# **Essay Front Page**

Please complete this sheet electronically and attach it as the front page

for each essay submitted.

## Candidate Number: XTPC0

\*\*\*\*Please remember that you must use your **candidate number** as the Submission Title on Turnitin\*\*\*\*

## Essay Title: Term 2 Assessment

## Essay Number: 3

## Module Code: POLS0010

## Module Title: Data Analysis

## Submission Date: 09/05/2023

## Word Count: 2992

Please tick this box if you **do not** agree to your essay being used anonymously in future teaching

\*\*\*\*Your name must not appear anywhere in this submission\*\*\*\*

**Extenuating Circumstances** If you are not in a position to complete your assessments by the deadline because you are unwell, have unexpected caring responsibilities or any other circumstances have disrupted your academic work, please fill in [an extenuating circumstances form](https://www.ucl.ac.uk/academic-manual/chapters/chapter-4-assessment-framework-taught-programmes/section-6-extenuating-circumstances) (also available on the Information Hub on Moodle) and request an extension on your essays via our [online form](https://forms.office.com/Pages/ResponsePage.aspx?id=_oivH5ipW0yTySEKEdmlwpfeoObLHPxCoDpDzst8RUtUQjdKWlJBWlBMT0QwQlVJQlVEWjQ2UUJGWS4u).

**QUESTION A:**

Introduction:

This analysis uses data from the 2015 Scottish Social Attitudes Survey of 1100 voters to identify the most supportive characteristics of Scottish independence. The dataset has a binary dependent variable to measure support for independence (1 if the respondent voted yes, 0 otherwise). In addition, the dataset includes several independent variables such as age in years, and binary measures like housing status (1 if homeowner, 0 otherwise), sex (1 if male, 0 if female), income quartile (1 if in highest income quartile, 0 if otherwise), and religious affiliation (1 if belonging to organized religion, 0 if otherwise). Other independent variables in the dataset can be measured on a scale, such as level of satisfaction with the National Health Service (NHS) from 1 (very dissatisfied) to 5 (very satisfied), self-reported left-right ideology scale from 1 (left-wing) to 5 (right wing), and the level of trust in the British and Scottish governments to work in Scotland’s long-term interest on a scale from 1 (almost never) to 4 (just about always). We will run three logit models and use the best-performing one to inform recommendations for targeting a new campaign to mobilize supporters of Scottish independence to vote.

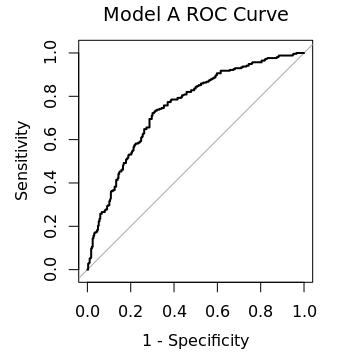
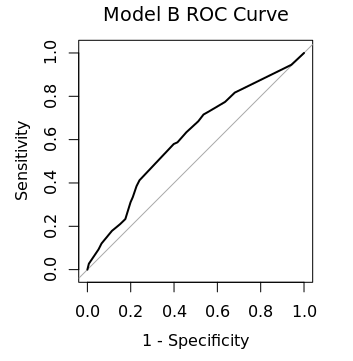
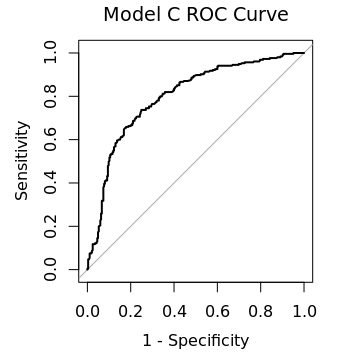
Question i)

The selection process in deciding which variables to use in these three models was random to ensure representativeness and avoids including too many variables to avoid overfitting. Our first model (Model A) included the predicting variables male, age, high income, trust in Scottish government, and left-right ideology. Model B included the variables Religious, satisfaction with NHS, and homeowner status. Our final model (Model C) includes the variables trust in UK government, age, satisfaction with NHS, and male sex to predict support for independence.

