Academic integrity statement

You must sign this (typing in your details is acceptable) and include it with each piece of work you submit.

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance.

I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance.

I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes.

I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Name	Anupam Bose
Student ID	201570198

MATH5743M: Statistical Learning: Assessed Practical 2 - Brexit

Anupam Bose, 201570198, School of Mathematics

Semester 2 2022

On June 23, 2016, the United Kingdom held a national referendum to determine whether it should leave the European Union ('Brexit'). The result, a win for the Leave campaign, surprised many political commentators, who had expected that people would vote to Remain. Based on the ages, earnings, education, and class of different electoral wards, the Guardian newspaper outlined some apparent demographic tendencies in the vote.

The dataset contains five possible input variables that could have influenced the decision to leave or stay in the European Union. We also have the target variable *voteBrexit* which is answer to the question, "Did the electoral ward vote for Brexit?" Hence, it is either of the two values: TRUE/FALSE.

By evaluating the model and selecting the optimal combination of input variables, we will explore the data using a logistic regression model to determine which input variables are relevant for explaining the output variable. In addition, determining which input variables have a significant impact on the output variable.

The libraries to import to do further analysis on data

```
library(Metrics)
library(MASS)
library(tidyverse)
library(sf)
library(tinytex)
```

The data has been imported into a dataframe (df) by using the **read.csv** function.

```
df = read.csv("brexit.csv")
```

The first 6 rows of our data will give us some idea about it.

```
head(df)
```

```
##        abc1   notBornUK medianIncome medianAge withHigherEd voteBrexit
## 1 0.1336406 0.012605042    0.2525773 0.5000000   0.08552632       TRUE
## 2 0.1290323 0.113445378    0.1082474 0.2727273   0.11184210       TRUE
## 3 0.1612903 0.004201681    0.1288660 0.6363636   0.11842105       TRUE
## 4 0.3225806 0.046218487    0.2268041 0.4545455   0.21710526       TRUE
## 5 0.3456221 0.058823529    0.2010309 0.5454545   0.24342105       TRUE
## 6 0.2211982 0.012605042    0.2319588 0.4545455   0.09210526       TRUE
```

Now, let's look into summary statistics of our data:

```
summary(df)
```

```
##       abc1          notBornUK        medianIncome      medianAge     
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.2857   1st Qu.:0.04622   1st Qu.:0.1804   1st Qu.:0.3636  
##  Median :0.4194   Median :0.09244   Median :0.2577   Median :0.5455  
##  Mean   :0.4315   Mean   :0.16068   Mean   :0.2923   Mean   :0.5103  
##  3rd Qu.:0.5622   3rd Qu.:0.18067   3rd Qu.:0.3814   3rd Qu.:0.6818  
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :1.0000  
##   withHigherEd    voteBrexit     
##  Min.   :0.0000   Mode :logical  
##  1st Qu.:0.1908   FALSE:107      
##  Median :0.2895   TRUE :237      
##  Mean   :0.3162                  
##  3rd Qu.:0.4095                  
##  Max.   :1.0000                  
```

Based on prior observations of a data set, logistic regression will be used which is a statistical analytic approach for predicting a binary outcome, such as TRUE or FALSE. It analyses the relationship between one or more existing independent variables to predict a dependent data variable. It helps in the classification of data into discrete classes by examining the link between a set of labelled data. It takes the given dataset and learns a linear relationship before adding non-linearity in the form of the Sigmoid function.

To compute **logistic regression**, use the R function glm() (generalised linear model). The parameter $family = binomial$ must be specified to tell R that we want to fit logistic regression.

logistic regression performed on data to find relevant input variables:

```
log_reg_model <- glm(voteBrexit ~., family = binomial, data = df)
print(summary(log_reg_model))
```

```
## 
## Call:
## glm(formula = voteBrexit ~ ., family = binomial, data = df)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.9793  -0.2296   0.3073   0.6032   2.0177  
## 
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   -0.1386     0.8477  -0.164 0.870122    
## abc1          17.5780     2.9114   6.038 1.56e-09 ***
## notBornUK      5.6861     1.8033   3.153 0.001615 ** 
## medianIncome  -6.3857     1.9217  -3.323 0.000891 ***
## medianAge      5.9209     1.4066   4.209 2.56e-05 ***
## withHigherEd -26.7443     3.5762  -7.478 7.52e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 426.52  on 343  degrees of freedom
## Residual deviance: 247.39  on 338  degrees of freedom
## AIC: 259.39
##
## Number of Fisher Scoring iterations: 6
```

The points to note from the summary:

- The P-value of all the input variables are less than 0.05 which makes all of them significant in predicting the target variable $voteBrexit$.
- The variable $withHigherEd$ has lowest P-value and highest magnitude of the coefficient(-26.7443) making it the most significant variable in predicting the target variable $voteBrexit$.
- The variable $notBornUK$ has the lowest magnitude of the coefficient(5.69) making it the least significant variable in predicting the target variable $voteBrexit$.

