OR-568 Applied Predictive Analytics

IBM Watson Marketing Data Analysis

Doddanaik Basavaraj Vakkund
Data Analytics Engineering
George Mason University

Mercy Ahalya Seelam
Data Analytics Engineering
George Mason University

Sindhusha Vempati
Data Analytics Engineering
George Mason University

Shiva Ram Kaushil Pabba
Data Analytics Engineering
George Mason University
spabba2@gmu.edu

Vineel Vishwanth Busi
Data Analytics Engineering
George Mason University
vbusi@gmu.edu

Abstract – To understand demographics of consumers and buying behavior. Using predictive analytics to determine which consumers are most profitable and how they communicate. Take tailored steps to improve competitive response, acquisition, and development for the customers.

In this project using the predictive analysis, we will predict the behavior of the customer and retain the customers using the customer information. There is only one dependent feature available in the dataset which is "response", which will give information whether the customer respond back based on the service he is offered.

I. INTRODUCTION

IBM Watson Analysis is an application that helps for identifying patterns and to know more details of the data. IBM Watson Analysis is also used in self- service data analysis and visualization. Because of its languages and processing this analysis having it feels like we are talking with data. This will produce response either a structured or unstructured way. This analysis contains the details of the customer, whether they are employed, what car are they using and customer Id. It analyzes all the customer data and develop some programs that helps in retrieving customers back so they can get products again and again.

II. OBJECTIVE

In this project we are going to understand customer demographics and predict the behavior of customer and retain the customers using the customer information. We will find out how to increase the customer retention, response and growth of the customers by analyzing the customer response based on the services they are offered. By doing the predictive analysis we can also find what features are significantly important to retain the customer.

III. DATASET

IBM Watson Marketing dataset is having the details of customers and it is collected to know whether the customer will respond back or not. This response variable is called as the dependent variable. As it is the only feature which is having the binary values as 0 and 1. This dataset have total of **9135** rows and **24** variables. This is in the format of CSV file. There is only one dependent variable in the dataset which is response which will give information about the customer response based on the service he is offered, and the rest of variables are either numerical or categorical data types. It not only contains the details of customer but also details of vehicle and policy they took and complaints they filed when they come back. [1]

Source: Taken from the Kaggle source which is owned by Google LLC.
Privacy: This dataset is intended for Public access and use.

S.No.	Data Field	Data Type	Definition	Example
1	Customer	Continuous	Unique ID of the customer	BU79786
2	State	Categorical	The state location of the customer	Washington
3	Customer Lifetime Value	Numerical	Value is a prediction of the net profit attributed to the entire future relationship with a customer	2763.519
4	Response	Categorical	The customer responded for the policy	No
5	Coverage	Categorical	The class of the policy	Extended
6	Education	Categorical	Education qualification of the customer	Bachelor
7	Effective to Date	Date	The start date of customer purchase	1/31/2011
8	Employment Status	Categorical	The employment status of the customer – Employed/Unemployed	Employed
9	Gender	Categorical	The gender details of the customer	F
10	Income	Numerical	The Per annum income of the customer	56274
11	Location Code	Categorical	The location level of the customer – Urban/Rural/Suburban	Suburban
12	Marital Status	Categorical	Marital status of the Customer	Married
13	Monthly Premium Auto	Numerical	Auto Loan monthly premium paid by the customer	69
14	Months Since Last Claim	Numerical	The Number of months where customer took the gap	32
15	Months Since Policy Inception	Numerical	The number of months since the policy taken	5
16	Number of Open Complaints	Numerical	Complaints raised by the customer	0
17	Number of Policies	Numerical	The Number of total policies	8
18	Policy Type	Categorical	The category of the policy	Corporate Auto
19	Policy	Categorical	The category of the policy – L2/L3	Corporate L3
20	Renew Offer Type	Categorical	Which offer type is used to renew the policy	Offer1
21	Sales Channel	Categorical	The channel by which the customer took the policy	Agent
22	Total Claim Amount	Numerical	Claimed amount by the customer	384.8111
23	Vehicle Class	Categorical	The Vehicle class details	Two-Door Car
24	Vehicle Size	Categorical	The size of the vehicle.	Med-size

The chosen dataset is real world data and on analysis will give the details which we are aiming for. The dataset has all the customer details like their Employment status, Education, Income, Monthly premium auto, Renew offer type, Sales channel along with the customer's demographic details and so forth.

The Dataset contains some **NaN** values which needs to be looked. Data cleaning needs to be done. After data cleaning process, the data exploration needs to be performed to obtain hidden information or insights for the decision making. The Dataset contains 24 fields with different data types but we need only few

important features for decision making and model building. The response feature can be used to predict the customer's behavior with regards to the services offered to them.