We assessed the predictive performance of our models by splitting the dataset into training and test data and estimating each model with only its training data. We obtained the total error rate, sensitivity, and specificity in a confusion matrix, and looked for the area under the ROC curve. As seen in Table 1, model C has the lowest error rate at 26.7%, which means it is the most accurate at predicting outcomes based on the given variables. Model C also had the closest value for the area under the curve to 1 (0.80) and has a ROC curve that is closest to the top-left corner of the plot (see Figure 1 below). This indicates that the model has the best ability to distinguish between positive and negative classes. It’s important to note that although Model C is the best out of the three models we pre-selected, the values are not exactly ideal. A specificity of 83.9% means that the model does well at correctly classifying people who did not support Scottish independence, however, a sensitivity of 61.6% means that the model did quite poorly at correctly classifying people who supported Scottish independence. Seeing as our aim is to tailor a campaign towards those who did vote yes, failure to correctly classify yes votes for support is worse than failure to classify no votes.

Table 1: Confusion Matrix Results for Support for Scottish Independence

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model and Variables** | **Error Rate (%)** | **Sensitivity**  **(%)** | **Specificity**  **(%)** | | **Area Under the Curve** |
| **Model A**  *Male, age, high income, trust in Scottish government, left-right ideology* | 31.4 | 62.5 | 25.9 | 0.75 | |
| **Model B**  *Religious, satisfaction with NHS, homeowner status* | 40.5 | 41.2 | 75.9 | 0.61 | |
| **Model C**  *Trusts UK government, age, satisfaction with NHS, male* | 26.7 | 61.6 | 83.9 | 0.80 | |

Figure 1: Plots of ROC Curves for Each Model Predicting Support for Scottish Independence

Question ii):

We then calculated how much the variables in Model C matter in explaining support for Scottish independence by calculating the average marginal effects (AME) for each independent variable. On average for the observations in our dataset, the AME shows us the effect of a one-unit change in each independent variable on the probability of our outcome variable (a Yes vote for Scottish independence) occurring. They give the impact of each variable, averaged across all possible profiles of the other independent variables. The AME is a preferred way of summarizing the impact of the independent variables, as opposed to marginal effects at the mean, which is based on the mean value of all the independent variables. This is because it is unlikely there is an observation like the average profile in reality. We also measured the uncertainty of our AME value using the Delta Method, which is reflected in the standard errors and respective significant levels in Table 2 below.

Table 2: Average Marginal Effect for Variables in Model C and Measures of Uncertainty

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Marginal Effect** | **Standard Error** | **P Value** |
| **Trust in UK Gov** | -0.283 | 0.029 | < 2.2e-16 \*\*\* |
| **Age** | -0.003 | -0.001 | 0.0004 \*\*\* |
| **Satisfaction with NHS** | 0.009 | 0.012 | 0.454 |
| **Male** | 0.069 | 0.027 | 0.009\*\* |
| Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1 | | | |

The marginal effect for Trust in UK Gov suggests that on average across our dataset, an increase of Trust in UK Gov by one unit is associated with a fall in the probability of voting yes for Scottish independence of 0.28, holding all other variables constant. The marginal effect is statistically significant with a p-value of < 2.2e-16 at the 0.001 level. The marginal effect for age demonstrates that an increase in age in one year is associated with a decrease in Scottish independence of 0.003, holding all other variables constant. It is indeed statistically significant, with a p-value of 0.0004 at the 0.001 level. The marginal effect for Satisfaction with the NHS demonstrates that with a one-unit increase in satisfaction, there is a 0.009 increase in the probability of a yes vote for Scottish Independence. However, it is statistically insignificant, with a p-value of 0.454 at the 1 level. The marginal effect for being male is associated with a 0.069 increase in a vote Yes for Scottish independence and is statistically significant with a p-value of 0.009 at the 0.01 level.

Question iii):

Based on the AME output in Table 2, an advertising campaign aimed at mobilizing supporters for Scottish independence to vote can be targeted at a specific demographic. The estimates suggest that a one-year increase in age is associated with a decrease in the probability of voting for independence. Therefore, an advertising campaign that targets younger voters will be more effective. The AME for male sex suggests that being male (compared to being female) is associated with an increase in the probability of voting for independence, so men should also be targeted. AME for trust in UK government suggests that a lack of trust in the UK government is associated with a higher probability of voting for independence. Therefore, an advertising campaign that seeks out populations who have expressed distrust in the UK government will be effective in increasing turnout among voters. While the AME for satisfaction with NHS is not statistically significant, it suggests that voters who are satisfied with the NHS may be less likely to support independence. Therefore, an advertising campaign that highlights the potential benefits of independence for healthcare may be effective in increasing turnout among this demographic.

