Assignment#08:Prediction of Credit Card Payment Default

**Team Members**

**Harika Addagada (hxa150930)**

**Ravali Nallamasu(rxn152730)**

**Aman Miryala (axm151830)**

**Himanshu Parashar(hxp151330)**

**Purpose:** To create a model that will predict the default of credit card payment for a given dataset using the techniques Decision Trees(Bagging and Random Forest) and Support Vector Machine.

**Dataset:**  Credit Card Payment dataset available in below link

<https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

**Data Preprocessing:**

* First, we removed the ID column from the dataset as it is the client identifier and doesn’t add value to the analysis
* Renamed the “default payment next month” attribute to “default\_payment” (response variable) for easy access of the column.
* Then we fixed the datapoint of education and marriage - combined few categories of Education together to get only 4 types i.e., 1 = graduate school; 2 = university; 3 = high school; 4 = others
* And, aggregated the marriage attribute values into 3 categories which are 1 = married; 2 = single; 3 = others
* Then, converted the character attributes Education, Marriage and Sex into factors for analysis.

**Some Observations:**

* From summary of data, we saw that there are more female than male in the dataset.
* There are more single people than married, i.e (53.2%:45.5%)
* From the correlation plots with demographic data (sex, education and marriage) against default\_payment we can observe that married woman are more likely to default payment.
* The default\_payment is approx 22% of the total recordings in the dataset.

**Approach/Model Building:**

**Decision Tree (Random Forest):**

**Without adjusting datapoint for education and marriage. And without removing attribute “ID”**

* First we installed the randomForest package then converted the default.payment.next.month or default\_payment in our case into factor.
* Then we divided out dataset into training and test data in ratio of 75:25 respectively. So out of 30000 records, we have 22500 records for our training data.
* Then we used randomForest function to form our decision tree model on training data, with number of variables randomly sampled at each split (mtry) set to 4 . We got estimate of error rate = 17.75 % and confusion matrix as

Confusion matrix:

0 1 class.error

0 16664 924 0.05253582

1 3069 1843 0.62479642

* Using importance function we saw variable importance measure as

MeanDecreaseGini

ID 492.41982

LIMIT\_BAL 370.51443

SEX 73.91762

EDUCATION 130.17241

MARRIAGE 89.56886

AGE 401.06139

PAY\_0 719.65505

PAY\_2 335.51049

PAY\_3 221.20978

PAY\_4 175.05407

PAY\_5 146.48878

PAY\_6 140.95033

BILL\_AMT1 424.58760

BILL\_AMT2 386.77554

BILL\_AMT3 367.62435

BILL\_AMT4 355.01603

BILL\_AMT5 350.89011

BILL\_AMT6 355.39707

PAY\_AMT1 368.04890

PAY\_AMT2 337.84828

PAY\_AMT3 326.43996

PAY\_AMT4 311.00667

PAY\_AMT5 311.67087

PAY\_AMT6 325.71731

* Then we used predict for this model on test data.
* The model did a good job and resulted in below predictions when compared against the original values available in the dataset:

Predicted

Actual 0 1

0 5461 315

1 1088 636

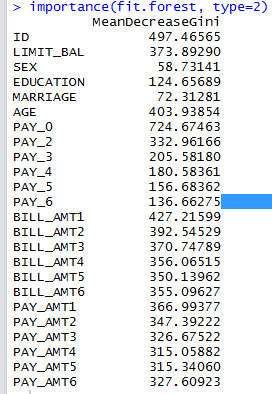
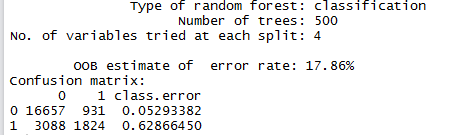
* To calculate our prediction accuracy from this matrix/table we used

(forest.pred[1] + forest.pred[4])/ (forest.pred[1] + forest.pred[2] + forest.pred[3] + forest.pred[4])

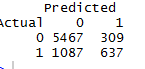
* And our accuracy is 0.8129333.

**Again trying random forest. This time by removing ID and fixing datapoint for education and marriage as mentioned in preprocessing.**

* Test and train data was split in the same manner as earlier (75:25) but mtry set to 6 this time in randomForest, we got accuracy of 81.24% so we again tried with mtry=4 and following is the snip of the result we got



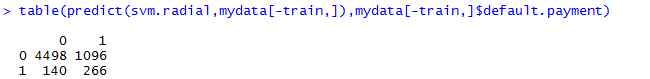
* Predictions on test data–



* Prediction accuracy we got this time = 0.8138667. which is almost same, just slightly better than previous one.

**SVM (Support Vector Machine):**

**With preprocessed data as mentioned in preprocessing**

* We first split the data into train and test data using the sample() function. We took 80%(24,000 rows out of 30,000 rows) as train data and the rest (6,000) as test data.
* We first built the svm model using the svm() function in e1071 package by passing in the train data, gamma=1,cost=1,kernel=”radial” as parameters.
* Then we built a table of predicted values of the “default payment next month” attribute obtained using the above model versus actual values for test data set and found that the accuracy is 79.4%
* 
* Next we tried building svm using several kernel types like “linear” and “polynomial”(with and without scaling). But it took a lot of time to finish and eventually we had to terminate the process.

**Summary:**

On comparing the values obtained after building the models using various techniques, we found SVM and Random Forest did a really good job in predicting the values with accuracy in predicting as 79.4% and 81.39% respectively. We also observed that we were able to build SVM with kernel type as “radial” only. hence SVM seemed to do a pretty good job in predicting the “default payment next month” attribute. When we tried to build SVM using kernel type as “linear” and “polynomial”, the process took a lot of time and we eventually terminated it. For this data we got better accuracy with Random Forest (Decision Tree) and also it took comparatively less time than SVM. In RF we saw that we can slightly improve accuracy by adjusting data point for some attributes where we can have relevant types only for that attribute, also by setting mtry (Number of variables randomly sampled as candidates at each split).