Summary stats for raw data before preprocessing:

```
summary(IBM_data)
   Customer            State       Customer.Lifetime.Value Response        C
AA10041:    1   Arizona   :1703   Min.    : 1898          No :7826    Basic
AA11235:    1   California:3150   1st Qu.: 3994          Yes:1308    Exten
AA16582:    1   Nevada    : 882   Median : 5780                      Premi
AA30683:    1   Oregon    :2601   Mean   : 8005
AA34092:    1   Washington: 798   3rd Qu.: 8962
AA35519:    1                     Max.   :83325
(Other):9128
                Education     Effective.To.Date       EmploymentStatus Gende
Bachelor             :2748   1/10/2011: 195     Disabled      : 405     F:465
College              :2681   1/27/2011: 194     Employed      :5698     M:447
Doctor               : 342   2/14/2011: 186     Medical Leave: 432
High School or Below:2622    1/26/2011: 181     Retired       : 282
Master               : 741   1/17/2011: 180     Unemployed    :2317
                             1/19/2011: 179
                             (Other)   :8019
    Income         Location.Code   Marital.Status Monthly.Premium.Auto
Min.   :     0   Rural    :1773   Divorced:1369   Min.   : 61.00
1st Qu.:     0   Suburban:5779   Married :5298   1st Qu.: 68.00
Median :33879   Urban    :1582   Single  :2467   Median : 83.00
Mean   :37657                                    Mean   : 93.22
3rd Qu.:62338                                    3rd Qu.:109.00
Max.   :99981                                    Max.   :298.00
NA's   :5                                        NA's   :4
Months.Since.Last.Claim Months.Since.Policy.Inception Number.of.Open.Comp
Min.   : 0.0            Min.   :  0.00                Min.   :-1.0000
1st Qu.: 6.0            1st Qu.: 24.00                1st Qu.: 0.0000
Median :14.0            Median : 48.00                Median : 0.0000
Mean   :15.1            Mean   : 48.13                Mean   : 0.3835
3rd Qu.:23.0            3rd Qu.: 71.00                3rd Qu.: 0.0000
Max.   :35.0            Max.   :640.00                Max.   : 5.0000
```

Figure: 1. Summary of data before preprocessing

From the summary stats we can observe that this data was collected on **9134** unique customers from 5 states along with other details like income, education, employment status, gender, location code, marital status. From summary stats we can observe that there are null values in features like income, monthly premium auto, and number of open complaints we have minimum value as -1 which makes no sense. By observing "Effective to Date" feature we can say that data was collected for 2 months which was not mentioned on the website.

Summary stats for cleaned-up data

```
> summary(IBM_data)
       State       Customer.Lifetime.Value Response        Coverage
 Arizona   :1703   Min.    : 1898          No :7826   Basic   :5568
 California:3150   1st Qu.: 3994          Yes:1308   Extended:2742
 Nevada    : 882   Median : 5780                     Premium : 824
 Oregon    :2601   Mean   : 8005
 Washington: 798   3rd Qu.: 8962
                   Max.   :83325

                Education     Effective.To.Date       EmploymentStatus Gender
 Bachelor             :2748   1/10/2011: 195     Disabled      : 405     F:4658
 College              :2681   1/27/2011: 194     Employed      :5698     M:4476
 Doctor               : 342   2/14/2011: 186     Medical Leave: 432
 High School or Below:2622    1/26/2011: 181     Retired       : 282
 Master               : 741   1/17/2011: 180     Unemployed    :2317
                              1/19/2011: 179
                              (Other)   :8019
    Income         Location.Code   Marital.Status Monthly.Premium.Auto
 Min.   :     0   Rural    :1773   Divorced:1369   Min.   : 61.00
 1st Qu.:     0   Suburban:5779   Married :5298   1st Qu.: 68.00
 Median :33890   Urban    :1582   Single  :2467   Median : 83.00
 Mean   :37657                                    Mean   : 93.22
 3rd Qu.:62320                                    3rd Qu.:109.00
 Max.   :99981                                    Max.   :298.00

 Months.Since.Last.Claim Months.Since.Policy.Inception Number.of.Open.Complaints
 Min.   : 0.0            Min.   :  0.00                Min.   :0.0000
 1st Qu.: 6.0            1st Qu.: 24.00                1st Qu.:0.0000
 Median :14.0            Median : 48.00                Median :0.0000
 Mean   :15.1            Mean   : 48.13                Mean   :0.3844
 3rd Qu.:23.0            3rd Qu.: 71.00                3rd Qu.:0.0000
 Max.   :35.0            Max.   :640.00                Max.   :5.0000

 Number.of.Policies          Policy.Type            Policy      Renew.Offer.Type
 Min.   :1.000       Corporate Auto:1968   Personal L3 :3426   Offer1:3752
```

Figure: 2. Summary of data after preprocessing

We can observe in the above figure that there is no null values in features like income, monthly premium auto as they are replaced by mean with respect to their data column and in number of open complaints feature we have minimum value as -1 which might have occurred because of human error so we replaced -1 with 1 as complaints can be either 0 or more than 0 but it can't be -1.

