Jeff Levesque

Professor Gates

IST 565

Homework #8

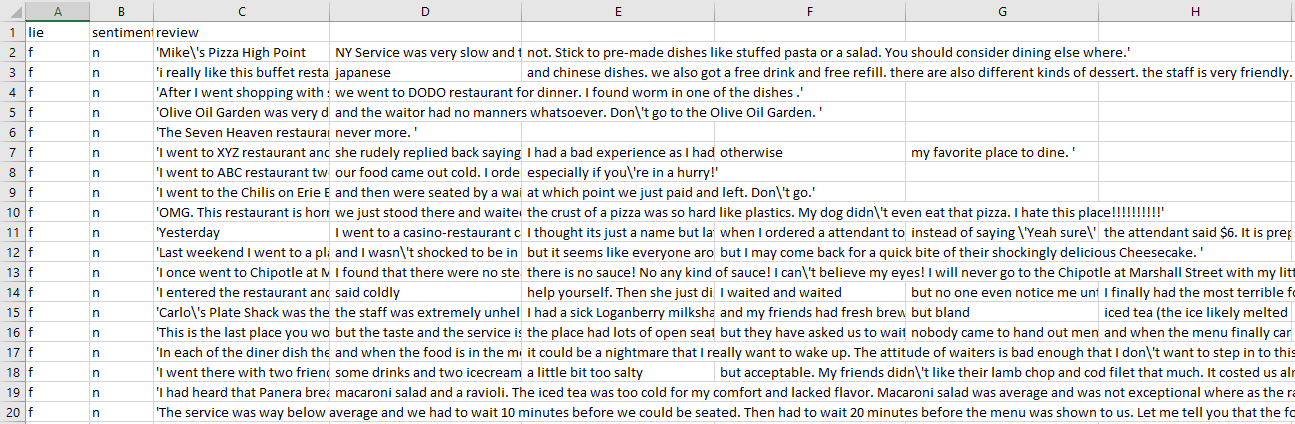
**Introduction**

**Analysis**

Data Preparation:

A single dataset was represented twice with two different file formats. The first was an arff file format, native to the weka application, and the second a csv file. This provided an option to use either weka, or an alternative csv reader. Since custom R code was written, the csv file was used.

When inspecting the csv dataset, it was immediately noticed there was 92 records, while the number of columns being non-standard. More generally, the number of columns varied in a given row, between three, to any 3+n columns. This was because the review could split into any number of columns:



Therefore, the 3+n columns needed to be collapsed into a single review column. Therefore, the column name needed to be read separately from successive rows. However, initial attempts to load the provided dataset, produced incorrect number of rows. More generally, the corresponding dataframe yielded only 45 entries:

> filepath = 'data/deception\_data\_converted\_final.csv'

> df.colnames = read.table(filepath, nrow=1, stringsAsFactors=FALSE, sep=',')

> df.full = read.table(filepath, skip=1, header=FALSE, sep='\n', fill=TRUE)

> count.fields(filepath, sep='\n', blank.lines.skip=FALSE)

[1] 1 NA NA NA 1 1 NA 1 NA 1 1 NA NA NA NA NA 1 NA NA NA NA 1 1 1 1 NA NA 1 1 NA NA 1 1 1 1 1

[37] 1 1 1 1 1 1 1 1 1 NA NA NA NA NA NA NA NA 1 NA NA NA 1 1 1 1 1 1 1 1 1 NA 1 NA NA NA NA

[73] NA 1 1 1 NA NA NA NA NA NA NA NA NA NA 1 1 1 NA NA 1 1

To account for missing attributes, additional attributes for read.table was required:

## import dataset

filepath = 'data/deception\_data\_converted\_final.csv'

df.colnames = read.table(filepath, nrow=1, stringsAsFactors=FALSE, sep=',')

df.full = read.table(

filepath,

skip=1,

header=FALSE,

sep='\n',

quote = '',

comment.char = ''

)

Then, the corresponding single column dataframe was exploded into three columns:

out = stri\_split\_fixed(str = df.full[, c(1)], pattern = ',', n = 3)

df.split = as.data.frame(do.call(rbind, out))

colnames(df.split) = df.colnames

df.split$review = as.character(df.split$review)

Once the dataframe was loaded, the feature set was reduced, to help decrease the sparsity of train, and test matrix. Specifically a feature set of 1462 was reduced to 118. This was done by removing any column with a sum less than 0.3:

df.merged = df.merged[, colSums(df.merged) > 0.3]

After the dataset preprocessing completed, the text2vec package was used to tokenize a bag of words vocabulary, into a document term matrix. Then, the term frequency-inverse document frequency was applied to the document term matrix:

it\_train = itoken(

df.split$review,

preprocessor = tolower,

tokenizer = word\_tokenizer

)

vocab = create\_vocabulary(it\_train)

vectorizer = vocab\_vectorizer(vocab)

model\_tfidf = TfIdf$new()

dtm\_tfidf = model\_tfidf$fit\_transform(create\_dtm(it\_train, vectorizer))

The gmum.r package for multiclass svm was initially attempted. However, the package was removed from the cran-r repository. Additionally, attempting to clone the corresponding git codebase, then load the package within R code, produced errors. More generally, the corresponding package could not be found, and loaded into memory. Therefore, the naivebayes, and e1071 packages were used to implement the naïve bayes, and svm models. Lastly, the FSelector package was used to determine the top20 classifiers, using the gain.ratio, and the chi2 functions.

**Results**

Due to time restriction, the svm case indicates a need to perform many multiple one vs one classifiers. This would be done with one classifier being trained to one feature, then compared collectively to the remaining features.

**Conclusions**