Final Project

Garrett Marshall
Northern Arizona University

STA 478
Dr. Benjamin Lucas
December 9, 2022

Introduction

The topic I chose to study for my final project is banking. Specifically, analyzing whether or not a customer will churn (stop using the bank). The data set I will be analyzing is called "Bank Customers Churn" from Kaggle (https://www.kaggle.com/datasets/adammaus/predicting-churn-for-bank-customers?datasetId=66163&language=R). I chose to study this topic simply because I found it the most interesting while searching for data sets on Kaggle. Bank churning analysis is very important to the banking industry. This is because it is much more costly for banks to obtain new customers than it is to keep existing customers. Making it very important to keep your customers happy and using your business. Determining whether a customer will churn can be a difficult task as their are many variables to consider in a bank and client relationship. In this project I will use the found data to analyze, create and explore hypothesis's, create visualizations, and build predictive models to better understand why customers churn.

Data Evaluation

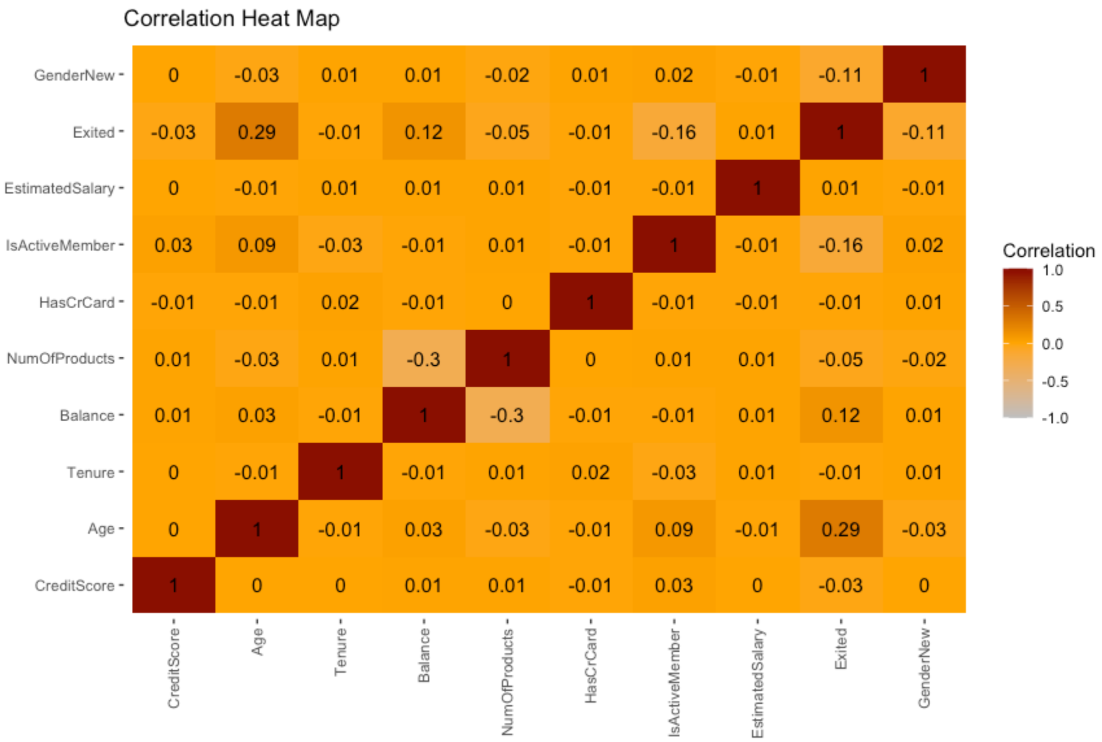
This data set contains 10,000 observations and 11 variables. The majority of the data is numerical except for the 'Geography' variable. There are quite a few binary variables like 'Gender', 'HasCrCard', 'IsActiveMember', and 'Exited'. All of the variables were pretty self explainatory except for 'NumOfProducts', but with some digging I found out that this just means the number of accounts or the other products the customers have with the bank like fixed deposits, car insurance, or home loans. Below is a breakdown of the variables in the data set.

