STATS/CSE 780
Assignment 2

Pao Zhu Vivian Hsu (Student Number: 400547994)

2023-10-17

Introduction

The goal of this study is to develop a model that predicts customer churn for a bank and provide suggestions that can help the bank reduce the rate of churn. Two models were built to predict customer churn; the first was built using logistic regression and the second was built using K-nearest neighbour (KNN) classification. A comparison of accuracy rates showed that the KNN model is a stronger fit than the logistic model.

Methods

The data for this study was downloaded from the Kaggle website (n.d.). It consists of 14 variables and 10002 observations from a bank that operates in France, Germany, and Spain. The data set includes a binary indicator for customer churn (called "exited") and a variety of columns describing the customer, such as age, credit score, and estimated salary. The categorical churn variable makes this data suitable for logistic regression and KNN classification.

Prior to any modelling, the data was first screened for missing data, duplicate rows, unexpected data ranges, and data type issues. These issues were addressed through data imputation and transformation. A summary of the data issues is available in Figure 2 in the Supplementary Materials. Additionally, row numbers, customer ids, and surnames were removed from the data set as they are not important for studying customer churn. The remaining 10 variables were used as predictors for classification.

Box plots, bar charts, and a correlation plot were created to visualize trends in the data. Of the six box plots shown in Figure 3, age, balance, and number of products appeared to have the greatest impact on churn. Those who churned tend to be older, have a greater bank balance, and have less products with the bank compared to those who did not churn. The box plots for credit score, age, and number of products also show potential outliers. These outliers were not removed since these customers do not appear to be outliers for many of the other variables. Furthermore, the values for these variables do not appear to be unreasonably large or small from a practical perspective. For the bar charts in Figure 4, customers who churned appeared to be slightly less active and female rather than male compared to those who did not churn. But overall, the bar charts did not appear to show any strong patterns. Finally, the correlation plot in Figure 5 showed that there was low correlation between the continuous variables.

Once the data was cleaned, it was randomly split into two equal parts for training and testing. A logistic regression model with a logit link was the first model that was built. Based on Harrell's 1:15 rule of thumb for a suitable number of predictor variables compared to observations (2015), all 10 variables were included in the model initially. The final model includes only 6 of these variables since 4 were removed with backward selection. A sensitivity and specificity analysis with an ROC curve was performed to choose the optimal cutoff value of 0.21 for interpreting the predictions. A summary of the final model and the ROC curve can be found in Figure 6 and Figure 7 of the Supplementary Materials respectively. Next, KNN classification was used to create another model to predict customer churn. K-fold cross validation was used to determine a neighbourhood size of k=7. Both models were tested using the hold-out set.

Results

The most important variable in the logistic model is the customer's active status in the bank. Its regression coefficient is -1.09, which suggests that active members are exp(-1.09)=66.32% less likely to churn compared to inactive members. The most important variable cannot be determined from a KNN classification model because the algorithm does not indicate the importance of variables unlike regression.

Figure 1 below shows a comparison of the two models after being tested with the hold-out set. The sensitivity is over 20% higher for the KNN model than the logistic regression model. On the other hand, the specificity is about 5% lower for the KNN model than the logistic regression model. Overall, the KNN classification model performed better than the logistic regression model since the miss-classification rate is lower and accuracy is higher than logistic regression model.

	logistic_regression	knn
Miss-classification error rate	0.27	0.19
Accuracy	0.73	0.81
Sensitivity	0.42	0.68
Specificity	0.88	0.83

Figure 1: Model performance comparison

Conclusion

- KNN model is better
- important predictors
- ways to apply results to real world
- suggestions for future models and improvement

Supplementary material

Data definitions for categorical variables

Here are the data definitions for categorical variables in the final data set. Definitions for the continuous variables remain the same as the original data source (Meshram (n.d.)).

