Classification

Code

9/25/22

Data Set Citation: Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml (http://archive.ics.uci.edu/ml)]. Irvine, CA: University of California, School of Information and Computer Science.

Linear Models for Classification: The target must be qualitative and measures the probability between zero and one of the positive class. Meaning that it measures the probability that something is true given something else. A drawback of this is that it cannot do multiclass classification because of the processing power required to do so. This is counterbalanced by the fact that the model does well when there is one target that needs to be guessed and many other qualitative data points to help you infer it.

Reading in the data and splitting it into 80/20

We read in the data and rename the columns because they were not named in the actual data file. We then had to factor one of our variables into being either 1 or 2 in order to make working with it possible.

Hide

```
adult <- read.csv('adult.csv')
colnames(adult) <- c("age","workclass","fnlwgt","education","education-num","marital-status","oc
cupation","relationship","race","sex","capital-gain","capital-loss","hours-per-week","native-cou
ntry","50k")
adult$`50k` <- as.factor(adult$`50k`)

### Splitting Data 80/20
set.seed(1234)
i <- sample(1:nrow(adult), nrow(adult)*.80, replace=FALSE)
train <- adult[i,]
test <- adult[-i,]
```

Manipulating the data

We used different functions to see and understand the data better and allow us to make more informed inferences on how to manipulate the data in the next parts of the project.

Hide

```
str(train)
```

```
'data.frame':   26048 obs. of  15 variables:
 $ age           : int  53 20 42 31 43 46 29 40 23 37 ...
 $ workclass     : chr  " Private" " Private" " Federal-gov" " Private" ...
 $ fnlwgt        : int  238481 154779 284403 43716 142682 153501 34383 220563 141264 186934 ...
 $ education     : chr  " Assoc-voc" " Some-college" " HS-grad" " Assoc-voc" ...
 $ education-num : int  11 10 9 11 9 9 11 8 10 7 ...
 $ marital-status: chr  " Married-civ-spouse" " Never-married" " Divorced" " Divorced" ...
 $ occupation    : chr  " Exec-managerial" " Sales" " Adm-clerical" " Machine-op-inspct" ...
 $ relationship  : chr  " Husband" " Other-relative" " Not-in-family" " Unmarried" ...
 $ race          : chr  " White" " Other" " Black" " White" ...
 $ sex           : chr  " Male" " Female" " Male" " Male" ...
 $ capital-gain  : int  0 0 0 0 0 0 0 0 3103 ...
 $ capital-loss  : int  1485 0 0 0 0 0 0 0 0 ...
 $ hours-per-week: int  40 40 40 43 30 40 55 40 40 44 ...
 $ native-country: chr  " United-States" " United-States" " United-States" " United-States" ...
 $ 50k           : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 1 2 ...
```

