# Machine Learning – Project 1
# Classification

**Submitted by *Vaisakh Babu***
**Batch : *DF2103CM***

## 1. Problem Statement and Scope of Study

**Problem Statement:** We have a dataset from a credit card company. The dataset includes the financial data for the past six months and some demographic information of the account holders. The last column of the dataset indicates whether the customer has failed to make the minimum payment. The problem is to predict whether the customer will be defaulted or not for the next month using the data of past six months.

**Scope of Study:** Default prediction is one of the major challenge faced by any financial institution. Correctly identifying the customers who will be defaulted can help the financial institutions to reduce their risk. The project is intend to identify the key drivers that helps in default prediction and to perform default prediction for the given dataset. Therefore, this study has a wider scope in financial sector.

## 2. Data exploration and cleaning

The dataset contains 30,000 records in 25 columns.

| Column Name | Description |
|---|---|
| ID | Account ID |
| LIMIT_BAL | Amount of credit provided |
| SEX | Gender (1=Male,2=Female) |
| EDUCATION | Education (1=Graduate school, 2=University, 3=High school, 4=Others) |
| MARRIAGE | 1=Married, 2=Single, 3=Other |
| AGE | Age in years |
| PAY_1 to PAY_6 | Past six month payment status. (-1=Pay duly, -2=No consumption, 1 to 9 = Payment delay in month) |
| BILL_AMT1 to BILL_AMT6 | Bill statement amount for the past six moths |
| PAY_AMT1 to PAY_AMT6 | Amount of previous payments for the past six moths |
| default payment next month | 1=Defaulted, 0=Paid duly |

**Table. 1** Column Description

## 2.1 Verification of data integrity

In this step, we are verifying whether the dataset matches with the description provided.

**Issues in the dataset :**

1. ID column contains duplicate entries
2. There are observation with all the values equal to zero except the ID column
3. EDUCATION column contains extra values 0,5 and 6
4. MARRIAGE column contains extra value 0
5. PAY_1 column contains value ‘Not available’

**Solution :**

1. The duplicated entries in ID column have all the other features as zero.
2. Therefore, we can remove the duplicate entries.
3. There is ‘Other’ label in EDUCATION column. We can map the values 0,5 and 6 to ‘Other’ label which is indicated by value 4.
4. There is ‘Other’ label in MARRIAGE column. We can map the value 0 to ‘Other’ label which is indicated by value 3.
5. We can map the ‘Not available’ values to NaN which can be further processed during missing value imputation step.

# 3. Exploratory Data Analysis

## 3.0 Variable Identification

Identification of datatype of all the variables and Identifying the predictors and target are carried out in this step.

‘default payment next month’ is the target column with labels 0 and 1. ID column is a unique identifier which can be of no use for modeling. All the columns except ID and the target column are the predictor variables.

## 3.1 Univariate Analysis

In this section, variables are analyzed one at a time. Distribution of all the variables are studied using both graphical and statistical methods.

**Observations from univariate analysis:**

1. Number of rows in the data matches with number of unique entries in the ID column. This implies that there is no presence of duplicate entries
2. Target variable has labels 0 and 1. Class imbalance is there with class 0 as the majority class(~80%). Stratified sampling should be carried out while performing the cross validation techniques.
3. LIMIT_BAL column has 166 outliers which is detected using IQR method (with $Q3+(1.5*IQR)=525000$)
4. Female and Male customers has a percentage count of 60 and 40.
5. More than 80% customers have University or Graduate school education.
6. Count of married and single customers are around 50% each with single as the class with highest count.
7. AGE column has 269 outliers which is detected using IQR method (with $Q3+(1.5*IQR)=60.5$)

8. 'PAY_1', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6' columns are categorical indicating payment delay in months as integer values.
9. BILL_AMT1-6 and PAY_AMT1-6 are numerical variables which have high skewness. Transformations have to be applied before the modelling.

### 3.2 Bivariate and Multivariate Analysis

In this section, relationships between variables are studied.