Correlation Matrix:

```
cor_mat <- df[, c(1,2,3,4,5,6)]
round(cor(cor_mat),2)
```

```
##                abc1 notBornUK medianIncome medianAge withHigherEd voteBrexit
## abc1           1.00      0.39         0.78     -0.17         0.89      -0.39
## notBornUK      0.39      1.00         0.56     -0.73         0.55      -0.40
## medianIncome   0.78      0.56         1.00     -0.35         0.74      -0.42
## medianAge     -0.17     -0.73        -0.35      1.00        -0.24       0.29
## withHigherEd   0.89      0.55         0.74     -0.24         1.00      -0.55
## voteBrexit    -0.39     -0.40        -0.42      0.29        -0.55       1.00
```

The result shows the correlation between all the variables in our dataset. Here we can see that the input variable $abc1$ is highly correlated with $withHigherEd$. The value of which is 0.89.

However, this will not suffice to bring our findings to a conclusion. To increase our confidence in our results, we must perform **cross-validation**. There are 344 observations in total. 50% of the data is used to form the training set. The rest of it will be used to form the test set. $seed()$ function is used for reproducibility of the random sample selected.

```
set.seed(1234)
idx = sample(1:344, 172)
train_data = df[idx, ]
test_data = df[-idx, ]
```

There are 5 input variables. Every possible combinations of input variables has been considered, $C^5_1 + C^5_2 + C^5_3 + C^5_4 + C^5_5 = 31$

Therefore, there are **31 possible combinations* of inputs.

```
formulas = c("voteBrexit ~ abc1", "voteBrexit ~ notBornUK", "voteBrexit ~ medianIncome","voteBrexit ~ me
             "voteBrexit ~ abc1 + notBornUK", "voteBrexit ~ abc1 + medianIncome", "voteBrexit ~ abc1 + m
             "voteBrexit ~ notBornUK + medianIncome","voteBrexit ~ notBornUK + medianAge","voteBrexit ~
             "voteBrexit ~ medianIncome + withHigherEd","voteBrexit ~ medianAge + withHigherEd",
             "voteBrexit ~ abc1 + notBornUK + medianIncome","voteBrexit ~ notBornUK + medianIncome + med
             "voteBrexit ~ abc1 + medianIncome + medianAge","voteBrexit ~ abc1 + medianAge + withHigherE
```