4 categorical variables: Geography, Gender, HasCrCard, and IsActiveMember.

6 continuous variables: CreditScore, Age, Tenure, Balance, NumOfProducts, EstimatedSalary.

Target variable: Exited (Binary).

Fortunately, the data is already pretty clean. One thing I had to do was remove irrelevant columns as they will not provide any important information in my analysis. These columns are 'RowNumber', 'CustomerId', and 'Surname'. I also made the 'Gender' variable numerical so it would be easier to work with. All data cleaning can be seen in chunk 2.1 in the appendix. After cleaning the data, I wanted to see the relationships between the variables. I decided to use a correlation heat map for this.



Looking at the heat map above we can conclude that none of the variables are dependent on one another. The most correlated variables are 'NumOfProducts' and 'Balance', but not significant whatsoever (Appendix 2.2). I decided to make a couple of tables to learn more about key variables.

Country	Exited	No.Exit
France	810	4204
Germany	814	1695
Spain	413	2064

This table shows that Germany has the most churns and Spain has the least. France makes up the most of customers who did not churn (Appendix 2.3).

Gender	Exited	No.Exit
Female	1139	3404
Male	898	4559

This table shows that females are more likely to churn than males (Appendix 2.4).

Has.Credit.Card	Exited	No.Exit
Yes	1424	5631
No	613	2332

This table shows that customers with a credit card are more likely to churn than customers without one (Appendix 2.5).

Is.Active	Exited	No.Exit
Yes	735	4416
No	1302	3547

This table shows that more active customers are less likely to churn than customers who are less active (Appendix 2.5).

Modeling Introduction

The modeling methods I will be using are logistic regression and k-nearest neighbors. According to researchers, "best results are obtained if we use 20-30% of the data for testing, and the remaining 70-80% of the data for training" (Gholamy Et al., 2018). Given this information, I have decided to use 25% of the data for testing and the remaining 75% for training (Appendix 3.1).

The first modeling method I applied was logistic regression. According to the class notes, logistic regression is used when we want to model a binary variable, which is what I am trying to do (Lucas, 2022). Logistic regression does this by applying the logistic function to linear regression, resulting in the output being bound to the interval (1,0). The first GLM (generalized linear model) I fit predicting 'Exited' I used all available parameters (Appendix 3.2). This model produced an AIC of 6532.3. I wanted to minimize the AIC so I tried removing some of the less significant parameters. I created a new GLM except with the parameters 'Tenure', 'NumOfProducts', 'HasCrCard', and 'EstimatedSalary'. The 'Tenure' variable being deemed as insignificant by my model did surprise be. I expected the length of the relationship between the customer and the bank to be a fairly good indicator of whether or not a customer will churn. The second GLM I produced did slightly reduce the AIC to 6527.2 (Appendix 3.3).

The second modeling method I applied was k-nearest neighbors (k-NN). According to bio-statistician Leif E. Peterson, k-NN is a simple classification method useful when there is little knowledge about the distribution of the data (Peterson, 2009). My one concern about using this method is its inferior efficiency when dealing with large data sets, considering my data set contains 10,000 observations. In k-NN, the output is a function of the k most similar examples in the training set, in the case of classification it is the majority class. I applied the k-NN algorithm, with $k = 10$, using the function 'knn()' in the 'class' package (Appendix 3.4).

I decided I was going to create another logistic regression model using the two predictors I think are the most influential. This model will be used in a hypothesis test to determine if it performs better than the null model. The two predictors I chose were 'CreditScore' and 'IsActiveMember' (Appendix 3.5). I chose 'CreditScore' as the first predictor because I think that it is a good indication of financial responsibility. My logic being that customers who are more financially responsible are less likely to change up their banking situation and vice-versa. I chose 'IsActiveMember' as the second predictor because I think that the activity of customers is a good indication of their satisfaction of the bank.

Analysis Results

The logistic regression model performed the best out of the two methods (Appendix 4.1). See the confusion matrix below.