Conclusion:

By running statistical tests to ensure the best prediction for support for Scottish independence and demonstrating statistical significance through hypothesis testing, we can be confident in our model’s performance against the other pre-selected models and the impact of each variable on support for Scottish independence. Although the estimate for Model C’s sensitivity is not ideal for our research question, we can be confident that it performed the best out of the three pre-selected models.

**QUESTION B:**

Introduction:

In this section, we are interested in the public’s support for income redistribution. Our data contains 6410 survey responses from British citizens aged 18 and above and is collected across all 366 local authorities in the UK. The dependent variable is a measure of each individual's level of support for redistribution on a scale from 1-5, where higher numbers indicate greater support. The independent variables include binary variables that take on a value of 1 if the individual belongs to each category, and 0 if they do not. These include belonging to female sex, and age categories from 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, and 80 and above. The remaining independent variables measure characteristics of the 366 local authorities and are standardized. These include the local authority population aged 65+ (%), the local authority population aged 18-24 (%), the average household income, the average amount of government benefits received per household (mean-centered), and the population density of each local authority.

Question i)

We will use a multilevel model with selected independent variables from the list above and random intercepts for local authorities to explain support for income redistribution across all 366 areas. The dependent variable in this study is the level of support for redistribution, and the independent variables include both individual and group-level predictors to provide a more nuanced understanding of the relationship between the predictors and the outcome variable. The individual level predictors we have selected are the respondent’s sex and age category. The group-level predictor is the percentage of the local authority area population aged 18-24. The use of both individual and group-level predictors in our model helps to avoid the issue of the ecological fallacy. We go further to enhance our model by including varying slopes by the percentage of 18–24-year-olds in each local authority. This is justified because it is plausible that the effect of age on support for redistribution may differ between lower-output areas with a high percentage of younger respondents versus those with a low percentage.

Question ii)

The results of the multilevel model can tell us a lot about how much the individual and group-level variables we have included (age, sex, percentage of population aged 18-24) affect support for redistribution. Produced below are the average effects across local authorities of each of the independent variables, also known as the fixed effects.

Table 3: Fixed Effect Estimates for Multilevel Model Predicting Support for Income Redistribution Across Local Authorities in England

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Variable** | **Estimate** | **Standard Error** | | **T-Value** |
| ***Intercept*** | 3.5 | 0.09 | 36.80 | |
| ***Sex***  Female | 0.02 | 0.02 | 0.82 | |
| ***Age***  20-29  30-39  40-49  50-59  60-69  70-79  80+ | -0.01  -0.19  -0.27  -0.24  -0.28  -0.44  -0.39 | 0.10  0.10  0.10  0.10  0.10  0.10  0.13 | -0.13  -1.95  -2.75  -2.47  -2.95  -4.53  -2.94 | |
| ***Population 18-24 (%)*** | 0.09 | 0.01 | 6.26 | |

The estimates for individual-level predictors, such as female and age categories, show that being female and being in older age groups are associated with lower levels of support for redistribution. For instance, being in the age group 70-79 is associated with a decrease of 0.44 in the level of support, compared to the reference group. The estimate for the area-level predictor, the percent of the local authority area population aged 18-24, is positive and significant, suggesting that higher levels of younger population in the area are associated with higher levels of support for redistribution.

Because we have included varying slopes for the percentage of 18 to 24-year-olds across authority areas, we can combine both fixed and random effects to determine how the effects of a younger population in a local authority on support differ. Bristol was the local output area with the highest coefficient, at 0.117, making it the local authority area where the percentage of 18-24-year-olds had the highest impact on support. Portsmouth had the lowest coefficient of 0.051, making it the local authority where the percentage of the population aged 18-24 had the lowest impact on support.

Question iii)

The next step was to produce post-stratified estimates of average support for redistribution across the 366 local authorities, using two sources of data- namely the predicted outcome for demographic groups in each local output area (from our multilevel model), and census data on the demographic composition of every local authority. We then weigh the predicted outcome for each demographic group by its percentage of the area’s population and post-stratify by adding up the weighted predictions for each area. The plot in Figure 2 below depicting post-stratification estimates across 366 local authorities is used to provide an overview of the distribution of the estimated proportion of individuals in each local authority who support redistribution. The plot allows for easy comparison of the estimated proportions of support for the policy or outcome across the local authorities, highlighting any geographic trends or patterns that may be present.