Hide

```
dim(train)
```

```
[1] 26048    15
```

Hide

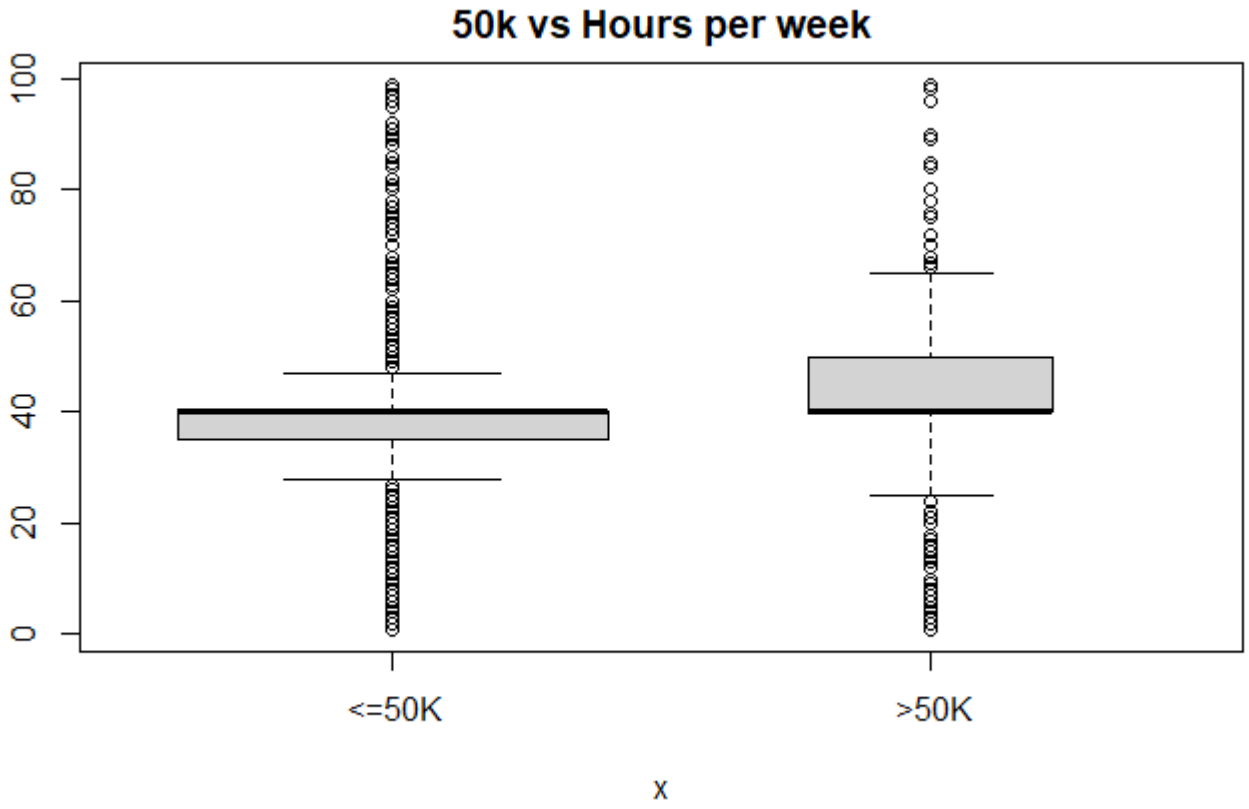
```
head(train, 1)
```

…	workclass	fnlwgt	education	education-num	marital-status	occupation	…
<int>	<chr>	<int>	<chr>	<int>	<chr>	<chr>	
7452 53	Private	238481	Assoc-voc	11	Married-civ-spouse	Exec-managerial	I

1 row | 1-9 of 15 columns

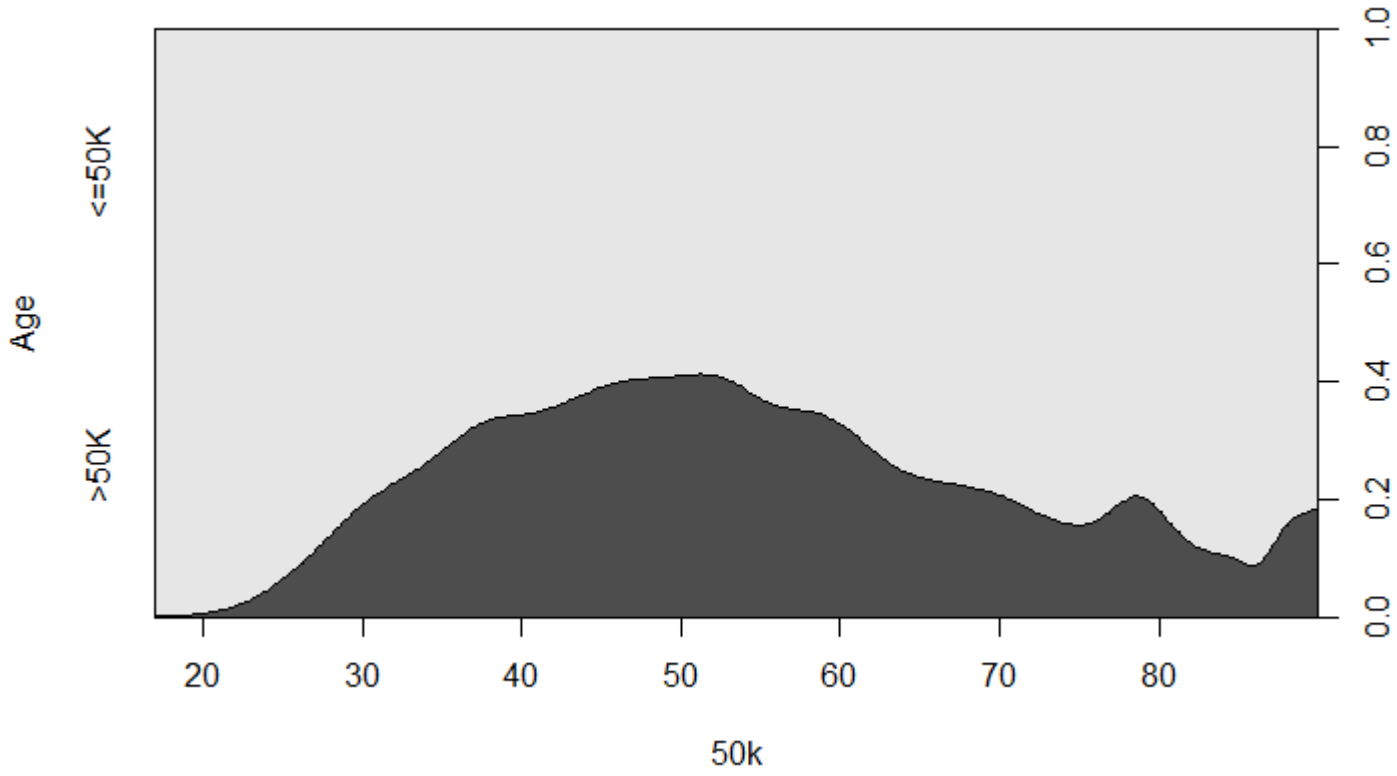
Hide

```
plot(train$`50k`, train$`hours-per-week`, main="50k vs Hours per week", ylab="", varwidth=TRUE)
```