- Exited: Whether the customer has a churned or not (0 = not churned, 1 = churned)
- Gender: The customer's gender (1 = female, 2 = male)
- Geography: Country where the customer lives (1 = France, 2 = Germany, 3 = Spain)
- Has credit card: Whether the customer has a credit card or not (0 = no, 1 = yes)
- Is active member: Whether the customer is active in the bank or not (0 = no, 1 = yes)

Figures

	data_type	min	max	nulls	blanks
RowNumber	integer	1	10000	0	0
CustomerId	integer	15565701	15815690	0	0
Surname	character	Abazu	Zuyeva	0	0
CreditScore	integer	350	850	0	0
Geography	character		Spain	0	1
Gender	character	Female	Male	0	0
Age	numeric	18	92	1	0
Tenure	integer	0	10	0	0
Balance	numeric	0	250898.09	0	0
NumOfProducts	integer	1	4	0	0
HasCrCard	integer	0	1	1	0
IsActiveMember	integer	0	1	1	0
EstimatedSalary	numeric	11.58	199992.48	0	0
Exited	integer	0	1	0	0

Figure 2: Summary of the original Kaggle data used prior to cleansing

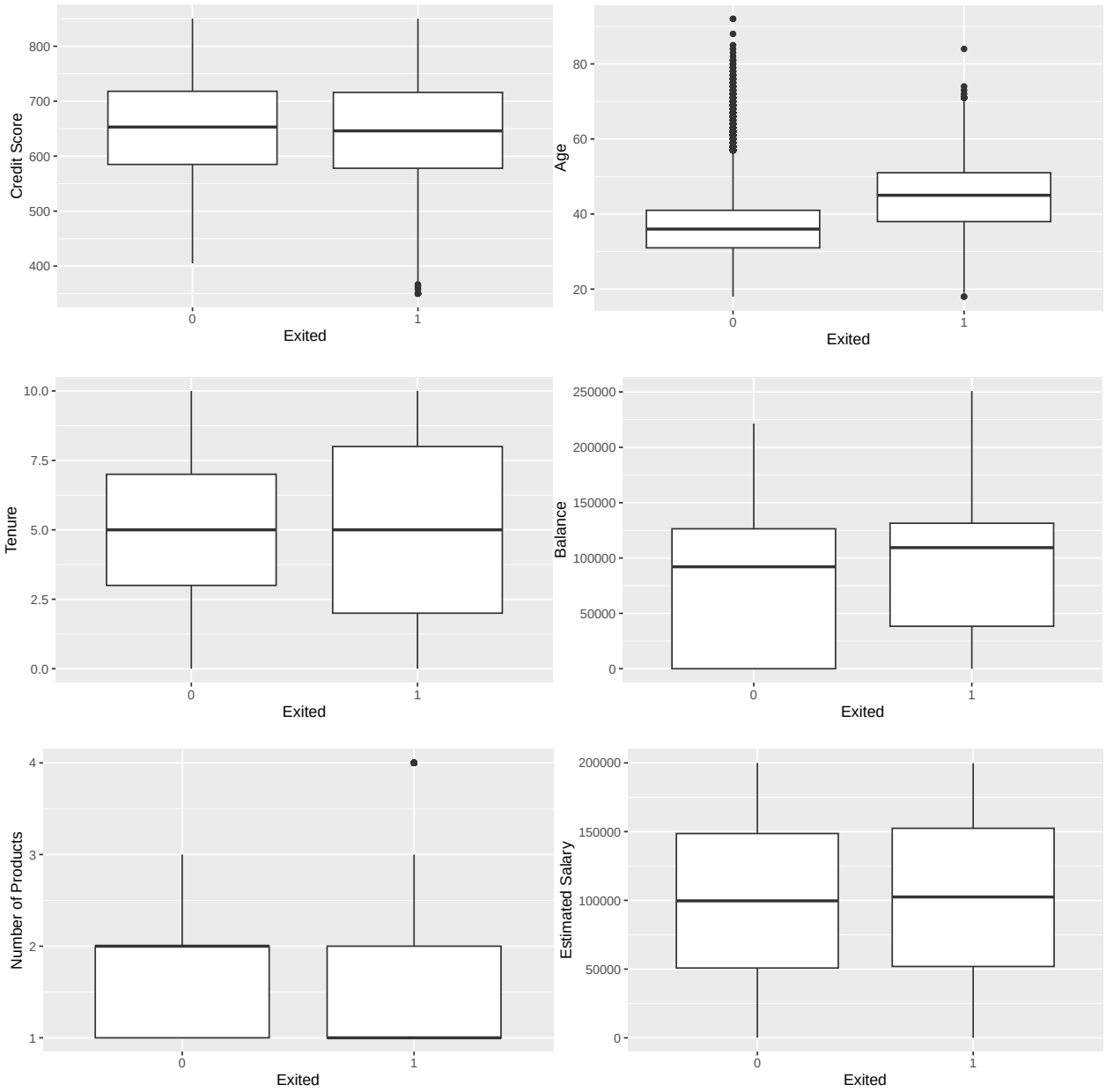

Figure 3: Boxplots of continuous variables

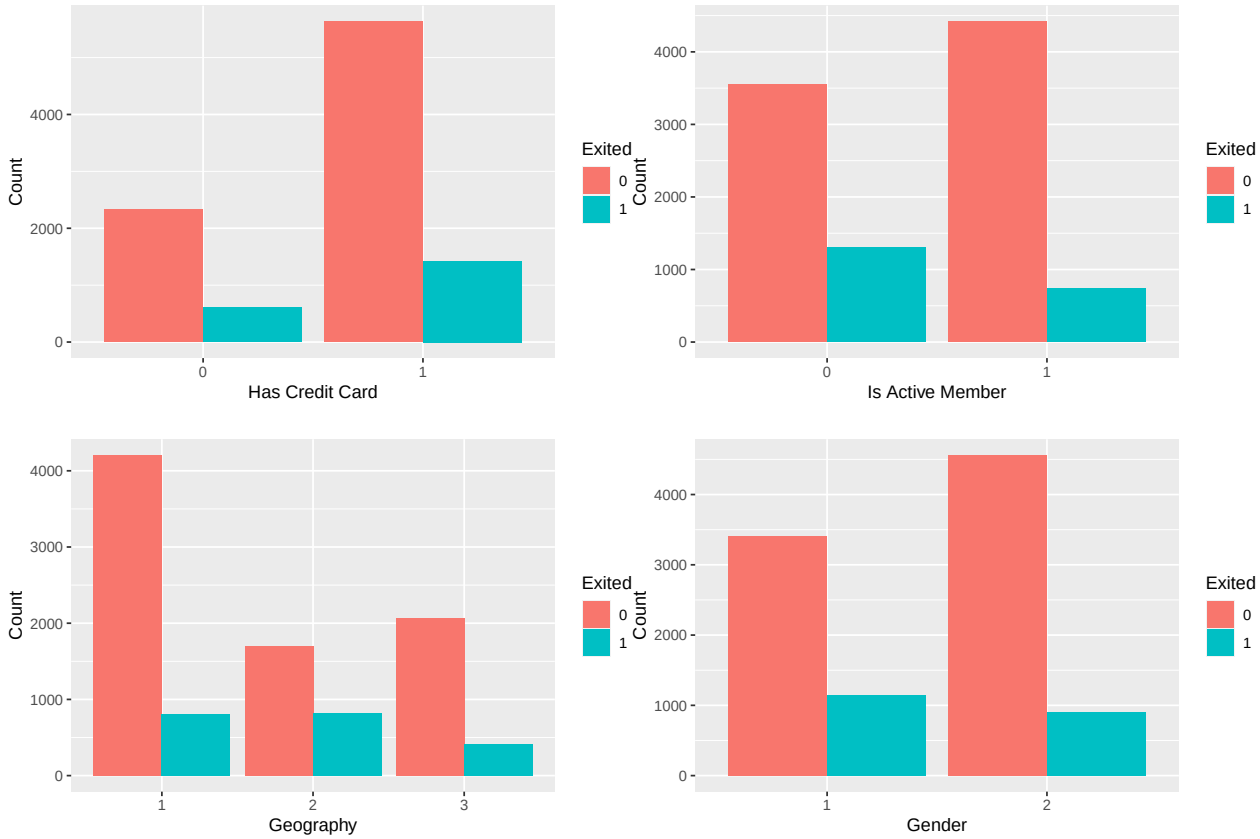

Figure 4: Bar charts of categorical variables

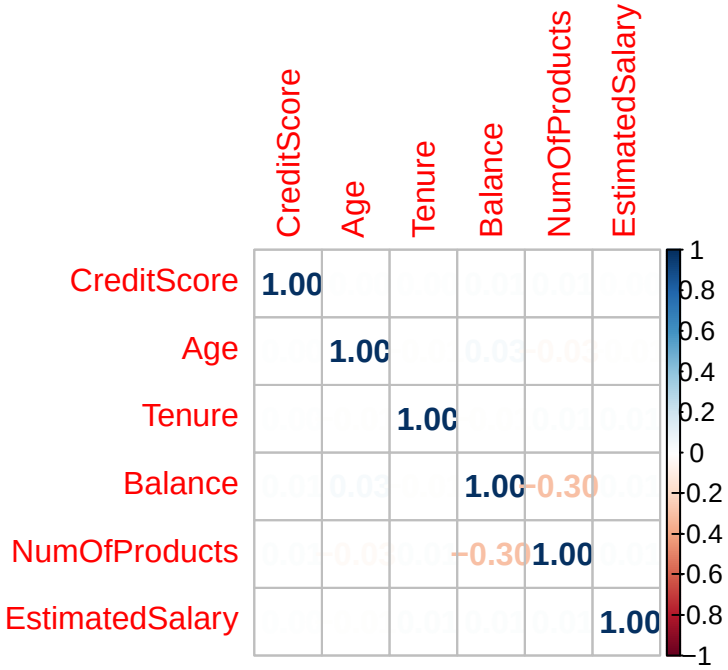

Figure 5: Correlation between continuous variables

```
glm(formula = Exited ~ CreditScore + Gender + Age + Tenure +
    Balance + IsActiveMember, family = binomial("logit"), data = train_data)
```

	estimate	std_error	z_value	pr(>|z|)
(Intercept)	-3.3037	0.3139	-10.5253	0.0000
CreditScore	-0.0009	0.0004	-2.2584	0.0239
Gender2	-0.4126	0.0780	-5.2871	0.0000
Age	0.0701	0.0036	19.2026	0.0000
Tenure	-0.0328	0.0136	-2.4203	0.0155
Balance	0.0000	0.0000	7.9794	0.0000
IsActiveMember1	-1.0882	0.0836	-13.0116	0.0000

