System Used:

MacBook Pro - 2.8 GHz Intel Core i7, 16 GB 2133 MHz LPDDR3

Overview:

The objective of this project is to create a model from the IMDB movie reviews data which can predict the sentiment of any future unseen reviews.
Input data used is a set of IMDB movie review which has already been tagged as positive or negative sentiments.

Output of the project is to get a ML model which can help predict the sentiments of the review with least error.

Preprocessing of Data/Customized Vocabulary:

Tokenization: Used tokenization to split the review strings into word tokens

Converted all the words to lower case words to make words treated as same irrespective of lower case or upper-case letters used.

Defined the list of some common stop words, which usually does not provide any additional meaning to the sentiment.

Used stop words: "i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours", "their", "they", "his", "her", "she", "he", "a", "an", "and", "is", "was", "are", "were", "him", "himself", "has", "have", "it", "its", "of", "one", "for", "the", "us", "this"

Used 'create_vocabulary' function in R from the library text2vec to create a word vocabulary. Vocabulary is built using ngram 1L & 2L (uni and bigram).

Using 'prune_vocabulary' function from the text2vec library, pruned the highly frequent and rare words to consolidate the vocabulary further. Parameters used: term_count_min = 5, doc_proportion_max = 0.5, doc_proportion_min = 0.001)

Used 'vocab_vectorizer' to map words to indices and create a vector for create document term matrix in the next step.

Used 'create_dtm' function to create document term matrix for the training data.
Used the tokenization and document term matrix steps for the test data.

Applied the screening method on the train data using two sample t-test and reduced the vocabulary size to 2500 using the t-statics.

Training/Technical Details:

Final Model: Used cv.glmnet to find the minimum lambda which I then used in the training using ridge regression using the function glmnet.

Other Models tried: Also tried lasso regression on the data with various lambda values and number of folds.

Model Validation:

Ran and evaluated the code using variable s=1,2,3 on three train and test splits.
This being a classification problem Area under the ROC Curve (AUC), which provides is a good measure of accuracy among all possible thresholds, is being used as evaluation criterion.

The accuracy numbers below for respective spilt are AUC results.

Performance for Split 1	0.9637
Performance for Split 2	0.9634
Performance for Split 3	0.9641
Vocab Size	2500
Total time taken in Split 3	146.541 seconds
Total time taken in Split 1	145.966 seconds
Total time taken in Split 2	143.023 seconds

Next steps:

Plan to work on further reducing the vocabulary without compromising the accuracy, rather try to get better results with smaller vocabulary by trying different ML algorithms. Would be interesting to apply advanced text retrieval techniques such as stemming, working with different n-gram models with the aim of improving the accuracy with reduced vocabulary size.