Classification

Aditi Chaudhari

2022-09-25

Classification

Classification is a typical supervised learning task that attempts to identify what class an observation falls into. To be more precise, the linear models in classification create linear boundaries to create regions in which most observations are of the same class. An advantage to using classification algorithms is that they help classify observations when the target variable is qualitative. However, classification algorithms are disadvantaged in that they are not as useful when our target variable is quantitative. Linear regression would be more beneficial in the latter case.

Data Exploration

Let's delve into exploring the logistic regression model!

First, data from the adult.csv file is read into a data frame. The data was obtained from https://www.kaggle.com/datasets/uciml/adult-census-income?resource=download.

```
df <- read.csv("adult.csv")
```

Firstly, let's simply see what our data looks like using the head() function, which selects the first n rows of a data frame. The target variable will be income, so understanding how the data is stored in the income variable is key.

```
head(df, n=10)
```

```
##    age   workclass fnlwgt    education education.num marital.status
## 1   90           ?  77053      HS-grad             9        Widowed
## 2   82     Private 132870      HS-grad             9        Widowed
## 3   66           ? 186061 Some-college            10        Widowed
## 4   54     Private 140359      7th-8th             4       Divorced
## 5   41     Private 264663 Some-college            10      Separated
## 6   34     Private 216864      HS-grad             9       Divorced
## 7   38     Private 150601         10th             6      Separated
## 8   74   State-gov  88638    Doctorate            16  Never-married
## 9   68 Federal-gov 422013      HS-grad             9       Divorced
## 10  41     Private  70037 Some-college            10  Never-married
##           occupation   relationship  race    sex capital.gain capital.loss
## 1                  ?  Not-in-family White Female            0         4356
## 2    Exec-managerial  Not-in-family White Female            0         4356
## 3                  ?      Unmarried Black Female            0         4356
## 4  Machine-op-inspct      Unmarried White Female            0         3900
## 5     Prof-specialty      Own-child White Female            0         3900
## 6      Other-service      Unmarried White Female            0         3770
## 7       Adm-clerical      Unmarried White   Male            0         3770
## 8     Prof-specialty Other-relative White Female            0         3683
```

```
## 9     Prof-specialty  Not-in-family White Female            0         3683
## 10      Craft-repair      Unmarried White   Male            0         3004
##    hours.per.week native.country income
## 1              40  United-States  <=50K
## 2              18  United-States  <=50K
## 3              40  United-States  <=50K
## 4              40  United-States  <=50K
## 5              40  United-States  <=50K
## 6              45  United-States  <=50K
## 7              40  United-States  <=50K
## 8              20  United-States   >50K
## 9              40  United-States  <=50K
## 10             60              ?   >50K
```

Next, let's take a look at the structure of the data frame. An important point to note is that income is of type character, but we would want it to be a factor.

```
str(df)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : chr  "?" "Private" "?" "Private" ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : chr  "HS-grad" "HS-grad" "Some-college" "7th-8th" ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: chr  "Widowed" "Widowed" "Widowed" "Divorced" ...
##  $ occupation    : chr  "?" "Exec-managerial" "?" "Machine-op-inspct" ...
##  $ relationship  : chr  "Not-in-family" "Not-in-family" "Unmarried" "Unmarried" ...
##  $ race          : chr  "White" "White" "Black" "White" ...
##  $ sex           : chr  "Female" "Female" "Female" "Female" ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
```

We can convert the income variable to be a factor using the as.factor() function.

```
df$income <- as.factor(df$income)
```

Now, we can randomly divide the data into a training set containing 80% of the original data and a test set containing 20% of the original data.

```
i <- sample(1:nrow(df), nrow(df) * 0.80, replace=FALSE)
train = df[i,]
test <- df[-i,]
```