A screen shot of a computer

Description automatically generatedFigure 2: Post Stratified Estimates of Predicted Support for Redistribution Across 366 Local Authority Areas in the UK

Question iv)

To assess our model’s performance across all 366 local authorities, we can calculate the mean absolute error (MAE). The MAE value is a measure of the average difference between the predicted values and actual values in the model, which we have obtained in a dataset that contains existing estimates of average support for redistribution in every local authority. Our multilevel model has produced an MAE value of 0.06, suggesting that on average, our predicted values are within 0.06 units of the actual values. This indicates that our model performs well in predicting support for redistribution across different local authority areas. In Figure 3 below, the actual estimates for predicted support are plotted against the post-stratified results of our multilevel model and demonstrate that our model may have slightly overestimated the results.

Figure 3: Actual Results of Predicted Support for Redistribution Plotted Against our Multilevel Post-stratified Results for Every Local Authority in UK

A screen shot of a computer screen

Description automatically generated with medium confidence

Conclusion:

In conclusion, post-stratified estimates of our predictive model did well in estimating the average support for redistribution across local authority areas when compared to the existing set of estimates, despite slight overestimation. We were able to achieve this by including random intercepts, varying slopes, and rich individual and group-level data in our multilevel model.

**QUESTION C:**

Introduction:

We have been commissioned by the British National Health Service to analyze reviews of GP surgeries and build a predictive tool that classifies reviews into negative and positive. The dataset consists of 1,986 online reviews that have been labeled as positive or negative on a binary scale, where 1= positive and 0= negative.

Question i)

Using appropriate tools, we analyzed the reviews to identify words associated with positive and negative sentiment, and how they varied between the two. After creating a document term matrix that followed proper data cleaning (see Table 1), we ended with a corpus that had 2,550 unique words. We also applied TF-IDF weighting to identify informative words for sentiment classification by weighing the frequency of occurrence of each word by its inverse document frequency. This also handles the issue of stop words that were not removed in the cleaning process. We then plotted the 25 most-used words in Figure 1 below to see how usage differed between positive and negative reviews.

|  |  |  |  |
| --- | --- | --- | --- |
| Corpus Criteria | Original text data | Lowercase, no numbers, no punctuation, remove stop words and rare words | After stemming |
| **N** | 166,934 | 3,508 | 2,550 |
|  |  |  |  |

Table 1: Number of Words in Corpus after each Text Data Cleaning Procedure

Figure 1: Plot of the Most Used Words in Final Corpus, Grouped by Type of Review

A picture containing text, screenshot, diagram, plot

Description automatically generatedA picture containing text, screenshot, diagram, plot

Description automatically generated

Question ii)

When creating our dictionary, we based our decision on which words to include from the analysis drawn from Figure 1 above. We chose not to include certain words that appeared many times in both positive and negative reviews that are not necessarily representative of a positive or negative sentiment, like “appointment”, “surgery”, “gp”, “practice”, “time”, and “doctor”. In our dictionary, we included 15 of the remaining words “thank”, “care”, “staff”, “help”, “alway”, “nurs”, “servic”, “friend”, “well”, “effici”, “feel”, “profession”, “great”, and “good” to represent positivity. The words “call”, “get”, “told” “phone”, “tri”, “day”, “wait”, “book”, “week”, “back”, “need”, “one”, “receptionist”, “go”, and “just” were used for negative reviews.

Question iii)

The next step was to use the lasso logit method to classify the reviews into negative and positive, which selects the most informative words or phrases to include in the model based on their contribution to the classification accuracy. This approach has the advantage of being able to detect subtle patterns in the text data that might be missed by a dictionary-based approach. This involved splitting our data set into testing and training data and evaluating the model performance on the test set to get an accurate estimate of how well the model can perform in the real world.