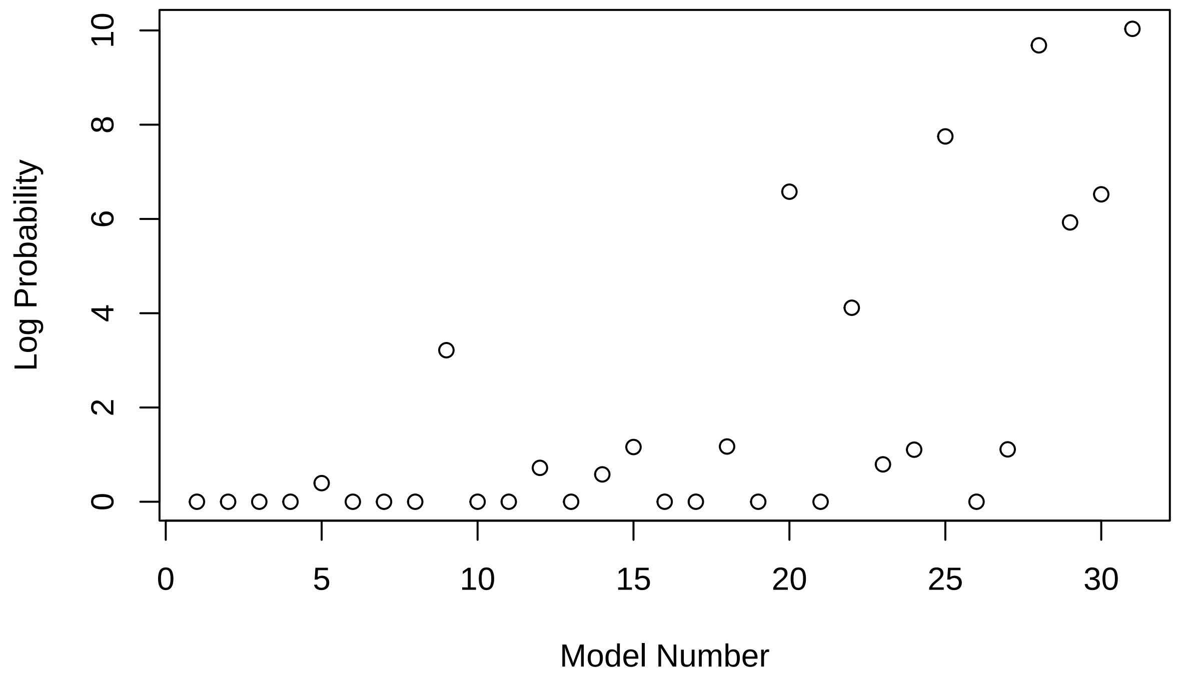
```
              "voteBrexit ~ abc1 + notBornUK + withHigherEd","voteBrexit ~ notBornUK + medianIncome + wi
              "voteBrexit ~ abc1 + medianIncome + withHigherEd",
              "voteBrexit ~ abc1 + notBornUK + medianIncome + medianAge","voteBrexit ~ notBornUK + media
              "voteBrexit ~ abc1 + medianIncome + medianAge + withHigherEd","voteBrexit ~ abc1 + notBorn
              "voteBrexit ~ abc1 + notBornUK + medianIncome + withHigherEd",
              "voteBrexit ~ abc1 + notBornUK + medianIncome + medianAge + withHigherEd")
```

The predicted (log-)probabilities of the actual test output is used as the measure of the prediction quality. The quality of the prediction is stored in a new variable *predictive_log_likelihood* by fitting and predicting with every possible combination of input variables on testing data.

```
predictive_log_likelihood = rep(NA, length(formulas))
for (i in 1:length(formulas)){
  #Fit logistic regression model with the training data
  current_model = glm(formula = formulas[i], data = train_data, family = binomial(link = 'logit'))
  #Evaluate the probability of the test outputs
  #predicted mean for each new data point
  ypredict_mean = predict(current_model, test_data,type='response')
  #Calculating the predictive log probability by summing the
  #log probability of each output value in the test data
  predictive_log_likelihood[i] = sum(dbinom(test_data$voteBrexit, ypredict_mean, size=172))
}
```

Plotting the predictive log likelihood gives the following result:

```
plot(1:length(formulas), predictive_log_likelihood,
xlab="Model Number", ylab="Log Probability")
```



