: Gender (1 = male; 2 = female).**

### **X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).**

### **X4: Marital status (1 = married; 2 = single; 3 = others).**

### **X5: Age (year).**

### **X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.**

### **X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.**

### **X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.**

**3. Introduction**

### **The credit platform adjusts their prices using a specific algorithm which is real time and dynamic. Now we know the most common issue there is in providing these kinds of deals are people not being able to pay the bills. These people are what we call “defaulters”. The algorithms automatically finds defaulters of credit card**

### **The machine learning algorithms generally outputs a multiplier which is adjusted along with the credit limit, payment history and educational bases ,age factor, marital status generate the defaulters prediction**

### **Our goal here is to build a predictive model, which could help defaulters predictions**

## **4. Reasons for defaulters**

**The reasons for Defaulters are:**

* **Intentionally they don’t want to pay**
* **They don’t have sufficient money to repay**

**5. Most Defaulters Age**

**1. most of the 27th age people defaulters**

**2.least defaulters were above 60age people were defaulters**

**6. Steps involved:**

* **Exploratory Data Analysis**

**After loading the dataset we performed this method by comparing** **our target** **variable that is Defaulters with Other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable**

* **Null values Treatment**
* **Our dataset contains NO null values which may tend to our accuracy at the beginning of our project in order to get a better result**.
* **Encoding of categorical columns**

**We used One Hot Encoding to produce binary integers of 0 and 1 to encode our categorical features are SEX, MARRIAGE, EDUCATION, because categorical features that are in string format cannot be understood by the machine and**

* **Feature Selection**

**In these steps we using information gaining method finding correlation coefficients between variables mostly effects that are removed from our data set i.e**  **"PAY\_SEPT","BILL\_AMT\_SEPT","PAY\_AMT\_SEPT" removed from the dataset**

* **APPLYING SMOTE**

**The given data set unbalanced data set for this we used SMOTE i.e** Synthetic Minority Oversampling Technique **resampling of to performing best results for randomly before after applying SMOTE**

**Original dataset shape Counter({0: 18691, 1: 5309})**

**Resample dataset shape Counter({1: 23364, 0: 23364})**

**AND fitting and applying different algorithms to it**

**The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.**

* **Fitting different models**

**For modeling we tried various classification algorithms like:**

1. **Logistic Regression**
2. **KNN Classifier**
3. **Random Forest Classifier**
4. **XGBoost classifier**

* **Tuning the hyperparameters for better accuracy**

**Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models**

**like Random Forest Classifier and XGBoost classifier**.

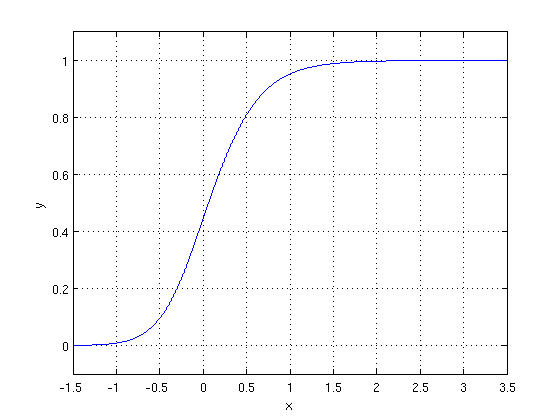
**7.1. Algorithms:**

1. **Logistic Regression:**

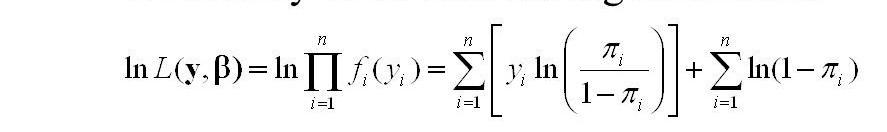
**Logistic Regression is actually a classification algorithm that was given the name regression due to the fact that the mathematical formulation is very similar to linear regression.**

**The function used in Logistic Regression is sigmoid function or the logistic function given by:**

**f(x)= 1/1+e ^(-x)**



**The optimization algorithm used is: Maximum Log Likelihood. We mostly take log likelihood in Logistic:**



**We have implemented logistic regression with Grid search cv. We get an accuracy score of approximately 62%. and precision score approximately is 62% and f1\_score is 62%and roc auc approximately is 62% As we have an imbalanced dataset, recall score is approximately 63% better parameter.**

.

