STA 141A Final Project

Credit Card Fraud

2022-06-06

Name	Email
Clayton Chan	clwchan@ucdavis.edu

Instructor: Emanuela Furfaro

STA 141A - Fundamentals of Statistical Data Science

University of California, Davis

Introduction

Credit cards, one of the most common methods of payment in this age. However, like most things, this method of payment is not flawless and can be susceptible to theft and fraudulent transactions. However, the goods news is that there are some characteristics we can analyze within a credit card and recent transactions to flag down these fraudulent transactions. This is exactly the main purpose of the project where we will utilize these features in supervised learning to predict whether a transaction is fraudulent or not so we can protect consumers from theft/fraud.

Data Description

For a full description of the dataset[1], we have 3075 observations with 10 predictor variables where some are quantitative and some qualitative and a binary qualitative response of whether a transaction is fraudulent or not.

Table 1. Data Description

Features	Descriptions
Merchant ID	Observation ID
Avg Amount per Day	Average Transaction Amount per Day
Transaction Amount	Amount of the Transaction
Is Declined?	Yes = The credit card is declined, No = The credit card is not declined
Number of Declines per Day	Total Number of Declines per Day
Foreign Transaction	Yes = Foreign Transaction, No = Not a Foreign Transaction
High Risk Country	Yes = high risk country, No = Not a high risk country
Daily Average Chargeback Amount	Daily Average Chargeback Amount
6 Month Average Chargeback Amount	6 Months Average of Chargeback Amount
6 Month Chargeback Frequency	Frequency of Chargebacks in 6 months
Is Fraudlent	Fradulent = Fradulent Transaction, Not Fradulent = Non-Fradulent Transaction

Questions

Before we begin the analysis, let's look some main questions we want to answer that will give us some insight.

1) Which subset of variables should we work with to detect credit card fraud?
2) Which classifier works best in terms of the error rate for classifying credit card fraud?
3) Which classifier should we use to make predictions on credit card fraud?

Data Exploration & Visualization

Now we finally begin the analysis by exploring our variables and initially screening for anything problematic or interesting about our data.

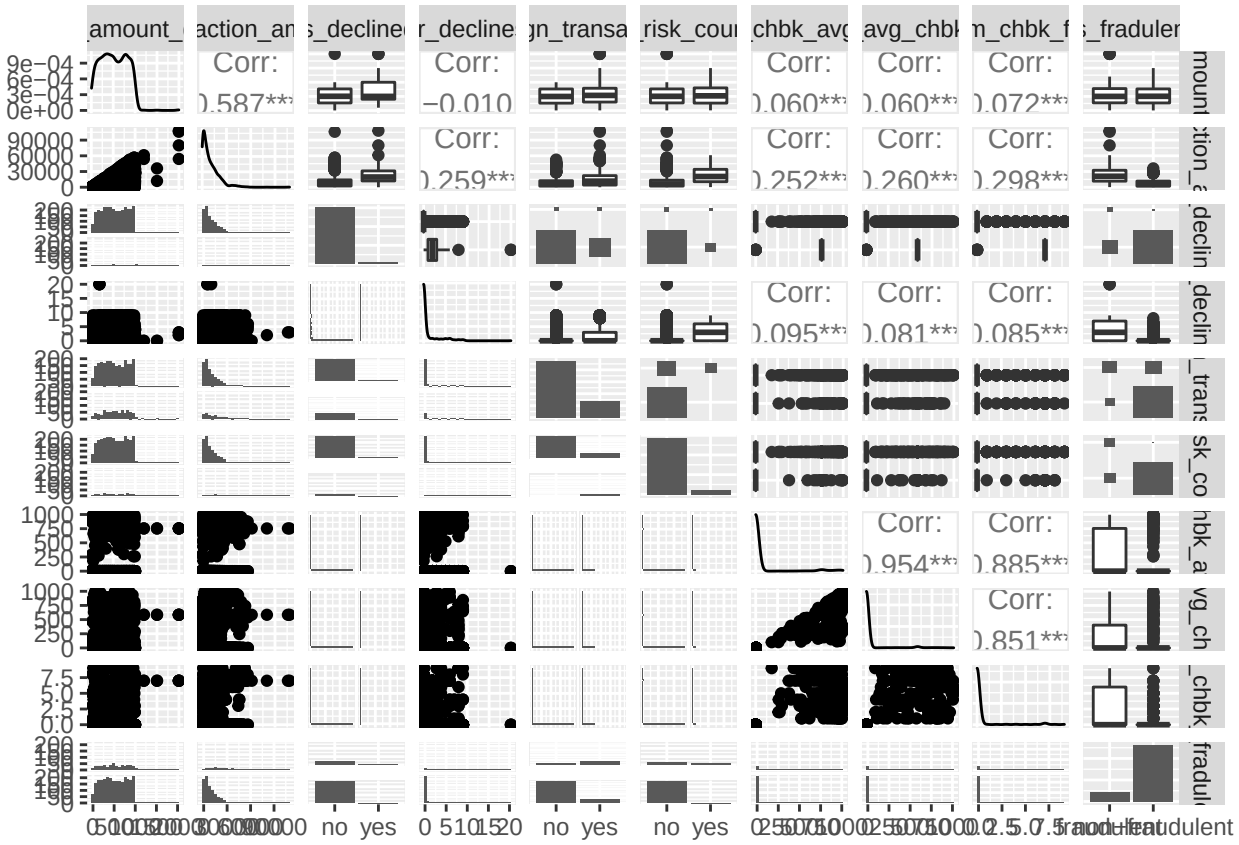
Table 2. Number of NA values in each variable