Let's take a look at the structure of the training data frame to see how the data type of the income variable has changed. It is now a factor with 2 levels, one of which is "<=50k" and the other is ">50k".

```
str(train)
```

```
## 'data.frame':    26048 obs. of  15 variables:
##  $ age           : int  49 19 43 17 36 38 60 37 60 44 ...
##  $ workclass     : chr  "Self-emp-not-inc" "?" "Private" "Private" ...
##  $ fnlwgt        : int  123598 230874 147110 152652 126569 108293 376973 348796 259803 125461 ...
##  $ education     : chr  "HS-grad" "Some-college" "Some-college" "11th" ...
##  $ education.num : int  9 10 10 7 10 14 9 13 13 14 ...
```

```
##  $ marital.status: chr  "Never-married" "Never-married" "Never-married" "Never-married" ...
##  $ occupation    : chr  "Craft-repair" "?" "Adm-clerical" "Handlers-cleaners" ...
##  $ relationship  : chr  "Not-in-family" "Own-child" "Unmarried" "Own-child" ...
##  $ race          : chr  "White" "White" "White" "White" ...
##  $ sex           : chr  "Male" "Female" "Male" "Male" ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  30 40 48 25 40 40 42 40 45 35 ...
##  $ native.country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 2 1 2 1 2 1 2 2 ...
```

Using the summary() function in R provides us with summary statistics for each column. It is important to note that there are more data points in "<=50K" level than there are in the ">50k" level for the income factor.

```
summary(train)
```

```
##       age         workclass             fnlwgt         education
##  Min.   :17.00   Length:26048       Min.   :  12285   Length:26048
##  1st Qu.:28.00   Class :character   1st Qu.: 117983   Class :character
##  Median :37.00   Mode  :character   Median : 178735   Mode  :character
##  Mean   :38.52                      Mean   : 190154
##  3rd Qu.:47.25                      3rd Qu.: 237549
##  Max.   :90.00                      Max.   :1484705
##  education.num   marital.status      occupation        relationship
##  Min.   : 1.00   Length:26048       Length:26048       Length:26048
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.09
##  3rd Qu.:13.00
##  Max.   :16.00
##      race               sex             capital.gain    capital.loss
##  Length:26048       Length:26048       Min.   :    0   Min.   :   0.00
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.00
##  Mode  :character   Mode  :character   Median :    0   Median :   0.00
##                                        Mean   : 1088   Mean   :  84.75
##                                        3rd Qu.:    0   3rd Qu.:   0.00
##                                        Max.   :99999   Max.   :4356.00
##  hours.per.week  native.country       income
##  Min.   : 1.00   Length:26048       <=50K:19778
##  1st Qu.:40.00   Class :character   >50K : 6270
##  Median :40.00   Mode  :character
##  Mean   :40.45
##  3rd Qu.:45.00
##  Max.   :99.00
```

Let's find the size of the training data set.

The nrow() function shows that there are 26,048 observations.

```
nrow(train)
```

```
## [1] 26048
```

The ncol() function shows that there are 15 variables

```
ncol(train)
```

```
## [1] 15
```

Using the colSums() function, we can see that there are no missing values in any of the columns. It is important to remove missing values prior to performing logistic regression.