Question iiii)

The last step is to compare the performance of our dictionary and the lasso logit model to then decide which would be best to classify reviews for the NHS. We completed this by running a confusion matrix and printed the performance results in table 2 below. Based on the results of the confusion matrix, it is evident that the lasso logit model performs much better than the dictionary in classifying reviews as positive or negative. The error rate for the lasso logit model is only 12.08%, which is significantly lower than the error rate of 80.9% for the dictionary. Additionally, the lasso logit model has a much higher specificity of 85.5%, meaning that it correctly identifies negative reviews. The sensitivity of the lasso logit model is also high at 89.1%, indicating that it correctly identifies positive reviews. On the other hand, the dictionary method has a very low specificity of 6.3% and a sensitivity of 31.4%, suggesting that it performs poorly in identifying both positive and negative reviews. Therefore, it can be concluded that the lasso logit model is a much more effective method for classifying reviews in text data analysis than the dictionary and should be used by the NHS in classifying future reviews.

Table 2: Confusion Matrix Results Between Dictionary and Lasso-Logit Classifying Positive and Negative Reviews

|  |  |  |
| --- | --- | --- |
|  | Dictionary | Lasso-Logit |
| Error Rate (%) | 80.9 | 12.08 |
| Specificity (%)  Sensitivity (%) | 6.3  31.4 | 85.5  89.1 |
|  |  |  |

Conclusion:

In conclusion, although we took steps to thoroughly clean and analyze our corpus, our selected dictionary performed much worse than the lasso-logit method. This is because lasso-logit is a machine-learning algorithm that can capture complex interactions between words, which may have been missed by our dictionary-based approach.

Appendix:

#PART 1

#load required packages

load("scottishindependence.Rda")

install.packages("lme4")

library(lme4)

install.packages("mfx")

library(mfx)

install.packages("pROC")

library("pROC")

#Question A question i)

#split dataset into test and training

training.rows <- sample(1100/2)

training.data <- ssa[training.rows,]

test.data <- ssa[-training.rows,]

#model 1 logit

mod1.glm <- glm(voteYes~male + age + highinc + trustSEgov + leftright,

family=binomial(link="logit"), data = training.data)

mod1.preds.test <- ifelse(predict(mod1.glm, test.data, type='response')>0.5,1,0)

table(mod1.preds.test,test.data$voteYes)

mod1.error.rate <- (96+74)/(212+96+74+160)

mod1.error.rate

mod1.sensitivity <- 160/ (160 + 96)

mod1.sensitivity

mod1.specificity <- 74/ (212+74)

mod1.specificity

mod1.probs.test <- predict(mod1.glm, test.data, type='response')

mod1.roc.plot <- roc(test.data$voteYes, mod1.probs.test)

mod.1.auc <- mod1.roc.plot$auc

mod.1.auc

plot(mod1.roc.plot, legacy.axes=T)

title('Model A ROC Curve', font.main=1, line=3)

#model 2 logit

mod2.glm <- glm(voteYes~religious + satisfiedNHS + housing,

family=binomial(link="logit"), data = training.data)

mod2.preds.test <- ifelse(predict(mod2.glm, test.data, type='response')>0.5,1,0)

table(mod2.preds.test,test.data$voteYes)

mod2.error.rate <- (151+71)/(221+151+70+106)

mod2.error.rate

mod2.sensitivity <- 106/ (106 + 151)

mod2.sensitivity

mod2.specificity <- 221/ (221+70)

mod2.specificity

mod2.probs.test <- predict(mod2.glm, test.data, type='response')

mod2.roc.plot <- roc(test.data$voteYes, mod2.probs.test)

mod.2.auc <- mod2.roc.plot$auc

mod.2.auc

plot(mod2.roc.plot, legacy.axes=T)

title('Model B ROC Curve', font.main=1, line=3)

#model 3 logit

mod3.glm <- glm(voteYes~trustUKgov + age + satisfiedNHS+ male,

family=binomial(link="logit"), data = training.data)

mod3.preds.test <- ifelse(predict(mod3.glm, test.data, type='response')>0.5,1,0)

table(mod3.preds.test,test.data$voteYes)

mod3.error.rate <- (98+46)/(239+98+46+157)

mod3.error.rate

mod3.sensitivity <- 157/ (157 + 98)

mod3.sensitivity

mod3.specificity <- 239/ (239+46)

mod3.specificity

mod3.probs.test <- predict(mod3.glm, test.data, type='response')

mod3.roc.plot <- roc(test.data$voteYes, mod3.probs.test)

mod.3.auc <- mod3.roc.plot$auc

mod.3.auc

plot(mod3.roc.plot, legacy.axes=T)

title('Model C ROC Curve', font.main=1, line=3)