merchan_id	avg_amount_days	transaction_amount	is_declined	number_declines_days
0	0	0	0	0

foreign_transaction	high_risk_countries	daily_chbk_avg_amt	X6m_avg_chbk_amt	X6m_chbk_freq
0	0	0	0	0

First, we observe that our data is clean of NA's so we do not have to do much manipulation in this area.

Figure 1. Plots between each pair of variables



However, notice that there are a few predictors that are correlated/have some sort of relationship with each other but not exactly in the typical straight line. The most problematic patterns we see is the triangle shape in the plot of "Average Amount per Day" and "Transaction Amount" which could be an indication of multicollinearity. Meanwhile notice that the Density plots for "Number of Declines per Day" , "Daily Average Chargeback Amount", "6 Month Average Chargeback Amount", and "6 Month Chargeback Frrequency" has this strong L shape or rightly skewed distribution suggesting too many 0's for this to make sense as a numeric variable. Let's confirm this.

Table 3. Number of zeros in each quantiative variable

merchan_id	avg_amount_days	transaction_amount	number_declines_days	daily_chbk_avg_amt	X6m_avg_chbk_amt	X6m_chbk_freq
0	0	91	2384	2857	2857	2857

We can see that the variables we mentioned earlier indeed do have a lot of zeros and that their scatter plots look very problematic due to being highly correlated but the graph doesn't depict any obvious relationship other than a lot of (0,0)'s. (For ex Daily Chargeback Average Amount vs 6 Month Average Chargeback Amount). So what we should do instead is transform them into categorical variables with 2 levels where 0 indicates the value of 0 units and 1 indicates the value of non-zero units.

Let's redefine our variables now.

For Is Declined, we have 2 levels where 0 represents no and 1 represents yes.

For Number of Declines per Day, we have 2 levels 0 = Zero declines per day; 1= Number of declines per day is non-zero.

For Average daily chargeback amount, 0 = Average daily chargeback amount is 0; 1 = Average daily chargeback amount is non-zero.

For 6 Month Average Chargeback Amount, 0 = 6 Months Average Chargeback Amount is 0; 1 = 6 Months Average Chargeback Amount is non-zero.

For 6 Month Chargeback Frequency, 0 = Frequency of Chargebacks in 6 months is 0; 1 = Frequency of Chargebacks in 6 months is non-zero.

For Is Fraudulent, 0 = non-fradulent transaction; 1 = fraudulent transaction.

Since the Average Transaction Amount per Day and the Amount of the Transaction are intuitively and statistically correlated as shown above, we can choose to drop one of them since having both of them is redundant. From the boxplots on the very last column, we can see that the Average Transaction Amount per Day has roughly the same boxplots for fraudulent and non-fraudulent transactions indicating a lack of effect while that is not the case for the Transaction Amount Boxplots where the boxplots are clearly different and that there might be a stronger effect compared to Average Transaction Amount per Day. Hence we will be dropping the Average Transaction Amount per Day due to multicollinearity issues and a seemingly lack of effect.

Another issue I have noticed is how Foreign Transactions and High Risk Countries should not be treated as 2 separate categorical variables and instead as one variable. Intuitively they are dependent because if you have a transaction in a high risk country, it is automatically a foreign transaction or if you have no foreign transaction you automatically do not have a transaction in a high risk country. So a better way to approach this is to combine these categorical variables into one variable with more levels. First let's look at our categorical variables of interest more closely.