	0	1
0	1923	426
1	59	92

This model has a accuracy of 80.6%, meaning it was able to accurately predict whether or not someone would churn 80.6% of the time.

The predictions produced by the k-NN algorithm did not perform as well as the logistic regression but it still did pretty well (Appendix 4.2). See the confusion matrix below.

	0	1
0	1932	496
1	50	22

k-NN has an accuracy of 78.2%, meaning it was able to accurately predict whether or not someone would churn 78.2% of the time.

Discussion of Final Models and Analysis

There are many reasons as to why logistic regression out-performed k-NN in my application. The first being the fact that logistic regression is a probabilistic model, meaning it can provide a measure of uncertainty when making predictions. This can be useful when making decisions based on the model's output, as it can help determine how much weight to give to the model's predictions. Secondly, logistic regression can be regularized, meaning it can be adjusted to prevent over fitting. Over fitting occurs when a model fits the noise or randomness of the data rather than the underlying trend, and it can lead to poor performance on new or unseen testing data. In contrary, k-NN is less efficient as the algorithm does not build a model until it is asked to make a prediction. This means that with every new observation, the algorithm must go through the entire training data to find the nearest neighbors, which can be very computationally expensive.

I conducted a hypothesis test regarding my logistic regression model with 'CreditScore' and 'IsActiveMember' as the predictors. The null hypothesis is that the two predictors are insignificant, $H_o\colon \beta_1 = \beta_2 = 0$. The alternative hypothesis is that the two predictors are significant and not equal to the null or zero, $H_a\colon \beta_1 \neq 0, \beta_2 \neq 0$. I calculated the p-value from the chi-squared statistic and got a p-value of ≈ 0 (Appendix 5.1). At a significance level of $\alpha = 0.05$ and with a p-value ≈ 0, I reject the null hypothesis in favor of the alternative. There is a statistically significant relationship between the combination of a customers credit score and whether they are active or not in predicting if they will churn.

Conclusion

In conclusion, my project showed that using logistic regression and K-nearest neighbors can be effective in predicting a bank's churn rate. My results indicated that these methods can accurately classify customers as either likely to churn or not likely to churn. This information can be valuable for banks in developing targeted retention strategies to prevent customer churn and improve their bottom line. Overall, this study demonstrates the usefulness of these modeling methods in predicting churn in the banking industry.

All in all, I found this project challenging but very interesting. I feel that I learned a whole lot by applying what I have learned in this statistics course as well as my previous courses to this study. In the future I think it would be worth it to apply a random forest to the data set, as that is a method I found intriguing and would like to learn more about in application.

Citations

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: a pedagogical explanation.

Lucas, B. (2022) Module 3: Overview of Statistical Learning [PowerPoint slides] Retrieved from https://bblearn.nau.edu/ultra/courses/_285455_1/cl/outline

Lucas, B. (2022) Module 4: Classification Methods [PowerPoint slides] Retrieved from https://bblearn.nau.edu/ultra/courses/_285455_1/cl/outline

Peterson, L. E. (2009). K-Nearest Neighbor. Scholarpedia. Retrieved December 8, 2022, from http://www.scholarpedia.org/article/K-nearest_neighbor