**Observations :**

1. Spearman's rank correlation is used to find the correlation between qualitative variables, which are PAY_1 to 6 and the target variable. Pay_1 to 6 is positively correlated to the target variable with Pay_1 being the variable with highest correlation(0.29) and PAY_2 to 6 gradually decreasing to 0.14.
2. There is a presence of high multicollinearity between the Pay features. Since, Pay features are important variable, building model without any of them is not advisable. Linear models are highly effected by multicollinearity. Therefore, we can use tree based classification models which are immune to multicollinearity.
3. Point Biserial correlation is used to find the correlation between Numerical variables and the target variable. All the features have very less positive correlation with the target variable. PAY_AMNT_1 has the highest correlation with correlation coefficient 0.07
4. Default rate within the SEX column shows that Males have a slightly higher default rate than Females with ~3.4% difference.
5. Default rate within EDUCATION columns indicates that customers with high school education have the highest default rate and 'Others' category has the lowest default rate.
6. There is no linear trend in categories of SEX,EDUCATION or MARRIAGE. Therefore, linear models won't be able to make use of these categorical variables effectively for prediction. For linear models we can one hot encode the variable and use it.

# 4. Building the Models

### 4.0 Model evaluation metrics

We are using F1 score and Accuracy as the metrics for evaluating the models. We are choosing F1 score as one the metric to give equal importance to both positive and negative classes. Accuracy is used as a secondary evaluation metric to avoid confusion if two models give almost same F1 score.

**F1 score** is the Harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

where, Precision = True positive upon total predicted positive
Recall = True positive upon total actual positive

**Accuracy** is the proportion of total correct prediction to total number of predictions

## 4.1 Logistic Regression

From the exploratory data analysis we have concluded that, because the presence of multicollinearity in the important features we should choose a non parametric models. Therefore, we are not expecting a good performance from logistic regression, which is a parametric model. But we are building logistic regression model to use as a base model. Performance of this logistic regression model will be used as lower limit of performance for choosing other models.

KNN imputer, Robust Scaler and Yea-johnson transformation are applied to the dataset before training the model using the Logistic regression inside the pipeline.

Using stratified K-fold(with 5 folds) we got an F1 score of 0.315

**Testing the individual effect of each variable using sklearn.feature_selection:**

- PAY_1 and PAY_2 have the highest F-statistic
- AGE, BILL_AMNT4 to 6 are insignificant variable since p-values are greater than 0.10

**The most important feature for predicting the default is PAY_1. That is whether the customer has defaulted for the previous month.**

Logistic regression model is trained(70%) with all the features except AGE, BILL_AMNT4 to 6.

| | | |
|---|---|---|
| Model Performance: | F1 | : 0.327 |
| | Accuracy | : 0.803 |

## 4.2 K-Nearest Neighbor Classifier

KNN classifier is a non-parametric machine learning algorithm. It stores all the training data and make prediction by using a specified number (K) of training data points which is nearest to the data point which needs to be predicted.

KNN imputer, Robust Scaler and Yea-johnson transformation are applied to the dataset before training the model using KNN classifier inside the pipeline.

Using grid search, we found the optimum value for the number of neighbors is 5.

| | | |
|---|---|---|
| Model Performance: | F1 | : 0.421 |
| | Accuracy | : 0.790 |

## 4.3 LDA Classifier

LDA classifier uses mean, variance and standard deviation of the target classes and apply Bayes' theorem to make the predictions. Since the algorithm assumes common variance, the function(discriminant function) is linear in nature. This also also assumes normal distribution of observations within the classes.

KNN imputer, Robust Scaler and Yea-johnson transformation are applied to the dataset before training the model using the LDA classifier inside the pipeline.

Model Performance: F1 : 0.334
Accuracy : 0.802

## 4.3 QDA Classifier

QDA classifier is similar to LDA classifier. But it uses class specific covariance matrix instead of common variance. Therefore the function appears as a quadratic function.