Table 4. Contingency Table between Foreign Transaction and High Risk Countries

Foreign Transaction / High Risk Country	no	yes
no	2369	0
yes	501	205

When we combine these variables, let's name this variable Location as in the location of the transaction and from this table we can see we have 3 levels with

0 for non-foreign transactions (both foreign transaction and high risk countries are 0) (least suspicious)

1 for foreign transactions that aren't in high risk countries (foreign transaction is 1 while high risk countries is 0) (more suspicious)

2 for foreign transactions in high risk countries (both foreign transaction and high risk countries are 1) (most suspicious)

Notice also that this categorical predictor is ordinal in the sense that there is a natural order where the level of suspicion increases from 0 to 2 which will be important to note in k-NN and our interpretations.

Selecting the Best Subset and Logistic Regression

After cleaning and manipulating the data set, we are ready to begin our analysis. One glaring question when doing our analysis is what variables we should use. Because we don't have too many predictors and a binary response, we can get away with the Best Subset Method[2] where we will be fitting 2^{10} logistic regressions on the training data and will use BIC as our criteria to select our best model.

Doing so, we figure out that the best subset of variables we should work with includes 5 variables which are the Transaction Amount, Total Number of Declines per Day, 2 dummy variables to account for the ordinal variable Location with 3 levels, and the 6 Month Chargeback Frequency.

Table 5. Summary of Logistic Regression Coefficients For The Best Subset Is Fradulent? ~ Trasnaction Amount + Number of Declines per Days + Location + 6 Month Chargeback Frequency

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-6.6351	0.33399	-19.86626	<2e-16
transaction_amount	0.00014	1e-05	11.09116	<2e-16
number_declines_days1	2.49593	0.24067	10.37078	<2e-16
location1	2.06403	0.25115	8.21839	<2e-16
location2	8.24123	0.77599	10.6203	<2e-16
X6m_chbk_freq1	3.73147	0.29275	12.74642	<2e-16

Also, looking at the Wald tests, even with a Bonferonni correction of $\alpha = 0.05/5 = 0.01$, we reject every null hypothesis that the corresponding coefficient is 0 and that every predictor is highly significant and important when it comes to our logistic regression.

Looking at the coefficients, we can see some behaviors that are associated with a higher probability of a fradulent transaction.

Firstly, for Transaction Amounts, due to a positive coefficient, transaction amounts are positively correlated with the probability of a fradulent transaction.

Next, with the positive coefficient, observations where the Number of Declines per day are nonzero have a higher probability of being fraudulent transactions compared to those where the card aren't declined.

Also, the coefficients for location line up with the ordinal nature of our variable. The more suspicious the location of transaction the higher the probability of the observation being a fradulent transaction. (Since 0 is our baseline level 1 has a positive coefficient and then level 2 has the largest coefficient)

Finally, the observations with a 6 Month Average of Chargeback Amount that are nonzero have a higher probability of being fraudulent transactions compared to those where the average is 0 i.e. no chargeback.

As a summary, red flags/behaviors that line up with a larger probability/risk of fraudulent credit card transactions tend to involve spending more in a transaction, cards being declined , transactions in suspicious locations, and charging back your credit card which is in line with what we see in the real world.

