Credit Card Prediction

Allison Gohl

Ann Mansour

Pandi Mengri

Maria Doda

Abstract

Our project aims to assist applicants that suffer from bad credit or the inability to gain a credit card understand which variable majorly impact their eligibility. In addition, it focuses on comparing different analytical models/methods to understand which one performs best in deciphering how a company approves or denies applicants requests for credit cards. Using data that display applicant's information such as such as credit score, income, hold balance, help identify the criteria used by a company to determine eligibility. This project compares four different models/methods: logistic regression (LR) model, linear discriminant analysis (LDA) model, k-nearest neighbors (KNN) algorithm and random forest method. Results show that random forest model yields to higher accuracy rate (99.4%) compared to other models (LR = 95.2%, LDA= 93.8%, KNN = 83.7%). The two top performing models helped determining that the most impactful variable to influence credit card eligibility is preferred customer probability.

Background

Many people apply for credit cards regularly, and while some of them get their requests approved others do not. Therefore, our project is designed to assist applicants that suffer from bad credit or the inability to gain a credit card understand what variables impact credit card approval process. In addition, we aim to compare different analytical models and understand which one performs best in deciphering how a company approves or denies individuals applications for credit cards. Specifically, our analysis will provide answers to the following questions:

- What is the main predictor on which a person would be approved or denied a credit card?
- Which model performs best in classifying an application as "approved" or "denied"?
- How important is credit score regarding approval odds, relative to other variables such as hold balance, region, etc.

The dataset we have selected for our analysis was found on Kaggle and it is structured to contain 10000 observations. These observations correspond to anonymized candidates' requests for credit cards. Each candidate's request is analyzed through a set of variables which are 12 in total. 11 of the variables examine various information about candidates such as credit score, income, hold balance, demographic, etc., whereas a single variable encompasses the final result which provides the answer as to whether the request for credit cards was approved or rejected.

Our project was organized as follows: collection of data, cleaning, analyzing and reporting. After performing a thorough exploratory analysis and meeting with the professor, we decided to compare different models in order to identify the one that presents more accuracy.

Methodology

Several models were used to analyze the factors and reasoning behind the result of the applicants for credit cards. Our dataset contained calculated fields that created a near perfect logistic model thus this spawned the need to test different models. This gave us an advantage point to spend the time comparing models instead of refining a single model.

We first cleansed the dataset using Python before importing it in R for analysis. First, we needed to combine the two files from the Kaggle link and remove duplicate rows. We dropped unwanted columns, took out nulls, renamed columns and exported a csv to be used in R. Within R, we decided to use the following packages to help us with our analysis: ggplot2 for graphing, dplyr for data manipulation, class for the KNN function, caret for the confusion matrix function, pROC for ROC graph, gridExtra for grouping the plots in one page, reshape2 was used to "melt" the correlation matrix, cowplot, randomForest and party for random forests model.

The first model used in our analysis was logistic regression. This model predicts a dependent variable by studying the connection between one or more independent variable(s). This model is easy to interpret, fast and the industry is very familiar and comfortable with it. However, it does not handle large number of categorical predictors well. To create our logistic regression model, we decided to use a backwards propagation approach. The benchmark used was the Pr(>|z|) of the predictors and if it fell below the limit of 0.05. The predictors that were included in the model were: demographics, est_income, hold_balance, preferred_cust_prob, and credit_score. Below is a code sample that shows the logistic regression model creation.

```
logisticRegressionModel <- glm(approved ~. -risk_score -imp_crediteval -axio_score -count ,
                               family = binomial,data=trainCredit)

logRegression_TestCreditResults <- mutate(testCredit,
                                  approvalPrediction = round(predict(logisticRegressionModel,testCredit,type="response"),digits = 0),
                                  approvalProbability = predict(logisticRegressionModel,testCredit,type="response"))
```