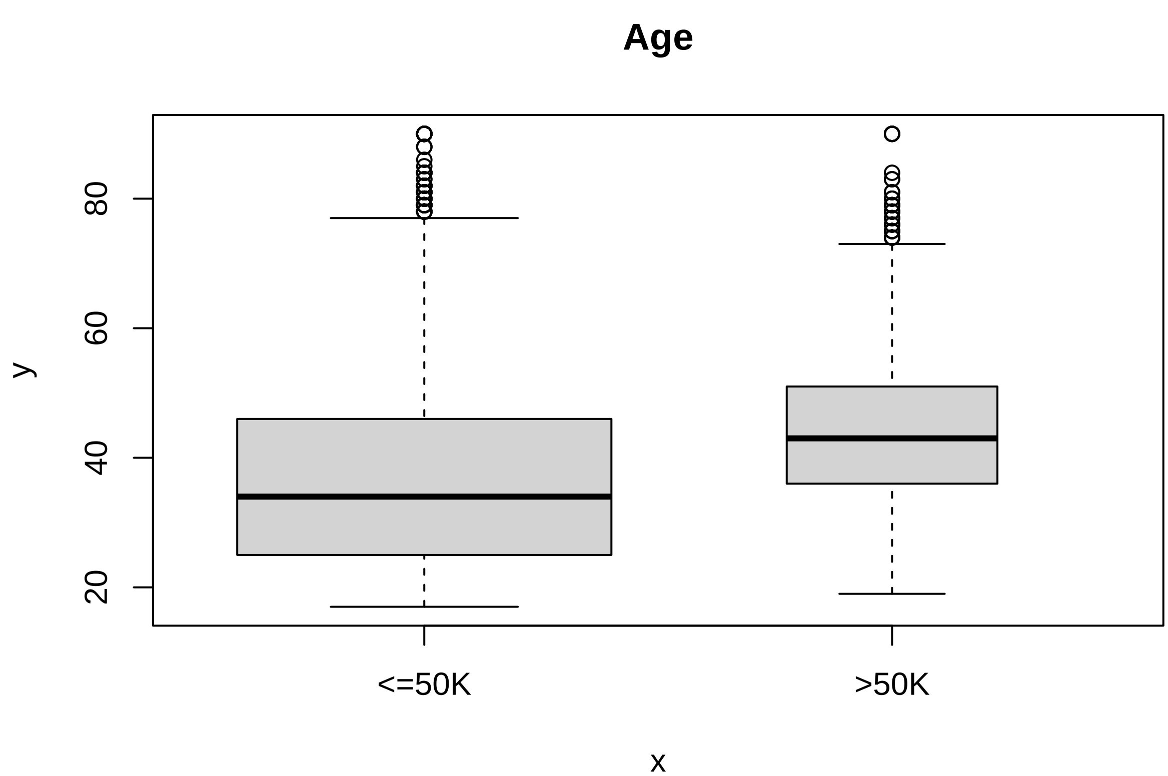
```
colSums(is.na(train))
```

```
##            age      workclass         fnlwgt      education  education.num
##              0              0              0              0              0
## marital.status     occupation   relationship           race            sex
##              0              0              0              0              0
##   capital.gain   capital.loss hours.per.week native.country         income
##              0              0              0              0              0
```

Data Visualization

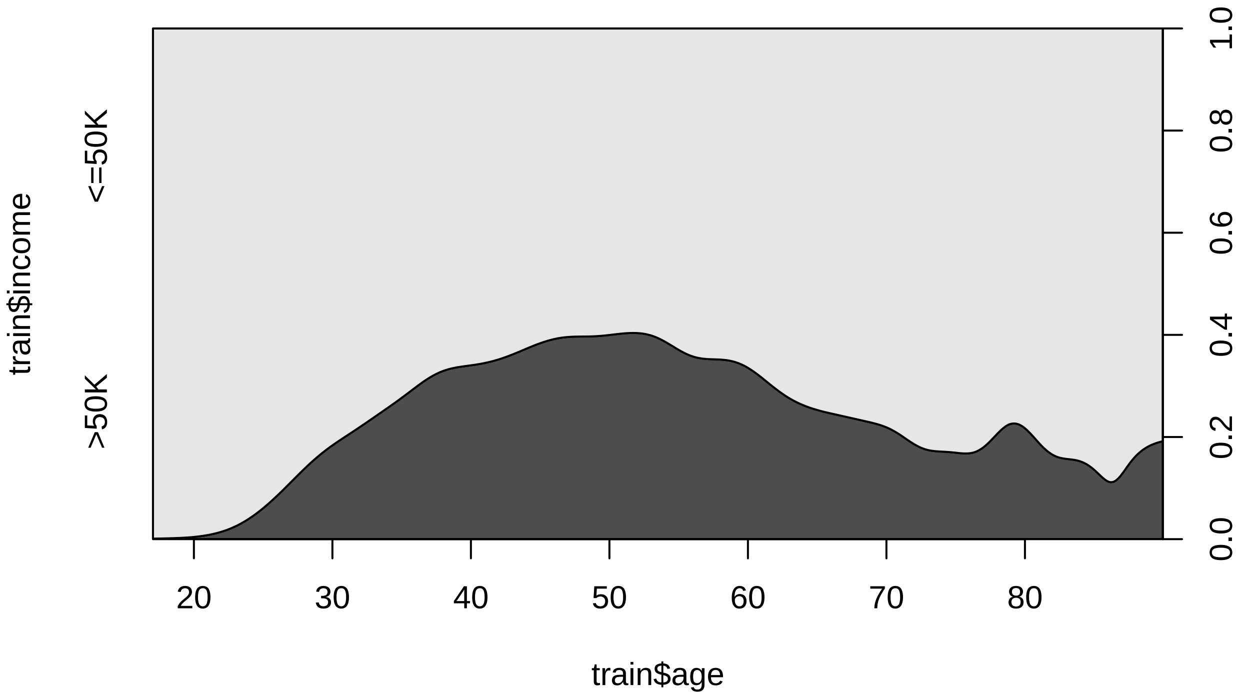
We can use a box-plot to visualize how age affects income. The graph below shows that <=50k is more common than >50k. More importantly, the box-plot shows that >50k observations are associated with those that are slightly older.

```
plot(train$income, train$age, data=train, main= "Age", varwidth=TRUE)
```