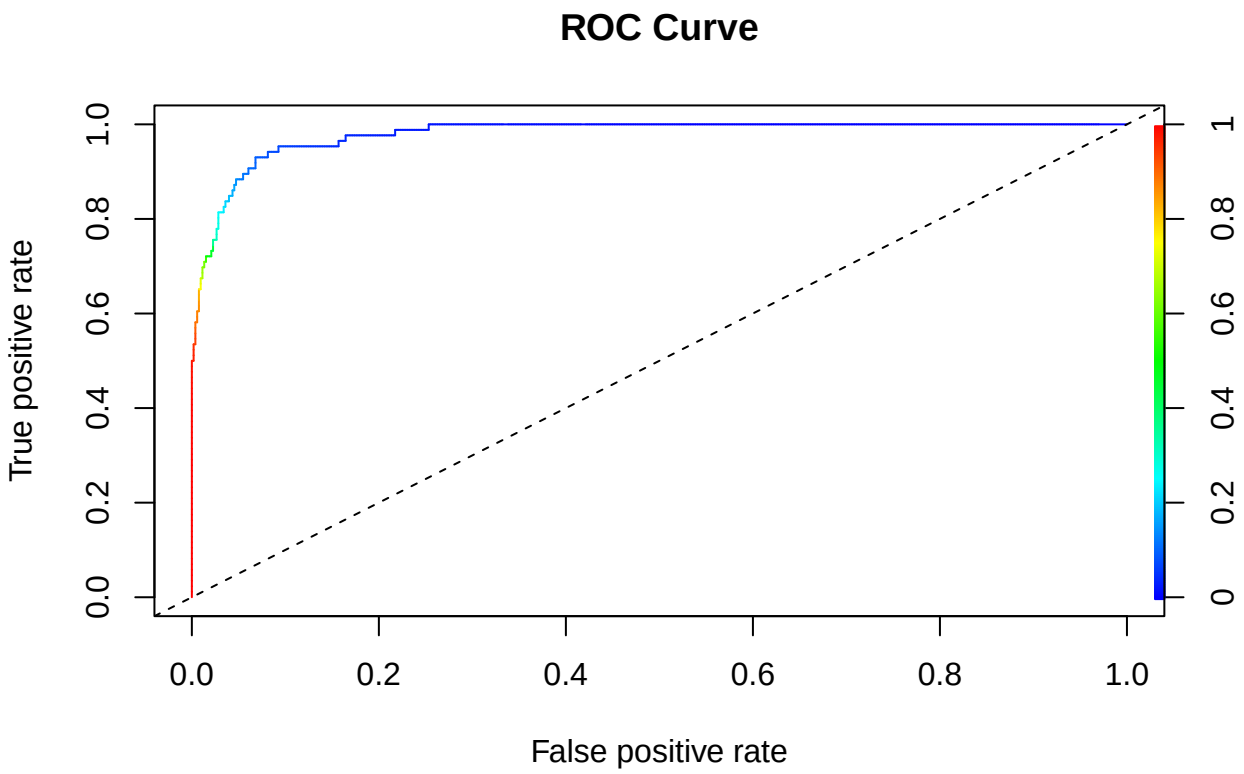
Addressing Proposal Feedback: ROC Curve Analysis

Although this model looks great, we can't really say much until we evaluate the ROC Curve and Confusion Matrix.

Table 6. Optimal Threshold Point and AUC

threshold	specificity	sensitivity	AUC
0.0933083	0.9319471	0.9302326	0.9790742

Figure 2. ROC Curve with color representing threshold



This is our ROC curve. Visually, the ROC curve is almost touching the top left corner which is an indication of an exceptionally performing model. This is confirmed by the fact that the AUC is 0.9791 which is almost 1, an indication that the model has a performance that is almost ideal. Furthermore, our optimal threshold is found to be 0.0933 where the TPR and FPR are optimized. Notice how sensitive our optimal model is due to the extremely low threshold. This is in fact desirable because even the slightest sign of fraud is something we should be paranoid about.

Table 7. Confusion Matrix for Logistic Regression. Negative = non-fraudulent, Positive = fraudulent.

True/Predicted	non-fraudulent	fraudulent
non-fraudulent	493	36
fraudulent	6	80

Table 8. Summarization of Errors

Accuracy	Error Rate	FPR	FNR	TPR	TNR
0.9317073	0.0682927	0.0680529	0.0697674	0.9302326	0.9319471

Let's observe that our logistic regression model with the optimal threshold has a confusion matrix that supports the fact that it performs well. The first good sign is the fact that the Error Rate at 0.0683 is extremely low. Furthermore, it also makes very few mistakes for both true postives and negatives with low False Positive Rates and False Negative Rates at 0.0681 and 0.0698 respectively. Overall, there seem to be very little flaws with this model.

k-NN

Another method we could employ is k-NN. To have comparable results, we will use the same variables we employed in our logistic regression which is the Transaction Amount, Number of Declines per Day, Location, and the 6 Month Chargeback Frequency. After careful assessment, the Euclidean distance will be suitable for these variables. Number of Declines per Day and the 6 Month Chargeback Frequency are binary so there will not be much issue. As for Transaction Amount, we must remember to standardize this variable as its units differ from the rest of the variables. Finally, the ordinal variable Location while most complicated, can be kept in its numeric form 0, 1, 2. As a recap

0 for non-foreign transactions (both foreign transaction and high risk countries are 0) (least suspicious)

