Logistic Regression on Brexit

Prajwal Amin

2023-05-24

In 2016, The UK had a national referendum to decide wether the country should leave or remain in the EU (‘Brexit’). This was declared through an election where votes came from different wards comprising of certain proportion of people across England, Scotland and Wales.

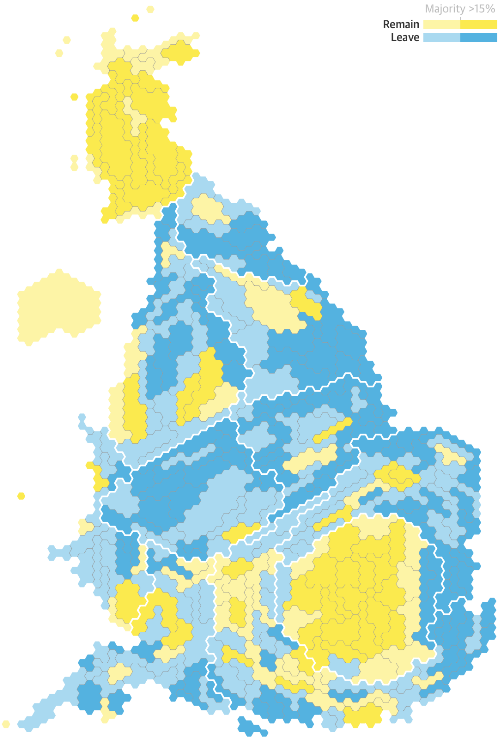


Figure 1: EU Referendum Results Map (The Guardian, 2016)

The Guardian newspaper presented some trends in modelling the outcome of voting. They have done the analysis but stopped at presenting the results graphically and commenting on the apparent patterns. Hence, we will use their analysis to compare our findings and perform some statistical analysis on the data.

The data contains 6 variables including the output variable ’*voteBrexit* which follows binary outcome. The variables are normalized so that the lowest value is 0 and highest is 1.

- abc1: proportion of individuals who are in the ABC1 social classes (middle to upper class)
- medianIncome: the median income of all residents
- medianAge: median age of residents
- withHigherEd: proportion of residents with any university-level education

- notBornUK: the proportion of residents who were born outside the UK

In the tasks below, we will fit a logistic regression model, explore the coefficients which have greater effect on the output, discuss the difficulties faced in interpreting these coefficients of the model and finally adapt an alternative approach to overcome these difficulties.

Task 1

In this task we will fit a logistic regression model to the data to model the outcome of *voteBrexit* using all inputs. Through the summary of the model, will find the direction and magnitude of each of the inputs. Out of which, we will also identify the inputs having strong effects on the outcome. Finally, we will discuss about the findings of the model and compare them with the plots featured on the Guardian.

We will first split the data into training and testing using the ratio of 80:20. Then, will apply *train* data while training the model and *test* data for evaluation.

```
# Splitting the data (80:20 ratio).
set.seed(2)
split <- sample.split(brexit, SplitRatio = 0.8)

train <- subset(brexit, split == "TRUE")
test <- subset(brexit, split == "FALSE")
```

We will be using **glm** command in R to fit the data to the model. The "." (dot) indicates that all variables except *voteBrexit* to be used as inputs. Additionaly, *family = "binomial"* specifies the model to perform logistic regression.

```
# Training the model
model <- glm(voteBrexit ~ ., family = binomial, data = train)
summary(model)
```

```
##
## Call:
## glm(formula = voteBrexit ~ ., family = binomial, data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.57568  -0.07988   0.32642   0.62410   1.93422
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.2414     0.9876  -0.244  0.80689
## abc1          15.7788     3.6001   4.383 1.17e-05 ***
## notBornUK      6.1156     2.2322   2.740  0.00615 **
## medianIncome  -5.8385     2.2713  -2.571  0.01015 *
## medianAge      6.2603     1.6714   3.746  0.00018 ***
## withHigherEd -25.0804     4.4940  -5.581 2.39e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 279.28  on 229  degrees of freedom
## Residual deviance: 165.45  on 224  degrees of freedom
## AIC: 177.45
##
## Number of Fisher Scoring iterations: 6
```

The estimates of the coefficients given by the model can be used to determine the magnitude and direction for each of the inputs. The magnitude indicates the strength of the impact a variable has on the outcome, whereas direction is whether an impact is positive or negative. Hence, given below are the inputs ranked on decreasing order of magnitude.