KNN imputer, Robust Scaler and Yea-johnson transformation are applied to the dataset before training the model using the QDA classifier inside the pipeline.

Model Performance: F1 : 0.489
Accuracy : 0.687

## 4.4 Decision Tree Classifier

Decision tree classifier divides the feature space into simple regions during the training stage. The new observations are predicted using these simple regions.

KNN imputer is applied to the dataset before training the model using the DT classifier inside the pipeline.

Using Grid search, we found the hyper-parameters

max_depth : 5
criterion : entropy
class_weight : balanced
splitter : best

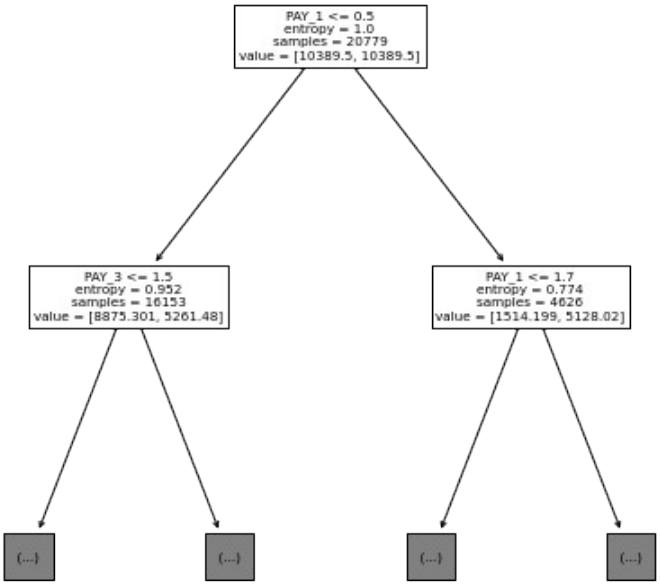


**Figure. 1**

Model Performance: F1 : 0.524
Accuracy : 0.776

## 4.5 Ensemble Models – Bagging

In ensemble techniques, we use multiple models for the same problem. In Bagging model, we build multiple base models using the random sub samples created with replacement from the dataset. The aggregated prediction from all these models will be our final model.

### 4.5.1 Random Forest Classifier

Random forest is the application of Bagging with Decision Trees. In order to add randomness to each model, we can limit the maximum number of features used while finding the nodes.

KNN imputer is applied to the dataset before training the model using the Random Forest classifier inside the pipeline.

Using Grid search, we found the hyper-parameters

max_depth : 7
criterion : entropy
class_weight : balanced
max_features : 5

Model Performance: F1 : 0.534
Accuracy : 0.770

### 4.5.2 Bagging with Logistic Regression

Model Performance: F1 : 0.331
Accuracy : 0.802

### 4.5.2 Bagging with QDA

Model Performance: F1 : 0.494
Accuracy : 0.705

## 4.6 Ensemble Models – Boosting

Boosting models are used to convert a weak learner/model to a good model. Multiple models are used in Boosting. But, in contrast with Bagging, the models are trained in series. Misclassified observations in the previous model will have more weightage in the next model.

### 4.6.1 Adaboost Classifier

KNN imputer is applied to the dataset before training the model using the Adaboost classifier inside the pipeline.

Model Performance: F1 : 0.441
Accuracy : 0.817

### 4.6.1 Gradient Boosting Classifier

Gradient Boosting is a generalization of Adaboost.

KNN imputer is applied to the dataset before training the model using the Gradient Boosting classifier inside the pipeline.