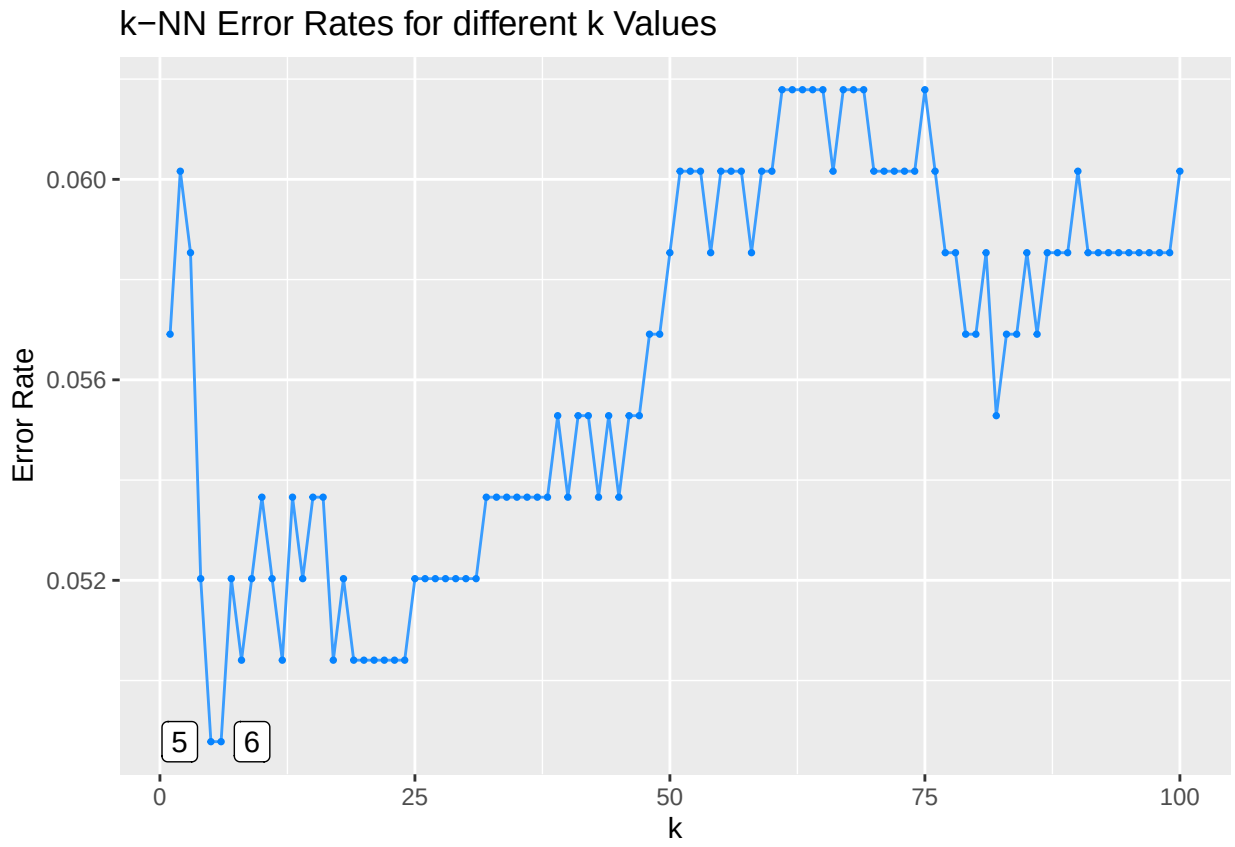
1 for foreign transactions that aren't in high risk countries (foreign transaction is 1 while high risk countries is 0) (more suspicious)

2 for foreign transactions in high risk countries (both foreign transaction and high risk countries are 1) (most suspcious)

The difference between 0 and 2 is more drastic as compared to 1 and 2 so we will keep it in numeric form to be able to represent stronger differences.

As a rule of thumb, we should take the square root of the number of observations in the training set and start searching values around that value to find the optimal k for our k-NN which is one of our goals here. We will define optimal as having the lowest error rate. In this case the square root of the number of observations is roughly 50. As a result we will be searching values from 1 to 100.

Figure 3. Plot of Error Rates for k-NN at various k values



After running k-NN for these values, looking at the plot and double checking the error rates, 5 and 6 are tied as the optimal k for k-NN. However we should go with 5 since 6 is a multiple of the number of classes

3 which is going to cause issues due to complete ties being possible.

Let's take a closer look at the confusion matrix and the summary for k-NN k=5.

Table 9. Confusion Matrix for k-NN k=5

True/Predicted	non-fraudulent	fraudulent
non-fraudulent	522	7
fraudulent	23	63

Table 10. Summarization of Errors

Accuracy	Error Rate	FPR	FNR	TPR	TNR
0.9512195	0.0487805	0.0132325	0.2674419	0.7325581	0.9867675

We will discuss the results even further when we compare our models in our research questions but first, we can see that this model performs relatively well. It has a low error rate, however one very alarming issue is the high FNR. That means, if we were to use this model, we would have a lot of false negatives where an excessively large proportion of criminals would get away with theft and credit card fraud which would have serious consequences.

Conclusion

The overall goal of this project was to utilize our dataset and find an optimal supervised learning technique to be able to catch criminals and their fraudulent transactions via credit card. In order to achieve that goal some questions had to be answered which are the following.

