Project Name - Santander Customer Transaction Prediction

**Problem Statement :**

In this challenge, we need to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted.

You are provided with an anonymized dataset containing numeric feature variables, the binary target column, and a string ID\_code column. The task is to predict the value of target column in the test set.

**SOLUTION :**

We are given train and test dataset and our task is to predict the target values in gtest dataset.The target variable in the train dataset contains either 0 or 1 values which simply implies that target is a categorical variable.

So,this is a classification problem.And for classification problem firstly ,we will predict the values using Logistic Regression Machine Learning Algorithm.

Before implementing this algorithm,we will determine various characterstics of the dataset for ex- no of observations and variables present,type/class of various variables present,summary of variables and dataset,finding whether missing values are present and if present then undergoing missing value analysis.After this we will perform various Machine Learning Algorithm on our dataset to predict the target variable values and depending on the performance of the model, we will choose the best model and those predicted values will be our final result.

Now,starting with our train dataset.Let’s first load the data and determine the characteristics and properties of our data present in train dataset.

**R Code :**

rm(list=ls()) #clean the RAM

setwd("C:/Users/user/Documents") #set working directory

getwd() #get current working directory

df=read.csv("C:/Users/user/Downloads/train.csv",header=T) #loading train dataset

df2=read.csv("C:/Users/user/Downloads/test.csv",header=T) #loading test dataset

dim(df) #getting no of observations and variables in train dataset

dim(df2) #getting no of observations and variables in test dataset

str(df) #getting structure variables names,their type and data of train dataset

str(df2) #getting structure variables names,their type and data of test dataset

colnames(df) #column names of train dataset

colnames(df2) #column names of test dataset

class(df$target) #type of variable target in train dataset

class(df2$var\_0) #type of variable var\_0 in test dataset

Now,let us start with our Missing Value Analysis :

miss\_val=data.frame(apply(df,2,function(target){

sum(is.na(target))

}))

miss\_val

This will give missing values present in each variable.If missing values found then we can undergo mean,median and KNN imputation method to determine which method is best to depict missing values using this dataset.This would be decided depending on the values nearance to the original value in the dataset.As,this dataset don’t contain any missing value so we don’t need to evaluate missing values through various imputation.Lets just find out which method is best for missing value imputation for this dataset.

df[20,3] #get value at 20th observation for 3rd variable in train dataset # 4.409

df[20,3]=NA #set value NA at 20th observation for 3rd variable in train dataset

df$var\_0[is.na(df$var\_0)]=mean(df$var\_0,na.rm=T) #Mean method

df[20,3] # 10.67995

df$var\_0[is.na(df$var\_0)]=median(df$var\_0,na.rm=T) #Median method

df[20,3] # 10.5248

require(DMwR) #library to impute KNN Imputation for Missing value Analysis

df=knnImputation(df, k=2) #Knn Imputation

sum(is.na(df))

df[20,3] # 11.0208

So,Median method value is nearest to the original value.Therefore,for this dataset it is better to use Median method for Missing Value Analysis.

Now,let us determine the independent variables which affect the target(dependent) variable mostly.Now, using Correlation Analysis for Feature Selection.Correlation Analysis is applicable on only numeric data.So,first convert the target variable into numerical variable.

df <- data.frame(df[,-1], row.names = df[,1]) #make the first column as index for the train dataset

df

df2 <- data.frame(df2[,-1], row.names = df2[,1]) #make the first column as index for the test dataset

df$target=as.numeric(df$target) #changing target to numeric type

require(corrplot) #library to plot correlation graph

M<-cor(df)

head(round(M,2))

corrplot(M1, method="circle") #plot correlation graph indicating circles

corrplot(M1, method="color") #plot correlation graph indicating colours

corrplot(M1, method="number") #plot correlation graph indicating numbers

Through this we will know the correlation among dependent and independent variables.

Now,perform various machine learning algorithms on our dataset in order to predict the values for target variable in test dataset.

Firstly,start with basic algorithm i.e. Logistic Regression Algorithm.

Logistic Regression Algorithm is same as Linear Regression.The only difference is that if the target variable is categorical and continuous then we go for Logistic Regression.It predict probability of particular outcomes.Uses logistic function to determine the prediction.It can be binomial,ordinal and multinomial.

It assumes that their is absence of multicollinearity,no outliers and independence of errors.

require(tibble) #library to add column at given location in dataset

df2=add\_column(df2,target=df$target,.before="var\_0") #adding target column in test dataset before var\_0

df2$target=as.factor(df2$target) #converting target variable to factor variable as it is categorical variable

df$target=as.factor(df$target) #converting target variable to factor variable as it is categorical variable

require(glmm) #library to perform Logistic Regression

logit\_model=glm(target~.,data=df,family="binomial")

summary(logit\_model)

pred1=predict(logit\_model,newdata=df2,type="response") #predicting values for test data

pred1=ifelse(pred1>=0.05,1,0) #if probability is > 0.05 then 1 else 0

pred1

conf=table(df$target,pred1) #build confusion matrix

conf

83/(83+55) #determine False Negative Rate #0.6014493

(55+7)/(55+83+7+5) #determine accuracy #0.4133333

(55)/(55+83) #determine Recall #0.3985507

Now,we will again predict the values and determine the performance using KNN Analysis Machine Learning Algorithm.This algorithm stores all available cases and classifies new cases based on a similarity measure.It is local heuristic.Can be used both for Regression and Classification data.

require(class) #library to impute knn analysis

KNN\_pred=knn(df,df2,df$target,k=1) #predict knn values

conf\_matrix=table(KNN\_pred,df$target) #build confusion matrix

sum(diag(conf\_matrix))/nrow(df2) #determine accuracy # 0.8733333

conf\_matrix

(12)/(131+12) #determine False Negative Rate #0.07801418

(130)/(130+11) #determine Recall # 0.9219858

Now,from the performance parameters values for Logistic Regression and Knn Analysis ML Algorithm. KNN Analysis is better Algorithm than Logistic Regression because FNR is much more lower in KNN Analysis,which makes the model error free.Also,the Accuracy and Recall of KNN Analysis algorithm is higher than Logistic Regression.

Now,we will again predict the values of test data using Naive Bayes Algorithm for getting much more accurate values.

Naive Bayes is practical learning method.It is a probabilistic classification model.It Bayes theorem of probability to predict the class of unknown dataset.It assumes each independent variable is contributing independently so all probability for each independent variable is found.

require(e1071) #library to perform Naive Bayes Analysis

NB\_model=naiveBayes(target~.,data=df) #takes train data

NB\_predict=predict(NB\_model,df2,type = 'class') #predict test data

NB\_predict

conf=table(observed=df[,1],predicted=NB\_predict) #build matrix

caret::confusionMatrix(conf) #print confusion matrix

1/(137+1) #determine FNR #0.007246377

(137)/(137+1) #determine Recall #0.9927536

137/(137+1+12+0) #determine Accuracy #0.9133333

Now,among all the 3 Machine Learning algorithms for predicting values Naive Bayes is the best algorithm because FNR value is lowest in Naive Bayes algorithm i.e. Naive Bayes reduce the tendency of errors in the model.Also,at the same time,Accuracy and Recall values are also highest for Naive Bayes Algorithm.This makes Naive Bayes Algorithm most accurate and efficient Machine Learning algorithm for predicting the values for this dataset.

So,at last save the predicted values by Naive Bayes model in the target variable for the test set.

df2$target=NB\_predict #save the predicted values in test set-target variable.

**NOTE :**

I have explained the outline of the project using only R code. Similarly, the Python code can also be used for this problem.I have attached both the R and Python code files along with my project.