---
title: "Classification"
output: html_notebook

The data set used in this notebook is named "Adult Data Set" and was acquired from the UCI Machine Learning Repository.

The data set can be found here: https://archive.ics.uci.edu/ml/datasets/Adult

It contains data on adults who participated in the 1994 census.

In using this data set, the objective is to create a classification model that can predict whether a given person had a net income greater than or less than fifty thousand dollars.

```{r}
library(caret)
library(naivebayes)
library(tidyverse)
library(dplyr)
library(ggplot2)

set.seed(12345)

adultsData <- read.csv("adult.data", header=FALSE)
names(adultsData) <- c("age", "workclass", "fnlwgt", "education",
"education_num", "marital_status", "occupation", "relationship", "race",
"sex", "capital_gain", "capital_loss", "hours_per_week", "native_country",
"income")

sapply(adultsData, function(x) sum(is.na(x)))

adultsData <- adultsData[(complete.cases(adultsData)),]
sum(is.na(adultsData))

adultsData$workclass <- factor(adultsData$workclass)
adultsData$education <- factor(adultsData$education)
adultsData$marital_status <- factor(adultsData$marital_status)
adultsData$occupation <- factor(adultsData$occupation)
adultsData$relationship <- factor(adultsData$relationship)
adultsData$race <- factor(adultsData$race)
adultsData$sex <- factor(adultsData$sex)
adultsData$native_country <- factor(adultsData$native_country)
adultsData$income <- factor(adultsData$income)

sample <- sample(c(TRUE,FALSE), nrow(adultsData), replace=TRUE,
prob=c(0.8,0.2))
train <- adultsData[sample, ]
test <- adultsData[!sample, ]

summary(train)
nrow(train)
ncol(train)
```

```
cor(train$age, as.numeric(train$income))
cor(train$capital_gain, as.numeric(train$income))
cor(train$capital_loss, as.numeric(train$income))

hist(train$age)
boxplot(train$age~train$income, xlab="Income", ylab="Age")

logReg1 <- glm(income ~ ., family=binomial, data=train)
summary(logReg1)
plot(logReg1)
```
Analysis of Logistic Regression:
There are some Notable outliers in the plots, for example instance 20177. In 20177's case, this is likely due to the fact that it has a large capital_gains value. In the summary of the model all variables are broken down. As expected, Age and capital gains/losses are good predictors for Income in this model. Interestingly, some values of native_country are better predictors that others, with "United-States" being the most useful for prediction and "Holand-Netherlands"[SIC] being the worst predictor. This is likely due the limited scope of this data set, as "United-States" is the most common value for native_country while there is only one instance of the value "Holand-Netherlands".

```{r}
bayes1 <- naive_bayes(income ~ ., data = train, usekernel = T)
summary(bayes1)
plot(bayes1)
```
Analysis of Naïve Bayes:
Naïve Bayes determines the likelihood of the outcome based on the probabilities of the prior variables. As such plotting the bayes model gives us a graph breaking down the distribution of each variable according to it's corresponding value of income. From analyzing these graphs we can observe that the relative peak of the ">50k" line on the age graph is much further along the x-axis then the relative peak of the "<=50k". This disparity reinforces what we already knew that age is a strong predictor for income.

```{r}
logProb <- predict(logReg1, test, type = "response")
predBayes <- predict(bayes1, test)

predLog <- ifelse(logProb>0.5, ">50K", "<=50K")

predLog <- factor(predLog)

logTab <- table(predLog, test$income)
logTab

logAcc <- (sum(diag(logTab))/sum(logTab))

print("Logistic regression: ")
print("Accuracy: ")
logAcc
```

```
print("Error rate: ")
(1 - logAcc)
print("Sensititvity: ")
(logTab[1,1] /(logTab[1,1] + logTab[2,1]))
print("Specificity: ")
(logTab[2,2] /(logTab[2,2] + logTab[1,2]))

(bayesTab <- table(predBayes, test$income))

bayesAcc <- sum(diag(bayesTab)) / sum(bayesTab)

print("Naive Bayes: ")
print("Accuracy: ")
bayesAcc
print("Error rate: ")
(1 - bayesAcc)
print("Sensititvity: ")
(bayesTab[1,1] /(bayesTab[1,1] + bayesTab[2,1]))
print("Specificity: ")
(bayesTab[2,2] /(bayesTab[2,2] + bayesTab[1,2]))
```

Logistic Regression vs Naïve Bayes Predictions:
Of the two models, the logistic regression model is overall more accurate. However, the Bayes model correctly identifies true positives more often than the logistic regression model, meaning the Bayes model has a higher sensitivity. Conversely the Bayes model's specificity is considerably lower than the regression model. Likewise, the logistic regression model's specificity is not great. Keeping this in mind, the logistic model's cut-off point was arbitrarily set to 0.5, it is possible a better cut-off point could be decided upon.

Strengths and Weaknesses of Each Model:
Overall, the regression model is more accurate, however it is more prone to false negatives than the Bayes model. While the Bayes model has better sensitivity, it also has a much higher rate of false positives.

On Classification metrics:
In terms of metrics, accuracy is the simplest to understand. Ultimately, accuracy and its complement, error rate, help give a good general idea of the model, but they only go so far as a means of understanding the model. Specificity and sensitivity go hand in hand, essentially acting as accuarcy for positive and negative results, respectively. They both give a good understanding of what results your model may be biased towards, but each only tells part of the story.