1) Which subset of variables should we work with to detect credit card fraud?

Essentially, we had to figure out what kind of information we needed before we even started building our models to achieve our overall goal. Logistic regressions can be used in a way where we figure out which variables should be employed in our model and after troubleshoting some issues with our data and fitting every possible model, we used some specific criteria (namely BIC) and determined the best model would need the following information/variables: Transaction Amount, Number of Declines per Day, Location, and 6 Month Chargeback Frequency.

2) Which classifier works best in terms of the error rate for classifying credit card fraud?

After assessing some assumptions required, we found that logistic regression and k-NN were suitable choices for our data. As seen in the confusion matricies and summaries, we were able to find that k-NN had the lowest error rate at 4.9% while logistic regression was almost just as good at 6.8%. However, we can't immediately conclude that k-NN is the better model that we should utilize because there are other factors to consider than solely the error rate.

3) Which classifier should we use to make predictions on credit card fraud?

First, here is a table comparing the results of the logistic regression and k-NN

Table 11. Comparison of Errors Made in Logistic Regression and k-NN

	k-NN	Logistic Regression
Accuracy	0.9512195	0.9317073
Error Rate	0.0487805	0.0682927
FPR	0.0132325	0.0680529
FNR	0.2674419	0.0697674
TPR	0.7325581	0.9302326
TNR	0.9867675	0.9319471

Before we choose our model, we need to figure out what criteria makes a good model. In this context, a good model should have...

i) A low error rate since we generally don't want to make too many mistakes.

ii) A reasonable FPR. It's okay to be extra careful of fraud and thus we can tolerate a higher FPR but it cannot be so high that we are constantly wasting time on innocent people.

iii) A very low FNR. This part matters the most since it's detrimental to have too many criminals getting away with their crimes.

As mentioned earlier, both models have a fairly low error rate where they aren't much different from each other where they could be roughly tied in this area.

Next, the logistic regression had a higher FPR at 6.8% while k-NN had a lower FPR at 1.3%. This means that k-NN is better at identifying negatives compared to the logistic regression. In this context a False Positive is when an innocent consumer is flagged down for credit card fraud. While it will be inconvenient for the innocent consumer, it is OK to have a higher FPR since it's okay to be safe rather than sorry when it comes to catching criminals. As long as the FPR is not obnoxiously high where we are constantly filing false reports, both of these models are fine in this area as 1% and 6% are fairly low.

However, the logistic regression had a FNR at roughly 7% while k-NN had a noticeably higher FNR at roughly 26.7%. This means that the logistic regression is incomparably better than k-NN at identifying positives and wins in this area. In this context, more criminals would get away with credit card fraud in k-NN as opposed to logistic regression. While someone innocent being accused can prove their innocence, a criminal getting away with something illegal is a more detrimental error.

As a summary, both models meet criteria i and ii pretty well. However k-NN performs terribly in criteria iii while logistic regression is the clear winner in this area. As a result, logistic regression meets all criteria while k-NN doesn't and thus it would be best to go with the logistic regression to make predictions.

References

1. Neural Designer (n.d.). Credit Card Fraud. from https://www.neuraldesigner.com/learning/examples/credit-card-fraud
2. A. I. McLeod, C. Xu (2010). bestglm: Best Subset GLM. from http://www2.uaem.mx/r-mirror/web/packages/bestglm/vignettes/bestglm.pdf