1. **KNN Classifier:**

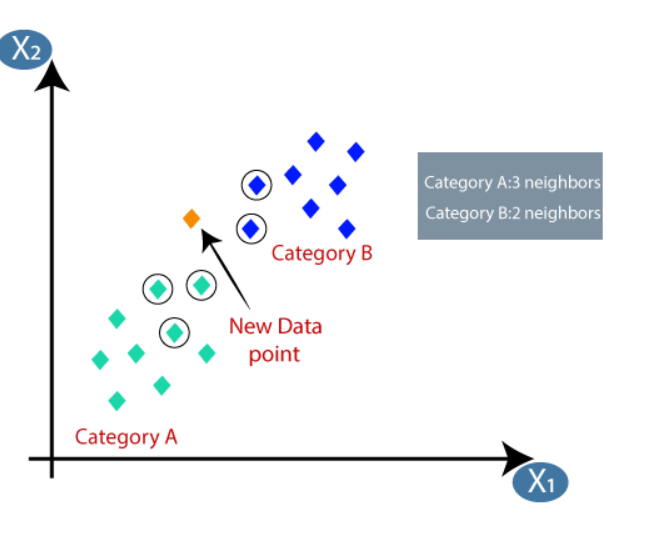
* **K-Nearest Neighbors is one of the simplest supervised learning algorithms.**

### **The KNN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.**

### **KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using KNN algorithm.**

### **KNN algorithms can be used for regression as well as for classification problems.**

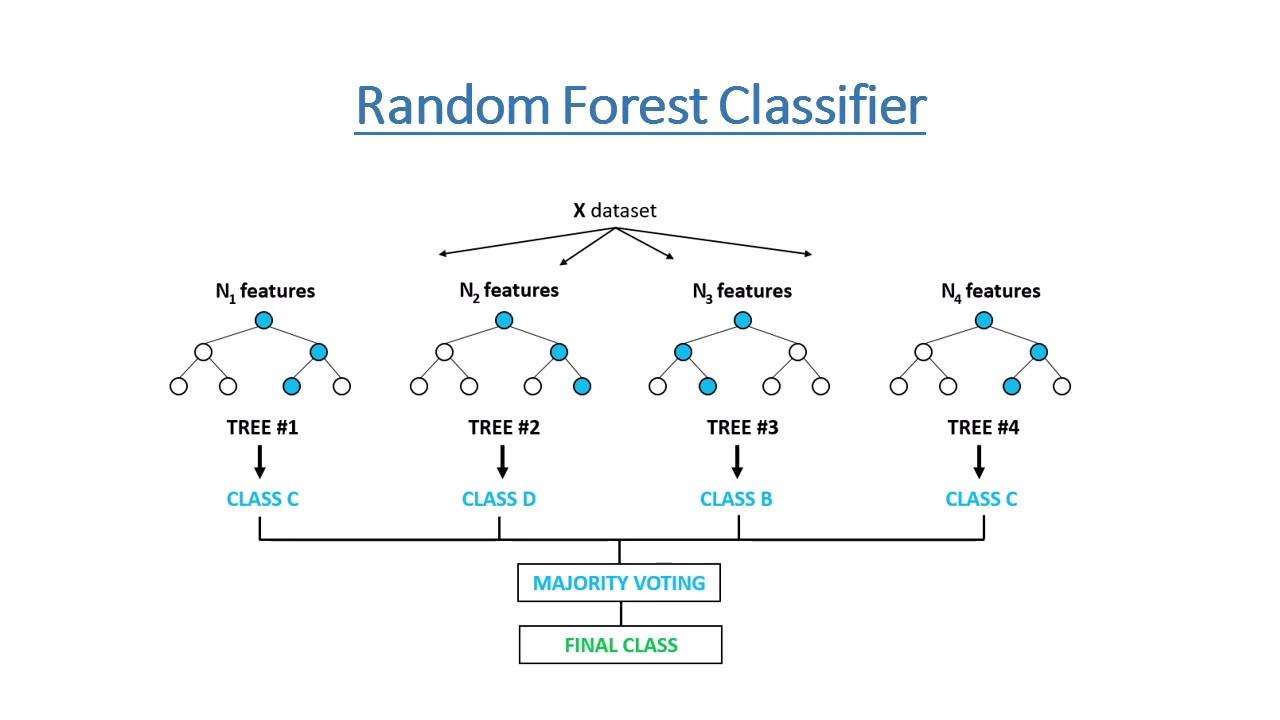
### **As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A**



**KNN and our accuracy score is approximately 75%. and precision score is approximately 70% and f1\_score is 76% and ROC\_AUC score is 75% ,recall score is approximately 88% better parameter**

1. **Random Forest Classifier:**

**Random Forest is a bagging type of Decision Tree Algorithm that creates a number of decision trees from a randomly selected subset of the** **training set,** **collects the labels from these subsets and then averages the final prediction depending on the most number of times a label has been predicted out of all.**



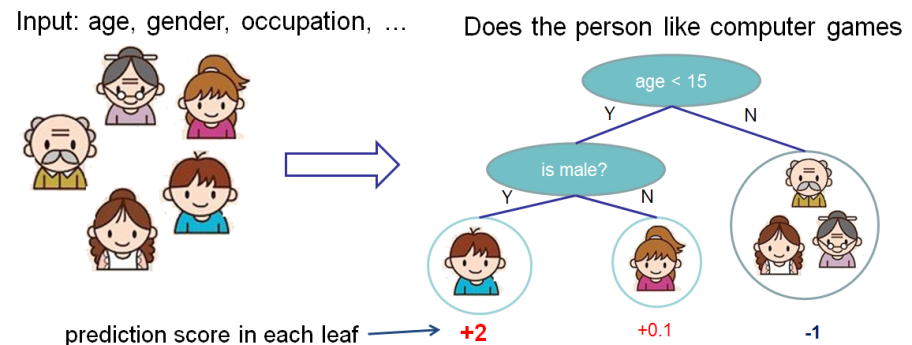
**Random Forest and our accuracy score is approximately 86%. and recall score is approximately 82% and f1\_score is 85% and ROC\_AUC score is 86% ,precision score is approximately90% better parameter. Let's go ahead with other models and see if they can give better result.**