IV. EXPLORATORY DATA ANALYSIS

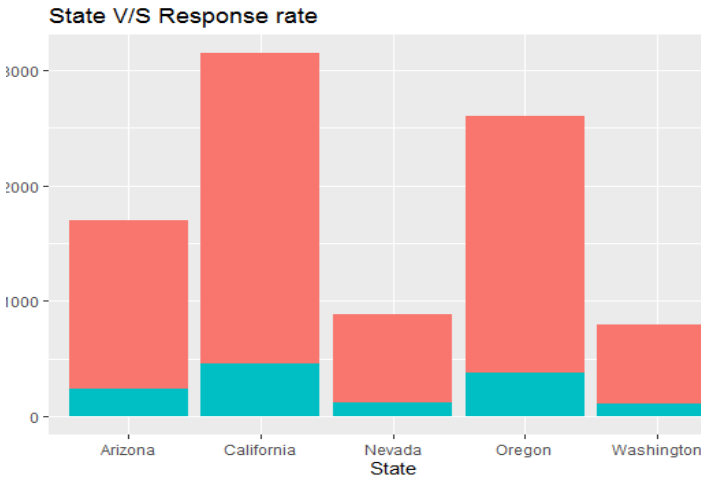


Figure: 3. State wise Response Rate

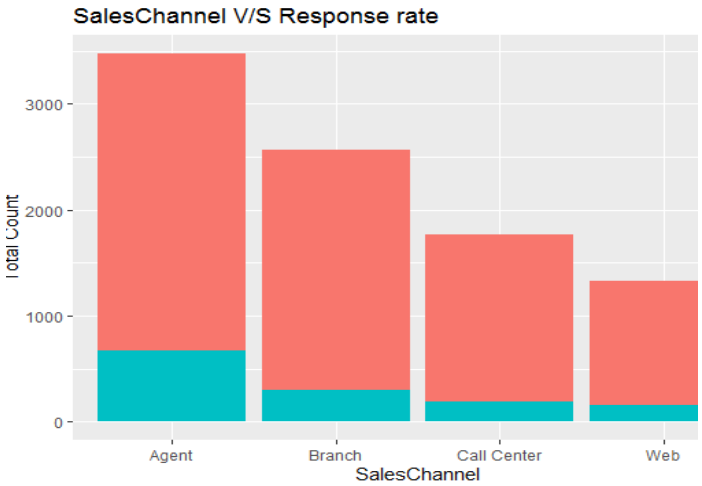


Figure: 4. Sales Channel vs Response Rate

Analyzed customer responses over the 5 States available in the dataset: From the plot in figure 3 we can see that California and Oregon have more customer who have responded positively to the various marketing schemes as compared to other states but they also have higher number of non- responders as data collected from those states is more in the dataset. Similarly when we observe the response rate with respect to sales channel as shown in figure 4 we can see that through agent medium customer were reached more and more customer have responded positively and 2nd medium which stands out is branch as compared to other mediums as call center and web.

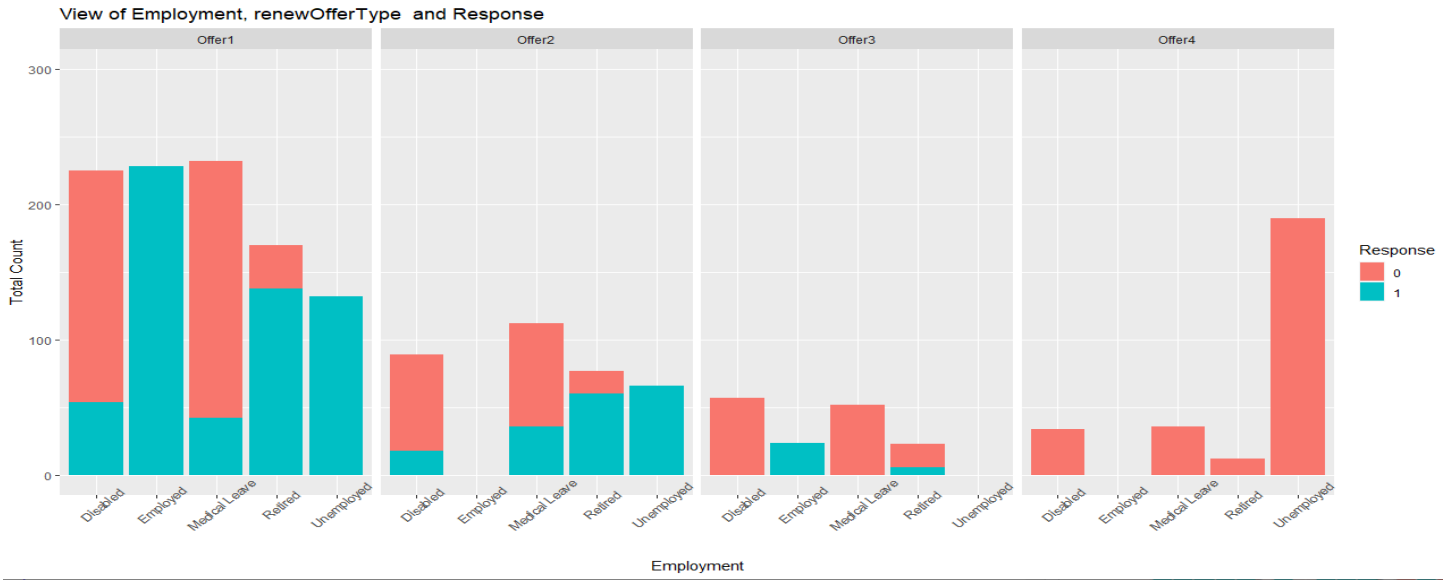


Figure: 5. View of Employment, renew Offer Type with Response

We tried to analyze how customers responded to 4 offers offered to them based on their employment status as shown in the plot in figure 5, we can observe most of the employed customers have responded to offer1 and

offer 3 and, we can also observe that employed customer's have not responded back to offer2 and offer4. Most of the unemployed customers have responded back positively to offer1 and offer2. These insights can be used for targeted marketing to retain customers.

