**Credit Card Default Prediction**

**Nargis Nasreen**

We are all aware what is credit card. It is type of payment payment card in which charges are made against a line of credit instead of the account holder's cash deposits. When someone uses a credit card to make a purchase, that person's account accrues a balance that must be paid off each month.

Credit card default happens when you have become severely delinquent on your credit card payments.Missing credit card payments once or twice does not count as a default. A payment default occurs when you fail to pay the minimum amount due on the credit card for a few consecutive months.

The main objective of our project is to predict which customer might default in upcoming months.

We have:-

* Number of instances: 30000 and
* Number of attributes: 25

The dataset contains features like:

* **ID:** ID of each client
* **LIMIT\_BAL:** Amount of given credit in NT dollars (includes individual and family/supplementary credit)
* **SEX:** Gender (1 = male, 2 = female)
* **EDUCATION:** (1 = graduate school, 2 = university, 3 = high school, 0,4,5,6 = others)
* **MARRIAGE:** Marital status (0 = others, 1 = married, 2 = single, 3 = others)
* **AGE:** Age in years

**Scale for PAY\_0 to PAY\_6** : (-2 = No consumption, -1 = paid in full, 0 = use of revolving credit (paid minimum only), 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)

* **PAY\_0:** Repayment status in September, 2005 (scale same as above)
* **PAY\_2:** Repayment status in August, 2005 (scale same as above)
* **PAY\_3:** Repayment status in July, 2005 (scale same as above)
* **PAY\_4:** Repayment status in June, 2005 (scale same as above)
* **PAY\_5:** Repayment status in May, 2005 (scale same as above)
* **PAY\_6:** Repayment status in April, 2005 (scale same as above)
* **BILL\_AMT1:** Amount of bill statement in September, 2005 (NT dollar)
* **BILL\_AMT2:** Amount of bill statement in August, 2005 (NT dollar)
* **BILL\_AMT3:** Amount of bill statement in July, 2005 (NT dollar)
* **BILL\_AMT4:** Amount of bill statement in June, 2005 (NT dollar)
* **BILL\_AMT5:** Amount of bill statement in May, 2005 (NT dollar)
* **BILL\_AMT6:** Amount of bill statement in April, 2005 (NT dollar)
* **PAY\_AMT1:** Amount of previous payment in September, 2005 (NT dollar)
* **PAY\_AMT2:** Amount of previous payment in August, 2005 (NT dollar)
* **PAY\_AMT3:** Amount of previous payment in July, 2005 (NT dollar)
* **PAY\_AMT4**: Amount of previous payment in June, 2005 (NT dollar)
* **PAY\_AMT5:** Amount of previous payment in May, 2005 (NT dollar)
* **PAY\_AMT6:** Amount of previous payment in April, 2005 (NT dollar)

**Target Variable : default payment next month**

* **default payment next month:** 1-Default, 0-Not default

**Banks are constantly using Machine Learning algorithms so that it can take proper action beforehand. They do this by understanding the dependency of Default with other relevant features to increase that can increase the chance to find which customer can be a defaulter in the given month.**

* **Exploratory Data Analysis**

After loading the dataset I changed the name of the target variable to 'Default' then I looked for duplicate and null values in the columns. There were none. So I performed EDA by comparing our target variable that is Default with other independent variables. This process helped me figuring out various aspects and relationships among the target and the independent variables. It gave me a better idea of which feature behaves in which manner compared to the target variable.

The result I get after performing EDA is that most of the users who are young i.e, 20-30 years of age or old i.e, above 50 and less educated are more likely to default. While the educated ones are making their payments on time.

* **Null values Treatment**

The dataset contains no null values.

* **Encoding of categorical columns**

I used One hot encoding Encoding for replacing the values of categorical variables by increasing the number of categories with same values. This was done to not increase the dimensions to the data set while also keeping the relationship of variables with Default into consideration.

* **Feature engineering**

Feature engineering is the process of tansforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved accuracy on unseen data.

And I did one hot encoding here to replace a categorical value by increasing the number of categories with same values. Any non-categorical columns are automatically dropped by the one hot encoder model. This can help improve machine learning accuracy since algorithms tend to have a hard time dealing with high cardinality columns.

* **Feature Selection**

It was not difficult to do feature select in this project. I kept all the original feaures features.

**For modelling I tried various classification algorithms like:**

* **Logistic Regression**
* **Decision Tree Classifier**
* **Random Forest Classifier**

Tuning the hyperparameters of respective algorithms is necessary for less error values,regularization and to avoid overfitting in case of tree based models.

**Logistic Regression:**

Logistic regression, despite its name, is a classification algorithm rather than regression algorithm. Based on a given set of independent variables, it is used to estimate discrete value (0 or 1, yes/no, true/false). It is also called logit or MaxEnt Classifier.

Basically, it measures the relationship between the categorical dependent variable and one or more independent variables by estimating the probability of occurrence of an event using its logistics function.

sklearn.linear\_model.LogisticRegression is the module used to implement logistic regression.

**Decision Tree Classifier:**

Decision tree classifiers are supervised machine learning models. This means that they use prelabelled data in order to train an algorithm that can be used to make a prediction. Decision trees can also be used for regression problems.

Decision tree classifiers work like flowcharts. Each node of a decision tree represents a decision point that splits into two leaf nodes. Each of these nodes represents the outcome of the decision and each of the decisions can also turn into decision nodes. Eventually, the different decisions will lead to a final classification.

The diagram below demonstrates how decision trees work to make decisions. The top node is called the root node. Each of the decision points are called decision nodes. The final decision point is referred to as a leaf node.



**Random Forest Classifier:**

Every decision tree has high variance, but when I combine all of them together in parallel then the resultant variance is low as each decision tree gets perfectly trained on that particular sample data and hence the output doesn’t depend on one decision tree but multiple decision trees. In the case of a classification problem, the final output is taken by using the majority voting classifier. In the case of a regression problem, the final output is the mean of all the outputs.

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as **bagging**. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

**Model performance:**

Model can be evaluated by various metrics such as:

* **F1-Score:**

F1 score, also known as balanced F-score or F-measure

The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal. The formula for the F1 score is:

F1 = 2 \* (precision \* recall) / (precision + recall)

* **Recall Score**:

The recall is the ratio tp / (tp + fn) where tp is the number of true positives and fn the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

The best value is 1 and the worst value is 0.

I choose Recall as the deciding factor to decide which model is giving the best results.

* **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.

I used Grid Search CV, for hyperparameter tuning. This also results in cross validation and in this case I divided the dataset into different folds.

* **Conclusion:**

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

After hyper parameter tuning, I have prevented overfitting and decreased errors by regularizing and reducing learning rate.

Using a Random Forest classifier, we can predict with ~85% accuracy, whether a customer is likely to default next month.

It mean out of 100 defaulters 85 will be correctly caught by Random Forest Classifier.

Random Forest outperforms Logistic Regression and Decision Tree if measured on their F1 scores.

The strongest predictors of default are the PAY\_AMTX, the LIMIT\_BAL & the BILL\_AMT\_X, PAY\_X, AGE and SEX on the basis of models used.

Demographics: We see that being Female, More educated, Single and between 30-40 years old means a customer is more likely to make payments on time.