Appendix

```
# import data
churn <- read_csv("Churn_Modeling.csv")
```

Section 2

```
# 2.1
# cleaning the data

# make gender numerical
churn$GenderNew <- if_else( churn$Gender == "Male", 1, 0 )

# remove irrelevant columns
churn <- churn %>% dplyr::select( c(-'RowNumber', -'CustomerId', -'Surname',
-'Gender') )

# 2.2
# create a correlation heat map

churn1 <- churn %>% dplyr::select( c(-'Geography') )
corr <- round( cor(churn1), 2 )
melted_corr <- melt(corr)
# head(melted_corr)

heat_map <- ggplot(data = melted_corr, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value), size = 4) +
  scale_fill_gradient2(low = "grey", high = "darkred",
                       limit = c(-1,1), name="Correlation", mid = "orange") +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        panel.background = element_blank()) +
  labs( title = "Correlation Heat Map") +
  scale_x_discrete(guide = guide_axis(angle = 90))

# 2.3
churn2 <- churn %>%
  group_by(Geography, Exited) %>% summarise(n = n())

## `summarise()` has grouped output by 'Geography'. You can override using th
e
## `.groups` argument.

churn2

## # A tibble: 6 × 3
## # Groups:   Geography [3]
##   Geography Exited     n
##   <chr>      <dbl> <int>
## 1 France         0  4204
## 2 France         1   810
## 3 Germany        0  1695
## 4 Germany        1   814
## 5 Spain          0  2064
## 6 Spain          1   413
```

```
geo_table <- data.frame( Country = c("France", "Germany", "Spain"),
                        Exited = c(810, 814, 413),
                        No.Exit = c(4204, 1695, 2064) )
#geo_table

# 2.4
churn2 <- churn %>%
  group_by(GenderNew, Exited) %>% summarise(n = n())
```

```
## `summarise()` has grouped output by 'GenderNew'. You can override using th
e
## `.groups` argument.
```

```
churn2
```

```
## # A tibble: 4 × 3
## # Groups:   GenderNew [2]
##   GenderNew Exited     n
##       <dbl>  <dbl> <int>
## 1         0      0  3404
## 2         0      1  1139
## 3         1      0  4559
## 4         1      1   898
```

```
gender_table <- data.frame( Gender = c("Female", "Male"),
                        Exited = c(1139, 898),
                        No.Exit = c(3404, 4559) )
#gender_table

# 2.5
churn2 <- churn %>%
  group_by(HasCrCard, Exited) %>% summarise(n = n())
```

```
## `summarise()` has grouped output by 'HasCrCard'. You can override using th
e
## `.groups` argument.
```

```
churn2
```

```
## # A tibble: 4 × 3
## # Groups:   HasCrCard [2]
##   HasCrCard Exited     n
##       <dbl>  <dbl> <int>
## 1         0      0  2332
## 2         0      1   613
## 3         1      0  5631
## 4         1      1  1424
```

```
card_table <- data.frame( Has.Credit.Card = c("Yes", "No"),
                        Exited = c(1424, 613),
                        No.Exit = c(5631, 2332) )
#card_table

# 2.6
churn2 <- churn %>%
  group_by(IsActiveMember, Exited) %>% summarise(n = n())
```

```
## `summarise()` has grouped output by 'IsActiveMember'. You can override usi
ng
## the `.groups` argument.
```

```
churn2
```

```
## # A tibble: 4 × 3
## # Groups:   IsActiveMember [2]
##   IsActiveMember Exited     n
##            <dbl>  <dbl> <int>
## 1              0      0  3547
## 2              0      1  1302
## 3              1      0  4416
## 4              1      1   735
```

```
active_table <- data.frame( Is.Active = c("Yes", "No"),
                        Exited = c(735, 1302),
                        No.Exit = c(4416, 3547) )
#active_table
```

