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. Logistic regression and K-nearest neighbour (KNN) classification models were built. Both models showed that X is an important predictor of churn. 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 Kaggle, but was originally sourced from an archived data set from the UCI Machine Learning Repository (n.d.). It consists of 14 variables and 10002 observations including a binary indicator of customer churn (i.e. exited status) and a variety of columns describing the customer, such as age, bank balance, and estimated salary. The presence of a categorical churn variable makes the data suitable for logistic regression and KNN classification.

Prior to modelling, the data quality was validated by checking for duplicated rows, missing data, unexpected data ranges, and data type concerns (see Figure 2 in Supplementary Materials). Data was cleansed to remove these issues. Bar charts of categorical variables and box plots of continuous variables were created to highlight differences between customers who have exited compared to those who have not (see Figure 3 and Figure 4 in Supplementary Materials). The boxplot of age versus churn status showed that there were potential outliers in the data in terms of age. These data points were not removed since the maximum age in the data set is 92 does not seem unreasonably large from a practical perspective. 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.

First, a logistic regression model of binomial family with a logit link was built. Based on Harrell's 1:15 rule of thumb for the number of predictor variables compared to observations in genearlized linear models (2015), all 10 variables were included in the model.

Next, KNN classification was used to create another model to predict customer churn. The final model was fit using k=7 nearest neighbours. This value of k was determined using k-fold cross validation.

Results

	knn	logistic_regression
Miss-classification error rate	0.18	0.26
Accuracy	0.82	0.74
Sensitivity	0.71	0.44
Specificity	0.83	0.88

Figure 1: Model performance comparison

Conclusion

Supplementary material

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 original Kaggle data used for data quality validation

