Tim Miller
15.071 | HW #6

PROBLEM 1A

```
> table(reviews$review_scores_rating)

  1    2    3    4    5
 62   56  156  708 3191
```

Using the results from above, it appears we have

- 62 “1” scores
- 56 “2” scores
- 156 “3” scores
- 708 “4” scores
- 3191 “5” scores

PROBLEM 1B

Aggregating by review_scores_rating, we find the following:

```
> aggregate(reviews$review_length, list(reviews$review_scores_rating), mean)
  Group.1        x
1       1 464.5968
2       2 597.7321
3       3 388.8013
4       4 276.4760
5       5 289.4184
```

The immediate takeaway is that lower review scores have more characters in the text feedback. This is not surprising. Typically when someone leaves a bad review they will include a long list of complaints – driven by anger or frustration – justifying their low rating.

PROBLEM 1C

corpus = tm_map(corpus, tolower)

This command changes all characters to lower case.

corpus = tm_map(corpus, removePunctuation)

This command removes all punctuation from the corpus.

corpus = tm_map(corpus, removeWords, stopwords("english"))

This command removes all stop works such as ‘the’ ‘and’ ‘but’ etc.

corpus = tm_map(corpus, removeWords, c("airbnb","apartment","location","place","room","host","stay"))

This command removes words commonly found in Airbnb reviews. Since they are common based on the context of an Airbnb review, we do not want them to influence our predictions.

corpus = tm_map(corpus, stemDocument)

This command returns the root of words (i.e., the word 'stem') in the corpus. This helps us remove some of the clutter and noise in the cluster and get to the most important part of what the review is saying.

Running the following command returns the following string for the first document:

```
> strwrap(corpus[[1]])
[1] "good issu direct instruct differ actual properti"
```

PROBLEM 1D

Positive reviews word cloud

Negative reviews word cloud

Comments

- Different colors for different words, makes it easy to read
- In R Studio we can hover over each word to see the number of times it occurs
- Interesting to see ‘clean’ appear in both positive and negative word clouds. Likely in the negative reviews ‘not’ appears in front of the word ‘clean’ but that was filtered out with the stop words cleaning step. It would be interesting to concatenate words to see if ‘not_clean’ would show up frequently.

PROBLEM 1E

```
# Calculate the frequency of each words over all reviews.
frequencies = DocumentTermMatrix(corpus)
frequencies

findFreqTerms(frequencies, lowfreq=900)
```

We find the following terms appear at least 900 times:

- recommend
- great
- clean
- nice

PROBLEM 1F

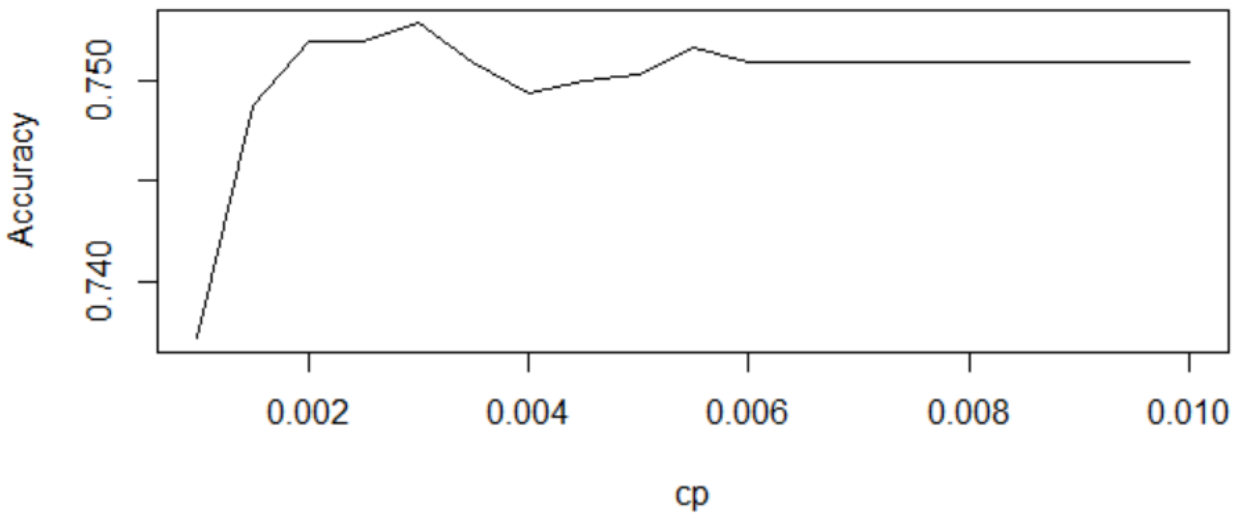
After removing the sparse words, we now only have 404 words in the corpus.

```
> sparse = removeSparseTerms(frequencies, 0.99)
> sparse
<<DocumentTermMatrix (documents: 4173, terms: 404)>>
Non-/sparse entries: 68927/1616965
Sparsity           : 96%
Maximal term length: 13
Weighting          : term frequency (tf)
>
```