We can also use a conditional density plot to visualize how age affects income. The rectangle is the total probability space with the lighter grey indicating <=50k and the darker grey indication <50k.

```
cdplot(train$income~train$age)
```



Logistic Regression

Let's fit a logistic regression model to the data using the glm() function. A summary of the glm model that was created reveals 4 things: the glm() call, the residual distribution, the coefficients with statistical significance metrics, and metrics for the model. The deviance residual is a mathematical transformation of the loss function and quantifies a given point's contribution to the overall likelihood. It can be used to form RSS-like statistics. We can see statistical significance metrics at the bottom of the output. The null deviance measures the lack of fit of the model only considering the intercept, whereas the residual deviance measures the lack of fit of the entire model. Since the residual deviance is lower than the null deviance, our model is a good fit. The AIC, which stands for the Akaike Information Criteria, is useful in comparing models and typically, the lower the AIC is, the better. The coefficient is 0.039647, which quantifies the difference in the log odds of a target variable.

```r
glm1 <- glm(income~age, data=train, family=binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = income ~ age, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5544  -0.7509  -0.5939  -0.4831   2.0661
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.774284   0.047945  -57.86   <2e-16 ***
```

```
## age          0.040305   0.001086   37.11   <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: 28752  on 26047  degrees of freedom
## Residual deviance: 27308  on 26046  degrees of freedom
## AIC: 27312
##
## Number of Fisher Scoring iterations: 4
```