Section 3

```
# 3.1
# splitting the data into training and testing
churn$Geography <- as.factor(churn$Geography)
churn$Geography <- as.numeric(churn$Geography)

# split data
size <- floor( .75 * nrow(churn) )
set.seed(456)
index_split <- sample( seq_len(nrow(churn)), size = size )
churn.train <- churn[index_split, ]
churn.test <- churn[-index_split, ]

#head(churn)

# 3.2
# logistic regression
lr.model.1 <- glm( Exited ~ ., data = churn.train, family = binomial )
summary(lr.model.1)
```

```
##
## Call:
## glm(formula = Exited ~ ., family = binomial, data = churn.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1280  -0.6684  -0.4648  -0.2788   2.8971
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -3.592e+00  2.862e-01 -12.553   <2e-16 ***
## CreditScore     -6.719e-04  3.230e-04  -2.080   0.0375 *
## Geography        7.906e-02  3.822e-02   2.069   0.0386 *
## Age              7.256e-02  2.948e-03  24.616   <2e-16 ***
## Tenure          -9.216e-03  1.077e-02  -0.856   0.3920
## Balance          4.718e-06  5.298e-07   8.905   <2e-16 ***
## NumOfProducts   -3.103e-02  5.408e-02  -0.574   0.5661
## HasCrCard       -5.203e-02  6.754e-02  -0.770   0.4411
## IsActiveMember  -1.091e+00  6.638e-02 -16.433   <2e-16 ***
## EstimatedSalary  6.023e-07  5.420e-07   1.111   0.2664
## GenderNew       -5.497e-01  6.245e-02  -8.803   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7558.4  on 7499  degrees of freedom
## Residual deviance: 6510.3  on 7489  degrees of freedom
## AIC: 6532.3
##
## Number of Fisher Scoring iterations: 5
```

```
# 3.3
# second logistic regression
lr.model.2 <- glm( Exited ~ CreditScore + Geography + Age + Balance + IsActiv
eMember + GenderNew, data = churn.train, family = binomial )
summary(lr.model.2)
```

```
##
## Call:
## glm(formula = Exited ~ CreditScore + Geography + Age + Balance +
##     IsActiveMember + GenderNew, family = binomial, data = churn.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1497  -0.6683  -0.4673  -0.2802   2.9030
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.665e+00  2.556e-01 -14.338   <2e-16 ***
## CreditScore    -6.775e-04  3.230e-04  -2.098   0.0359 *
## Geography       7.835e-02  3.820e-02   2.051   0.0403 *
## Age             7.260e-02  2.945e-03  24.656   <2e-16 ***
## Balance         4.805e-06  5.116e-07   9.393   <2e-16 ***
## IsActiveMember -1.089e+00  6.631e-02 -16.426   <2e-16 ***
## GenderNew      -5.499e-01  6.241e-02  -8.811   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7558.4  on 7499  degrees of freedom
## Residual deviance: 6513.2  on 7493  degrees of freedom
## AIC: 6527.2
##
## Number of Fisher Scoring iterations: 5
```

```
# 3.4
# K-NN method using the knn() function

set.seed(12)
knn_preds <- knn( churn.train, churn.test, as.numeric(churn.train$Exited), k=
10 )

# 3.5
# this model will be used for a hypothesis test against the null model
lr.model.3 <- glm(Exited ~ CreditScore+IsActiveMember, data = churn.train, fa
mily = binomial)
summary(lr.model.3)
```

```
##
## Call:
## glm(formula = Exited ~ CreditScore + IsActiveMember, family = binomial,
##     data = churn.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8370  -0.7780  -0.5581  -0.5340   2.0279
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.6303918  0.1986357  -3.174  0.00151 **
## CreditScore    -0.0005845  0.0003019  -1.936  0.05287 .
## IsActiveMember -0.7919851  0.0595130 -13.308  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7558.4  on 7499  degrees of freedom
## Residual deviance: 7369.3  on 7497  degrees of freedom
## AIC: 7375.3
##
## Number of Fisher Scoring iterations: 4
```

Section 4

```
# 4.1
# Logistic regression results
glm_probs <- predict(lr.model.2, churn.test, type="response")
glm_preds <- rep(0,2500)
glm_preds[glm_probs > .5] = 1
glm_predictions <- table(glm_preds,churn.test$Exited)
glm_predictions

##
## glm_preds    0    1
##         0 1923  426
##         1   59   92

accuracy_glm <- mean(glm_preds == churn.test$Exited)
accuracy_glm

## [1] 0.806

# 4.2
# knn predictions
knn_predictions <- table(knn_preds,churn.test$Exited)
knn_predictions

##
## knn_preds    0    1
##         0 1932  496
##         1   50   22

accuracy_knn <- mean(knn_preds == churn.test$Exited)
accuracy_knn

## [1] 0.7816
```

Section 5

```
# 5.1
# calculating the p-value from the Chi-squared test statistic
1-pchisq(7558.4-7369.3, 7499-7497)

## [1] 0

#1-pchisq(75-73, 74-73)
```