PROBLEM 1G

We want to model with the highest accuracy. So we plot cp vs Accuracy:

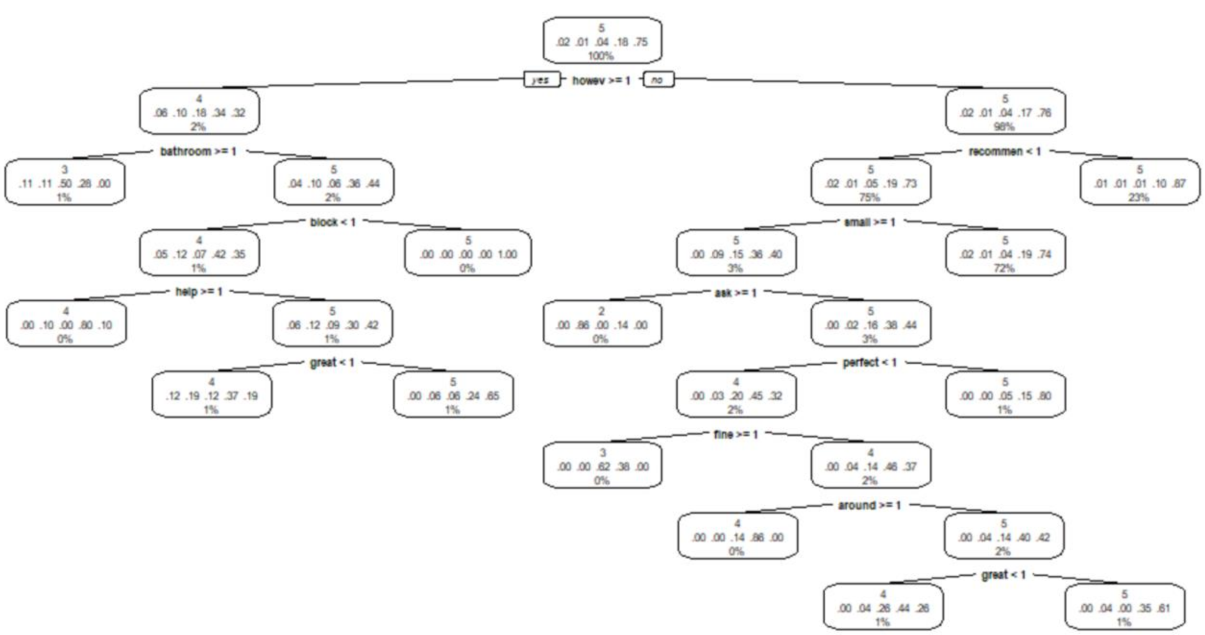
```
plot(cv.trees$results$cp, cv.trees$results$Accuracy, type = "l", ylab = "Accuracy", xlab = "cp")
```



We get a value of 0.0030 for cp.

PROBLEM 1H

Plot of the tree with cp = 0.0030.



Looking at the tree there are 2 leaves with a score of 3 and 1 leaf with a score of 2. Breakdown of features for the node with predicted score of 2:

- ‘recommen’ is not mentioned
- ‘small’ is mentioned greater than or equal to 1 time
- ‘ask’ is mentioned greater than or equal to 1 time
- ~86% of the responses that fall in the bucket will be predicted to have a score of 2
- Remaining ~14% of the responses that fall in the bucket will be predicted to have a score of 4
- Contains <1% of responses

PROBLEM 1I

Yes, the tree aligns with my intuition. Looing at the predicted score of “2” node we see that the reviews will not contain “recommend, ” “small” & “ask will be mentioned more than 1 time. I can see a scenario where someone writes a review where they don’t include the word recommend because it was a bad experience, perhaps due to it being small. They might also include “ask” in there because they had to ask for a refund.

