ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CS410: Final Project Report

IMDB Movie Review Sentiment Analysis

Student:
Suganya Somu (NetID: ssomu2)
Raghavendran Ramasubramanian (NetID: rr26)

Table of Contents

1. Introduction

1.1 Background

We are using the IMDB movie dataset. The core dataset contains 50,000 reviews split evenly into 25k train and 25k test sets. The overall distribution of labels is balanced (25k pos and 25k neg). We also include an additional 50,000 unlabeled documents for unsupervised learning.

In the entire collection, no more than 30 reviews are allowed for any movie because reviews for the same movie tend to have correlated ratings. Further, the train and test sets contain a disjoint set of movies, so no significant performance is obtained by memorizing movie-unique terms and their association with observed labels. In the labeled train/test sets, a negative review has a score <= 4 out of 10, and positive thinking has a score >= 7 out of 10. Thus reviews with more neutral ratings are not included in the train/test sets. In the unsupervised set, reviews of any rating are included, and there are an even number of reviews > 5 and <= 5.

1.2 Objective

We aim to develop a binary classification model with a vocabulary size of fewer than 1000 words that can accurately predict the opinion expressed in a review. We will utilize the same vocabulary for each of the five training and test datasets, and our goal should be to achieve an overall AUC for the five test datasets equal to or greater than 0.95.

2. Dataset

2.1 Input

Our input data is tab-delimited and begins with a header row, followed by 25,000 rows containing an id, a sentiment, and the full text of a review. We are going to separate these data into five sets, and then we will create a document-term matrix. In this matrix, each row will represent a different document, each column will represent another word, and each entry will represent the number of times that particular word appears. This will help us develop a vocabulary, so keep it handy. In addition, we will provide the model with the training data from its initial iteration.

The data set, **alldata.tsv**, has 50,000 rows (i.e., reviews) and 4 columns:

- Col 1: "id", the identification number;
- Col 2: "sentiment", 0 = negative and 1 = positive;
- Col 3: "score", the 10-point score assigned by the reviewer. Scores 1-4 correspond to negative sentiment; Scores 7-10 correspond to positive sentiment. This data set contains no reviews with scores of 5 or 6.
- Col 4: "review".

2.2 Output

Output is the probability of positive sentiment for each document.

3. Implementation:

3.1 Vocab Creation

- Using text2vec packages, we created a document-term matrix from the first split of training data to generate the vocabulary.
- All of the reviews have removed their HTML tags; they have been tokenized and converted to lowercase; the vocabulary was initially constructed using an n-gram of size four, and then the stop words were taken out.
- We calculated the t-statistics for each vocabulary from the various sentiment groups, and then we extracted the first 2000 words that had the greatest significant absolute value of t-statistics.
- By applying logistic regression with an L1 penalty, we reduced the vocabulary, even more, bringing it down to a total of 998 words with an n-gram size of 2.

3.2 Model Implementation:

- For our training data, we are now utilizing the custom vocabulary file that was previously mentioned.
- Our five-fold cross-validation study employed logistic regression with Ridge (L2 penalty) and a minimal lambda value.

3.3 Tuning Parameter:

- As we could not achieve the ideal AUC when utilizing Logistic Regression with Ridge while utilizing the default Lambda min, we were required to engage in cross-validation for a total of five folds to obtain the desired lambda.
- Since the present split data set will not change, the values have been locked in at 0.06. This brings the total execution time of the model down to a more manageable level. Before the execution of the model, the real-world design will use cross-validation to locate the minimum value of lambda.

4. Project code:

The final project code is available in the GitHub location - https://github.com/raghavan7/CS410_Final_Project

```
├── main.R                                          <-- R file containing model execution and AUC check
├── vocab_creation.R                                <-- R file containing Vocabulary creation from first split
├── run_docker.sh                                   <-- sh file containing docker run command (mention below in
step 3)
├── Dockerconfig
|   |
|   └── requirements.R                              <-- library requirement file to install dependencies in doc
ker image
├── data
|   |
|   ├── split*
|   |   ├── myvocab.txt                             <-- bag of words generated from vocab_creation.R script
|   |   ├── prediction_results.txt                  <-- prediction results on test data from model run
|   |   ├── test_y.tsv
|   |   ├── test.tsv                                <-- test data for model
|   |   └── test.tsv                                <-- train data for model
|   ├── alldata.tsv                                 <-- full imdb dataset
|   └── splits.csv                                  <-- file with split information to generate split folders
├── Project Progress Report - Sentients.pdf
├── Project Proposal - Sentients.pdf
├── Dockerfile                                      <-- configuration for Docker image
└── README.md
```

5. Execution:

Below are steps to perform the execution and evaluation of the model designed for sentiment analysis to generate the required AUC score. The project code setup is dockerized for ease of setup and use.