Figure 6: Final logistic regression model

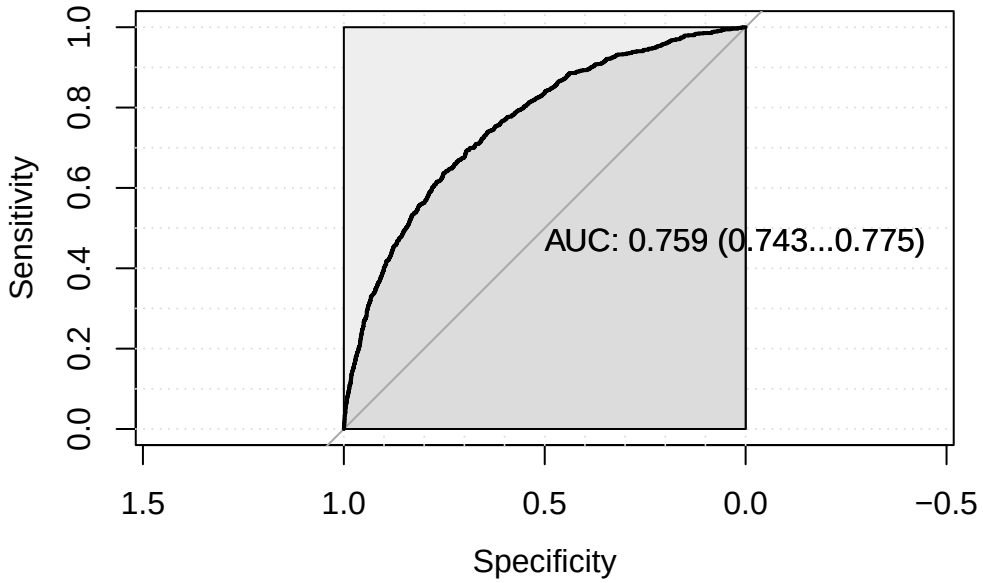

Figure 7: ROC curve

Code

```
# ----- LOAD PACKAGES AND DATA ----- #
packages <- c("knitr", "dplyr", "tidyverse", "DescTools", "ggplot2", "corrplot",
              "class", "pROC", "glmnet", "caret")
lapply(packages, library, character.only = TRUE)
bank_raw <- read.csv("Churn_Modelling.csv")


# ----- DATA EXPLORATION ----- #
# Check data types, min, max, and missing data
data_type <- sapply(bank_raw,class)
min <- sapply(bank_raw, function(col){min(col,na.rm=TRUE)})
max <- sapply(bank_raw, function(col){max(col,na.rm=TRUE)})
nulls <- sapply(bank_raw, function(col){sum(is.na(col))})
blanks <- sapply(bank_raw,
                 function(col){ifelse(is.na(sum(col == "")), 0, sum(col == ""))})
data_summary <- data.frame(row.names = names(nulls), data_type=data_type,
                           min=min, max=max, nulls=nulls, blanks=blanks)
kable(data_summary)

# Check dimensions and reason for numeric Age
dim(bank_raw) # There are duplicate rows
bank_raw$Age[round(bank_raw$Age) != bank_raw$Age] # There are decimals

# ----- CLEANSE DATA ----- #
bank <- bank_raw %>%

  # Remove columns that are not important for analysis
  dplyr::select(-c("RowNumber", "CustomerId", "Surname")) %>%

  # Impute missing values with mean, median, and mode
```

```
  mutate(Age = replace_na(Age, round(mean(Age,na.rm=TRUE),0)),
         HasCrCard = replace_na(HasCrCard, median(HasCrCard,na.rm=TRUE)),
         IsActiveMember = replace_na(IsActiveMember, median(HasCrCard,na.rm=TRUE)),
         Geography = ifelse(Geography == "", Mode(Geography,na.rm=TRUE)[1], Geography)) %>%

  # Change data types
  mutate(Age = as.integer(round(Age)),
         Geography = as.factor(unclass(as.factor(Geography))),
         Gender = as.factor(unclass(as.factor(Gender))),
         HasCrCard = as.factor(HasCrCard),
         IsActiveMember = as.factor(IsActiveMember),
         Exited = as.factor(Exited))

# Remove duplicated rows
bank <- bank[!duplicated(bank), ]

# ----- DATA VISUALIZATION  ----- #
# Create a boxplot for each continuous variable
ggplot(bank, aes(x = Exited, y = CreditScore)) + geom_boxplot() + ylab("Credit Score")
ggplot(bank, aes(x = Exited, y = Age)) + geom_boxplot()
ggplot(bank, aes(x = Exited, y = Tenure)) + geom_boxplot()
ggplot(bank, aes(x = Exited, y = Balance)) + geom_boxplot()
ggplot(bank, aes(x = Exited, y = NumOfProducts)) + geom_boxplot() +
  ylab("Number of Products")
ggplot(bank, aes(x = Exited, y = EstimatedSalary)) + geom_boxplot() +