Naive Bayes Model

Naive Bayes is another classification algorithm. The prior for income, called A-priori, below is 0.759137 for <= 50k and 0.240863 for >50k. The likelihood data is shown in the output as conditional probabilities. Discrete data, such as sex, is broken down into <=50k and >50k for each attribute. For instance, if someone is making >50k, they are 15% likely to be female or 85% likely to be male according to the Naive Bayes model shown below. For continuous variables, such as age, we are given the mean and standard deviation for the two classes. The Naive Bayes model shown below reveals that the mean age for those making <=50k is around 36, while the mean age for those making >50k is around 44.

```
library(e1071)
nb1 <- naiveBayes(income~.,data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     <=50K      >50K
## 0.7592905 0.2407095
##
## Conditional probabilities:
##        age
## Y           [,1]     [,2]
##   <=50K 36.69926 13.94973
##   >50K  44.26890 10.56660
##
##        workclass
## Y                   ?  Federal-gov    Local-gov Never-worked      Private
##   <=50K 0.0661340884 0.0235109718 0.0597128122 0.0003539286 0.7188795632
##   >50K  0.0256778309 0.0456140351 0.0757575758 0.0000000000 0.6338118022
##        workclass
## Y       Self-emp-inc Self-emp-not-inc    State-gov  Without-pay
##   <=50K 0.0196177571     0.0729598544 0.0383254121 0.0005056123
##   >50K  0.0792663477     0.0937799043 0.0460925040 0.0000000000
##
##        fnlwgt
## Y           [,1]     [,2]
##   <=50K 190988.3 106877.5
```

```
##   >50K  187521.6 102177.7
##
##       education
## Y              10th         11th         12th      1st-4th      5th-6th
##   <=50K 0.0352917383 0.0445444433 0.0161795935 0.0063707149 0.0130447972
##   >50K  0.0074960128 0.0079744817 0.0039872408 0.0007974482 0.0019138756
##       education
## Y           7th-8th          9th   Assoc-acdm    Assoc-voc    Bachelors
##   <=50K 0.0249772474 0.0200222469 0.0327636768 0.0418646981 0.1264536354
##   >50K  0.0043062201 0.0035087719 0.0347687400 0.0443381180 0.2851674641
##       education
## Y         Doctorate      HS-grad      Masters    Preschool  Prof-school
##   <=50K 0.0046516331 0.3548892709 0.0323086257 0.0019718880 0.0065729599
##   >50K  0.0390749601 0.2108452951 0.1239234450 0.0000000000 0.0545454545
##       education
## Y       Some-college
##   <=50K 0.2380928304
##   >50K  0.1773524721
##
##       education.num
## Y           [,1]     [,2]
##   <=50K  9.604712 2.448536
##   >50K  11.636364 2.373247
##
##       marital.status
## Y          Divorced Married-AF-spouse Married-civ-spouse Married-spouse-absent
##   <=50K 0.1604307817      0.0006067348       0.3367377895          0.0154211750
##   >50K  0.0596491228      0.0012759171       0.8529505582          0.0038277512
##       marital.status
## Y       Never-married    Separated      Widowed
##   <=50K  0.4117706543 0.0394883204 0.0355445444
##   >50K   0.0649122807 0.0068580542 0.0105263158
##
##       occupation
## Y                  ? Adm-clerical Armed-Forces Craft-repair Exec-managerial
##   <=50K 0.0664880170 0.1331782789 0.0004044898 0.1297401153    0.0847911821
##   >50K  0.0256778309 0.0644338118 0.0001594896 0.1172248804    0.2524720893
##       occupation
## Y       Farming-fishing Handlers-cleaners Machine-op-inspct Other-service
##   <=50K    0.0350894934      0.0511174032      0.0696228132  0.1275154212
##   >50K     0.0140350877      0.0106858054      0.0331738437  0.0167464115
##       occupation
## Y       Priv-house-serv Prof-specialty Protective-serv        Sales
##   <=50K    0.0062190312   0.0930832238    0.0175447467 0.1068358782
##   >50K     0.0001594896   0.2387559809    0.0251993620 0.1251993620
##       occupation
## Y       Tech-support Transport-moving
##   <=50K 0.0263929619     0.0519769441
##   >50K  0.0349282297     0.0411483254
##
##       relationship
## Y          Husband Not-in-family Other-relative   Own-child   Unmarried
##   <=50K 0.296187683   0.301041561    0.037314187 0.201789868 0.130245728
##   >50K  0.752153110   0.109569378    0.004944179 0.008612440 0.027751196
```

```
##        relationship
## Y              Wife
##   <=50K 0.033420973
##   >50K  0.096969697
##
##        race
## Y       Amer-Indian-Eskimo Asian-Pac-Islander       Black       Other
##   <=50K        0.010820103        0.031449085 0.110122358 0.009960562
##   >50K         0.004625199        0.035247209 0.051674641 0.002711324
##        race
## Y             White
##   <=50K 0.837647892
##   >50K  0.905741627
##
##        sex
## Y          Female      Male
##   <=50K 0.3855294 0.6144706
##   >50K  0.1535885 0.8464115
##
##        capital.gain
## Y            [,1]       [,2]
##   <=50K  149.0977   983.0557
##   >50K  4048.1057 14711.1505
##
##        capital.loss
## Y            [,1]     [,2]
##   <=50K  52.31161 307.8936
##   >50K  187.05805 584.1778
##
##        hours.per.week
## Y           [,1]     [,2]
##   <=50K 38.87410 12.26736
##   >50K  45.41675 10.96410
##
##        native.country
## Y                  ?     Cambodia       Canada        China     Columbia
##   <=50K 1.784811e-02 3.539286e-04 3.286480e-03 2.426939e-03 2.275255e-03
##   >50K  1.897927e-02 7.974482e-04 5.263158e-03 2.551834e-03 1.594896e-04
##        native.country
## Y               Cuba Dominican-Republic      Ecuador  El-Salvador      England
##   <=50K 3.084235e-03       2.528061e-03 1.011225e-03 3.943776e-03 2.426939e-03
##   >50K  3.189793e-03       3.189793e-04 6.379585e-04 1.435407e-03 4.306220e-03
##        native.country
## Y             France      Germany       Greece    Guatemala        Haiti
##   <=50K 5.561735e-04 3.741531e-03 7.584184e-04 2.426939e-03 1.668521e-03
##   >50K  1.913876e-03 5.741627e-03 9.569378e-04 3.189793e-04 6.379585e-04
##        native.country
## Y       Holand-Netherlands     Honduras         Hong      Hungary        India
##   <=50K       5.056123e-05 4.044898e-04 6.572960e-04 4.550511e-04 2.325817e-03
##   >50K        0.000000e+00 0.000000e+00 9.569378e-04 3.189793e-04 5.103668e-03
##        native.country
## Y               Iran      Ireland        Italy      Jamaica        Japan
##   <=50K 9.101021e-04 8.089797e-04 1.921327e-03 2.881990e-03 1.617959e-03
##   >50K  1.754386e-03 6.379585e-04 2.870813e-03 1.275917e-03 3.189793e-03
```

```
##        native.country
## Y                Laos       Mexico    Nicaragua Outlying-US(Guam-USVI-etc)
##   <=50K 8.089797e-04 2.467388e-02 1.415714e-03               6.572960e-04
##   >50K  3.189793e-04 4.146730e-03 3.189793e-04               0.000000e+00
##        native.country
## Y                Peru  Philippines        Poland     Portugal  Puerto-Rico
##   <=50K 1.213470e-03 5.410052e-03 1.820204e-03 1.466276e-03 4.297705e-03
##   >50K  3.189793e-04 7.336523e-03 1.754386e-03 4.784689e-04 1.754386e-03
##        native.country
## Y            Scotland        South       Taiwan     Thailand Trinadad&Tobago
##   <=50K 4.044898e-04 2.679745e-03 1.365153e-03 5.561735e-04    6.572960e-04
##   >50K  3.189793e-04 2.073365e-03 2.711324e-03 3.189793e-04    1.594896e-04
##        native.country
## Y       United-States      Vietnam   Yugoslavia
##   <=50K  8.892709e-01 2.629184e-03 3.033674e-04
##   >50K   9.129187e-01 7.974482e-04 9.569378e-04
```