The k-nearest neighbors (KNN) algorithm is used for classification and regression models. The inputs are all the training observations, and using a calculation, finds the k closest neighbors with the output depending on the most "votes". KNN is useful when dealing with nonlinear data and is a simple algorithm to understand and implement. However, it is a slow algorithm that relies on the user to select the best k value. KNN was the trickiest model to implement for selecting neighbors. Eventually all predictors, besides demographics (nonnumeric), was used as the Euclidean distance was used to find the closets neighbors. Once the predictors were selected it was time to determine the best k value for optimal results. Several values were tested including the sqrt of the number of records (k=89) and smaller values. Ultimately a k value of 10 was decided upon based on the performance in terms of speed and accuracy.

```
#KNN mode only accepts numeric values. Here we are making new train and test data sets with numeric columns only!
knnTrainCredit <- dplyr::select(trainCredit,est_income:approved)
knnTestCredit <- dplyr::select(testCredit,est_income:approved)
#build the knn model
knnModel <- knn(train=knnTrainCredit,test=knnTestCredit,cl=knnTrainCredit$approved,k=15)
knn
knnTestPlusModel <- data.frame(knnTestCredit$approved,knnModel)

knnResults <- table(knnTestCredit$approved,knnModel)

#Print the confusionmatrix. We see 85% accuracy
print(confusionMatrix(knnResults))
```

Linear Discriminant Analysis (LDA) was also used in our project. The model allows a set of observations to be characterized into two or more classes and each observation receives a score for how well it fits in each class. LDA is a great model by being easy to implement and results in fast classification. A downside is that the training time required is long and that the inputs for the

model involves complex matrices. LDA was an easy model to build with the only predictor not included is demographic as it is nonnumeric. The model performed well with all these predictors thus no need to remove predictors for performance improvements.

```
creditLDA <- lda(approved~ ., trainCredit[1:9])
ldaPredictions <- predict(creditLDA, testCredit)
```

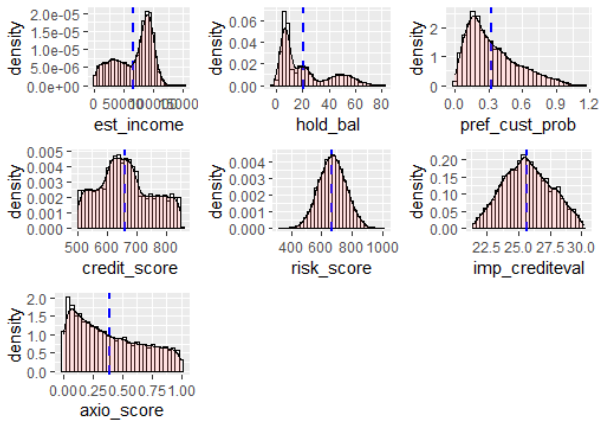
The random forest method is a learning algorithm that randomly creates and merges multiple decision trees into one forest or group. The goal for this method is to collect all the decisions to improve accuracy and not depend on one decision tree. The model is easy to explain and understand and is useful since the classifications are highly accurate. For random forest all predictors were used since it did not matter if they were numeric or not. The decision to decide on number of trees was based on performance and accuracy. With 100 trees we were able to achieve almost 100% but took a long time to compute. Instead a value of 10 for the number of trees was used since it had a high accuracy but was also quick to compute.

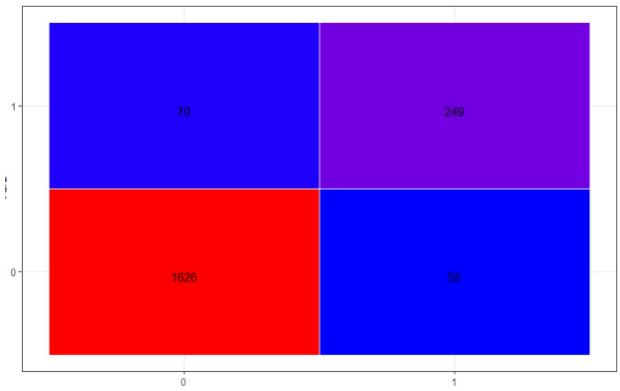
```
randomForest_Credit <- randomForest(approved ~., data=trainCredit,ntree=10,proximity=TRUE)
randomForest_Credit$importance

rfPredictions <-predict(randomForest_Credit,testCredit)
rfPredictionsDF <-data.frame(rfPredictions,testCredit$approved)
```