Hide

```
cdplot(train$age, train$`50k` , ylab = "Age", xlab = "50k")
```



Logistic Regression

In this part we see how the education people has effects their income and whether or not that income is greater or less than 50k.

Hide

```
glm1 <- glm(`50k`~education, data=train, family=binomial)
summary(glm1)
```

```

Call:
glm(formula = `50k` ~ education, family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6688  -0.6586  -0.5953  -0.3015   2.4954

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)             -2.799831   0.157068 -17.826  < 2e-16 ***
education 11th          -0.071472   0.214016  -0.334    0.738
education 12th           0.308534   0.257520   1.198    0.231
education 1st-4th       -0.268222   0.446194  -0.601    0.548
education 5th-6th       -0.208324   0.334841  -0.622    0.534
education 7th-8th        0.008665   0.245048   0.035    0.972
education 9th           -0.168017   0.277923  -0.605    0.545
education Assoc-acdm     1.726337   0.175816   9.819  < 2e-16 ***
education Assoc-voc      1.727567   0.171552  10.070  < 2e-16 ***
education Bachelors      2.453391   0.160063  15.328  < 2e-16 ***
education Doctorate      3.906640   0.202725  19.271  < 2e-16 ***
education HS-grad        1.159317   0.159846   7.253 4.08e-13 ***
education Masters        3.042968   0.166060  18.324  < 2e-16 ***
education Preschool    -10.766236  80.716423  -0.133    0.894
education Prof-school    3.807846   0.188196  20.233  < 2e-16 ***
education Some-college   1.381698   0.160511   8.608  < 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: 28932  on 26047  degrees of freedom
Residual deviance: 25528  on 26032  degrees of freedom
AIC: 25560

Number of Fisher Scoring iterations: 12
```

Hide

```
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, 1, 0)
acc <- mean(pred==as.integer(test$`50k`))
print(paste("accuracy = ", acc))
```

```
[1] "accuracy =  0.0310196560019656"
```

Hide

```
table(pred, test$`50k`)
```

```
pred  <=50K  >50K
   0   4818  1180
   1    202   312
```

Our Data here shows that for some of the education backgrounds there is a strong correlation between that and the amount of money that they make annually. However the low accuracy means that it is very difficult for a single line to be made that represents the entirety of the data.

Naive Bayes Model

In this part we use a Bayes model to see the different probabilities and how properly the data represents the population as a whole.

Hide

```
library(e1071)
nb1 <- naiveBayes(`50k`~., data=train)

p1 <- predict(nb1, newdata=test, type="class")
table(p1, test$`50k`)
```

```
p1        <=50K  >50K
   <=50K   4690   720
   >50K     330   772
```

Hide

```
mean(p1==test$`50k`)
```

```
[1] 0.8387592
```

Hide

```
p1_raw <- predict(nb1, newdata=test, type="raw")
head(p1_raw)
```

```
            <=50K         >50K
[1,] 9.170158e-01 8.298422e-02
[2,] 4.546486e-01 5.453514e-01
[3,] 6.337766e-42 1.000000e+00
[4,] 1.043668e-06 9.999990e-01
[5,] 3.340485e-01 6.659515e-01
[6,] 9.999892e-01 1.076255e-05
```

The data here shows that the Bayes model has a very high degree of accuracy and can accurately predict the correct income in the vast majority of cases.

The difference between the accuracy of the Bayes model and the linear regression is very high with the linear regression being .03 and the Bayes being .83. This shows that the Naive Bayes model is much more accurate for this set of data and can find an accurate prediction model that is much better than the linear regression. This does not mean that Bayes is better in every situation but in this one Bayes has a better time creating predictions.