Appendix

```
library(kableExtra)
library(bestglm)
library(ROCR)
library(pROC)
library(GGally)
library(ggplot2)
library(tidyverse)
library(dplyr)
library(class)
library(formatR)
library(knitr)
knitr::opts_chunk$set(echo = FALSE)
opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
Info <- t(as.matrix(c(Name = "Clayton Chan", Email = "clwchan@ucdavis.edu")))

(the_kable <- kable(Info, "latex") %>%
    kable_styling(full_width = T))
credit <- read.csv("creditcard-fraud.csv", sep = ";", header = T,
    stringsAsFactors = TRUE)
Features <- c("Merchant ID", "Avg Amount per Day", "Transaction Amount",
    "Is Declined?", "Number of Declines per Day", "Foreign Transaction",
    "High Risk Country", "Daily Average Chargeback Amount", "6 Month Average Chargeback Amount",
    "6 Month Chargeback Frequency", "Is Fraudlent")
Descriptions <- c("Observation ID", "Average Transaction Amount per Day",
    "Amount of the Transaction", "Yes = The credit card is declined, No = The credit card is not declin
    "Total Number of Declines per Day", "Yes = Foreign Transaction, No = Not a Foreign Transaction",
    "Yes = high risk country, No = Not a high risk country",
    "Daily Average Chargeback Amount", "6 Months Average of Chargeback Amount",
    "Frequency of Chargebacks in 6 months", "Fradulent = Fradulent Transaction, Not Fradulent = Non-Fra

(the_kable <- kable(cbind(Features, Descriptions), "latex", booktabs = T))
(nas <- kable(t(as.matrix(colSums(is.na(credit))[1:5])), "latex",
    booktabs = T) %>%
    kable_styling(full_width = T))

(nas2 <- kable(t(as.matrix(colSums(is.na(credit))[6:10])), "latex",
    booktabs = T) %>%
    kable_styling(full_width = T))
ggpairs(credit[, -1])
zeros <- c()
for (i in names(credit)) {
```

```
    if (class(credit[[i]]) == "numeric" | class(credit[[i]]) ==
        "integer") {
        zeros[i] <- sum(credit[[i]] == 0)
    }
}

(zeros <- kable(t(as.matrix(zeros)), "latex", booktabs = T) %>%
    kable_styling(latex_options = c("scale_down", "hold_position")))
credit$is_declined <- factor(ifelse(credit$is_declined == "yes",
    1, 0))
credit$number_declines_days <- factor(ifelse(credit$number_declines_days ==
    0, 0, 1))
credit$daily_chbk_avg_amt <- factor(ifelse(credit$daily_chbk_avg_amt ==
    0, 0, 1))
credit$X6m_avg_chbk_amt <- factor(ifelse(credit$X6m_avg_chbk_amt ==
    0, 0, 1))
credit$X6m_chbk_freq <- factor(ifelse(credit$X6m_chbk_freq ==
    0, 0, 1))
credit$is_fradulent <- factor(ifelse(credit$is_fradulent == "fraudulent",
    1, 0))
cat_table <- table(credit$foreign_transaction, credit$high_risk_countries,
    dnn = c("Foreign Transaction", "High Risk Country"))
kable(matrix(c("Foreign Transaction / High Risk Country", "no",
    "yes", "no", 2369, 0, "yes", 501, 205), 3, 3, byrow = T),
    "latex") %>%
    kable_styling(full_width = F, latex_options = c("striped",
        "hold_position"))

credit$location <- factor(ifelse(credit$foreign_transaction ==
    "yes" & credit$high_risk_countries == "yes", 2, ifelse(credit$foreign_transaction ==
    "no" & credit$high_risk_countries == "no", 0, 1)))
credit <- credit[, -c(6, 7)]  #Drop columns foreign transaction and high risk countries since it is now
credit <- credit[, c(1:8, 10, 9)]  #Rearrange columns for bestglm function
set.seed(6)
index <- sample(1:3075, 3075/5)
train <- credit[-index, ]
test <- credit[index, ]
bestmodel <- bestglm(Xy = train[, -c(1, 2)], family = binomial,
    IC = "BIC")
bestmodel$BestModel
# We will exclude from the search Merchant ID as it is just
# an ID number and Avg Amount of Days due to reasons we
# stated earlier
thebestmodel <- glm(is_fradulent ~ transaction_amount + number_declines_days +
    location + X6m_chbk_freq, family = binomial, data = train)
sumlog <- cbind(round(summary(thebestmodel)$coefficients[, 1:3],
    5), rep("<2e-16", 6))
colnames(sumlog)[4] <- "Pr(>|z|)"
kable(sumlog, "latex", booktabs = T) %>%
    kable_styling(latex_options = c("hold_position", "striped"))
prob <- predict(thebestmodel, test, type = "response")
roc_curve <- roc(test$is_fradulent, prob, plot = F, legacy.axes = T,
    xlab = "FPR", ylab = "TPR")
```

```
opt <- coords(roc_curve, x = "best", input = "threshold", best.method = "youden")
kable(data.frame(opt, AUC = roc_curve$auc), booktabs = T) %>%
    kable_styling(full_width = T, latex_options = c("hold_position",
        "striped"))
