ST447: SUMMATIVE PROJECT 2021-22

Candidate Number 39719

December 3rd 2021

INTRODUCTION :

XYZ has been learning to drive for a while and is considering taking the practical automobile test in the United Kingdom.There are two viable options:

1. Take the practical test at the nearest test centre to his or her residence
2. Take it at the LSE's nearest exam centre, i.e. Wood Green

PROFILE GENERATION :

The profile of XYZ :

Age: 21 | Gender: Male | Home address: Tolworth (London)

DATA PREPARATION :

The Data was extracted for both locations, i.e., Tolworth and Wood Green for a 7-year period. The data for the past 7 years only has been used mainly because there has been a significant change in modern automobiles in terms of driver and passenger safety features that have transformed the way modern vehicles are driven.

Also, as per this notification from UK Government, there were major changes in the way tests are conducted 2017 onwards, so much of the data for previous years has been omitted and only the data from recent years is used for our analysis.

The data preparation was done in Excel entirely and the following transformations have been followed to manipulate the data for fitting the model -

Gender	Value Specified
Male	1
Female	0

Outcome	Value Specified
Pass	1
Fail	0

Location	Value Specified
Tolworth	1
Wood Green	0

CREATING A DATAFRAME :

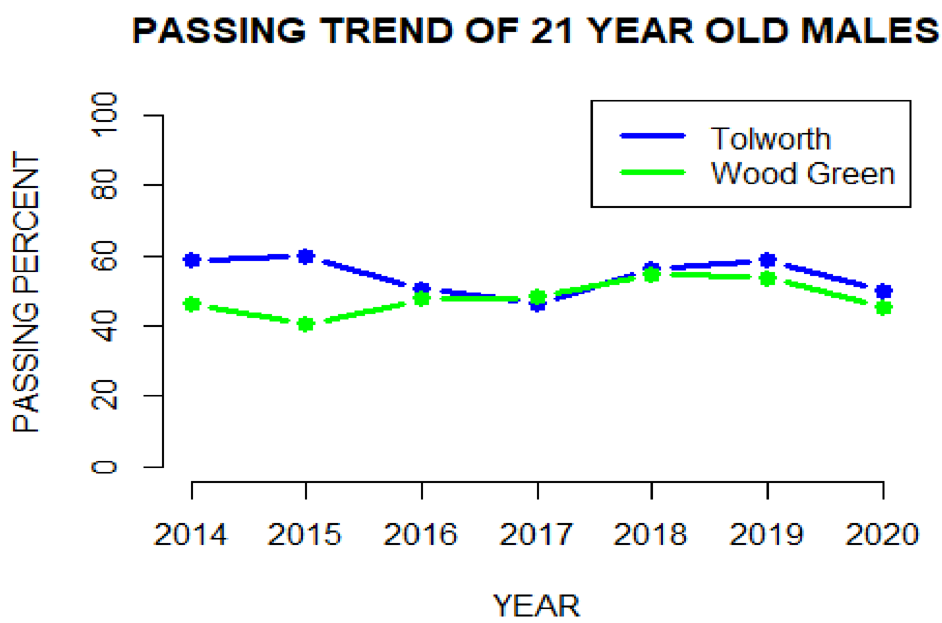
```
#FIRST LOAD THE DATA INTO A VARIABLE
combined_data = read.csv("CombinedData.csv", header = TRUE)

#SEE THE EXTRACTED DATA AND ITS STRUCTURE
head(combined_data)

##   SNO. YEAR AGE OUTCOME AGECAT GENDER LOC
## 1    1 2020  17       1      1      1   1
## 2    2 2020  17       1      1      1   1
## 3    3 2020  17       1      1      1   1
## 4    4 2020  17       1      1      1   1
## 5    5 2020  17       1      1      1   1
## 6    6 2020  17       1      1      1   1

str(combined_data)

## 'data.frame':    57687 obs. of  7 variables:
##  $ SNO.   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ YEAR   : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ AGE    : int  17 17 17 17 17 17 17 17 17 17 ...
##  $ OUTCOME: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ AGECAT : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ GENDER : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ LOC    : int  1 1 1 1 1 1 1 1 1 1 ...

#CHECK FOR MISSINNG AND NA VALUES
nrow(combined_data[is.na(combined_data)])

## NULL
```

Great! We have no missing values in our dataset.