5.1 Docker Creation and Run:

Build Docker

Step 1. `docker build --rm --force-rm -t rstudio/sentients .`

Step 2. `docker image list | grep sentients`

Run Docker

Step 3. `DATA_DIR=${PWD}/Data docker run -it --rm -m 4g --name sentients -e USERID=$UID -v $DATA_DIR:/home/rstudio/data rstudio/sentients /bin/bash`

5.2 Vocab Creation and Model Execution:

Test Model

Once the step 3 run is successful, the root directory is shown in the command line.

Step 4. `cd /home/rstudio`

Step 5. `Rscript vocab_creation.R`

Step 6. `Rscript main.R`

6. Results:

The model uses a vocabulary list of 998 words – it performs well on the test data and gets us the target AUC of greater or equal to 0.96.

6.1 Model Evaluation Results:

Splits	AUC	Runtime (in Seconds)
1	**0.9608**	**11.58**
2	**0.9639**	**11.21**
3	**0.9639**	**12.99**
4	**0.9642**	**11.39**
5	**0.9637**	**10.97**

7. Technical Specs:

Using a 2020 model of MacBook Pro (which has a 2.3 GHz Dual-Core Intel i7 CPU with 16 GB RAM and macOS Catalina operating system), the average running time for 5 Splits is about **11.63 seconds.**

8. Model Limitations:

A bag of words is unable to capture sequential information. We could not tune the Logistic regression with the L2 penalty for a greater AUC. It lacks flexibility.

8.1 Error Explanation:

Sentiment analysis technologies can identify and analyze numerous parts of the text quickly and automatically. However, computer programs have difficulty perceiving sarcasm and irony, negations, jokes, and exaggerations, which a human would have no trouble identifying. And failure to acknowledge them can affect outcomes.

Example 1: False Positive [Actual Rating by user: 3/10 and Sentiment: 0 (Not a good movie)]

*"I've almost forever been against including songs in a movie. My belief was that the quality of the film would automatically be improved if only those extremely annoying songs would be axed. However, things have quickly changed after watching that horrible Black (no songs) & this movie, Page 3 (plenty of songs). While Black was weak to an extreme, Page 3 delivers a gripping story with some strong acting & good direction. The songs were almost incidental & blended in almost seamlessly with the film. There certainly weren't any women getting sprayed with water for no apparent reason from mysterious water sources while gyrating wildly on the streets at night.

I was pleasantly surprised with the bold and unabashed approach used by the director. There was no glossing over of anything and almost every scene was completely believable.

I'd recommend this films for Hindi-speaking people with at least a slight understanding of Mumbai life. The former because the English subtitling was below par and contained many errors which, at times, completely reversed the meaning of the actual statement. The latter because you'll definitely appreciate the accuracy of the depiction once you've lived it yourself.

I'd definitely rank this as a work worthy of international recognition. The scenes with the gossiping drivers was a nice touch and it served simultaneously as a source of genuine humour as well as another perspective on the whole mishmash. The movie does fall short in a few places though, where the characters sometimes say the most inexplicable things which detract from the overall direction of the film.

I also thought that a couple of the sadder scenes were not done very well. It was a touch amusing to watch, rather than arouse any feelings of sadness & the whole scene tended to come across as a bit foolish. These are minor issues though, because the film, on the whole, is truly a rare treat to watch.

Overall, it's a cynical, pessimistic outlook and a refreshing one at that! Actors, not 'heroes' - that's the key. A chance to glimpse believable human beings in an*

*extraordinary setting - everyday life. A behind-the-scenes look at the extent of the depravity and a rare ray of hope for Indian cinema.