Evaluating the Test Data

Evaluating the logistic regression model with the test data shows a 75% accuracy. The error rate is about 25%.

```r
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, 2, 1)
acc1 <- mean(pred==as.integer(test$income))
print(paste("glm1 accuracy = ", acc1))
```

```
## [1] "glm1 accuracy =  0.740826040227238"
```

```r
table(pred, as.integer(test$income))
```

```
##
## pred    1    2
##    1 4801 1547
##    2  141   24
```

A confusion matrix is created. 4829 is True Positive, in which the items are true and were classified as true. 1545 is False Positive, in which the items were false and classified as true. 117 is False Negative, in which the items were true and classified as false. Finally, 22 is True Negative, in which the items were false and classified as false.

The sensitivity, which is the true positive rate, is 97.6%. The specificity, which is the true negative rate, is approximately 1.4%.

Let's now evaluate the Naive Bayes model with the test data.

```r
p2 <- predict(nb1, test)
(tab2 <- table(p2, test$income))
```

```
##
## p2      <=50K >50K
##   <=50K  4612  766
##   >50K    330  805
```

```r
acc2= sum(diag(tab2)/sum(tab2))
print(acc2)
```

```
## [1] 0.8317212
```

The accuracy for Naive Bayes is about 83% and is slightly higher than the accuracy for logistic regression. The Naive Bayes model may have outperformed the logistic regression model due to the fact that Naive Bayes models tend to perform better with smaller data sets.

Strengths and Weaknesses of Logistic Regression and Naive Bayes

The strengths of the logistic regression model are that it separates classes decently if the classes are linearly separable, it is not computationally expensive, and it provides a nice probabilistic output. The weakness of the logistic regression model is that it is prone to underfitting. The strengths of the Naive Bayes model are that it works well with smaller data sets, its easy to implement and interpret, and it handles high dimensions well. The weaknesses for Naive Bayes are that other classifiers may outperform it for larger data sets, guesses are made for values in the test set that did not occur in the training set, and the predictors must be independent for good performance.

Benefits and Drawbacks of Classification Metrics

Classification can be evaluated using many metrics. In this notebook, we used accuracy, sensitivity, and specificity. Accuracy is the number of correct predictions divided by the total number of predictions. It is a good measure, but does not give information on the true positive rate and the true negative rate. Sensitivity gives information on the true positive rate, Specificity gives information on the true negative rate.