Additionally, for the node that predicts a score of 3 we see “however” and “bathroom” are mentioned more than 1 time. It is safe to assume that a reviewer might have started out the review saying the Airbnb was fine but then qualified it with “however” to add some additional detail e.g., the bathroom was not clean or had some other issue.

PROBLEM 1J

Model accuracy for training set

```
# Assessing the out-of-sample performance of the CART model, training set
predictions.cart <- predict(cart.mod, newdata=train, type="class")
matrix.cart = table(train$review_score, predictions.cart) # confusion matrix
accuracy.cart = (matrix.cart[1,1]
                 + matrix.cart[2,2]
                 + matrix.cart[3,3]
                 + matrix.cart[4,4]
                 + matrix.cart[5,5])/nrow(train)
accuracy.cart
```

```
> matrix.cart
   predictions.cart
       1    2    3    4    5
  1    0    0    2    2   50
  2    0    6    2    5   32
  3    0    0   14   10  110
  4    0    1    8   32  520
  5    0    0    0   11 2383
```

We find the model accuracy for the training set is **0.7638018**.

Model accuracy for test set

```r
# Assessing the out-of-sample performance of the CART model
predictions.cart <- predict(cart.mod, newdata=test, type="class")
matrix.cart = table(test$review_score, predictions.cart) # confusion matrix
accuracy.cart = (matrix.cart[1,1]
                 + matrix.cart[2,2]
                 + matrix.cart[3,3]
                 + matrix.cart[4,4]
                 + matrix.cart[5,5])/nrow(test)
accuracy.cart
```

```
> matrix.cart
   predictions.cart
     1   2   3   4   5
  1  0   0   0   0   8
  2  0   0   1   1   9
  3  0   0   1   4  17
  4  0   0   1  15 131
  5  0   0   2   6 789
```

We find the model accuracy for the test set is **0.8172589**.

PROBLEM 1K

Code for calculating out of sample accuracy by star level:

```r
# Assessing the out-of-sample performance of the CART model, test set
predictions.cart <- predict(cart.mod, newdata=test, type="class")
matrix.cart = table(test$review_score, predictions.cart) # confusion matrix

# 1-star out of sample accuracy

accuracy.cart = (matrix.cart[1,1]) / nrow(test[test$review_score == "1", ])
accuracy.cart

# 2-star out of sample accuracy

accuracy.cart = (matrix.cart[2,2]) / nrow(test[test$review_score == "2", ])
accuracy.cart

# 3-star out of sample accuracy

accuracy.cart = (matrix.cart[3,3]) / nrow(test[test$review_score == "3", ])
accuracy.cart

# 4-star out of sample accuracy

accuracy.cart = (matrix.cart[4,4]) / nrow(test[test$review_score == "4", ])
accuracy.cart

# 5-star out of sample accuracy

accuracy.cart = (matrix.cart[5,5]) / nrow(test[test$review_score == "5", ])
accuracy.cart
```

Accuracy values by star level:

- 1 star – 0
- 2 star – 0
- 3 star - 0.04545455
- 4 star - 0.1020408
- 5 star - 0.9899624

4 star and 5 star reviews have the highest out of sample accuracy. This makes sense given we have the most training data for those two categories.

PROBLEM 1L

Bag of words representation might fail when there is a temporal component to the prediction. In other words, you cannot just create random bag of words: the order that the words occur in have some impact on the type of prediction that is made. For example, if you are predicting the sentiment of some time of text, you may need the words in a specific order to ensure you are pulling out the right context that is driving the sentiment of word source.

To improve bag of words, I would include some type of time tag or indicator on the training data to force the ordering of words and ensure that the model was trained with the right level of context.