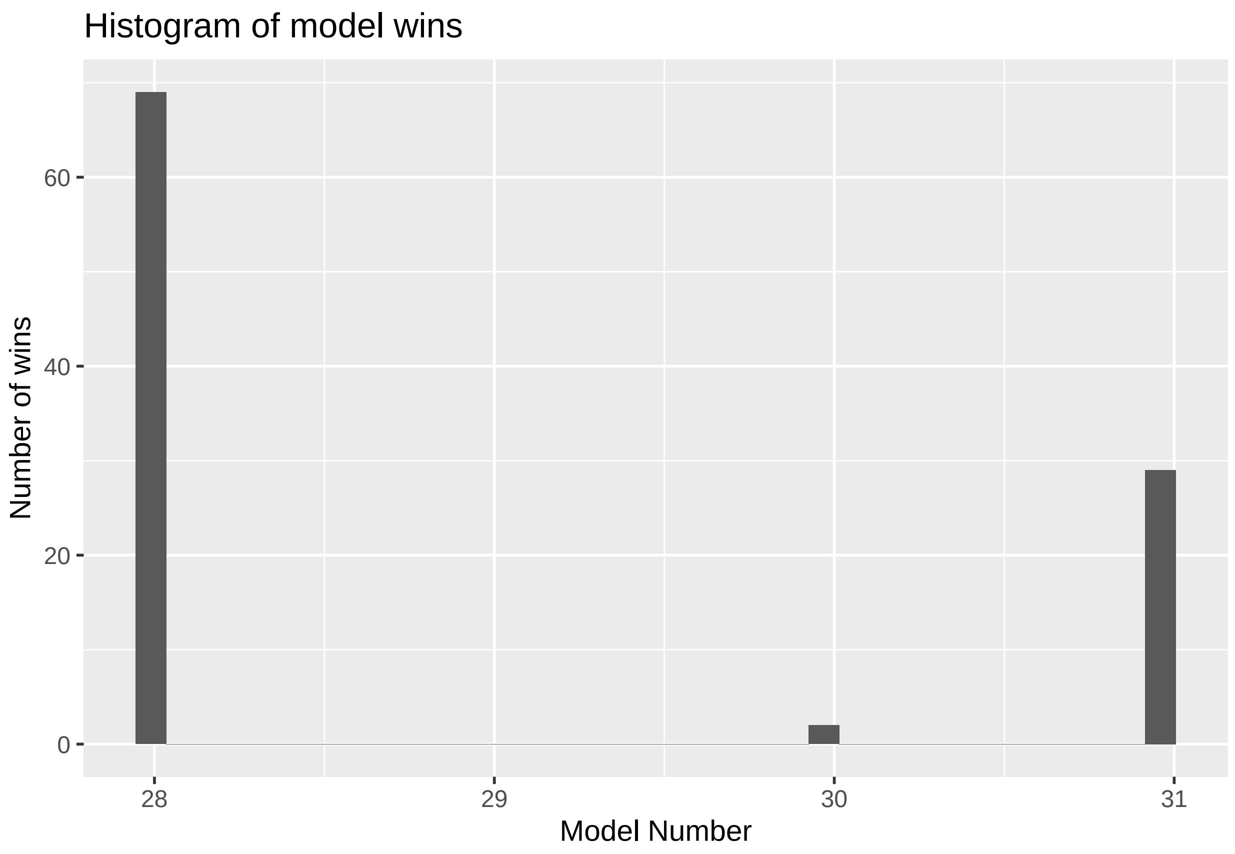
The plot shows that the model 31 has the highest log-probability for the test outputs which considers all the input variables. The model 31 is 'voteBrexit ~ abc1 + notBornUK + medianIncome + medianAge + withHigherEd', which tells that all the input variables are important. But it is not sufficient to conclude this as the best model. The model may be wrong and still predict better on a particular dataset.

To make the cross-validation more robust, the whole procedure of choosing a new random split in the data is repeated 100 times. The winning model with particular combination of input variables for every iteration is stored in a new variable *winner*. It stores the model with best predictive log-likelihood in every iteration.

```r
winner = rep(NA, 100)
#repeating cross validation process 100 times
for (iteration in 1:100){
  #Make a new random training data - test data split
  idx = sample(1:344, 172)
  train_data = df[idx, ]
  test_data = df[-idx, ]
  predictive_log_likelihood = rep(NA, length(formulas))
  for (i in 1:length(formulas)){
    #Fit model with the training data
    current_model = glm(formula = formulas[i], data = train_data, family = binomial(link = 'logit'))
    #Evaluate the probability of the test outputs
    #Predicted mean for each new data point
    ypredict_mean = predict(current_model,test_data,type='response')
    #Calculating the predictive log probability by summing the
    #log probability of each output value in the test data
    predictive_log_likelihood[i] = sum(dbinom(test_data$voteBrexit, ypredict_mean, size=172))
  }
  #Winning model for this iteration is stored
  winner[iteration] = which.max(predictive_log_likelihood)
}
```

Plotting the histogram of how often each model wins gives the following result:

```
ggplot(data=as.data.frame(winner), aes(winner)) +
  geom_histogram(bins = 30,binwidth = 0.09)+
  ggtitle("Histogram of model wins")+
  ylab("Number of wins")+
  xlab("Model Number")
```



The following observations can made from the histogram:

- The model 28 turns out to be the winner maximum number of times followed by model 31. There are some occasions when model 30 wins as well.
- It is interesting to observe that the model 28 with 4 input variables ($abc1$,$medianIncome$,$medianAge$,$withHigherEd$) outperforms model 31 which consists of all the input variables. Thus, overfitting of the model has been avoided by excluding the input variable which is not so significant in predicting the target variable.
- Previously, we had found out that the model 31 would be a good fit for prediction. But by performing quite a few cross-validation data splits, we found a better model. It also gives more confidence in choosing it.