  ylab("Estimated Salary")

# Create a bar chart for each categorical variable
ggplot(bank, aes(x = HasCrCard, fill = Exited)) + geom_bar(position = "dodge") +
  labs(y = "Count", x = "Has Credit Card")
ggplot(bank, aes(x = IsActiveMember, fill = Exited)) + geom_bar(position = "dodge") +
```

```
  labs(y = "Count", x = "Is Active Member")
ggplot(bank, aes(x = Geography, fill = Exited)) + geom_bar(position = "dodge") +
  labs(y = "Count")
ggplot(bank, aes(x = Gender, fill = Exited)) + geom_bar(position = "dodge") +
  labs(y = "Count")


# Check for correlation between continuous variables
corr_matrix <- cor(bank %>% dplyr::select(-Geography, -Gender, -HasCrCard,
                                          -IsActiveMember, -Exited))
corrplot(round(corr_matrix,2), method = "number")


# ----- SPLIT INTO TRAIN & TEST DATA ----- #
set.seed(2023780)
train_index <- sample(1:nrow(bank), round(nrow(bank)/2, 0), replace = FALSE)

# Training set
train_data <- bank[train_index, ]
train_x <- dplyr::select(train_data, -Exited)
train_y <- dplyr::pull(train_data, Exited)

# Testing set
test_data <- bank[-train_index, ]
test_x <- dplyr::select(test_data, -Exited)
test_y <- dplyr::pull(test_data, Exited)


# ----- LOGISTIC REGRESSION ----- #
set.seed(2023780)
```

```r
# Include all variables as predictors in the initial model
log_mod0 <- glm(Exited ~ ., family = binomial("logit"), data = train_data)
summary(log_mod0)

# Perform backward selection to select most significant predictors
log_mod0 <- update(log_mod0, ~ . -HasCrCard)
summary(log_mod0)
log_mod0 <- update(log_mod0, ~ . -Geography)
summary(log_mod0)
log_mod0 <- update(log_mod0, ~ . -EstimatedSalary)
summary(log_mod0)
log_mod1 <- update(log_mod0, ~ . -NumOfProducts)
summary(log_mod1)

# Final model
print(log_mod1$call)
summary <- data.frame(round(summary(log_mod1)$coefficients, 4))
colnames(summary) <- c("estimate", "std_error", "z_value", "pr(>|z|)")
kable(summary)

# Predict outcome probabilities using test set
log_mod1_y_prob <- predict(log_mod1, newdata = test_data, type = "response")

# To label the outcomes, find the optimal cut-off value using ROC curve
log_mod1_pROC <- roc(test_y, log_mod1_y_prob, smoothed = TRUE, ci=TRUE, ci.alpha=0.9,
                     plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
                     print.auc=TRUE, show.thres=TRUE)
cutoff <- coords(log_mod1_pROC, "best")$threshold

# Assign labels to prediction results using cut-off value
log_mod1_y <- ifelse(log_mod1_y_prob > cutoff, 1, 0)
```

```r