1. withHigherEd: This input variable has the highest magnitude compared to others. However, this has a negative impact on the outcome. As there is one unit increase in *withHigherEd*, the log-odds of voting for Brexit (TRUE) decreases by 25.08.

2. abc1: This variable has both higher magnitude as well as positive impact on the outcome. A single unit increase in this variable results in an increase of 15.78 in the log-odds of voting for Brexit.

3. medianAge: medianAge is a positive coefficient however the magnitude is not drastically high. An increase in one unit of this variable results in the increase of log-odds of voting for Brexit by 6.26.

4. notBornUK: This variable is also similar to *medianAge* considering its effect on the outcome. This variable has an increasing effect on the outcome by 6.12.

5. medianIncome: This variable has the least magnitude among all others, and the impact too is negative. A unit increase in medianIncome will decrease the log-odds of voting for Brexit by 5.84.

Apart from this, we can also observe that **p-value** for all of the inputs are below < 0.05 which suggests that the inputs are statistically significant. Although, the best predictor in modelling the outcome is **withHigherEd** (proportion of individuals with higher degree). The magnitude of this variable is exceptionally high and the p-value is quite low too.

Evaluation

In order to evaluate the model we will first need to perform predictions using the trained model. Note that, while making predictions *test* data will be used.

```
# Making predictions
pred <- predict.glm(model,test, type = c("response"))

# Converting predictions to 0 or 1
prediction_probs <- ifelse(pred > 0.5, 1, 0)
```

We will evaluate using **confusion matrix** metric from *caret* package which also provides the accuracy for the model.

```
# Create a confusion matrix
cm <- confusionMatrix(table(test$voteBrexit, prediction_probs))
cm$table
```

```
##    prediction_probs
##      0  1
##   0 27 12
##   1  3 72
```

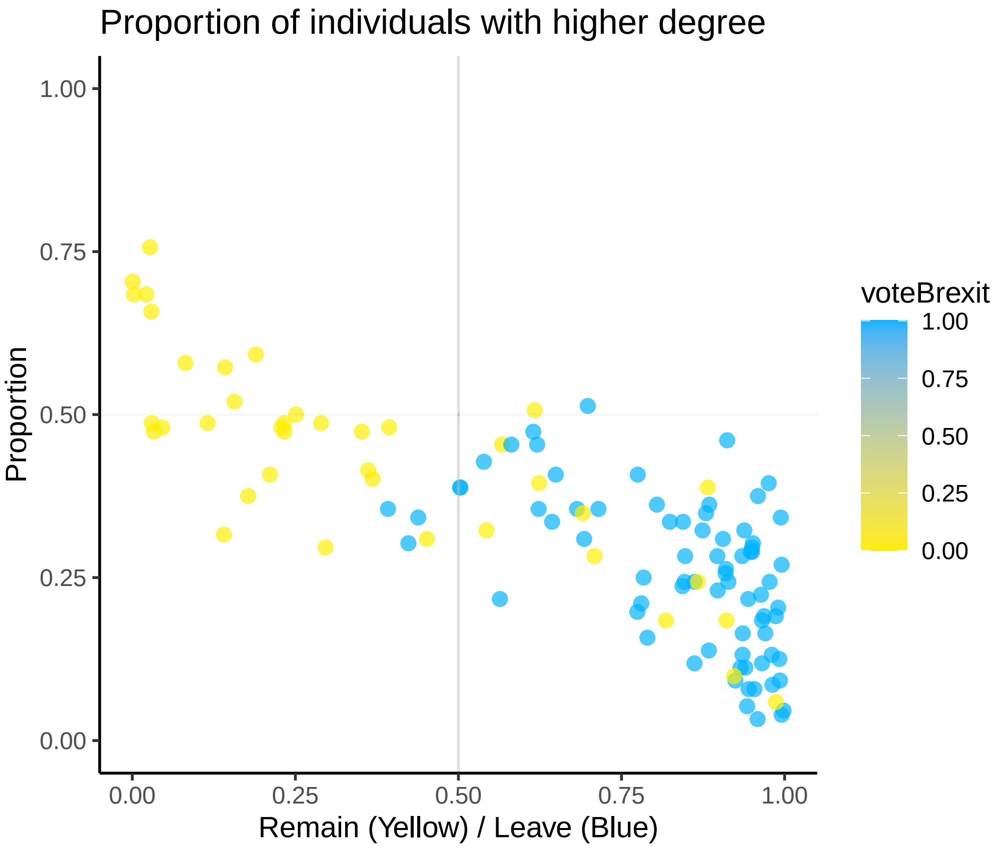
```
cm$overall["Accuracy"]
```

```
##  Accuracy
## 0.8684211
```