Now that we have found the optimum combination of the input variables, we can now train our final model with it:

```
final_model = glm("voteBrexit ~ abc1 + medianIncome + medianAge + withHigherEd", data = df, family = bi
final_model_pred = predict(final_model,test_data,type='response')
summary(final_model)
```

```
##
## Call:
## glm(formula = "voteBrexit ~ abc1 + medianIncome + medianAge + withHigherEd",
##     family = binomial(link = "logit"), data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9262  -0.1837   0.3421   0.6059   2.0583
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.6787     0.6024   2.787  0.00533 **
## abc1          14.3968     2.5737   5.594 2.22e-08 ***
## medianIncome  -4.6058     1.7570  -2.621  0.00876 **
## medianAge      2.7218     0.8382   3.247  0.00116 **
## withHigherEd -22.1413     2.9802  -7.430 1.09e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 426.52  on 343  degrees of freedom
## Residual deviance: 256.95  on 339  degrees of freedom
## AIC: 266.95
##
## Number of Fisher Scoring iterations: 6
```

The key points to note from the summary are:

- The P-value of all the input variables are less than 0.05 which makes all the input variables significant to predict target variable.
- The coefficient of the predictor variable $withHigherEd$ is -22.1413, magnitude of which is highest(strong negative) among all other features with lowest P-value of 1.09×10^{-13} and the standard error of 2.98. Hence, it is most significant factor in favor of votes against brexit, that is to remain in European Union.
- The second most important feature is $abc1$ with coefficient of 14.40 and standard error of 2.57. As the value is positive, it is second most significant feature for votes to leave from European union(In favor of Brexit).
- The coefficient of the predictor variable $medianIncome$ is -4.61 with the standard error of 1.76. This feature also impacts the vote against brexit.
- The coefficient of the predictor variable $medianAge$ is 2.72, magnitude of which is lowest among all other features with the standard error of 0.84 and P-value of 0.00116. Hence, the variable is least significant for the vote in the favor of brexit.
- The intercept is 1.68.

To confirm the significance of the predictor variables, we calculate confidence interval of their coefficient:

$$95\% C.I. : Estimate \pm z_{2.5} \times StandardError$$

Calculating critical value to find confidence interval:

```
#Get critical value of z_2.5% (should be 1.96)
zc = qnorm(0.975)
```

Confidence interval of $abc1$:

```
#C.I of abc1
#Calculating confidence interval and printing it
CI_abc1 = summary(final_model)$coefficients[2,1] + c(-1,1)*zc*summary(final_model)$coefficients[2,2]
print(CI_abc1)
```

```
## [1]  9.35235 19.44126
```

Confidence interval of $medianIncome$:

```
#C.I of medianIncome
CI_medianIncome = summary(final_model)$coefficients[3,1] + c(-1,1)*zc*summary(final_model)$coefficients
print(CI_medianIncome)
```

```
## [1] -8.049507 -1.162025
```

Confidence interval of $medianAge$:

```
#C.I of medianAge
CI_medianAge = summary(final_model)$coefficients[4,1] + c(-1,1)*zc*summary(final_model)$coefficients[4,
print(CI_medianAge)
```

```
## [1] 1.079063 4.364553
```

Confidence interval of $medianAge$:

```
#C.I of withHigherEd
CI_HighEd = summary(final_model)$coefficients[5,1] + c(-1,1)*zc*summary(final_model)$coefficients[5,2]
print(CI_HighEd)
```

```
## [1] -27.98229 -16.30030
```

From the confidence intervals of all the variables we can see that the zero does not lie in any of the intervals. Hence, all the variables are significant and impacts the target variable.

The table below shows the inputs in terms of decreasing effect of importance:

```
##
## | Input Variable | Coefficient      |In favor of brexit  |
## |----------------|:----------------:|-------------------:|
## | withHigherEd   | -22.14           | No                 |
## | abc1           | 14               | Yes                |
## | medianIncome   | -4.61            | No                 |
## | medianAge      | 2.72             | Yes                |
```

The model was picked as the best with high accuracy, but it may not be explaining these predictions correctly because the predictor variables ‘withHigherEd’ and ‘abc1’ are highly correlated to each other (0.89), with one strongly opposing Brexit and the other strongly supporting it. When the input variables are highly correlated, this problem arises.

The proportion of residents with a degree, according to the **Guardian website**, is the best predictor of a vote for remain (against Brexit). We came at the same conclusion as the input variable *withHigherEd*, indicates the proportion of residents with a university education.