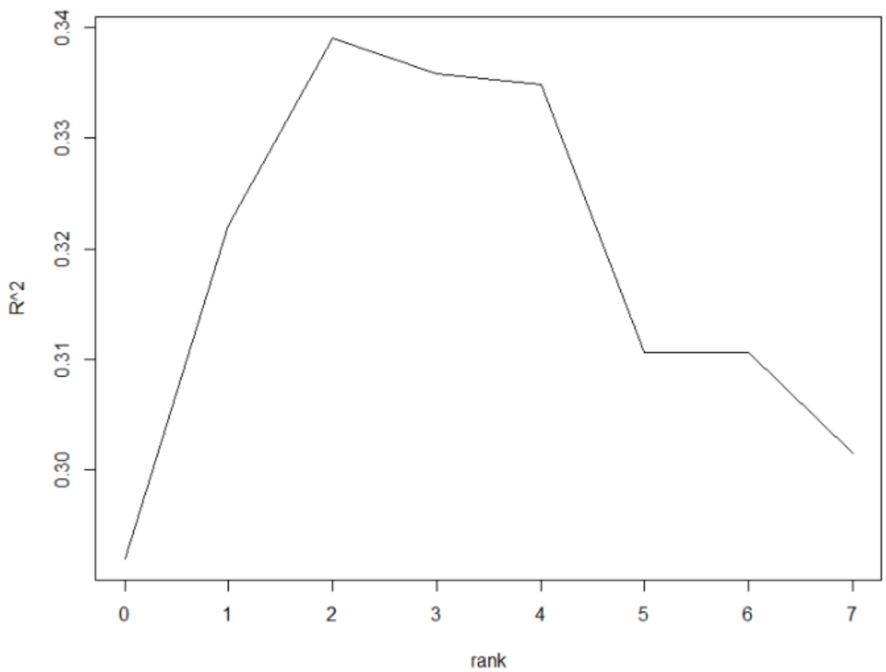
PROBLEM 2A

```
# Evaluate ranks
set.seed(789)
rank.info <- cf.evaluate.ranks(music.train, 0:7, prop.validate=0.05)

rank.info

plot(rank.info$rank, rank.info$r2, type = "l", ylab = "R^2", xlab = "rank")
```

Plotting R^2 vs rank we get



Given we want to maximize R^2 we select a rank of 2.

Intuitively this makes sense. Thinking of how my friends and I listen to music, we typically prefer 2 or 3 different genres and focus on those. In the 'Songs.csv' database we see there are 7 different possible genres. With the optimal 2 ranks we have identified; the model is likely grouping people by their preference for ranking 2 – 3 genres. As stated before, this matches how I typically think people focus their music listening habits.

PROBLEM 2B

```
# Fit collaborative model using center matrix
set.seed(157)
fit <- softImpute(mat.scaled, rank.max=2, lambda=0, maxit=1000)

# Make out-of-sample prediction
pred_outsample_0 <- impute(fit, music.test$userID, music.test$songID, unscale = TRUE)

# look at distribution
hist(pred_outsample_0)

#set min of 1 and max of 4
pred_outsample <- pmax(pmin(pred_outsample_0, 4), 1)

#check updated output and confirm bounds
hist(pred_outsample)

# Calculate osr^2
R2_outsample <- 1 - sum((pred_outsample-music.test$rating)^2)/sum((mean(music.test$rating) - music.test$rating)^2)
R2_outsample
```

The out of sample R^2 for the model is 0.3084477.

Note: in the above code you will see I set prediction scores at a minimum of 1 and a maximum of 4. This was based on the input data from 'MusicRatings.csv' where there was no score below 1 and no score above 4.

PROBLEM 2C

Top 10 songs with greatest negative difference

```
> # Top 10 greatest negative difference
> head(songs[order(songs$diff),], 10)
    songID                   songName year              artist      genre      score1       score2       diff
513    513           Sayonara-Nostalgia 2005       Base Ball Bear       Rock -0.16923460  0.152875011 -0.3221096
701    701                 16 Candles 1988           The Crests       Rock -0.12969976  0.181726596 -0.3114264
690    690 Video Killed The Radio Star 1979          The Buggles       Rock -0.11724456  0.177819117 -0.2950637
537    537      Better To Reign In Hell 2003      Cradle Of Filth       Rock -0.12784414  0.166751783 -0.2945959
448    448                 ReprÃ©sente 1999     Alliance Ethnik        Rap -0.22974017  0.064517690 -0.2942579
635    635      Make Love To Your Mind 1975        Bill Withers        RnB -0.21431737  0.075559983 -0.2898773
673    673           The Big Gundown 2009         The Prodigy Electronic -0.07924804  0.194646175 -0.2738942
790    790 A Beggar On A Beach Of Gold 1995 Mike And The Mechanics       Rock -0.09675220  0.157814552 -0.2545667
637    637                    Invalid 2002            Tub Ring       Rock -0.25473214 -0.008063062 -0.2466691
54      54               You're The One 1990        Dwight Yoakam    Country -0.34325587 -0.096992048 -0.2462638
>
```

We see that archetype 2 has positive values for rock while archetype 1 generally has negative scores. Based on this sample of 10 songs we can generally say that archetype 2 likes rock while archetype 1 does not prefer it.

