STATS769 Lab02

Yujie Zhang 130011770 yzhb915

September 1, 2019

The data are again electronic bicycle or scooter trips, but we will access the data from the original source: data.austintexas.gov, the official City of Austin data portal.The "Dockless Vehicle Trips" data set that we have been using has its main page here: https://data.austintexas.gov/Transportation-and-Mobility/Dockless-Vehicle-Trips/7d8e-dm7r We will use the API that is described here: https://dev.socrata.com/foundry/data.austintexas.gov/7d8e-dm7r NOTE: you will need to register for an app token in order to use the API.

1. Scrape data from the City of Austin data portal and create a data frame with two columns: trip_distance and trip_duration. You should extract 10,000 scooter trips that occurred in 2018 (your API request must specify records with year equal to 2018 and vehicle_type equal to "scooter").

   library(httr)

   library(jsonlite)

   token <- "MMXi47FhZZ4LYZtvJF7HyFa5S"

   requestUrl <- "https://data.austintexas.gov/resource/7d8e-dm7r.json?vehicle_type=scooter&year=2018&$limit=10000&$select=trip_duration,trip_distance"

   htmlResponse <- GET(requestUrl, add_headers('X-App-Token' = token))

   response <- content(htmlResponse,"text")

   data_json <- fromJSON(response)

   trip_distance <- as.numeric(data_json$trip_distance)

   trip_duration <- as.numeric(data_json$trip_duration)

   trips <- data.frame(trip_distance, trip_duration)

   write.csv(trips, "/Users/yujzhang/Documents/scotter_trips.csv")

2. Subset only trips with non-negative distances and durations, log the durations, and create a new "long trip" variable (where "long" means that the trip distance was greater than 1000m).

```
trips <- read.csv("./scotter_trips.csv")
set.seed(123)
subsetIndex <- trips$trip_duration > 0 & trips$trip_distance > 0
subset <- trips[subsetIndex,]

subset['long trip'] <- subset$trip_distance > 1000

test_index <- sample(1:nrow(subset), round(nrow(subset)*0.2))
test_subset <- subset[test_index,]
train_subset <- subset[-test_index,]

log_testDuration <- log(test_subset$trip_duration)
log_trainDuration <- log(train_subset$trip_duration)

test_longtrip <- test_subset$`long trip`
```

3. Fit a logistic regression model to the training data and calculate the accuracy of the model on the test data.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
fitGLM <- glm(y ~ x, data.frame(x=log_trainDuration, y=train_subset$`long trip`), family="binomial",na.a
predGLM <- predict(fitGLM, data.frame(x=log_testDuration), type="response")
glmDiag <- confusionMatrix(factor(predGLM > .5),
                factor(test_subset$`long trip`))
glmDiag$overall["Accuracy"]
```

```
##  Accuracy
## 0.7817552
```

4. Fit a k-nearest neighbours model and calculate its accuracy. You might need to try a few different values of k.

```
library(class)
library(caret)

result = c()
for(i in 1:70)
{
  knnPred <- knn(matrix(log_trainDuration, ncol=1),
  matrix(log_testDuration, ncol=1),train_subset$`long trip`,k=i, prob=TRUE)
  knnProb <- ifelse(knnPred == TRUE,attr(knnPred, "prob"),1 - attr(knnPred, "prob"))
  knnDiag <- confusionMatrix(knnPred, factor(test_subset$`long trip`))
  result[i] = knnDiag$overall["Accuracy"]
}

