Problem 5

Matthew Leong

Problem 5:

For this problem, we have data from ReutersC50 stored in a train and test set consisting of 50 authors with 50 documents each making for 2500 documents total in each set. The task at hand was given a document, to predict the author who wrote it. To accomplish this, we first processed and read the train and test list of documents into R using code similar to the tm_examples. Essentially it was an iteration that went through the lists to clean up the document names. During this step, we also created another dataframe for storing the author names for both the training and test set. Afterwards, we set up a corpus to then do some pre-processing using the library(tm) package. We decided to make everything lower case, remove numbers, remove punctuation, remove excess white space, and remove basic english stop words. This resulted in the following train document matrix:

```
## <<DocumentTermMatrix (documents: 2500, terms: 32570)>>
## Non-/sparse entries: 537861/80887139
## Sparsity           : 99%
## Maximal term length: 44
## Weighting          : term frequency (tf)
```

As we can see from the summary, the are 32570 terms across all the documents. However, there is a significant portion of those that only occur in 1 or 2 documents. Thus, we decided to cut those words out by removing any terms with a frequency of 0 in greater than 99% of documents.

```
## <<DocumentTermMatrix (documents: 2500, terms: 3393)>>
## Non-/sparse entries: 422971/8059529
## Sparsity           : 95%
## Maximal term length: 44
## Weighting          : term frequency (tf)
```

As seen in the summary, we managed to significantly cut down on the number of terms while also not losing that much sparsity of the matrix. After this process, the test document matrix

```
## <<DocumentTermMatrix (documents: 2500, terms: 33373)>>
## Non-/sparse entries: 545286/82887214
## Sparsity           : 99%
## Maximal term length: 45
## Weighting          : term frequency (tf)
```

One thing that we can already note is that the test set of documents has more terms than the training set which proves to still be the case even after removing 99% of the terms that have 0 occurences in greater than 99% of the documents.

```
## <<DocumentTermMatrix (documents: 2500, terms: 3448)>>
## Non-/sparse entries: 428509/8191491
## Sparsity           : 95%
## Maximal term length: 43
## Weighting          : term frequency (tf)
```

The test document term matrix also only lost 4% sparsity. Using these document matrices, we also constructed two tfidf_matrices and would base our analysis off of that measurement. However before we could begin our analysis, we had to tackle on how to deal with the term differences in the two documents as well as how to reduce the number of terms so we could make our predictive models. To reduce dimensions, we decided to do principal component analysis but only considered the first 50 PCAs to reduce computational time. We only did PCA on the training tf_idf matrix as the test would differ. To deal with term differences, we decided to only consider the terms that matched when calculating principal components. Additionally, we got rid of any terms that did not have any tf-idf values.

```
## Importance of first k=50 (out of 2500) components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     5.3158 4.33866 4.16045 3.91968 3.85121 3.83592 3.72543
## Proportion of Variance 0.0095 0.00633 0.00582 0.00517 0.00499 0.00495 0.00467
## Cumulative Proportion  0.0095 0.01583 0.02165 0.02682 0.03180 0.03675 0.04142
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     3.64014 3.48497 3.39092 3.27607 3.21116 3.19639 3.15146
## Proportion of Variance 0.00446 0.00408 0.00387 0.00361 0.00347 0.00344 0.00334
## Cumulative Proportion  0.04587 0.04996 0.05382 0.05743 0.06090 0.06434 0.06768
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     3.07907 3.04087 3.01378 2.96655 2.94492 2.89852 2.84844
## Proportion of Variance 0.00319 0.00311 0.00305 0.00296 0.00292 0.00282 0.00273
## Cumulative Proportion  0.07086 0.07397 0.07703 0.07999 0.08290 0.08573 0.08845
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     2.84250 2.81215 2.80766 2.76085 2.75120 2.73171 2.70868
## Proportion of Variance 0.00272 0.00266 0.00265 0.00256 0.00255 0.00251 0.00247
## Cumulative Proportion  0.09117 0.09383 0.09648 0.09904 0.10159 0.10410 0.10657
##                           PC29    PC30    PC31    PC32    PC33    PC34    PC35
## Standard deviation     2.65667 2.62913 2.60306 2.59740 2.58401 2.56659 2.55465
## Proportion of Variance 0.00237 0.00232 0.00228 0.00227 0.00225 0.00221 0.00219
## Cumulative Proportion  0.10894 0.11126 0.11354 0.11581 0.11806 0.12027 0.12246
##                           PC36    PC37    PC38    PC39    PC40    PC41    PC42
## Standard deviation     2.54741 2.53301 2.51021 2.50517 2.48770 2.47605 2.47245
## Proportion of Variance 0.00218 0.00216 0.00212 0.00211 0.00208 0.00206 0.00206
## Cumulative Proportion  0.12465 0.12680 0.12892 0.13103 0.13311 0.13518 0.13723
##                           PC43    PC44    PC45    PC46    PC47    PC48    PC49
## Standard deviation     2.47032 2.43510 2.42235 2.41439 2.41254 2.39042 2.38598
## Proportion of Variance 0.00205 0.00199 0.00197 0.00196 0.00196 0.00192 0.00191
## Cumulative Proportion  0.13928 0.14128 0.14325 0.14521 0.14717 0.14909 0.15100
##                           PC50
## Standard deviation     2.38523
## Proportion of Variance 0.00191
## Cumulative Proportion  0.15292
```