#Question A question ii)

mod3.glm <- glm(voteYes~trustUKgov + age + satisfiedNHS+ male, family=binomial(link="logit"), data = training.data)

#what are the mean values for the x variables?

mean(ssa$age)

mean(ssa$trustUKgov)

#test the coefficients for the marginal effects at the mean to make comaprisons the the AME

logitmfx(mod3.glm, data=ssa)

#now test the AME

logitmfx(mod3.glm, atmean=F, data=ssa)

mod3.glmpp <- glm(voteYes~trustUKgov + age + male, family = binomial(link="logit"), data=training.data)

library(arm)

library(boot)

set.seed(1)

votesims <- sim(mod3.glmpp, n.sims=1000)

coefs<- coef(votesims)

#profile for men

values1 <- c(1, mean(ssa$age), 1, )

#Question B question i)

load("2023qb\_surveydata\_corrected.Rda")

#select an appropriate mltilevel model that has varying slopes, and explore the output

m1 <- lmer(support~ female + age2029 + age3039 + age4049 + age5059 + age6069 + age7079 + age80pl + la\_pc1824\_cen + (1 + la\_pc1824\_cen | localauthority), data=surveydata)

summary(m1)

coef(m1)

View(coef(m1))

#detemrine in which local authority the effect of a young population is strongest and weakest on support for redistribution

min(coef(m1)[["localauthority"]][["la\_pc1824\_cen"]])

max(coef(m1)[["localauthority"]][["la\_pc1824\_cen"]])

#poststratification

install.packages("data.table")

library(data.table)

load("2023qb\_poststratdata\_corrected.Rda")

poststratdata$prediction2 <- predict(m1,newdata=poststratdata,type="response",allow.new.levels=TRUE)

poststratdata$weight.pred2 <- poststratdata$prediction2\*poststratdata$value

results2 <- data.table(poststratdata)[ , .(final.est.pred = sum(weight.pred2)), by = .(localauthority)]

results2

#demonstrate how these perform on a plot

pdf(file="poststratified\_results.pdf")

plot(results2$final.est.pred,

ylim=c(3,4),

xlab="Local Authority Area",

ylab="Predicted Support for Redistribution",

col="blue")

points(results2$final.est,col="red")

points(existing\_estimates$final.est, col="blue")

legend("topright",

c("Multilevel Model", "Actual Values"),

pch=c(1,1),lty=c(1,1),

col=c("red", "blue"))

dev.off()

#now lets load the existing dtaa and compare with MAE

load("2023qb\_existingestimates.Rda")

new\_data <- cbind(results2[,"final.est.pred"], existing\_estimates[,"final.est"])

MAE <- mean(apply(new\_data, 1, function(x) abs(x[1] - x[2])))

MAE

#PART 2

#question i)

load("nhs\_reviews.Rda")

install.packages("quanteda.textstats")

library(quanteda.textstats)

install.packages("quanteda")

library(quanteda)

install.packages("ggplot2")

library(ggplot2)

library(stringr)

install.packages("glmnet")

library(glmnet)

# Count the number of words in each review

word\_counts <- str\_count(nhs\_reviews$review\_text, "\\S+")

# Sum the word counts across all reviews in the column

total\_word\_count <- sum(word\_counts)

total\_word\_count

#create a corpus and document term matrix where everything is lowercase, numbers are removed, as well as punctuation and stop words

ReviewCorpus <- corpus(nhs\_reviews$review\_text, docvars = nhs\_reviews)

dfm\_reviews <- ReviewCorpus %>% tokens(remove\_numbers=T,

remove\_punct=T,

include\_docvars=T) %>%

tokens\_remove(stopwords("en")) %>%

tokens\_wordstem() %>%

dfm(tolower=T)

nfeat(dfm\_reviews)

dim(dfm\_reviews)

#remove rare words that only appear in a single speech

dfm\_reviews <- dfm\_reviews %>% dfm\_trim(min\_docfreq=2)

nfeat(dfm\_reviews)

dim(dfm\_reviews)

#what are the ten most common words?

textstat\_frequency(dfm\_reviews)[1:10]

#use the frequency of words as a proportion of total words in each document for comparison

dfm\_reviews\_weighted<- dfm\_weight(dfm\_reviews, scheme="prop")

#now what are the ten most common words?