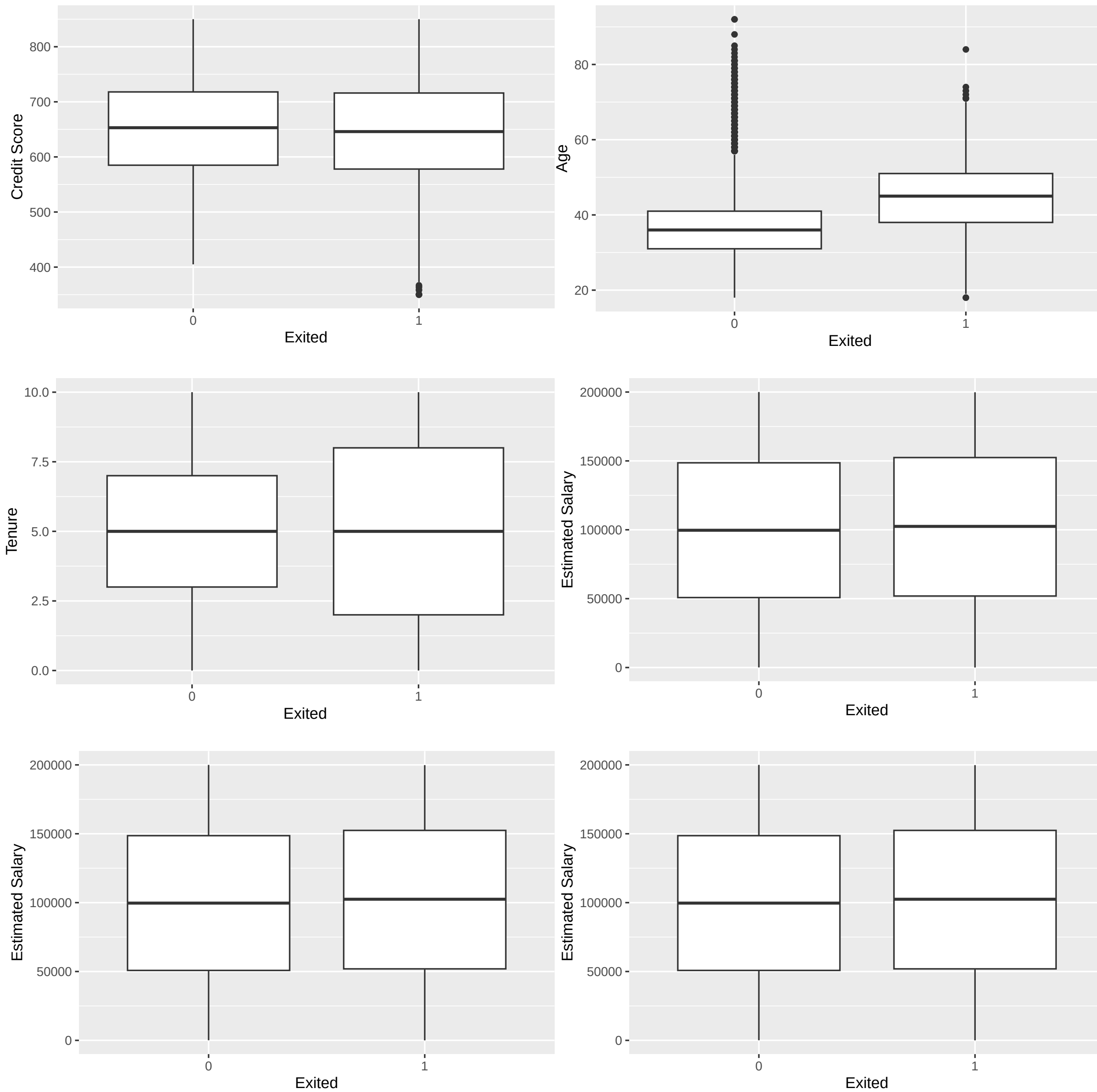

Figure 3: Boxplots of continuous variables

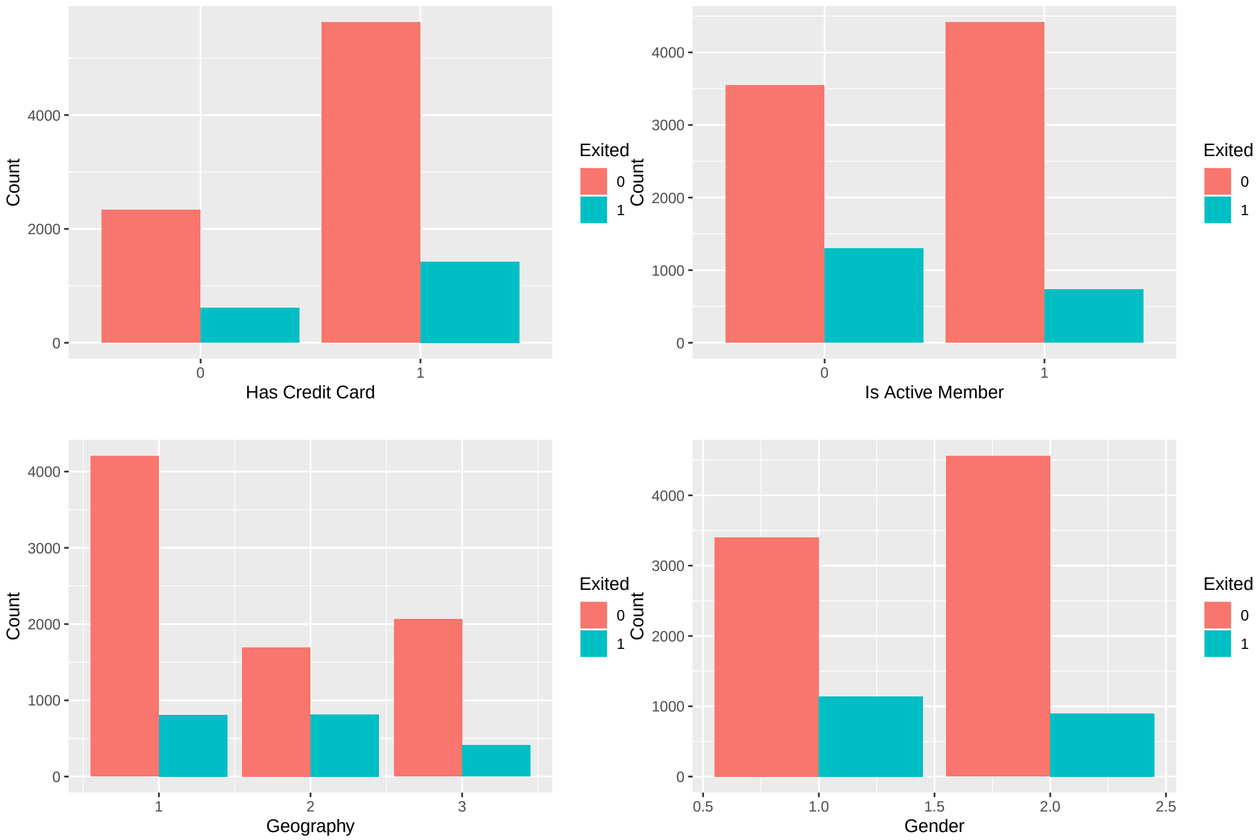

Figure 4: Bar charts of categorical variables

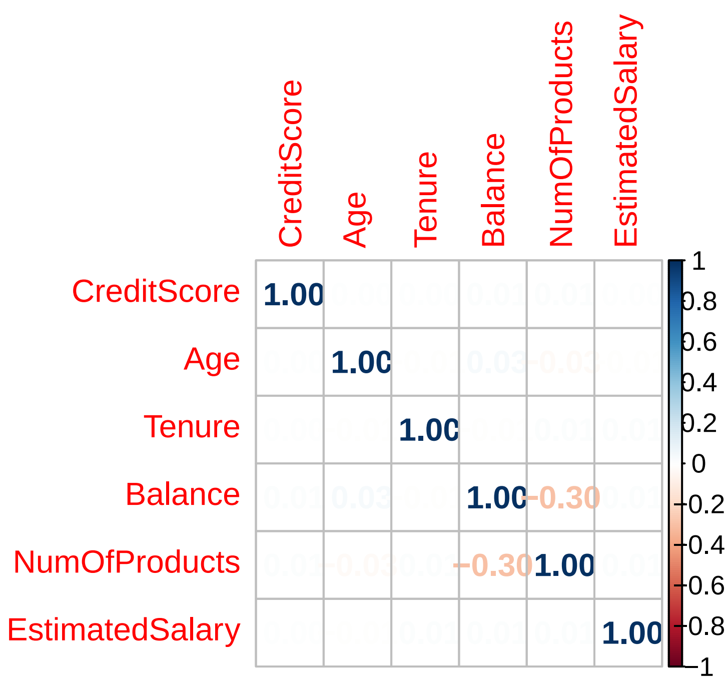

Figure 5: Correlation between continuous variables

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

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -3.638e+00  1.989e-01 -18.292  < 2e-16 ***
Gender          -4.142e-01  7.791e-02  -5.316 1.06e-07 ***
Age              7.006e-02  3.645e-03  19.221  < 2e-16 ***
Balance          5.139e-06  6.405e-07   8.022 1.04e-15 ***
IsActiveMember1 -1.086e+00  8.339e-02 -13.027  < 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: 4809.6  on 4999  degrees of freedom
Residual deviance: 4207.7  on 4995  degrees of freedom
AIC: 4217.7

Number of Fisher Scoring iterations: 5
```

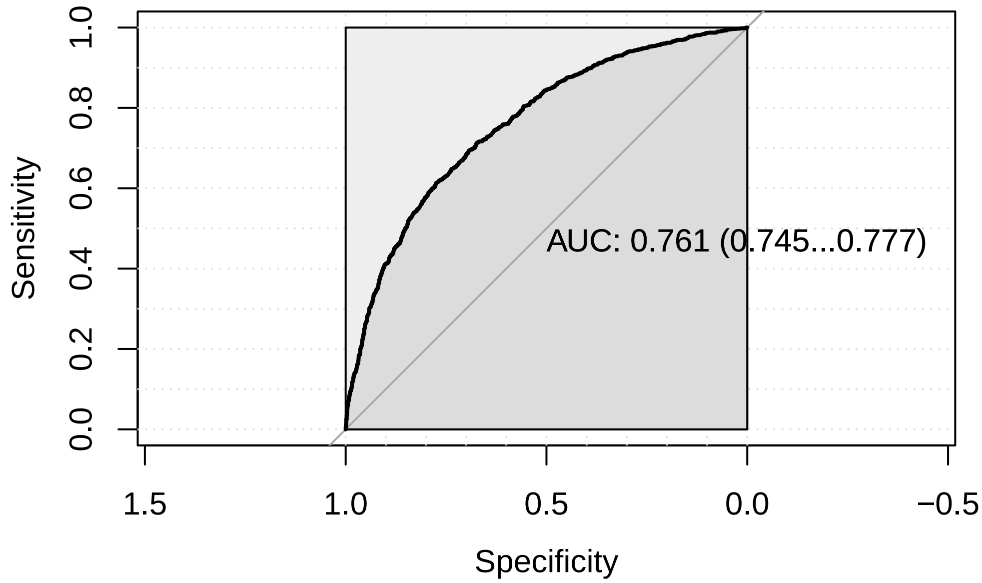

Figure 6: ROC curve

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)
```

```r
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 = unclass(as.factor(Geography)),
         Gender = unclass(as.factor(Gender)),
         HasCrCard = as.factor(HasCrCard),
         IsActiveMember = as.factor(IsActiveMember),
         Exited = as.factor(Exited)
         )

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

```r
# ----- DATA VISUALIZATION  ----- #
# Create bar charts for categorical variables
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")

# 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")



# 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)
```

```r
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)

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

set.seed(2023780)

# Remove predictors with p-values that are not significant (i.e. > 0.05)
log_mod1 <- update(log_mod0, ~ . -CreditScore -Tenure -NumOfProducts
                   -HasCrCard -EstimatedSalary -Geography)
summary(log_mod1)

# 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)

# ----- K-NEAREST NEIGHBOUR CLASSIFICATION ----- #
set.seed(2023780)

# Find optimal k value using k-fold cross validation and build the KNN model
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
```

```r
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
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"),
                         "knn" = round(knn_mod1_stats, 2),
                         "logistic_regression" = round(log_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.