DATA VISUALIZATION

```r
# INDEXING DATA FOR VISUALIZATION

#First, lets extract data for 21 year old males in Tolworth and create a data
frame from it
criterion1 =(combined_data$LOC == 1) & (combined_data$GENDER ==1) &
(combined_data$AGE == 21)
df_c1 = data.frame(combined_data[criterion1,])

#Now lets create a new data frame of mean passing rates in Tolworth
mean_tolworth = data.frame(aggregate(df_c1$OUTCOME, list(df_c1$YEAR),
FUN=mean))
colnames(mean_tolworth)<- c("Year", "Pass Percentage")

#Lets convert this mean value to a percent value
mean_tolworth$`Pass Percentage` = mean_tolworth$`Pass Percentage`*100
head(mean_tolworth)

##   Year Pass Percentage
## 1 2014        58.69565
## 2 2015        60.09852
## 3 2016        50.44643
## 4 2017        46.45669
## 5 2018        56.06061
## 6 2019        58.97436

#Next, we index data for 21 year old males in Wood Green and create its data
frame
criterion2 =(combined_data$LOC == 0) & (combined_data$GENDER ==1) &
(combined_data$AGE == 21)
df_c2 = data.frame(combined_data[criterion2,])

#Again, we create a new data frame of mean passing rates in Wood Green
mean_woodgreen = data.frame(aggregate(df_c2$OUTCOME, list(df_c2$YEAR),
FUN=mean))
colnames(mean_woodgreen)<- c("Year", "Pass Percentage")
mean_woodgreen$`Pass Percentage` = mean_woodgreen$`Pass Percentage`*100
head(mean_woodgreen)

##   Year Pass Percentage
## 1 2014        46.50206
## 2 2015        40.62500
## 3 2016        47.74775
## 4 2017        48.14815
## 5 2018        54.80226
## 6 2019        53.84615

#Finally, lets plot this on the graph and see any trends
plot(mean_tolworth, xlab = "YEAR", ylab = "PASSING PERCENT", col = "blue",
     type = "b" , main = "PASSING TREND OF 21 YEAR OLD MALES", lwd =3,
     bty = "n", ylim = c(0,100), pch = 19 )
```

```r
#Lets add Wood Green data on this and add a legend
lines(mean_woodgreen, col = "green", lwd = 3, type = "b", pch = 19)

#Lastly, we add a legend to our plot
legend(x = "topright", legend = c("Tolworth","Wood Green"), col =
c("blue","green"), lwd = 3)
```



MODELLING THE DATA :

STATISTICAL METHOD USED - MULTIPLE LOGISTIC REGRESSION

Since our variables are categorical in nature and the model needs to tell us the best possible choice out of two options, it is best to try fitting the model using a logistic regression.

```r
#We need to convert some of our data points to factors before we model them
combined_data$OUTCOME <- as.factor(combined_data$OUTCOME) #To be predicted,
dependent variable

#We use the Generalized Linear Model function in R to do the regression on
the combined data.
CO_MODEL = glm(OUTCOME ~ AGE + GENDER + LOC ,data = combined_data, family =
binomial(link = logit))

#Now lets see the model results:
summary(CO_MODEL)

##
## Call:
## glm(formula = OUTCOME ~ AGE + GENDER + LOC, family = binomial(link =
logit),
##     data = combined_data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3256  -1.1407  -0.9479   1.1800   1.4438
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.40524    0.07364   5.503 3.73e-08 ***
## AGE         -0.04051    0.00346 -11.709  < 2e-16 ***
## GENDER       0.27744    0.01683  16.486  < 2e-16 ***
## LOC          0.34784    0.01754  19.831  < 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: 79899  on 57686  degrees of freedom
## Residual deviance: 78894  on 57683  degrees of freedom
## AIC: 78902
##
## Number of Fisher Scoring iterations: 4
```

MODEL INTERPRETATION :

We see that the p-values for each of our variables is less than 0.05, indicating that each of them is indeed significant for our model! However, we would like to have the odds ratio and 95% confidence interval, instead of the log-transformed coefficient. Hence, we implemented the following code to exponentiate the coefficient:

```
exp(coefficients(CO_MODEL))

## (Intercept)         AGE      GENDER         LOC
##   1.4996640   0.9602997   1.3197507   1.4160056

exp(confint(CO_MODEL))

##                 2.5 %    97.5 %
## (Intercept) 1.2981198 1.7325286
## AGE         0.9538083 0.9668318
## GENDER      1.2769362 1.3640169
## LOC         1.3681629 1.4655437
```

The above figures can be better understood with the following table :

Variable	Coefficient	Lower 95	Upper 95
AGE	0.9602997	0.9538083	0.9668318
GENDER	1.3197507	1.2769362	1.3640169
LOC	1.4160056	1.3681629	1.4655437

ODDS RATIO :

Taking Age as an example, after adjusting for all the other variables in the model, the odd ratio is 0.96 with the 95% Confidence interval being 0.954 and 0.966.