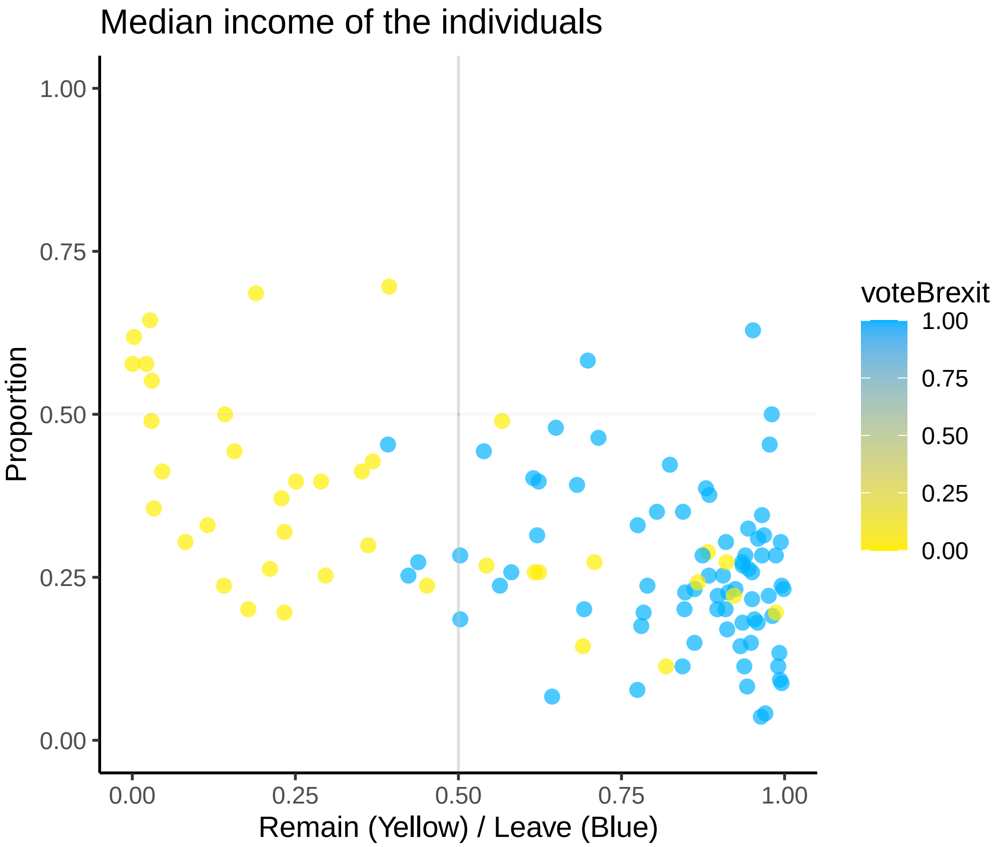
The TP and TN rate is high and the accuracy is $\approx 87\%$ which is satisfactory for this model.

