Classification

Aloksai Choudari, Alekhya Pinnamaneni

2022-10-08

Select a dataset

Data set: Adults

Source: https://archive.ics.uci.edu/ml/datasets/Adult

Load in the data

```
adult <- read.csv("adult.data", header=FALSE)

# Adds columns names to the data table
colnames(adult) <- c('age', 'workclass', 'fnlwgt', 'education', 'education_num', 'marital_status', 'occu
```

Data Cleaning

```
# Changes the character columns to numeric columns
adult$workclass <- as.numeric(as.integer(factor(adult$workclass)))
adult$education <- as.numeric(as.integer(factor(adult$education)))
adult$marital_status <- as.numeric(as.integer(factor(adult$marital_status)))
adult$occupation <- as.numeric(as.integer(factor(adult$occupation)))
adult$relationship <- as.numeric(as.integer(factor(adult$relationship)))
adult$race <- as.numeric(as.integer(factor(adult$race)))
adult$sex <- as.numeric(as.integer(factor(adult$sex)))
adult$native_country <- as.numeric(as.integer(factor(adult$native_country)))
adult$predicted_salary_range <- as.numeric(as.integer(factor(adult$predicted_salary_range)))
```

Split data into train and test data

```
set.seed(1234)
sample <- sample(1:nrow(adult), nrow(adult)*0.8, replace=FALSE)
train <- adult[sample,]
test <- adult[-sample,]
```

Explore the training data statistically and graphically

```r
# Prints the first 10 rows of the train data for adults
head(train, n=10)
```

```
##       age workclass fnlwgt education education_num marital_status occupation
## 7452   17         5 110798         2             7              5         13
## 8016   34         5 202450        12             9              3          4
## 7162   24         5 259351        16            10              5          4
## 8086   67         1  81761        12             9              1          1
## 23653  25         5 109532         3             8              5          4
## 9196   24         5 237928        10            13              5         11
## 623    65         5 109351         7             5              7         10
## 15241  44         5 368757        16            10              3          8
## 10885  45         5 189225        12             9              5          9
## 934    23         5 375871        12             9              3          2
##       relationship race sex capital_gain capital_loss hours_per_week
## 7452             4    5   1            0            0             20
## 8016             1    5   2            0            0             55
## 7162             5    1   2            0            0             40
## 8086             4    5   2            0            0             20
## 23653            4    5   2            0            0             40
## 9196             2    5   2            0            0             39
## 623              5    3   1            0            0             24
## 15241            1    5   2            0            0             40
## 10885            5    3   1            0            0             40
## 934              6    5   1            0            0             40
##       native_country predicted_salary_range
## 7452              40                      1
## 8016              40                      2
## 7162              27                      1
## 8086              40                      1
## 23653             40                      1
## 9196              40                      1
## 623               40                      1
## 15241             40                      1
## 10885             40                      1
## 934               27                      1
```

```r
# Prints the mean of education_num
mean(train$education_num)
```

```
## [1] 10.08561
```

```r
# Prints the median of hours worked per week for adults
median(train$hours_per_week)
```

```
## [1] 40
```

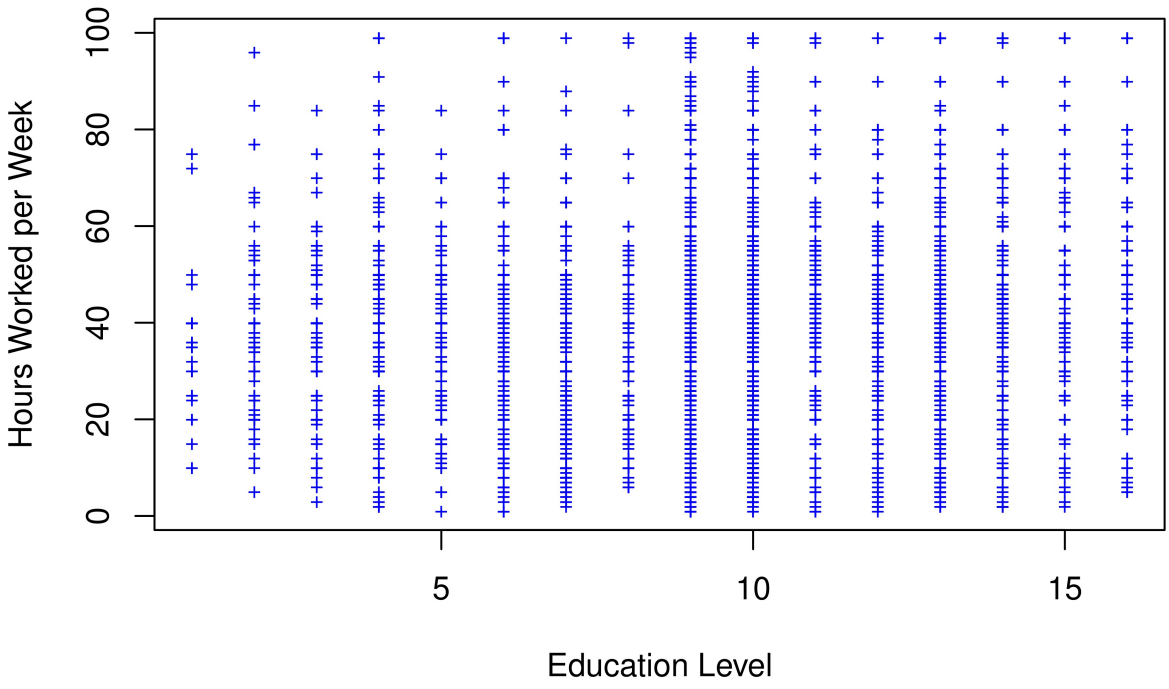
```r
# Prints the smallest and largest capital_gain across the adults
range(train$capital_gain)
```

```
## [1]     0 99999
```

```r
# Prints statistics for the age across the adults data
summary(train$age)
```