1. **XGBoost-Classifier**

**To understand XG Boost we have to know gradient boosting beforehand.**

* **Gradient Boosting-**

**Gradient boosted trees consider the special case where the simple model is a decision tree**

****

**In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above** example, **T=3 and w=[2, 0.1, -1]).**

**When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A ‘greedy’ way to do this is to consider every possible split on the remaining features (gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.**

**XG Boost is one of the fastest implementations of gradient boosting. trees. It does this by tackling one of the major inefficiencies of gradient boosted trees: considering the potential loss for all possible splits to create a new branch (especially if you consider the case where there are thousands of features, and therefore thousands of possible splits). XG Boost tackles this inefficiency by looking at the distribution of features across all data points in a leaf and using this information to reduce the search space of possible feature splits.**

**XGBOOST WITH GRID SEARCH CV and we getting accuracy score is approximately 84%. and recall score is approximately 78% and f1\_score is 83% and ROC\_AUC score is 84% ,precision score is approximately 89% better parameter**

**7.2. Model performance:**

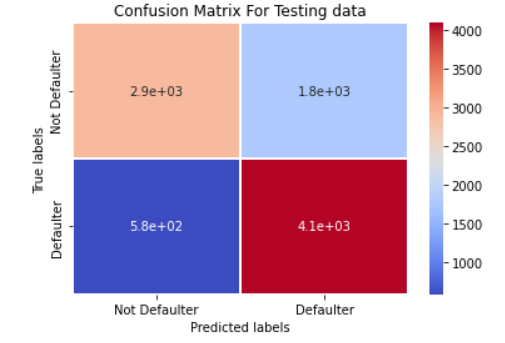
**Model can be evaluated by various metrics such as:**

1. **Confusion Matrix**-

**The confusion matrix is a table that summarizes how successful the classification model is at predicting examples belonging to various classes. One axis of the confusion matrix is the label that the model predicted, and the other axis is the actual label.**

**Out of all models Random forest classifier with Grid search CV performed very well its confusion matrix of test data is**





1. **Precision/Recall**-

**Precision is the ratio of correct positive predictions to the overall number of positive predictions : TP/TP+FP**

**Recall is the ratio of correct positive predictions to the overall number of positive examples in the set: TP/FN+TP**

**Out of them Random forest classifier with Grid search CV ( tuned hyperparameters gave) the best result. it's Highest**

**Precision score is approximately 90%,**

**Recall score is approximately 82%**

1. **Accuracy**-

**Accuracy is given by the number of correctly classified examples divided by the total number**

**of classified examples. In terms of the confusion matrix, it is given by: TP+TN/TP+TN+FP+FN**

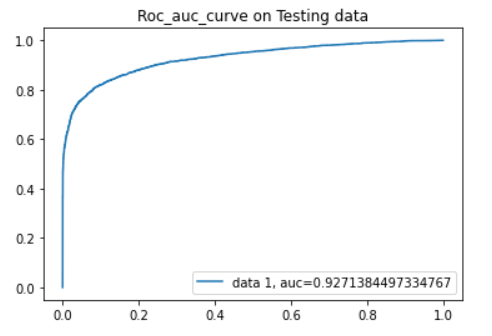
**Out of themRandom forest classifier Grid search CV ( tuned hyperparameters gave) the best result. it's Highest**

**Accuracy Score approximately 86%,**

1. **Area under ROC Curve(AUC)**-

**ROC curves use a combination of the true positive rate (the proportion of positive examples predicted correctly, defined exactly as recall) and false positive rate (the proportion of negative examples predicted incorrectly) to build up a summary picture of the classification performance.**

**Out of them Random forest classifier with Grid search CV ( tuned hyperparameters gave) the best result. it's Highest** **ROC\_AUC score is approximately 86%,**

****

**7.3. Hyper parameter tuning:**

**Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific** **hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.**

**We used Grid Search CV, for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds. The best performance improvement .**

1. **Grid Search CV-Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.**

**8. Conclusion:**

**That's it! We reached the end of our exercise.**

**Starting with loading the data so far we have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.**

**In all of these models our accuracy revolves in the range of 62 to 92%.**

**And there is no such improvement in accuracy score even after hyperparameter tuning.**

**We used different type of Classification algorithms to train our model like, Logistic Regression, Random Forest Classifier, KNN Classifier, XG boost Classifier. and Also we tuned the parameters of Random forest classifier and XG boost classifier ,KNN\_ Out of them Random forest classifier with Grid search CV ( tuned hyperparameters gave) the best result. it's Highest**

**Precision score is approximately 90%,**

**Recall score is approximately82%**

**ROC\_AUC score is approximately 86%,**

**Accuracy Score is approximately 86%,**

**and It's F1\_score approximately 85%**