2020

ST447 Project

DRIVING TEST PASS RATE PREDICTION

CANDIDATE NUMBER: 15688

XYZ Profile

The profile of XYZ:

- Age: 17

- Gender: Male

- Home address: Edinburgh (Currie)

Executive Summary

This report aims to predict the probability of passing the driving test at different test centers, i.e. the nearest test center to XYZ's home and the nearest test center to LSE, by analyzing the dataset DVSA1203 held by the Driver and Vehicle Standards Agency. After cleaning and reformatting the datasets, a permutation test is used to test the null hypothesis that pass rates at different test centers are the same, and logistic regression is implemented to predict the pass rate while linear discriminant analysis is used as well for comparison purpose.

Data Processing

After autofilling the locations of the test centers via Excel and basic data cleaning with R, i.e. removing missing values and empty columns, and renaming the columns, an additional feature 'year' is added to each dataframe before combing all the sheets in the dataset. As it is widely believed that the driving test routes around some centers are probably more difficult than others, only the data for Wood Green (London) and Edinburgh (Currie) is selected for further analysis.

```
'data.frame':   34704 obs. of  12 variables:
 $ location       : chr  "Aberdeen North" "Aberdeen North" "Aberdeen North" "Aberdeen North" ...
 $ age            : num  17 18 19 20 21 22 23 24 25 17 ...
 $ conducted-m    : num  304 164 105 92 76 59 61 66 59 384 ...
 $ pass-m         : num  221 93 67 57 44 35 44 36 29 271 ...
 $ pass rate-m    : num  72.2 56.7 63.8 62 57.9 ...
 $ conducted-f    : num  350 197 121 93 105 100 101 90 67 348 ...
 $ pass-f         : num  205 103 67 58 52 45 51 42 36 209 ...
 $ pass rate-f    : num  58.6 52.3 55.4 62.4 49.5 ...
 $ conducted-total: num  653 361 226 185 181 159 162 156 126 732 ...
 $ pass-total     : num  425 196 134 115 96 80 95 78 65 480 ...
 $ pass rate-total: num  65 54.3 59.3 62.2 53 ...
 $ year           : num  2018 2018 2018 2018 2018 ...
```

Figure 1 Information about the cleaned dataset.

Empirical Results and Analysis

1. Permutation Test

Permutation test is a non-parametric method if two distributions are the same without

assumptions on the distribution and parameters. In this case, permutation test is used to test if the pass rate at Wood Green (London) and Edinburgh (Currie) are the same. As the impact of testing routes would influence all the age groups, both genders, and all the time, total pass rates are used in this hypothesis testing.

Let $\theta_1, \dots, \theta_{54}$ be the total pass rates at Wood Green (London) and $p_1, \dots, p_{108}$ be the total pass rates at Edinburgh (Currie). Under H_0, $\{\theta_1, \dots, \theta_{54}, p_1, \dots, p_{108}\}$ form a sample of size 162 from the same distribution. The test statistic T chosen is $T = |\bar{p} - \bar{\theta}|$. The approximate p-value is $\frac{1}{5000} \Sigma_{i=1}^{5000} I(T_i > t_{obs})$, where t_{obs} is the observed value for the test statistic, and T_i is the calculated test statistic for each permutation. p-value is approximately 0 in this case, which means that it's highly unlikely to get the observed t_{obs} or higher values under null hypothesis, thus null hypothesis is rejected. We conclude that it is indeed more difficult to pass the driving test in Wood Green (London).

```
H_0: the pass rate at Wood Green (London) and Edinburgh (Currie) are the same.
H_1: the pass rate at Edinburgh (Currie) is higher than Wood Green (London)
```{r}
pass_l = london[['pass rate-total']]
pass_e = edinburgh[['pass rate-total']]

tobs = abs(mean(pass_e)-mean(pass_l)) #observed test statistic
pass = c(pass_l, pass_e) #under null hypothesis, pass_l and pass_e follows the same dist.

k = 0
for (i in 1:5000){
  zp = sample(pass, 162)
  T = abs(mean(zp[55:162])- mean(zp[1:54]))
  if (T > tobs) k = k+1
}
cat("p-value: ", k/5000, '\n')
```

```
p-value:  0
```

Figure 2 Relevant code for permutation test.

2. Logistic Regression

The dataset in reformatted to a dataframe with 5 variables, i.e. gender, age, year, location and pass, for individual candidates before implementing logistic regression. Among these variables, location, gender and pass are categorical variables represented by 0 and 1, and pass is the response binary variable.
```

```
location       age             year      pass       gender
0:48867   Min.   :17.00   Min.   :2007   0:38382   0:37370
1:23728   1st Qu.:18.00   1st Qu.:2011   1:34213   1:35225
          Median :19.00   Median :2014
          Mean   :19.99   Mean   :2013
          3rd Qu.:22.00   3rd Qu.:2016
          Max.   :25.00   Max.   :2018
```

