STAT 6333: Statistical Learning

Case Study 2 Instructions
Fall 2021
Due: 28th Nov 2021
100 points

Expectations of the Case Study Analysis:

The main goal of this case study is to gets hands-on experience in applying the logistic regression, Random Forest and Support Vector Machines to solve a classification problem.

We will be using a marketing example from the insurance industry. The data contains information on customer responses to a historical direct mail marketing campaign. Our goal is to improve the performance of future waves of this campaign by targeting people who are likely to take the offer.

We will do this by building a "look-alike" model to predict the probability that a given client will accept the offer, and then use that model to select the target audience going forward. Obviously, we want a model that is accurate so that we can find the best possible target audience.

Dataset:

DataThe dataset has 68 predictive variables and 20k records. For modeling and validation purposes, we split the data into 2 parts:

- 10k records for training. This dataset will be used to estimate models.
- 10k records for testing. This dataset will be kept in a vault to the very end and used to compare models.

The success of the model will be based on its ability to predict the probability that the customer takes the offer (captured by the PURCHASE indicator), for the validation dataset.

Most variables contain credit information, such as number of accounts, active account types, credit limits, and utilization. The dataset also captures the age and location of the individuals.

The area under the ROC curve can be used to evaluate model performance. In order to make the comparison as fair as possible, we used the same set of variables for each model. The variables were selected using the following procedure given in Steps 2 and 3.

1. First, download the "CaseStudy2. Rdata" (R Workspace) and save it in your R working directory. Please use the following code to read the data into your R workspace. Please change the path (in red font) according to your working directory. Thos will allow you to access the train and validation data sets.

setwd("C:/Users/18133 /Fall 2021/MATH 6333/MATH 6392 Materials/Case Studies/Case Study 2/gampost-master")

```
load("CaseStudy2.RData")
```

Check the dimensions of the data using the following.

```
dim(train) #10000 by 69
dim(valid) #10000 by 69
```

2. **Information value (IV)** provides a great framework for exploratory analysis and variable screening for binary classifiers. IV have been used extensively in the credit risk world for several decades, and the underlying theory dates back to the 1950s.

Begin your analysis by removing all variables with an information value (IV) less than 0.05.

You can use the Information Package to calculation information values. (Please visit: https://github.com/klarsen1/Information, README.me for more details on IV and the Information package)

Use the following R code to perform this "Step 2" and recreate a new training and validation data sets with a reduced number of predictor variables. [10 points]

```
install.packages("Information")
library(Information)
IV <- create_infotables(data=train, y="PURCHASE", ncore=2)
View(IV$Summary)

train_new <- train[,c(subset(IV$Summary, IV>0.05)$Variable, "PURCHASE")]
dim(train)

valid_new <- valid[,c(subset(IV$Summary, IV>0.05)$Variable, "PURCHASE")]
dim(valid)
```

3. Eliminate highly correlated variables using variable clustering (ClustOfVar package). You can generate 20 clusters and picked the variable with the highest IV within each cluster.

Use the following R code to perform this "Step 3" and to **select the most informative 20 variables** to be used for the classification. [10 points]

```
######## Variable clustering
install.packages("ClustOfVar")
install.packages("reshape2")
install.packages("plyr")
library(ClustOfVar)
library(reshape2)
library(plyr)
```

```
tree <- hclustvar(train_new [,!(names(train_new)=="PURCHASE")])
nvars <- 20
part_init<-cutreevar(tree,nvars)$cluster
kmeans<-
kmeansvar(X.quanti=train_new[,!(names(train_new)=="PURCHASE")],init=part_init)
clusters <- cbind.data.frame(melt(kmeans$cluster), row.names(melt(kmeans$cluster)))
names(clusters) <- c("Cluster", "Variable")
clusters <- join(clusters, IV$Summary, by="Variable", type="left")
clusters <- clusters[order(clusters$Cluster),]
clusters$Rank <- ave(-clusters$IV, clusters$Cluster, FUN=rank)
View(clusters)
variables <- as.character(subset(clusters, Rank==1)$Variable)
variables  #This will give you the the final 20 variables that you will use for classification purposes
```

Now, you can use the selected 20 variables as the inputs for all the following classification models.

4. Create a new response variable called "NEWPurchase" using "PURCHASE" variable in the train data set. If **PURCHASE is equal 1**, make **NEWPurchase=1**, else use **NEWPurchase=-1**.

Perform the same procedure to create NEWPurchase variable in the valid data set. Add this new variable into the existing train and valid datasets by naming them as "train_new$NEWPurchase" and "valid_new$ NEWPurchase", repectively. (Hint: Use ifelse() in R) [5 points]

5. [30 points]

I. Build up a Random forest model using 10,001 trees using train data (use the randomForest package). Use different "mtry" values varying from 1 to 13. Evaluate the Out-of-Bag (OOB) error for each model.

II. Use the model with the lowest OOB error (best model) to create a variable importance plot to decide on the importance of the variables.

III. Make the predictions (i.e predict the potential class of each individual,-1 or 1) for the individuals in the valid data set by utilizing the above selected best model.

IV. Evaluate the confusion table and calculate the **Sensitivity** (the proportion of customers that were predicted to take the offer out of those who actually took the offer), **Specificity** (the proportion of customers that were predicted not to take the offer out of those who actually did not take the offer), and **Accuracy** (percentage of all correct predictions that were made by the model) values using the valid data set predictions.

V. Create the ROC curve and evaluate the AUC value.

(Hint: Before running the Random Forest model, use set.seed() to get the reproducible results).

6. [30 points]

I. Built two SVM classification models, one using a polynomial kernel with degree 3 and the other one using a Gaussian radial kernel (use the e1071 package) with train data. Use the cost value 0.01 for both models. Use gamma=0.000001 for the second model. Make the prediction for both models using the valid data set.

Hint: R Code for training:

```
svm.model <- svm(NEWPurchase ~., data=train_new[,c("NEWPurchase ", variables)],
cost=0.01, kernel="polynomial", degree=3, probability=TRUE)
```

R Code for Prediction:

```
predict(svm.model,newdata=valid_new,probability=TRUE)
```

II. Evaluate the confusion table and calculate the **Sensitivity**, **Specificity** and **Accuracy** using the valid data set predictions.

III. Create the ROC curve and evaluate the AUC value.

Based on the different classification models that you created in Steps 5 and 6, comment on the improvements in Accuracy, Sensitivity, Specificity and AUC values (higher these values, the better the model performance is).

After the completion of the analysis, you need to hand over a small report (at least with 5 pages or more) with your findings. Make sure to comment/add justifications on all your important results.

The front page should include your name. Make sure to attach your R code at the end of the report.

[Proper Details and Report Completion: 15 points]