Using Grid search, we found the hyper-parameters

learning_rate : 0.1
n_estimators : 200
max_depth : 4
subsample : 0.5

Model Performance: F1 : 0.469
Accuracy : 0.812

**Feature Importance:**

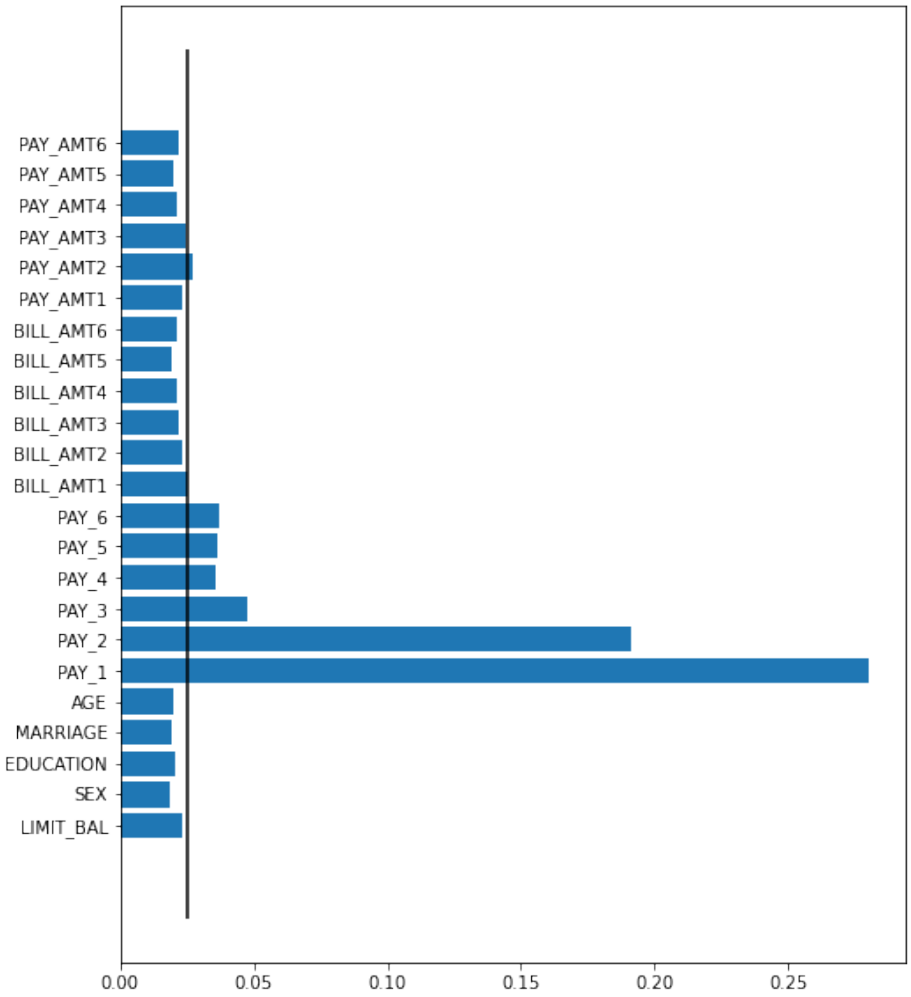


**Figure. 2**

PAY_1, PAY_2 and PAY_3 are the features that effect prediction the most.

# 5. Model Comparison

We have fitted 10 different models using the dataset. The performances of these models are given in **Table. 2**

| Model | F1 score | Accuracy |
|---|---|---|
| Logistic Regression | 0.327 | 0.803 |
| LDA Classifier | 0.334 | 0.802 |
| QDA Classifier | 0.489 | 0.687 |
| KNN Classifier | 0.421 | 0.790 |
| Decision Tree Classifier | 0.524 | 0.776 |
| **Random Forest Classifier** | **0.534** | **0.802** |
| Bagging with Logistic Regression | 0.331 | 0.802 |
| Bagging with QDA | 0.494 | 0.705 |
| Adaboost Classifier | 0.441 | 0.817 |
| Gradient Boosting Classifier | 0.469 | 0.812 |

**Table. 2**

According to our primary evaluation metric, F1 score, the best model is Random Forest Classifier. We have got the highest accuracy with Adaboost Classifier and Gradient Boosting Classifier. But F1 scores are less for them. In conclusion, we can choose Random Forest Classifier as the best model for our problem.