Plotting the data

The predictions contain the probability of each electoral ward voting for Brexit. Hence, using the predictions, we can visualize and model the output for each of the inputs. First we will use *withHigherEd* variable against the model predictions to identify any patterns that is evident. We will also imagine a decision boundary so that if probability is > 0.5, the individual is likely to vote and if it is < 0.5 he will not.



The graph clearly shows that a ward with less than 50% of people with higher degree, is more likely to vote for leave. The correlation of the variables is strongly negative (≈ -0.79).



Overall, *withHigherEd* seems like a good predictor for *voteBrexit*. In comparison to the findings featured on the Guardian website, it appears that their plots agree with the plots produced in this report and the difference is insignificant. The possible reasons for the patterns to vary might be due to the data that has been used to generate the plots or adapting different approaches in training the model. Further, the data points plotted on the website are classified much accurately compared to this report.

Task 2

Since we have used a logistic regression model, there are certain factors that affect the interpretability of the coefficients fitted to the model, these are:

1. Linearity Similar to linear models, logistic regression assumes that there exists some linearity in the relationship between the variables. However, if non-linear then the model might not capture true effect of the variables, which might be misleading while interpreting the coefficients.

2. Sample Size It is always a good idea to perform predictions using large sample size, the results obtained using small sized data sets may sometimes produce inaccurate results. The standard errors of the coefficients might be high, thus leading the coefficients to be unstable.

3. Normalization Normalization rescales the input variables so that they have similar scales, which can help in interpreting the coefficients. But, sometimes interpretation might be difficult especially when variables with non-linear relationship have been normalized. Moreover, different variables will be measured in different scales which will get affected due to this process.

4. Collinearity This occurs when the correlation between the input and output variables are exceptionally high, which might make it hard to determine the individual effect of each variable and finding out which variable is relevant for the output.

Although, it is discussed in *Task 1* that the coefficients are interpreted in such a way that as one one unit increase in that variable increases the log-odds of the *output* by *coefficient value* when all other variables are held constant. This should not be considered as the only decision rule in determining the relationship between the input and output variables, as there might be other demographic factors that influence the decision to vote for or against Brexit.

Further, taking into account the factors mentioned above, it is not always reliable to determine the input variables based on their decreasing effect on the output. If done so there might be some uncertainty in the ordering. However, there is one approach through which we can address the concept of **collinearity**. For this, we will use **VIF** (Variance Inflation Factor) from the *car* package.

```
library(car)
```

```
## Loading required package: carData
```

```
vif(model)
```

```
##         abc1    notBornUK medianIncome    medianAge withHigherEd
##    10.872783     3.346563     2.580604     3.090579     9.841184
```

Here, it is clearly noticeable that the VIF values for *abc1* and *withHigherEd* is above 5 which is considered to be as normal. This indicates that these two variables have high collinearity with the output. In Contrary, the other features have low VIF values which shows that they have greater independence on predicting the outcome.

Therefore, considering the factor of collinearity as well as the nature of the data, we might determine the ordering of the relevant features in decreasing order of their effect:

1. medianIncome Income level often signifies the level of importance in voting for an individual.

2. medianAge Age determines different political behavior among humans, hence the importance of this variable might be relevant.

3. notBornUK An individual whether born or not in UK might have only considerable effect.

4. withHigherEd An individual with a higher degree will be well equipped with political knowledge, however VIF is high.

5. abc1 The individuals belonging to different socioeconomic groups possess different political attitudes or behaviour. Considered least as it is subject to high VIF.