textstat\_frequency(dfm\_reviews\_weighted)[1:10]

#plot the different in top 25 words, by rating for NHS

dfm\_reviews\_tfidf <- dfm\_reviews %>% dfm\_tfidf()

byreview <- textstat\_frequency(dfm\_reviews\_tfidf,25,groups=review\_positive,force=T)

ggplot(byreview[byreview$group=="0",],

aes(x=frequency,y=reorder(feature,frequency))) +

geom\_point() +

ylab("") +

xlab("Frequency (TF-IDF Weighted)") +

ggtitle("Negative")

ggplot(byreview[byreview$group=="1",],

aes(x=frequency,y=reorder(feature,frequency))) +

geom\_point() +

ylab("") +

xlab("Frequency (TF-IDF Weighted)") +

ggtitle("Positive")

nfeat(dfm\_reviews)

dim(dfm\_reviews)

nfeat(dfm\_reviews\_tfidf)

#create my own dictionary based on these words

neg.words <- c("thank", "care", "staff", "help", "alway", "nurs", "servic", "friend", "well", "effici", "feel", "profession", "great", "good")

pos.words <- c("call", "get", "told", "phone", "tri", "day", "wait", "book", "week", "back", "need", "one", "receptionist", "go", "just")

mydict <- dictionary(list(negative = neg.words,

positive = pos.words))

#apply our dictionary to the the matrix using the dfm lookup() function

sentiment <- dfm\_lookup(dfm\_reviews\_tfidf,dictionary=mydict)

#convert the dictionary table into a data frame

sentiment <- convert(sentiment,to="data.frame")

#Classify the reviews as negative or positive, assigning 1 (positive) to reviews with more positive words than negative, and 0 otherwise

sentiment$score <- ifelse((sentiment$positive - sentiment$negative)>0,1,0)

#Use our variable together with the original pre-labeled classifications, to calculate the test error rate, sensitivity and specificity of our dictionary-based classifier.

cm<- table(sentiment$score,nhs\_reviews$review\_positive)

TP <- cm[2, 2] # True positives

TN <- cm[1, 1] # True negatives

FP <- cm[1, 2] # False positives

FN <- cm[2, 1] # False negatives

sensitivity <- TP / (TP + FN)

specificity <- TN / (TN + FP)

error\_rate <- (FP + FN) / sum(cm)

# Print the results

print(paste0("Sensitivity: ", sensitivity))

print(paste0("Specificity: ", specificity))

print(paste0("Error rate: ", error\_rate))

install.packages("glmnet")

library(glmnet)

dfm\_reviews\_tfidf <- as.matrix(dfm\_reviews\_tfidf)

dfm\_reviews\_tfidf <- cbind(nhs\_reviews$review\_positive, dfm\_reviews\_tfidf)

dim(dfm\_reviews\_tfidf)

#divide the matrix into half cross-validation and half test set, and use the cross-validation set to train a classifier using logit lasso.

set.seed(1)

cv.rows <- sample(nrow(dfm\_reviews\_tfidf),(nrow(dfm\_reviews\_tfidf)/2))

cv.data <- dfm\_reviews\_tfidf[cv.rows,]

test.data <- dfm\_reviews\_tfidf[-cv.rows,]

lasso.rev <- cv.glmnet(x=cv.data[,2:ncol(dfm\_reviews\_tfidf)],y=cv.data[,1],

family="binomial",type.measure="class")

rev.preds <- predict(lasso.rev,

test.data[,2:ncol(dfm\_reviews\_tfidf)],

type= "class")

#run confusion matrix to calculate error rate, sensitivity, specificity, of our classifier for the test set

cm.logit <- table(rev.preds,test.data[,1])

TP.2 <- cm.logit[2, 2] # True positives

TN.2 <- cm.logit[1, 1] # True negatives

FP.2 <- cm.logit[1, 2] # False positives

FN.2 <- cm.logit[2, 1] # False negatives

sensitivity.2 <- TP.2 / (TP.2 + FN.2)

specificity.2 <- TN.2 / (TN.2 + FP.2)

error\_rate.2 <- (FP.2 + FN.2) / sum(cm.logit)

# Print the results

print(paste0("Sensitivity: ", sensitivity.2))

print(paste0("Specificity: ", specificity.2))

print(paste0("Error rate: ", error\_rate.2))