As we can see from the principal component results, the first 50 or so principal components explain about 15% or so of the variation of the original documents. Considering that there were originally 2500 documents, this dimension reduction is rather successful. We then decided to look into the first 5 principal components to determine what they mean.

```
##       beijing        china       leader         hong      chinese       chinas 
##   -0.10432160  -0.09435837  -0.08996598  -0.08656697  -0.08595910  -0.08481126 
## prodemocracy    democracy    political         kong     beijings        kongs 
##   -0.08297024  -0.08099828  -0.08096041  -0.08015108  -0.07805502  -0.07797614 
##          rule    communist       colony     analysts      cheehwa      million 
##   -0.07606513  -0.07400720  -0.07238052   0.07150891  -0.07113945   0.07102543 
##         party   democratic        share  legislature      percent      analyst 
##   -0.07095791  -0.07056207   0.06929569  -0.06925280   0.06901021   0.06846258 
##         human 
##   -0.06744353
```

The first principal component seems to weight against any Chinese documents and weights positively towards financial documents. Thus we think this principal component is Chinese vs Finance documents

```
##           gms           uaw       workers        plants          auto 
##  0.0007957783 -0.0031189501 -0.0041831858  0.0018396106  0.0054238012 
##         parts          mich     automaker    automakers        strike 
##  0.0027702230 -0.0004995527  0.0059741744  0.0061297187 -0.0039886232 
##         plant      detroits         index        motors         truck 
## -0.0009485902  0.0026724392  0.0270799346  0.0087792380  0.0085632212 
##        points     guarantee          ohio        stocks       strikes 
##  0.0239817939 -0.0077512681  0.0021959873  0.0379000190 -0.0017974563 
##           new       pattern     agreement       percent      contract 
##  0.0310056722  0.0016954278 -0.0070022265  0.0690102134 -0.0004560420
```

Our second principal component seems to focus on workers rights in the automobile industry.

```
##           gms           uaw       workers        plants         parts 
##  0.0007957783 -0.0031189501 -0.0041831858  0.0018396106  0.0027702230 
##     automaker          auto         plant         truck          mich 
##  0.0059741744  0.0054238012 -0.0009485902  0.0085632212 -0.0004995527 
##      detroits        strike    automakers       strikes        motors 
##  0.0026724392 -0.0039886232  0.0061297187 -0.0017974563  0.0087792380 
##       pattern      assembly          ohio     guarantee        pickup 
##  0.0016954278 -0.0200067639  0.0021959873 -0.0077512681  0.0166672367 
##      contract    guarantees    employment         motor        trucks 
## -0.0004560420 -0.0047957897  0.0008447635  0.0103179476  0.0092861970
```

Our third principal component also deals with the same issue and seems to wieght things very similarly to our second one.

```
##     computer      quarter     software    microsoft      cheehwa  legislature 
##  0.045453137 -0.028412335  0.043070762  0.038127942 -0.031347243 -0.027499606 
##        kongs    computers      windows          inc     machines      elected 
## -0.032489449  0.035501603  0.022945034  0.029689496  0.024891776 -0.027598214 
##         corp         tung     colonial     personal  provisional        chris 
##  0.063241891 -0.026970240 -0.025775633  0.022094989 -0.025678265 -0.023010076 
## prodemocracy    selection         wall   microsofts       fourth         hong 
## -0.028245494 -0.024323755 -0.008618346  0.019534922 -0.023156826 -0.034328330 
##        sales 
## -0.009913769
```

Our fourth principal component seems to be dealing with technology news and the Hong Kong protests. It weights tech news positively and the protests negatively. This is most likely a technology vs Hong Kong news type of deal.

```
##    democrats    elections         vote    coalition       senate   democratic
## -0.047276460 -0.054927335 -0.036283361 -0.043041111 -0.045968246 -0.042142659
##        house     majority   opposition        polls        seats       tonnes
## -0.040029299 -0.038489656 -0.031860457 -0.035606678 -0.034629629 -0.023025053
##       vaclav      beijing        china      chinese       voting   candidates
## -0.042726098 -0.035611637 -0.031679691 -0.033112142 -0.029323768 -0.029574084
##      elected        klaus  legislature      imports       chinas         said
## -0.027598214 -0.039988757 -0.027499606 -0.027550215 -0.041919615 -0.001614367
##       shares
## -0.056214470
```