V. FEATURE SELECTION:

Data splitting: Two separate datasets were created from the original dataset, one for the training and one for validation purpose. The dataset used for training consists of 80% of the rows in the original dataset, and the other 20% of the rows were used for creating the validation dataset. The first column that stores the unique customer ID number were also removed as it does not have any significance in any kind of analysis.

To get the statistical and detailed analysis, we have used some algorithms which give us the results for our analysis. The first algorithm which we have taken is Logistic Regression, where the dependent variable class is predicted using the independent variables. We say it as the binary logistic regression as the dependent variable has two categories as "yes" and "no". In this model, we have used binomial as a family for the IBM data. Also, we used glm() function to run a binary logistic regression model. We are modelling the training data over here. Then we have used stepAIC() function on the regression model results. This function performs model trimming using backward model selection, starting from the most maximum model. All the unwanted variables are sliced down, and the most predicting values are included in the outcome. Based on that, we get an estimation. The selected features can be observed in the below figure 6.

```
Step:  AIC=4692.26
Response ~ Education + EmploymentStatus + Income + LocationCode +
    MaritalStatus + MonthlyPremiumAuto + MonthsSinceLastClaim +
    NumberofOpenComplaints + NumberofPolicies + RenewOfferType +
    SalesChannel + TotalClaimAmount + VehicleSize

                          Df Deviance    AIC
<none>                         4638.3 4692.3
- NumberofPolicies         1   4640.4 4692.4
- NumberofOpenComplaints   1   4640.6 4692.6
- Income                   1   4643.0 4695.0
- MonthsSinceLastClaim     1   4643.1 4695.1
- Education                4   4653.5 4699.5
- MonthlyPremiumAuto       1   4653.8 4705.8
- TotalClaimAmount         1   4661.1 4713.1
- VehicleSize              2   4665.1 4715.1
- MaritalStatus            2   4668.7 4718.7
- SalesChannel             3   4696.5 4744.5
- LocationCode             2   4763.4 4813.4
- EmploymentStatus         4   4990.4 5036.4
- RenewOfferType           3   5231.0 5279.0
```

Figure: 6. Output of stepwise backward logistic regression.

The features we use for further analysis are: " Education, Employment status, Income, Location code, Marital status, Monthly premium auto, Number of open complaints, Renew Offer type, Sales channel, Total claim Amount, Vehicle size".

VI. PREDICTION MODEL

We used four different methods to build model to predict customer response with respect to other significant independent variables obtained in feature selection. The methods used are:

1. Logistic Regression
2. Naïve Bayes
3. GBM – Gradient Boosting
4. Random Forest

- **Logistic Regression** - A statistical method for analyzing a dataset that contains one or more independent variables which determine the result. The effect is calculated against a dichotomous variable.

  Using Response as the dependent variable and the factors from stepwise regression as the independent variables, Logistic regression was performed on the given dataset. The whole data as mention above is split into 80% training data and 20 % test data. The model gave an accuracy of 88.12 % and the Area under the Curve for ROC as 0.799. The model could correctly classify 1610 instances and 217 instances incorrectly. To check the validity of the model performed, cross-validation with k = 5 and tuneLength = 5 is also performed. The accuracy of this model is 88.06% which is 0.06% less than the normal Binary Logistic regression model.

Accuracy: 88.12%

AUC: 0.799

Confusion matrix:

	0	1
0	1567	197
1	20	43

Cross Validation (k=5)