Figure 3 Information about reformatted dataset

Logistic regression is used to model the probability that Y belongs to a particular class, pass or fail in our case. Let $P(Pass \mid age, location, year, gender) = P(Y = 1 \mid X) = p(X)$. Then under logistic regression,

$$log\frac{P(x)}{1-P(x)} = \beta_0 + \beta_1 * location + \beta_2 * age + \beta_3 * year + \beta_4 * gender$$

```
# fit the logistic regression
set.seed(929)
n = length(df_log$location)
train = sample(1:n, n/2) #index for training data set

fit = glm(pass~., data = df_log, subset = train, family = 'binomial')
summary(fit)
```

```
Call:
glm(formula = pass ~ ., family = "binomial", data = df_log, subset = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3437  -1.1236  -0.9571   1.2031   1.4968

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -55.745200   7.037946  -7.921 2.36e-15 ***
location1    -0.364102   0.025511 -14.273  < 2e-16 ***
age          -0.048935   0.004203 -11.643  < 2e-16 ***
year          0.028121   0.003498   8.039 9.07e-16 ***
gender1       0.211976   0.021251   9.975  < 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: 50186  on 36296  degrees of freedom
Residual deviance: 49652  on 36292  degrees of freedom
AIC: 49662

Number of Fisher Scoring iterations: 4
```

Figure 4 Code and result of logistic regression

According to the result of logistic regression in Fig 4, it's more difficult to pass the driving test in London, which can be shown by the negative coefficient of location1. Also, the pass rate increases over time indicated by the positive correlation between the pass rate and year.

The pass rates predicted by this logistic regression model are 52.9% and 60.8% for taking the test in Wood Green (London) and Edinburgh (Currie) respectively, which aligns with the result of the permutation test.

Besides, bootstrap, a non-parametric computational method for estimating standard errors and confidence intervals, is used to estimate the confidence intervals for estimated pass rates. The logistic regression model is refitted 1000 times with 1000 bootstrap samples of the original datasets. And the confidence intervals are approximated with predictions with these models. The percentile interval for pass rate in London is [0.503, 0.523] while that for pass rate in Edinburgh is [0.590, 0.611]. Also, the probability of passing a driving test in Edinburgh is much higher than in London as clearly shown in Fig 5.

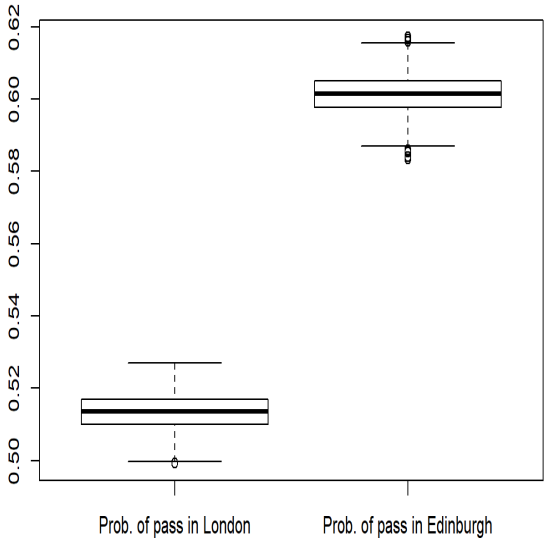

Figure 5 Boxplot of estimated pass rates

However, the accuracy rate of this logistic regression model is only 55% calculated by cross validation. This rate is similar to the accuracy rate of linear discriminant analysis that is used for comparison purpose in our case. This means that the reliability of the predicted probability by this logistic regression is questionable.

Conclusion

Based on the results of both the permutation test and logistic regression, the conclusion that XYZ should take the driving test in Edinburgh (Currie) can be easily drawn. However, the logistic model is inadequate as $\frac{D}{n-p} = 1.368 \gg 1$, where D is the residual deviance, p is the number of regressors and n is the number of observations, according to *Introduction to Linear Regression Analysis by Montgomery, Peck, Vining, 5E* and the high AIC value (49662). This inadequacy can be explained by the limitations of the scope of regressors and can be probably resolved by adding more relevant variables such as time spent on practicing.