Again we have a focus on political news especially regarding china. Judging from these 5 principal components, it seems that the best factors to determine author attribution to a document is basically whether they wrote about China or not. We then decided to run these principal components into multiple models including logistic regression, naive bayes, KNN, and a random forest m=5 tree model. To do so, we would combine our principal component analysis matrix with the author names in the training set into a training dataframe. The test dataframe would be constructed from the tf_idf test matrix and the list of authors in the testing set. Using the training dataframe, we would then make our model predicting author attribution based on the principal components. We would then test our model's prediction with the test dataframe and obtain an accuracy score. Out of all our models, the random forest tree model with m=5 performed the best.

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##      0.7436000      0.7383673      0.7260048      0.7606228      0.0200000
## AccuracyPValue  McnemarPValue
##      0.0000000            NaN
```

We used the library(caret) to create the confusion matrix. We found that our overall accuracy with random forests came out to be 73.84% which is better than our other models.

```
##      AaronPressman AlanCrosby AlexanderSmith BenjaminKangLim
## PC1      17.612654  15.795288      12.006282       17.490581
## PC2      16.931106  16.134970       9.973627       14.631515
## PC3      12.583953  19.874618      11.128195        6.827245
## PC4       9.731132  15.469098      13.917585       10.932236
## PC5      11.317921  15.870766      12.128144       14.607463
## PC6      22.470734  14.788033      14.707704       17.915866
## PC7      12.055065  13.926718      10.320445       10.236841
## PC8      13.835686  17.632847      15.342520       11.624441
## PC9      11.889261  14.709951      11.941013        8.633093
## PC10      7.471171   8.499273       8.237410        8.921593
## PC11     13.373647  12.629573       6.153249        9.195214
## PC12      8.946666  12.334494       8.778362        7.553729
## PC13      5.662194   9.407982       8.891912        5.219134
## PC14     13.396287  15.496796      12.255844       13.419992
## PC15      8.349312  11.614698       7.433326        8.071319
## PC16      7.509920  15.834781       8.916264        4.181743
## PC17      6.741958  13.211678       7.843687        5.815882
## PC18     10.482153  11.962798      10.580271        5.339238
```

```
## PC19     16.091749  13.164146       10.749003        11.091284
## PC20      7.957302   7.623295        7.515050         8.832686
## PC21      9.666791   9.896004        8.270912        13.496553
## PC22      7.649707  12.664022        9.935589         6.460846
## PC23      7.553912   7.172974        6.540325         3.993601
## PC24      8.639484   9.403717       10.025271         6.755617
## PC25      3.224262  12.751619        6.953862         8.292905
## PC26      9.990184  10.061676       12.075232         8.103367
## PC27      3.728726   7.779554        6.298783         5.028819
## PC28      6.976749   6.709181        7.259237         5.747794
## PC29      6.104193   7.632999        7.128862         4.184955
## PC30      4.619498   9.183978        7.251418         2.089856
## PC31     11.420106   6.477974        2.214721         1.801607
## PC32      8.692751   6.885868        4.216211         2.713873
## PC33      4.173999   4.565598        4.464669         4.317682
## PC34      4.861895   8.133085        5.644852         5.141520
## PC35      7.707828   6.888508        3.642108         1.338485
## PC36     13.213275   8.178398        9.515422         4.212463
## PC37      7.765796   7.051561        6.014672         2.930375
## PC38      5.349947   6.077594        9.158199         4.666177
## PC39      4.010756   8.055317        6.438026         3.444055
## PC40      3.854542   7.994113        5.239380         2.434877
## PC41      6.886660   9.879456        5.520163         6.682257
## PC42      6.384519   7.762481        5.843479         1.770156
## PC43      3.717220   7.542598        6.829686         3.494190
## PC44      4.130232   5.056386        6.038712         4.524827
## PC45      3.525186   8.168620        4.742220         5.131516
## PC46      1.969196   6.316022        7.546086         3.198902
## PC47      4.339983   4.784811        7.430795         3.607729
## PC48      4.248921   8.940835        5.796638         3.433510
## PC49      3.890851   9.204208        3.744290         4.418726
## PC50      2.959039  10.627811        4.357266         2.777786
```

From this snippet of the output of the importance function, we can see that the principal components importance vary across the authors but the first ones tend to be quite important towards predicting authors which makes sense considering those first principal components are the most important factors in the document.