Accuracy: 88.06%

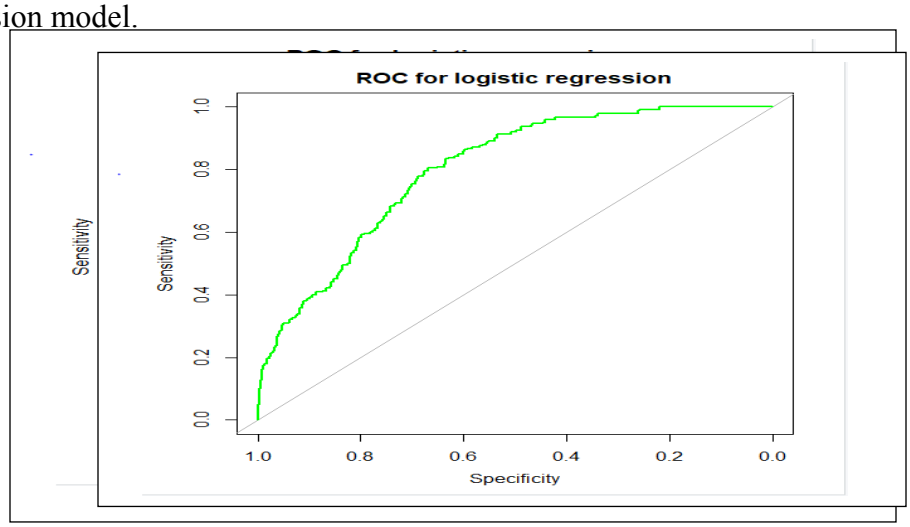


Figure: 7. ROC curve for binary logistic regression

- **Naïve Bayes -** A Naïve Bayes classification model of Bayes is an algorithm which uses theorem of Bayes to classify objects. Naive Bayes classifiers assume a strong, or naïve, independence between data point attributes.

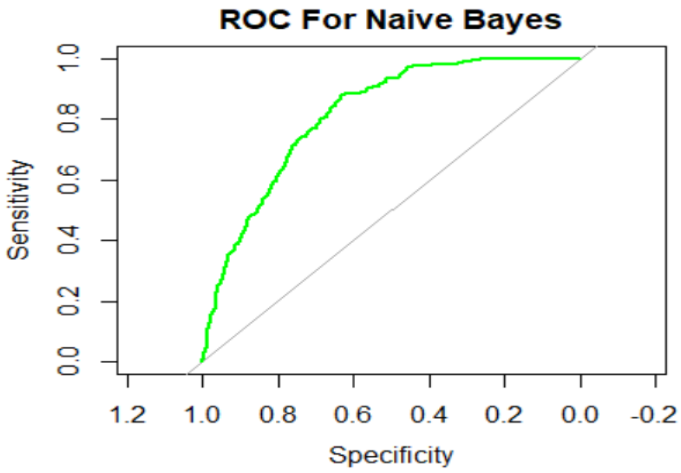
  There are several packages like e1071, Naïve Bayes which are used for, we used both the e1071 and caret packages for this model. First, we apply a 10-fold cross validation which we got 86% accuracy of observations in out training set not attrite. We had few parameters that can be used for tuning the naïve Bayes model they are User Kernel, adjust and FL. Use kernel allows to use kernel density to estimate for continuous variables with the Gaussian density. Adjust allows us to the bandwidth of kernel and FL acts as a Laplace smoother. In this model we can tune our model with the above-mentioned parameters, but we were not able to increase the accuracy of the model it is still the same the accuracy as model which we did not tool. [1]

Before Tuning

```
> confusionMatrix(predic, test$Response)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1574  235
         1   11    8

               Accuracy : 0.8654
                 95% CI : (0.8489, 0.8807)
    No Information Rate : 0.8671
    P-Value [Acc > NIR] : 0.5983

                  Kappa : 0.0426

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99306
            Specificity : 0.03292
         Pos Pred Value : 0.87009
         Neg Pred Value : 0.42105
             Prevalence : 0.86707
         Detection Rate : 0.86105
   Detection Prevalence : 0.98961
      Balanced Accuracy : 0.51299

       'Positive' Class : 0
```

After Tuning

```
> confusionMatrix(pred, test$Response)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1572  234
         1   13    9

               Accuracy : 0.8649
                 95% CI : (0.8483, 0.8802)
    No Information Rate : 0.8671
    P-Value [Acc > NIR] : 0.6246

                  Kappa : 0.0469

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.99180
            Specificity : 0.03704
         Pos Pred Value : 0.87043
         Neg Pred Value : 0.40909
             Prevalence : 0.86707
         Detection Rate : 0.85996
   Detection Prevalence : 0.98796
      Balanced Accuracy : 0.51442

       'Positive' Class : 0
```



Area under curve is 0.815

Accuracy % of Naïve Bayes 0.86

Figure: 8. ROC curve for Naïve Bayes

Here the prediction matrix of this is almost same before and the after tuning of the parameters where all correctly classified are 1582 and wrongly classified as 246 where after tuning there is 1 number of increment in both of the values like 1581, 247 respectively. We has also draw a ROC curve for this Naïve Bayes model which we can see in the below diagram area under the curve give the value as 0.815

- **GBM Gradient Boosting -** This produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

  For gradient boosting we have performed a 10 -fold cross validation method where we performed it on training dataset, and we could get the accuracy of 88.56% we did not perform any tuning on these parameters thinking that it will make no changes in the accuracy by observing no change in Naïve Bayes. Here we got the classification matrix as correctly classified as 1619 and wrongly classified as 219 where this leads to the accuracy of 88% which is little greater than Naïve byes and Logistic regression models comparatively. We have also plotted ROC curve for Gradient Boosting where AUC value is given as 0.882 where this is also greater than all the over models but less than the random forest. Hence, we can consider this as the second best model for analysis

Modeling:

Confusion Matrix:

gbm_ITV2	0	1
0	1574	198
1	11	45

Accuracy % of GBM: 88.56

AUC under curve: 0.8827

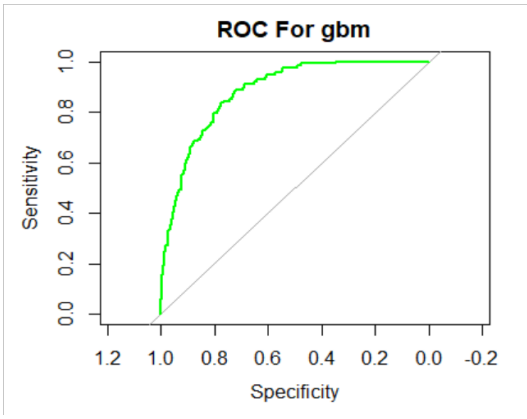


Figure: 9. ROC curve for Gradient Boosting

- **Random Forest -** It is a method for grouping, regression and other actions performed by constructing a multitude of decision trees at the time of training and generating groups that are class mode or mean trees prediction.

  As Random variable is best suited for data having imbalanced classes and also is effective compared to the rest of the models in avoiding over-fitting or under-fitting, Random forest algorithm is used on the given dataset with 80:20 split and the same predictor and response variables as used in Logistic Regression model. The model gave an accuracy of 93.37 % which is the highest of all the models used for prediction. The AUC for ROC curve is 0.9857 which is also the highest among all the models performed. The model was able to correctly classify 1706 instances while 113 instances were incorrectly classified [2].

  Variable importance plot is plotted to know the important variable effecting the response variable. From the variable importance plot it can be observed that Income, Total Claim Amount and Employment Status are top three important variables in determining the response variable while Location Code and Number of open Complaints variables are the least two important factors in predicting the response variable.

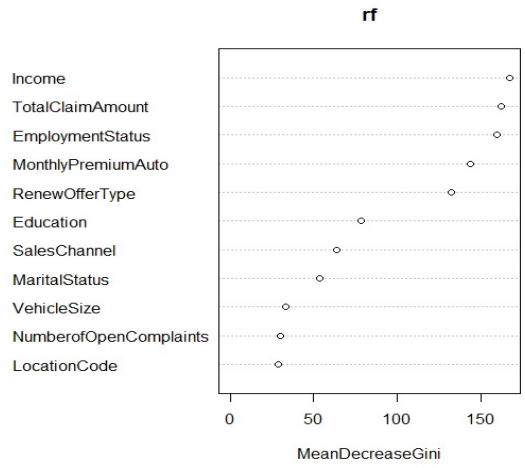


Figure: 10. Variable Importance plot

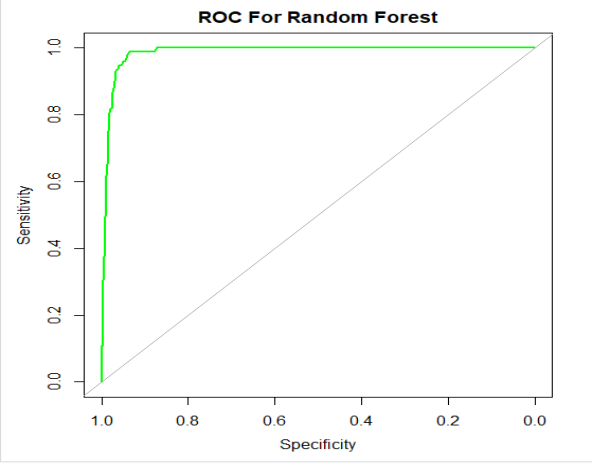


Figure: 11. ROC curve for Random Forest

Number of Trees: 100

Node size: 25

Model Evaluation:

Accuracy: 93.37%

AUC: 0.9857

Confusion matrix

	0	1
0	1569	103
1	18	137

VII. RESULTS AND DISCUSSION:

Four models are performed on the data. Their accuracies and AUC values of ROC as well their ROC curves are compared to choose the best model to predict the outcome variable which is 'Response' for the given dataset. Random Forest Model gave the best accuracy which is 93.37% while the lowest is for Naïve Bayes Model which is 86.54%.While the AUC value is also highest for Random Forest Model which is 0.985 the lowest is for Logistic regression which is 0.799. The figure below shows the ROC curves plotted for each of these models.

From the ROC curves it can be observed that Naive Bayes and Logistic Regression models performed equally well in the predicting the response variable and Gradient Boosting model performed better than these two models. Random Forest model performed the best in predicting the response variable.