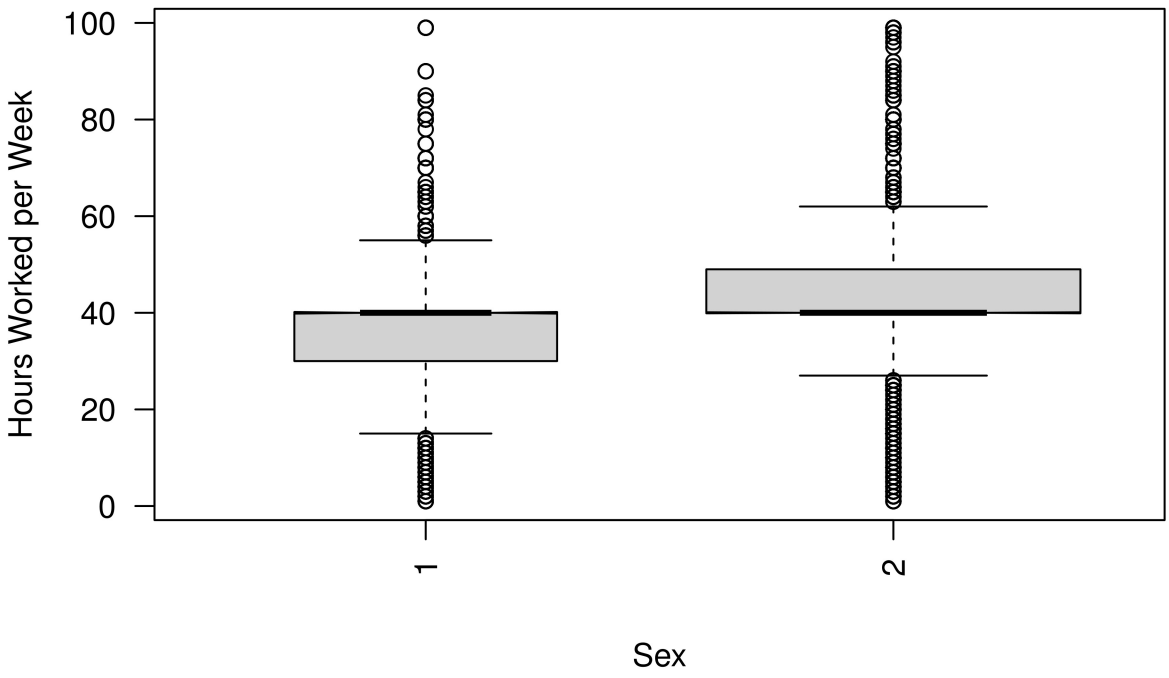
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   28.00   37.00   38.59   48.00   90.00
```

```r
# Scatterplot of education level vs. hours worked per week
plot(train$education_num, train$hours_per_week, pch='+', cex=0.75, col="blue", xlab="Education Level", y
```




```r
# Boxplot of hours worked per week based on sex
boxplot(train$hours_per_week~train$sex, varwidth=TRUE, notch=TRUE, xlab="Sex", ylab="Hours Worked per W
```

```
## Warning in (function (z, notch = FALSE, width = NULL, varwidth = FALSE, : some
## notches went outside hinges ('box'): maybe set notch=FALSE
```



Perform Logistic Regression

```
# Find model
glm1 <- glm(predicted_salary_range~education_num, data=train)
# Predict results
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>1.5, 2, 1)
# Calculate accuracy of results
acc <- mean(pred==as.integer(test$predicted_salary_range))
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.772301550744665"
```

Perform kNN for Classification

```
library(class)
# Find model
knn1 <- knn(train=train[, 1:14], test=test[, 1:14], cl=train[, 15], k=3)
# Predict results
results2 <- knn1 == test[, 15]
# Calculate accuracy of results
```

```
acc2 <- length(which(results2==TRUE)) / length(results2)
print(paste("accuracy = ", acc2))
```

```
## [1] "accuracy =  0.753723322585598"
```

Perform Decision trees

```
library(tree)
# Find model
tree1 <- tree(as.factor(predicted_salary_range)~., data=train)
# Predict results
pred2 <- predict(tree1, newdata=test, type="class")
# Calculate accuracy of results
acc3 <- mean(pred == test$predicted_salary_range)
print(paste("accuracy = ", acc3))
```

```
## [1] "accuracy =  0.772301550744665"
```

Analysis

The logistic regression algorithm had the same accuracy as the decision trees algorithm. The decision tree algorithm involves splitting the observations into smaller and smaller partitions until the observations in a group are similar. Meanwhile, logistic regression uses methods like gradient descent to iteratively find the most optimal linear model. Since both of these algorithms involve taking increasingly smaller steps towards the most optimal solution, it is understandable as to why they have the same accuracy. However, the kNN algorithm had a slightly lower accuracy than the logistic regression and decision tree algorithms. This can be explained by the fact that the kNN algorithm takes a more indirect approach by predicting the class of a data instance based on its neighbors' classes. This explains why the kNN aglorithm had a slightly lower accuracy.