Code used for this project

```
##### library the packages needed for data cleaning #####
library(readODS); library(dplyr); library(purrr); library(janitor)

##### import the dataset and data cleaning #####
path = 'dvsa1203.ods'
sheets = ods_sheets(path)
data = map(seq_along(sheets), read_ods, path = path, skip = 6) %>%
   set_names(sheets)

##### clean the data #####
for (df in data[-1]) print(dim(df)) # check the dim of each dataframes
data1 = data[-1] #remove the note
DataClean = function(df){
   df = remove_empty(df, which = 'cols') #remove empty columns
   # print(dim(df))
   df = na.omit(df) #remove NA values
   names(df) = col_name #rename the columns
   df[df == '..'] = NA # replace .. with NA
   df = cbind(df['location'], sapply(df[,-1], function(x) round(as.numeric(x), digits = 4)))
   #convert the columns to numeric and round to three decimal palces
   return (df)
}
data1 = lapply(data1, DataClean)

# creat a column of year in each dataset
year = seq(2018, 2007, -1)
for(i in 1:length(year)){
   data1[[i]]['year'] = year[i]
}
driving = do.call(rbind, data1) #combine all the dataframes
driving = driving[order(driving$location), ] #sort the dataframes according to location
row.names(driving) = NULL

##### Permutation test #####
london = driving[which(driving$location == 'Wood Green (London)'),]
edinburgh = driving[which(driving$location == 'Edinburgh (Currie)'), ]
edinburgh = na.omit(edinburgh) # remove the NA value in Edinburgh dataset
pass_l = london[['pass rate-total']]
pass_e = edinburgh[['pass rate-total']]

tobs = abs(mean(pass_e)-mean(pass_l))
pass = c(pass_l, pass_e)
```

```
k = 0
for (i in 1:5000){
   zp = sample(pass, 162)
   T = abs(mean(zp[55:162])- mean(zp[1:54]))
   if (T > tobs) k = k+1
}
cat("p-value: ", k/5000, '\n')

##### logistic regression #####
# construct a dataframe with individual driving test data instead of counts of each group
# combine london and edinburgh dataframes together and convert location to dummy variable
df = rbind(edinburgh, london)
df['location'] =as.factor(ifelse(df$location == 'Wood Green (London)', 1, 0))

#convert counts to individuals
### data for male candidates
df_male = df[,c(1:4, 12)]
df_male1 = df_male[rep(seq_len(nrow(df_male)), df_male$`pass-m`),] #data for passed individual
df_male1['pass'] = 1; df_male1['gender'] = 1
df_male1$`conducted-m` = NULL; df_male1$`pass-m` = NULL
df_male2 = df_male[rep(seq_len(nrow(df_male)), df_male$`conducted-m`- df_male$`pass-m`),]
df_male2['pass'] = 0; df_male2['gender'] = 1
df_male2$`conducted-m` = NULL; df_male2$`pass-m` = NULL
df_male = rbind(df_male1, df_male2)

### data for female candidates
df_female = df[,c(1:2, 6:7, 12)]
df_female1 = df_female[rep(seq_len(nrow(df_female)), df_female$`pass-f`),] #data for passed individual
df_female1['pass'] = 1; df_female1['gender'] = 0
df_female1$`conducted-f` = NULL; df_female1$`pass-f` = NULL
df_female2 = df_female[rep(seq_len(nrow(df_female)), df_female$`conducted-f`- df_female$`pass-f`),]
df_female2['pass'] = 0; df_female2['gender'] = 0
df_female2$`conducted-f` = NULL; df_female2$`pass-f` = NULL
df_female = rbind(df_female1, df_female2)
df_log = rbind(df_male, df_female)
df_log['gender'] = as.factor(df_log$gender); df_log['pass'] = as.factor(df_log$pass)

# shuffle the data
set.seed(929)
index = sample(nrow(df_log))
df_log = df_log[index, ]
summary(df_log)
```

```
# fit the logistic regression
set.seed(929)
n = length(df_log$location)
train = sample(1:n, n/2) #index for training dataset
fit = glm(pass~., data = df_log, subset = train, family = 'binomial')
summary(fit)

# predict the value of testing data set
pred_pass = predict(fit, newdata = df_log[-train, ], type = "response")
pass_status = rep(0, n/2)
pass_status[pred_pass>0.5] = 1
table(pass_status, df_log[-train, ]$pass) #confusion matrix

accuracy = (13461+6530)/(n/2) #calculate the accuracy rate of logistic regression
accuracy

##### calculate the prob. of passing the driving test with this logistci model #####
xyz = data.frame('age' = c(17, 17),
                 'gender' = as.factor(c(1, 1)),
                 year = c(2020, 2020),
                 location = as.factor(c(1, 0)))

pass_rate_london = predict(fit, newdata = xyz[1,], type = 'response')
pass_rate_edinb = predict(fit, newdata = xyz[2,], type = 'response')
cbind(pass_rate_london, pass_rate_edinb)

##### use bootstrap to calculate the confidence interval for the prob. of passing #####
pass_lon = 1:1000; pass_edin = 1:1000
for (i in 1:1000){
   train.i = sample(1:n, n/2)
   fit.i = glm(pass~., data = df_log, subset = train.i, family = 'binomial')
   pass_lon[i] = predict(fit.i, newdata = xyz[1,], type = 'response')
   pass_edin[i] = predict(fit.i, newdata = xyz[2,], type = 'response')
}
boxplot(pass_lon, pass_edin, names = c("Prob. of pass in London", "Prob. of pass in Edinburgh"))

pass_lon = sort(pass_lon); pass_edin = sort(pass_edin)
i = as.integer(0.025* 1000)
cat("95% Bootstrap confidence intervals for estimated pass rate in London", "\n")
cat("Percentile interval:", pass_lon[i], pass_lon[1000-i], "\n")
cat("95% Bootstrap confidence intervals for estimated pass rate in Edinburgh", "\n")
cat("Percentile interval:", pass_edin[i], pass_edin[1000-i], "\n")

v = 49652/(n/2-4) # adequacy of the logistic model
```