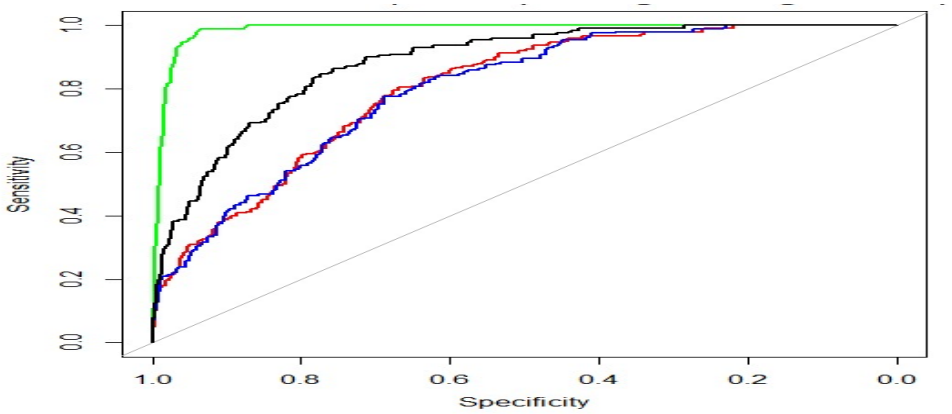


Figure: 12. ROC curve*s for Random Forest (Green) vs Logistic Regression (red) vs Naïve Bayes (Blue) vs Gradient Boosting (Black) Models*

Models	***Accuracy***	***AUC***
Binary Logistic Regression	88.12%	0.799
Naïve Bayes	86.54%	0.815
Random Forest	93.37%	0.985
Gradient Boosting	88.56%	0.882

VIII. CONCLUSION

There were few human errors in dataset which we found out and rectified. From our analysis Random forest Model gave the best accuracy in classifying the Response of the customer. It also have highest area under ROC curve. Income, Total Claim Amount and Employment Status are top three important variables in determining the response variable which is 'Response' in our dataset.

References

[1] "Kaggle," [Online]. Available: https://www.kaggle.com/pankajjsh06/ibm-watson-marketing-data-analysis-prediction. [Accessed May 2020].

[2] "UC Business Analytics," [Online]. Available: https://uc-r.github.io/naive_bayes. [Accessed May 2020].

[3] A. Garg, "Rpubs," May 2017. [Online]. Available: https://rpubs.com/Aakansha_garg/aakansha_cancer. [Accessed May 2020].