8/10."*

Example 2: False Negative [Actual Rating by user: 10/10 and Sentiment: 1 (Good movie)]

*"This movie is stuffed full of stock Horror movie goodies: chained lunatics, pre-meditated murder, a mad (vaguely lesbian) female scientist with an even madder father who wears a mask because of his horrible disfigurement, poisoning, spooky castles, werewolves (male and female), adultery, slain lovers, Tibetan mystics, the half-man/half-plant victim of some unnamed experiment, grave robbing, mind control, walled up bodies, a car crash on a lonely road, electrocution, knights in armour - the lot, all topped off with an incredibly awful score and some of the worst Foley work ever done.

The script is incomprehensible (even by badly dubbed Spanish Horror movie standards) and some of the editing is just bizarre. In one scene where the lead female evil scientist goes to visit our heroine in her bedroom for one of the badly dubbed: \That is fantastical. I do not understand. Explain to me again how this is...\" exposition scenes that litter this movie, there is a sudden hand held cutaway of the girl's thighs as she gets out of bed for no apparent reason at all other than to cover a cut in the bad scientist's \"Mwahaha! All your werewolfs belong mine!\" speech. Though why they went to the bother I don't know because there are plenty of other jarring jump cuts all over the place - even allowing for the atrocious pan and scan of the print I saw.

The Director was, according to one interview with the star, drunk for most of the shoot and the film looks like it. It is an incoherent mess. It's made even more incoherent by the inclusion of werewolf rampage footage from a different film The Mark of the Wolf Man (made 4 years earlier, featuring the same actor but playing the part with more aggression and with a different shirt and make up - IS there a word in Spanish for \"Continuity\"?) and more padding of another actor in the wolfman get-up ambling about in long shot.

The music is incredibly bad varying almost at random from full orchestral creepy house music, to bosannova, to the longest piano and gong duet ever recorded. (Thinking about it, it might not have been a duet. It might have been a solo. The piano part was so simple it could have been picked out with one hand while the player whacked away at the gong with the other.)

This is one of the most bewilderedly trance-state inducing bad movies of the year so far for me. Enjoy.

Favourite line: \"Ilona! This madness and perversity will turn against you!\" How true.

Favourite shot: The lover, discovering his girlfriend slain, dropping the candle in a cartoon-like demonstration of surprise. Rank amateur directing there."*

Examples 1 and 2 are reviews that are misclassified by the model. If we read the review, we will realize the author in Example 1 scoffs at the film, and in Example 2 is, appreciating the movie plot and make. As we use the bag of words for our model, which doesn't consider the sequential information, and the review does have many positive words in Example 1 and many negative comments in Example 2, it is reasonable to have this error.

9. Interpretability:

Our model is based on a simple Logistic Regression. Thus it is easy to comprehend how the vocabulary is being built and utilized to score the model. Different ML algorithms often employ distinct learning methodologies, resulting in model transparency and complexity variations. A neural network, for instance, operates in a black box-like manner when producing a result, meaning that the transparency is inadequate, i.e., users cannot understand explicitly why the output is produced. In contrast, rule-based knowledge representations can explain the reason for delivering work clearly, making it more explicit.

Ours is a regression model, and we provided some instances above of how we use the vocab to score the model. We can always clarify the logical reasons for a specific sentiment score if a scenario requires an explanation.

10. Did we Achieve the Objective?

Overall we achieved the objective of our project, reaching the AUC > 0.95 with a vocabulary size of fewer than 1000 words. We modified the distribution from 70:30 split to 80:20 split to generate the bag of words. And have to perform cross-validation to identify the ideal lambda value (0.06) as default for logistic regression was not yielding expected results.

11. Team effort:

Suganya Somu (NetID: SSOMU2) - Focused on pre-processing data, working on Logistics regression model, and generating vocabulary and report compilation.
Raghavendran Ramasubramanian (NetID: RR26) - Focused on computing accuracy for five splits and performance run time, report compilation.

12. Acknowledgments:

- Campuswire posts
- [text2vec] R vignettes: https://cran.r-project.org/web/packages/text2vec/vignettes/text-vectorization.html