Similarly, all other variables constant, the odds ratio for Gender(Male-to-female) is 1.32 with the 95% Confidence interval being 1.277 and 1.364.

Lastly, all other variables constant, the odds ratio for Location(Tolworth-to-WoodGreen) is 1.416 with the 95% Confidence interval being 1.368 and 1.465.

ODDS RATIO AS A PERECNTAGE :

```
#Since odds ratios can be daunting, lets convert them into percentages to
develop a better understanding of these variable relationships.
(exp(CO_MODEL$coefficients[-1])- 1)*100

##       AGE    GENDER       LOC
## -3.970033 31.975072 41.600559
```

> *This figure means that the odds of a candidate passing decrease by 3.97 % for a 1-year increase in Age.*
>
> *Additionally, since our gender coding is as 1 for male and 0 for females, this implies that the odds of males passing are 31.97% more than female's odds.*
>
> *And finally, since our location coding is as 1 for Tolworth and 0 for Wood Green, we can infer that the odds of a candidate passing increase by 41.60 % if they take the test in the Tolworth.*

NOW LET'S PREDICT VALUES FOR OUR DATASET USING OUR OWN MODEL

```
#We store the predicted values in a vector
R = predict(CO_MODEL, newdata = combined_data, type = "response")

#Lets take a look at the head of our predicted values
head(R)

##         1         2         3         4         5         6
## 0.5846413 0.5846413 0.5846413 0.5846413 0.5846413 0.5846413

#Now let's round up these values to compare it to our original model
R$converted.to.binary <- ifelse(R >= 0.5, 1, 0)

#Lets have one final look at our predicted values
head(R$converted.to.binary)

## 1 2 3 4 5 6
## 1 1 1 1 1 1
```

MODEL ACCURACY :

```
#Let's calculate the total predictions that were right and take the mean of
all observations to see the accuracy of our model.

accuracy <- mean((combined_data$OUTCOME) == (R$converted.to.binary))
print(accuracy)

## [1] 0.561409
```

Hence, we see that our model has an accuracy of **56.1408983% !**

EVALUATING BOTH THE OPTIONS :

```
#TEST 1 - SUCCESS RATE FOR TOLWORTH DRIVING CENTER -

friend = data.frame(AGE = 21, GENDER = 1, LOC = 1)
predicted_value_tolworth = predict(CO_MODEL, friend, type = "response")
print(predicted_value_tolworth)

##         1
## 0.5448333

#TEST 2 - SUCCESS RATE FOR WOOD GREEN DRIVING CENTER -

my_guy = data.frame(AGE = 21, GENDER = 1, LOC = 0)
predicted_value_woodgreen = predict(CO_MODEL, my_guy, type = "response")
print(predicted_value_woodgreen)

##         1
## 0.4580926
```

FINAL COMMENTS AND SUGGESTIONS :

We used Multiple Logistic Regression Analysis over categorical variables like Age, Gender and Location of Testing center to conclude the following:

1. XYZ's expected passing rate at the nearest test centre to his home is *54.4833349 %*
2. XYZ's expected passing rate at the nearest test centre to the LSE is *45.8092649 %*
3. Our friend has a better chance of passing the driving test if he gives it in the testing center near his home, i.e., Tolworth.
4. As seen from past data, he has better odds of passing since he is Male.
5. However, his chances decrease by roughly 4% every year he choses not to give the test, so he should give it as soon as possible.

FURTHER IMPROVEMENTS :

STRENGTHS:

- We found that the accuracy of our model is roughly around 56%.
- Logistic Regression requires average or no multicollinearity between independent variables, our variables may be highly correlated to one another due to repetition of values in dataset.
- Logistic Regression assumes that independent variables are linearly related to the log odds (log(p/(1-p)) .
- It not only provides a measure of how appropriate a predictor(coefficient size)is, but also its direction of association (positive or negative).

WEAKNESSES :

- It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm.
- It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.
- This may be further improved by using more variables that might not be available. For instance, do passing rates vary if the vehicle driven has automatic transmission or manual? Since automatic vehicles don't need constant changing of gears, that can have a significant effect on your result.
- We could possibly look into other traditional statistical alternatives to this such as Log-Binomial Regression, Poisson Regression, Cox Regression etc. but each have their own set of drawbacks

CITATIONS:

Car driving test data by test centre - GOV.UK (www.gov.uk)

Find a driving test centre - GOV.UK (www.gov.uk)

United Kingdom driving test - Wikipedia