APPENDIX: R CODE:

```
install.packages('stringr')
library(stringr)

Watson <-read.csv('C:/Users/ahaly/Downloads/IBM_watson.csv')
summary(Watson)
dim(Watson)
head(Watson)
colSums(is.na(Watson))
str(Watson)

Watson <- Watson[,-c(1)]

colnames(Watson)
colnames(Watson) <- str_replace_all(colnames(Watson),"[.]","")
colnames(Watson)
head(Watson)
#Replacing Null values with Mean in each of its column
for(i in 1:ncol(Watson)) {
  Watson[ , i][is.na(Watson[ , i])] <- mean(Watson[ , i], na.rm = TRUE)
}

colSums(is.na(Watson))


# Found some -1 values instead of 1 might got mistyped so replacing them with 1
unique(Watson$NumberofOpenComplaints)
Watson$NumberofOpenComplaints[Watson$NumberofOpenComplaints=="-1"] <- 1

#Watson$Response[Watson$Response=="No"] <- 0

summary(Watson)

head(Watson)

levels(Watson$Response) <- 0:1

table(Watson$Response)
names(Watson)


#Data analysis
names(Watson)
```

```
ggplot(Watson,aes(x = State,fill=Response)) +
  geom_bar() +
  ggtitle("State V/S Response rate")+
  xlab("State") +
  ylab("Total Count") +
  labs(fill = "Response")


ggplot(Watson,aes(x = SalesChannel,fill=Response)) +
  geom_bar() +
  ggtitle("SalesChannel V/S Response rate")+
  xlab("SalesChannel") +
  ylab("Total Count") +
  labs(fill = "Response")

#States with response
ggplot(Watson, aes(x = State, fill = Response))+
  geom_bar() +
  facet_wrap(~SalesChannel,nrow = 1) +
  ggtitle("View of State, SalesChannel  and Response") +
  xlab("State") +
  ylab("Total Count") +
  ylim(0,300) +
  labs(fill = "Response")

#  Employment status and renewOffer
ggplot(Watson, aes(x = EmploymentStatus, fill = Response))+
    geom_bar() +
    facet_wrap(~RenewOfferType,nrow = 1) +
    ggtitle("View of Employment, renewOfferType  and Response") +
    xlab("Employment") +
    ylab("Total Count") +
    ylim(0,300) +
    labs(fill = "Response")  +
  theme(axis.text.x = element_text(angle = 45))


# Create Training Data
n <- nrow(Watson)
f <- ncol(Watson)
# Number of rows for the training set (80% of the dataset)
n_train <- round(0.80 * n)
# Create a vector of indices which is an 80% random sample
set.seed(123)
train_indices <- sample(1:n, n_train)
# Subset the Diabetes data frame to training indices only
train <- Watson[train_indices, ]
# Exclude the training indices to create the test set
test <- Watson[-train_indices, ]


#continuous
```

```
names(train)
train<-train[,-c(6)]

mylogit<-glm(Response~.,data=train,family = "binomial")
summary(mylogit)

library(caret)
library(MASS)

stepAIC(mylogit,direction="backward")

names(Watson)

#logistic regression

mylogit2<-glm(Response ~ Education + EmploymentStatus + Income +
            LocationCode + MaritalStatus + MonthlyPremiumAuto + NumberofOpenComplaints +
              RenewOfferType + SalesChannel + TotalClaimAmount + VehicleSize, family = "binomial", data =
train)
summary(mylogit2 )

pred <- predict(mylogit2,test,type = "response")
x<-table(round(pred), test$Response)
sum(diag(x))/sum(x)

#after backwar regression 0.88122

#cv
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)

mod_fit <-train(Response ~ Education + EmploymentStatus + Income +
              LocationCode + MaritalStatus + MonthlyPremiumAuto + NumberofOpenComplaints +
              RenewOfferType + SalesChannel + TotalClaimAmount + VehicleSize ,data = Watson,
            method="glm", family="binomial",trControl = ctrl, tuneLength = 5)

pred <- predict(mod_fit,test,type = "raw")
x<-table((pred), test$Response)
sum(diag(x))/sum(x)

ROC_logit<- roc(test$Response, pred[,2])
plot(ROC_logit, col = "green", main = "ROC For logistic")
ROC_logit_auc <- auc(ROC_logit)
paste("Accuracy % of  logistic: ",
    mean(test$Response == round(pred[,2], digits = 0)))


#random forest(
library(randomForest)
rf<-randomForest(Response ~ Education + EmploymentStatus + Income +
              LocationCode + MaritalStatus + MonthlyPremiumAuto + NumberofOpenComplaints +
              RenewOfferType + SalesChannel + TotalClaimAmount + VehicleSize,data=train)
summary(rf)
```

```
pred <- predict(rf,test,type = "response")
x<-table((pred), test$Response)
sum(diag(x))/sum(x)
#accuracy 99.4

ROC_rf <- roc(test$Response, pred[,2])
plot(ROC_rf, col = "green", main = "ROC For rf")
ROC_rf_auc <- auc(ROC_rf)
paste("Accuracy % of  Random Forest: ",
     mean(test$Response == round(pred[,2], digits = 0)))


#Naive bais
nB_model <- naiveBayes(Response ~ Education + EmploymentStatus + Income +
                         LocationCode + MaritalStatus + MonthlyPremiumAuto + NumberofOpenComplaints +
                         RenewOfferType + SalesChannel + TotalClaimAmount + VehicleSize,
                       data = train)
summary(nB_model)
pred <- predict(nB_model, test, type="class")
N<-table(pred, test$Response)
sum(diag(N))/sum(N)
#Accuracy:0.8653

# set up 10-fold cross validation procedure
train_control <- trainControl(method = "cv", number = 10)
metric <- "Accuracy"
x <- train[,c(5,6,8,9,10,11,14,18,19,20,22)]
y<- train$Response
nb.m1 <- train(
  x = x,
  y = y,
  method = "nb",
  trControl = train_control
)

# results
confusionMatrix(nb.m1)

predic <- predict(nb.m1,test)
confusionMatrix(predic, test$Response)

search_grid <- expand.grid(
  usekernel = c(TRUE, FALSE),
  fL = 0:5,
  adjust = seq(0, 5, by = 1)
)

# train model
nb.m2 <- train(
  x = x,
  y = y,
  method = "nb",
```

```
  trControl = train_control,
  tuneGrid = search_grid

)

confusionMatrix(nb.m2)

pred <- predict(nb.m2,test,type='prob')
confusionMatrix(pred, test$Response)

library(pROC)

ROC_nb <- roc(test$Response, pred[,2])
plot(ROC_nb, col = "green", main = "ROC For Naive Bayes")
ROC_nb_auc <- auc(ROC_nb)
paste("Accuracy % of Naive Bayes: ",
     mean(test$Response == round(pred[,2], digits = 0)))

#gbm
library(gbm)
fitControl <- trainControl(method = "repeatedcv", number = 4, repeats = 10)
gbmFit1 <- train(Response ~ Education + EmploymentStatus + Income +
                   LocationCode + MaritalStatus + MonthlyPremiumAuto + NumberofOpenComplaints +
                   RenewOfferType + SalesChannel + TotalClaimAmount + VehicleSize,
                 data = train, method = "gbm", trControl = fitControl,verbose = FALSE)
summary(gbmFit1)
gbm_ITV2 <- predict(gbmFit1, test ,type= "prob")
N<-table(gbm_ITV2, test$Response)
sum(diag(N))/sum(N)

ROC_gb <- roc(test$Response, gbm_ITV2[,2])
plot(ROC_gb, col = "green", main = "ROC For gbm")
ROC_gb_auc <- auc(ROC_gb)
paste("Accuracy % of  GBM: ",
     mean(test$Response == round(gbm_ITV2[,2], digits = 0)))

plot(ROC_rf, col = "green", main = "ROC For Random Forest (GREEN) vs Logistic Regression (RED) vs
                                Gradiant boosting (BLACK)vs Naive byes (BLUE) ")
lines(ROC_logit, col = "red")+lines(ROC_nb,col= "blue")+lines(ROC_gb,ncol="black")+lines()
#accracy:0.8702
```