which.max(result)
```

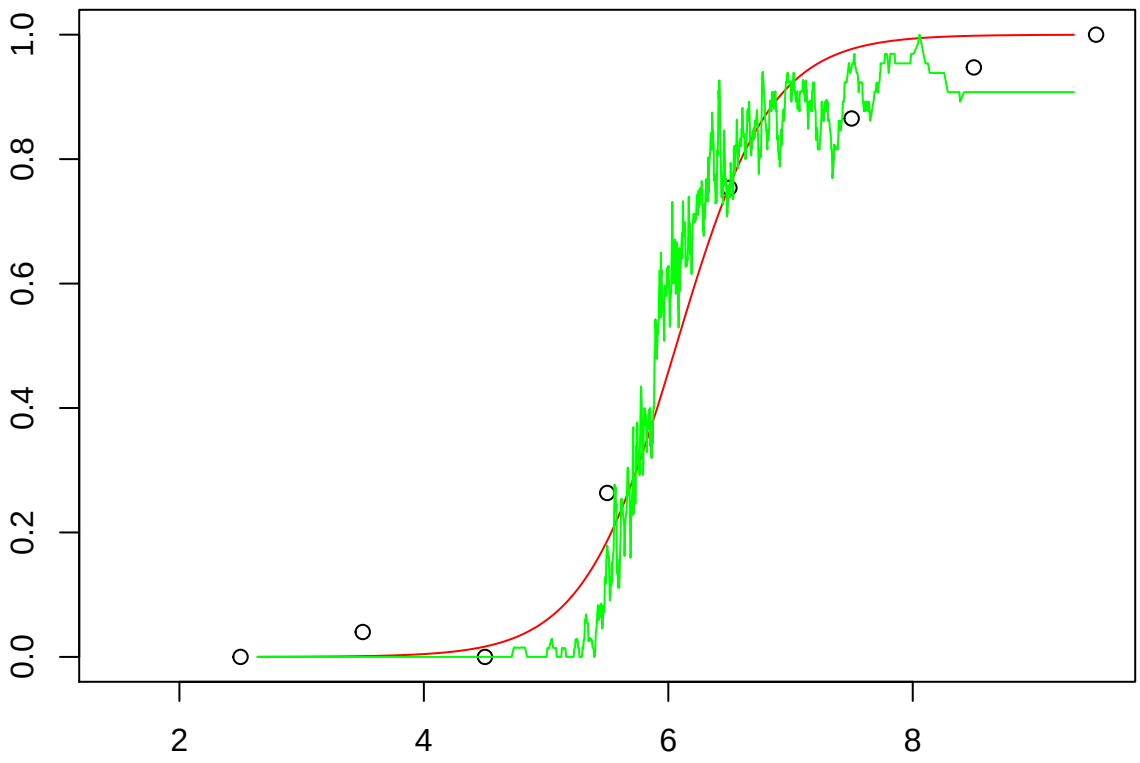
```
## [1] 65
```

```
knnPred1 <- knn(matrix(log_trainDuration, ncol=1),
matrix(log_testDuration, ncol=1),train_subset$`long trip`,k=which.max(result), prob=TRUE)
knnProb1 <- ifelse(knnPred1 == TRUE,attr(knnPred1, "prob"),1 - attr(knnPred1, "prob"))
knnDiag1 <- confusionMatrix(knnPred1, factor(test_subset$`long trip`))
knnDiag1$overall["Accuracy"]
```

```
##  Accuracy
## 0.7938799
```

5. Produce a plot based on the test data that shows the predictions from the logistic regression model and the predictions from the k-nearest neighbours model.

```
par(mar=c(3, 3, 2, 2))
breaks = seq(1,10,1)
midbreaks <- breaks[-1] - diff(breaks)/2
props <- tapply(test_longtrip, cut(log_testDuration, breaks=breaks), mean)
plot(midbreaks, props)
o <- order(log_testDuration)
lines(log_testDuration[o], predGLM[o], col="red")
lines(log_testDuration[o], knnProb1[o], col="green")
```



Conclusion

The data for this exercise came from website. We made http requests to get the data in json fromat and then extracted the data of trips for analyse. We created logistic model and knn model to predict the probability of long trip with different parameters.And then we caculated the accuracy of the models to evaluate.The accuracy of logistic model is about 0.79 and it is about 0.81 for knn model. Finally, we created a plot to display the models and it seems the two models are generating almost the same result.