As always these observations are subjective and the actual ordering varies in different scenarios. Moreover, factors such as sample size, linearity and normalization could also influence the ranking of these variables. In the upcoming task, we will deal with this problem of misinterpretation of the variables by following a alternative approach resolving the issue of **collinearity**.

Task 3

An alternative approach to carry out the analysis for task 1 would be to create another logistic regression model, however by excluding the variables which are subject to collinearity. Out of the variables we observed: *abc1* and *withHigherEd*, we will only remove *abc1* while training the model because even though *withHigherEd* is higlhy correlated it also a good predictor of the output compared to the rest of the variables.

We will create a subset of the data by removing the *abc1* column using *dplyr* package

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Removing 'abc1' column from the dataset
brexit_reduced <- select(brexit, -abc1)
```

We will split the data once again using the same split (80:20).

```
# Splitting the data into training & testing (80:20)
set.seed(2)
train_reduced <- subset(brexit_reduced, split == "TRUE")
test_reduced <- subset(brexit_reduced, split == "FALSE")
```

Now we can fit the model using training data

```
# Training the model
model_new <- glm(voteBrexit ~ ., data = train_reduced, family = "binomial")
```

```
summary(model_new)
```

```
##
## Call:
## glm(formula = voteBrexit ~ ., family = "binomial", data = train_reduced)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4398  -0.3021   0.3980   0.6829   1.4785
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.7719     0.8431   2.102  0.03558 *
## notBornUK      1.6623     1.7657   0.941  0.34650
## medianIncome   0.4884     1.7786   0.275  0.78365
## medianAge      3.6357     1.3674   2.659  0.00784 **
## withHigherEd  -9.2550     1.8487  -5.006 5.55e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 279.28  on 229  degrees of freedom
## Residual deviance: 190.16  on 225  degrees of freedom
## AIC: 200.16
##
## Number of Fisher Scoring iterations: 5
```

```
vif(model_new)
```

```
##    notBornUK medianIncome    medianAge withHigherEd
##     2.638910     1.922341     2.372904     2.052459
```

The VIF values for all the variables are low, and surprisingly, notice that *withHigherEd* has also reduced.

Evaluation

We can now use this model to make predictions, then evaluate the model performance using confusion matrix.

```
# Making predictions
predictions_new <- predict(model_new, newdata = test_reduced, type = "response")

# Converting predictions to 0 or 1
prediction_conv <- ifelse(predictions_new > 0.5, 1, 0)

# Evaluate using confusion matrix
cm2 <- confusionMatrix(table(test_reduced$voteBrexit, prediction_conv))
cm2$table
```

```
##    prediction_conv
##      0  1
##   0 21 18
##   1  4 71
```

```
cm2$overall["Accuracy"]
```

```
##  Accuracy
## 0.8070175
```

This signifies that the number of instances which have been correctly classified is less compared to the model from task 1. The accuracy of both the models proves this as the former model produced ≈ 87% and the latter ≈ 81%. The second model was trained with fewer variables to avoid collinearity, which might be the reason in achieving less accuracy, thus making the model less dependable in making predictions.

As we have followed this alternative approach, it comes with certain advantages and disadvantages which are listed down below:

Advantages

- Interpretability: Addressing collinearity by removing highly correlated variables makes it easier to interpret the coefficients of the model.
- Improves Performance: The model becomes much stable in making predictions and also reduces the being prone to overfitting.
- Feature selection: Simplifies the process of selecting features that are relevant for the model to perform better.

Disadvantages

- Loss of Information: In some cases there might be only few variables remaining after removing collinear variables which provides lack of information to the model.
- Loss of strong predictors: There might certain variables which act as strong predictors to the outcome and these might be removed due to collinearity.
- Poor results on small sample sizes: The values obtained by VIF could be unstable with small sample sizes, as sufficient data is required to provide accurate estimates.