Top 10 songs with greatest positive difference

```
> # Top 10 greatest positive difference
> head(songs[order(-songs$diff),], 10)
    songID                    songName year               artist      genre        score1      score2       diff
439    439                     Secrets 2009          OneRepublic       Rock -0.0438074433 -0.18390572 0.14009828
221    221            Livin' On A Prayer 1986             Bon Jovi       Rock  0.0118257995 -0.10943791 0.12126371
562    562                   Alejandro 2009            Lady GaGa        Pop -0.0106344575 -0.12960010 0.11896564
630    630                    Marry Me 2009                Train        Pop -0.0460526673 -0.16336591 0.11731324
368    368                 Bulletproof 2009              La Roux        Pop -0.0106450523 -0.12000622 0.10936117
761    761                 Cosmic Love 2009 Florence + The Machine       Rock -0.0028855932 -0.11178834 0.10890274
498    498             Creep (Explicit) 1993            Radiohead       Rock  0.0201216187 -0.08548524 0.10560685
736    736             I Gotta Feeling 2009      Black Eyed Peas        Pop  0.0264158180 -0.07767821 0.10409403
751    751                Electric Feel 2007                 MGMT       Rock -0.0001860719 -0.10103645 0.10085038
10      10 Harder Better Faster Stronger 2007            Daft Punk Electronic  0.0373182160 -0.05805891 0.09537713
>
```

We see that archetype 1 and archetype 2 are quite different in their scoring on pop songs. Perhaps this suggests that archetype 1 might like pop sounding songs a little bit more than archetype 2. Interesting that archetype 2 has negative scores for rock songs, while previously they had positive scores for rock songs. My guess is that some of the bands classified as rock should really be under pop, therefore the more hardcore rock fans in archetype 2 are ranking them negatively.

PROBLEM 2D

Data manipulation

```
# aggregate difference by artist
artist_diff = aggregate(songs$diff, by=list(songs$artist), FUN=sum)

# aggregate by number of songs per artist
song_count = aggregate(songs$artist, by=list(songs$artist), FUN=length)


# filter for artists with 4 songs
songs_min = song_count[which(song_count[,2]>=4),]

#confirm correct filtering
songs_min[order(-songs_min$x),]

# filter for artists with 4 or more songs
artist_diff_2 = filter(artist_diff,
                       Group.1 %in% songs_min$Group.1)

# Top 5 greatest negative difference
head(artist_diff_2[order(artist_diff_2$x),], 5)

# Top 5 greatest positive difference
head(artist_diff_2[order(-artist_diff_2$x),], 5)
```

<u>Top 5 greatest negative differences</u>

```
> head(artist_diff_2[order(artist_diff_2$x),], 5)
                Group.1          x
38      Octopus Project -1.0038446
46 the bird and the bee -0.5896226
43               Skream -0.3994692
42   Simian Mobile Disco -0.3671752
5             Boys Noize -0.3339435
>
```

<u>Top 5 greatest positive differences</u>

```
> head(artist_diff_2[order(-artist_diff_2$x),], 5)
         Group.1         x
9       Coldplay 0.8981082
3         Beirut 0.8619104
48   The Killers 0.7029997
44 Sleater-kinney 0.4368333
26      Lady GaGa 0.4342801
```

We see that the top 5 negative / positive differences are different than what we saw the previous result in problem 2c. For the top 5 negative differences we see several electronic artists including Skream, Simian Mobile Disco, and Boys Noize. This suggests that archetype 2 likes electronic more than archetype 1.

For the top 5 positive differences we see different results than what we got in problem 2c. Most of the artists on this list are classified mostly as rock. Therefore this suggests overall that archetype 1 likes rock more than archetype 2.

PROBLEM 2E

Unsure how to apply the model here

PROBLEM 2F

Unsure how to apply the model here