The goal of the experiments was to find the best model which is why confusion matrices and ROC/AUC curves were used to measure performance. The models were developed with the intent to make these as best as possible on our dataset which was easier for most due to the calculated predictors. This involved predictor selection and other variable manipulation such as values for *k* and *ntree*. Cross validation was used to split our data into test and train datasets. Because of this we were able to create confusion matrices from the predicted and actual values, allowing us to gain the accuracy of the models. The ROC and AUC curves also depicted which models were correctly classifying the testing observations. We took our analysis a step further and decided to check which predictors have the most effect of the approval decision based on our top two performing models.

```
roc(logRegression_TestCreditResults$approved,logRegression_TestCreditResults$approvalProbability, plot = TRUE, legacy.axes=TRUE, percent = TRUE,
    xlab = "False Positive Percentage", ylab="True Positive Percentage",
    col="#377eb8",lwd=4, print.auc=TRUE, print.auc.x=90, print.auc.y = 80,
    auc.polygon=TRUE, auc.polygon.col="#377eb822")


#PLOT LDA
#We see that the AUC for LDA = 97.8%
plot.roc(testCredit$approved, as.numeric(ldaPredictions$x), percent=TRUE,col="#4daf4a",lwd=4,
         print.auc=TRUE, add=TRUE, print.auc.x=90, print.auc.y=70)

#PLOT RANDOM FOREST
plot.roc(testCredit$approved,rfPredictionsDF$rfPredictions, percent=TRUE, col="red",lwd=4,
         print.auc=TRUE, add=TRUE,print.auc.x=90, print.auc.y = 90)


#PLOT KNN
plot.roc(knnTestCredit$approved,as.numeric(knnModel), percent=TRUE, col="orange",lwd=4,
         print.auc=TRUE, add=TRUE, print.auc.x=90,print.auc.y=60)
```

Results

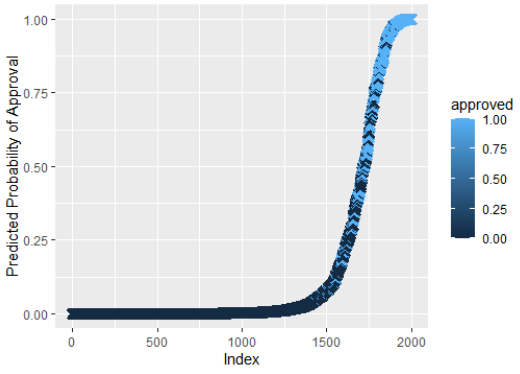




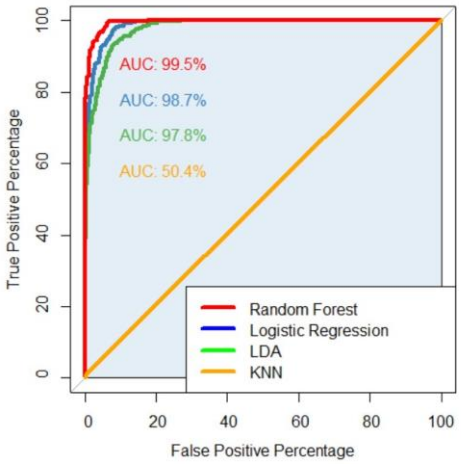
The mean for each predictor is as followed:
LDA total predictions with the misclassified values in blue and the correctly classified values in red and purple.

> mean(credit$est_income) [1] 65853.36
> mean(credit$hold_bal) [1] 20.96262
> mean(credit$pref_cust_prob) [1] 0.3294195
> mean(credit$credit_score) [1] 662.5488
> mean(credit$risk_score) [1] 670.0429
> mean(credit$imp_crediteval) [1] 25.69216
> mean(credit$axio_score) [1] 0.3932109

Logistic regression Model: LDA Model:



ROC/AUC curves for the models:



The predictors that were most important for our top performing models, random forest and logistic regression

```
> randomForest_Credit$importance
                   IncNodePurity
demographic_slice       47.44311
est_income             146.07554
hold_bal               118.56323
pref_cust_prob         594.75977
credit_score            28.12790
risk_score              25.56372
imp_crediteval          25.71585
axio_score              21.76777
```

```
> varImp(logisticRegressionModel, scale=FALSE)
                            Overall
demographic_sliceBWEsk45   4.121099
demographic_sliceCARDIF2   1.468594
demographic_sliceDERS3w5   3.702079
est_income                22.073558
hold_bal                   8.741273
pref_cust_prob            30.740550
credit_score               7.231638
```

Conclusion

Our project consisted of a thorough application and comparison of different statistical techniques and models to determine which one performs best in classifying credit cards' applications approvals or denials decision from a financial institution/company. Our focus was also on establishing the main predictor on which a person would be approved or denied a credit card.

By comparing different models and using confusion matrices and ROC and AUC curves, we were able to verify that random forest is the most accurate model among the one studied for classifying a candidate's request for credit card as "approved" or "non-approved". In fact, the random forest model was able to yield accuracy score of 99.4%, which is significantly high whereas the other models' outcomes are as follows: LR = 95.2%, LDA= 93.8%, KNN = 83.7%. The random forest was especially useful to measure the attribute importance for the given data set.

In addition, based on our top performing models, which are logistic regression and random forest, we were able to identify which predictors impact majorly the approval decision for a credit card. These are respectively preferred customer probability, estimated income, hold balance, demographic and credit score. This result was surprisingly unexpected and reverse a common idea that credit score is not the most relevant among the predictors to determine an applicant's eligibility for a credit card.

Working on this project has been challenging for several reasons. First, we were unable to meet in person as we did the previous weeks due the coronavirus outbreak in Michigan and the consequent closure of schools, libraries and businesses. Second, the different school and work schedules of each of the team members and the consequent need to rearrange them due the lockdown imposed in the state, required us additional efforts to be able to virtually meet. Lastly, after meeting with the professor, we were suggested to provide answers to additional questions for the project and that prompted the need to add further researching and coding to be able to find the model that yield the highest accuracy score.

However, we were able to overcome all the above difficulties and deliver a very well elaborated project that provided us with insight on factors that impact credit card eligibility and generally we gained knowledge on credit card prediction. Though, most importantly, we were able to apply the knowledge acquired during the Data Science and Analytics course to answer relevant question regarding credit card prediction.

References

milesh1, wanshun1. “Receiver Operating Characteristic (ROC) Curve in R.” Kaggle, Kaggle, 6 Aug. 2018, www.kaggle.com/milesh1/receiver-operating-characteristic-roc-curve-in-r

“K Nearest Neighbor.” *RPubs*, 11 July 2018, https://rpubs.com/awanindra01/knn

“ROC.” *Function | R Documentation*, www.rdocumentation.org/packages/pROC/versions/1.16.1/topics/roc

“Plot ROC.” *Function | R Documentation*, www.rdocumentation.org/packages/pROC/versions/1.16.1/topics/plot.roc

W, Amar. “Credit Card Eligibility.” *Kaggle*, 26 Dec. 2018, www.kaggle.com/amarvw/customercreditcard

Wickham, Hadley. “Dplyr v0.7.8.” *Dplyr Package | R Documentation*, www.rdocumentation.org/packages/dplyr/versions/0.7.8

“Create Elegant Data Visualisations Using the Grammar of Graphics.” *Create Elegant Data Visualisations Using the Grammar of Graphics • ggplot2*, https://ggplot2.tidyverse.org/

“Knn Function.” *Function | R Documentation*, www.rdocumentation.org/packages/class/versions/7.3-16/topics/knn

Robin, Xavier. “Package ‘pROC”, 19 March 2020. https://cran.r-project.org/web/packages/pROC/pROC.pdf

Breiman, Leo. “Package ‘randomForest”. 22 March 2018. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

Appendix

Project Dataset Definitions

Column Name	Column Description
demographic_slice	Demographic of the customers
est_income	Estimated income for each customer
hold_bal	Hold balance a customer has on their accounts
pref_cust_prob	Preferred customer probability
credit_score	Customers credit score
risk_score	A threshold scores for approving credits
imp_crediteval	Credit evaluation
axio_score	Acceptance score
approved	Approved for credit or not