# ----- K-NEAREST NEIGHBOUR CLASSIFICATION ----- #
# Find optimal k value using k-fold cross validation and build the KNN model
set.seed(2023780)
knn_ctrl <- trainControl(method = "repeatedcv", number = 15, repeats = 3)
knn_mod1 <- train(Exited ~ ., data = train_data, method = "knn", trControl=knn_ctrl,
                  preProcess = c("center", "scale"))

# Predict outcome using test set
knn_mod1_y <- predict(knn_mod1, newdata = test_data)



# ----- CLASSIFIER PERFORMANCE ----- #
# Miss-classification error rate: % of churn incorrectly predicted
# Accuracy: % of churn correctly predicted
# Sensitivity: % correctly predicted as churned
# Specificity: % correctly predicted as not churned

# --- LOGISTIC REGRESSION MODEL PERFORMANCE --- #
log_mod1_cmatrix <- table(log_mod1_y, test_y) # Confusion matrix
log_mod1_mcerate <- mean(log_mod1_y != test_y) # Miss-classification error rate
log_mod1_accuracy <- mean(log_mod1_y == test_y) # Accuracy
log_mod1_sensitivity <- log_mod1_cmatrix[2,2]/sum(log_mod1_cmatrix[2,]) # Sensitivity
log_mod1_specificity <- log_mod1_cmatrix[1,1]/sum(log_mod1_cmatrix[1,]) # Specificity
log_mod1_stats <- c(log_mod1_mcerate, log_mod1_accuracy, log_mod1_sensitivity,
                    log_mod1_specificity)

# --- KNN MODEL PERFORMANCE --- #
knn_mod1_cmatrix <- table(knn_mod1_y, test_y) # Confusion matrix
knn_mod1_mcerate <- mean(knn_mod1_y != test_y) # Miss-classification error rate
knn_mod1_accuracy <- mean(knn_mod1_y == test_y) # Accuracy
```

```r
knn_mod1_sensitivity <- knn_mod1_cmatrix[2,2]/sum(knn_mod1_cmatrix[2,]) # Sensitivity
knn_mod1_specificity <- knn_mod1_cmatrix[1,1]/sum(knn_mod1_cmatrix[1,]) # Specificity
knn_mod1_stats <- c(knn_mod1_mcerate, knn_mod1_accuracy, knn_mod1_sensitivity,
                    knn_mod1_specificity)

# --- COMPARE MODELS --- #
comparison <- data.frame(row.names = c("Miss-classification error rate", "Accuracy",
                                       "Sensitivity", "Specificity"),
                         "logistic_regression" = round(log_mod1_stats, 2),
                         "knn" = round(knn_mod1_stats, 2))
kable(comparison)
```

References

Harrell, F. E. (2015). Multivariable modeling strategies. In *Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis* (pp. 63–102). Springer International Publishing. https://doi.org/10.1007/978-3-319-19425-7_4

Meshram, S. (n.d.). *Bank Customer Churn Prediction*. Kaggle. https://www.kaggle.com/datasets/shubhammeshram579/bank-customer-churn-prediction/

Xie, Y., Dervieux, C., & Riederer, E. (2020). *R markdown cookbook*. CRC Press.