pred <- prediction(prob, test$is_fradulent)
plot(performance(pred, "tpr", "fpr"), colorize = T, main = "ROC Curve")
abline(a = 0, b = 1, lty = 2)

# Function for Confusion Matrix and its Summarization This
# will be very useful because we are going to be comparing
# classification models based on the criteria in the
# summary

# A warning for this function is that the inputs true and
# pred are recommended to have levels 0 and 1 so that the
# summary is accurate. It is still possible to rename the
# rows and columns so this should not be a major issue.

CMsum <- function(true, pred) {
    summarylist <- list()
    summarylist$conf_m <- table(true, pred, dnn = c("True", "Predicted"))
    summaryrates <- c()
    summaryrates["Accuracy"] <- sum(diag(summarylist$conf_m))/sum(summarylist$conf_m)
    summaryrates["Error Rate"] <- 1 - summaryrates["Accuracy"]
    summaryrates["FPR"] <- summarylist$conf_m[1, 2]/(summarylist$conf_m[1,
        1] + summarylist$conf_m[1, 2])
    summaryrates["FNR"] <- summarylist$conf_m[2, 1]/(summarylist$conf_m[2,
        1] + summarylist$conf_m[2, 2])
    summaryrates["TPR"] <- 1 - summaryrates["FNR"]
    summaryrates["TNR"] <- 1 - summaryrates["FPR"]
    summarylist$summaryrates <- summaryrates
    return(summarylist)
}
predicted <- ifelse(prob > opt$threshold, 1, 0)
CM_sum <- CMsum(true = test$is_fradulent, pred = predicted)
thecm <- matrix(c("True/Predicted", "non-fraudulent", "fraudulent",
    "non-fraudulent", 493, 6, "fraudulent", 36, 80), 3, 3)
kable(thecm, "latex") %>%
    kable_styling(latex_options = c("striped", "hold_position"))
(the_kable <- kable(t(as.matrix(CM_sum$summaryrates)), "latex",
    booktabs = T) %>%
    kable_styling(latex_options = "hold_position"))
# Coercing a factor to numeric does not give me the desired
# 0 and 1 but rather 1 and 2 so to solve this I created
# this function that coerces the factor into a character
# and then numeric.
factor_to_num <- function(x) {
    if (class(x) == "factor") {
        y <- as.character(x)
        z <- as.numeric(y)
        return(z)
    } else {
        stop("Entry must be a factor")
```

```
    }
}
train_pred <- train[, c(3, 5, 8, 9)]
test_pred <- test[, c(3, 5, 8, 9)]
train_resp <- train[, 10]
train_pred$transaction_amount <- (train_pred$transaction_amount -
    mean(train_pred$transaction_amount))/sd(train_pred$transaction_amount)
test_pred$transaction_amount <- (test_pred$transaction_amount -
    mean(test_pred$transaction_amount))/sd(test_pred$transaction_amount)
train_pred$number_declines_days <- factor_to_num(train_pred$number_declines_days)
train_pred$X6m_chbk_freq <- factor_to_num(train_pred$X6m_chbk_freq)
train_pred$location <- factor_to_num(train_pred$location)
test_pred$number_declines_days <- factor_to_num(test_pred$number_declines_days)
test_pred$X6m_chbk_freq <- factor_to_num(test_pred$X6m_chbk_freq)
test_pred$location <- factor_to_num(test_pred$location)
sqrt(nrow(train_pred))
k_values <- 1:100
Error_rates <- c()
for (k in k_values) {
    predictions <- knn(train_pred, test_pred, train_resp, k = k,
        prob = F)
    Results <- CMsum(true = test$is_fradulent, pred = predictions)
    Error_rates[as.character(k)] <- Results$summaryrates["Error Rate"]
}
Error_rates[Error_rates == min(Error_rates)]
error_data <- data.frame(k = k_values, Error_Rate = Error_rates)
ggplot(data = error_data, aes(x = k, y = Error_Rate)) + geom_line(color = "#3B9DFE") +
    geom_point(color = "#0683FE", size = 0.625) + ggrepel::geom_label_repel(data = dplyr::filter(error_
    Error_Rate == min(Error_Rate)), aes(label = k)) + labs(title = "k-NN Error Rates for different k Va
    y = "Error Rate")
predictions <- knn(train_pred, test_pred, train_resp, k = 5,
    prob = F)
Results <- CMsum(true = test$is_fradulent, pred = predictions)
knn_cm <- c("True/Predicted", "non-fraudulent", "fraudulent",
    "non-fraudulent", 522, 23, "fraudulent", 7, 63)
kable(matrix(knn_cm, 3, 3), "latex") %>%
    kable_styling(latex_options = c("striped", "hold_position"))
kable(t(as.matrix(Results$summaryrates)), booktabs = T) %>%
    kable_styling(latex_options = c("hold_position"))
k_NN <- Results$summaryrates
logreg <- CM_sum$summaryrates
comparison <- cbind(k_NN, logreg)
colnames(comparison) <- c("k-NN", "Logistic Regression")
kable(comparison, booktabs = T) %>%
    kable_styling(full_width = T)
```