Paper 5012-2020

# MOVIE REVIEWS: TO READ OR NOT TO READ!
## Spoiler Detection with Applied Machine Learning

Sreejita Biswas, Oklahoma State University;
Goutam Chakraborty, Oklahoma State University

## ABSTRACT

Would you still watch The Avengers if you found out that Iron Man dies at the end? Would you watch the final season of Game of Thrones if somebody told you Bran was going to be the king of the North? Highly unlikely! Given that many invest a great deal of time and emotions into movies and T.V. shows, having the experience ruined by, otherwise harmless, scrolling through Facebook can be frustrating. Having always been a movie lover myself, I have experienced first-hand how spoilers can take away from the enjoyment of a movie. Along with being detrimental to viewer experience, these revelations may also hurt the entertainment industry by causing people to lose interest in a particular film, resulting in the loss of revenue for the filmmakers.

This research is aimed at spotting spoilers in online movie reviews in order to protect the reader's movie watching experience. To this end, the paper will explore the likelihood of a spoiler based on the movie's genre, rating, and year of release. Few reviewers were identified who always post spoilers in their reviews. The Jupyter notebook within SAS® Viya has been leveraged for this project along with various concepts of Natural Language Processing to predict if the model and features considered are able to predict spoilers in a movie or not. The paper also builds and compares models such as Neural Network, Gradient Boosting, and Random Classifier to score new reviews datasets to predict spoilers.

## INTRODUCTION

Reviews and comments about a movie can often play a significant role in its popularity. This is because prior to deciding whether to watch a movie or show, potential viewers may read multiple reviews of it. One unintended consequence of doing this however, is accidentally reading information which uncovers important parts of the plot – popularly known as "spoilers". Spoilers can ruin the excitement and surprise of a movie and make it less enjoyable for audiences. This in turn can also result in a lesser number of people watching

the movie, which of course can hurt the revenues of movie studios. Given the effects of spoilers, on both viewers and the entertainment industry, it is beneficial to reduce the number of spoilers that are able to reach audiences.

To reduce spoilers, certain review websites allow reviewers to tag their reviews for spoilers, unfortunately, few reviewers utilize this feature.

Existing researches on spoiler detection were practiced on datasets where the spoiler is tagged. The tag defines which line in the review or comment contains the spoiler. Another approach taken is to extract features like 'movie spoiler ratio' or 'user spoiler ratio' to train the neural network learn from the past. But this method may not be effective when the movie or the reviewer is new, since no past data regarding these features is available to train the model. Yet another method used for detecting spoilers is to define a dictionary of 'spoiler words' based on the genre of the movies. For example, if it is a thriller or a murder mystery, then the word 'death' or words defining relationships like 'father' or 'husband' are considered spoilers. This raises some problems as every reviewer is unique and would write differently, making it hard for rules to be created for every possible word that could be used. Additionally, a very precise domain of knowledge regarding movies is required to apply this method across all kinds of scenarios.

## DATA DESCRIPTION

The dataset is extracted from https://www.kaggle.com/rmisra/imdb-spoiler-dataset, and comprises of two parts: movie details and user reviews.

A sample of the Movie Details dataset is shown in Figure 1. The movie_id variable is unique for each row. Plot summary is the summary of the movie which does not contain spoilers. Duration is the runtime of the movie. Plot synopsis is the movie's plot and contains spoilers.

| duration | genre | movie_id | plot_summary | plot_synopsis | rating | release_date |
|---|---|---|---|---|---|---|
| 1h 53min | [Sci-Fi, Thr | tt0289879 | Evan Treborn grows | In the year 1998, E | 7.7 | 1/23/04 |
| 1h 41min | [Drama] | tt1723811 | Brandon is a 30-som | Brandon (Michael I | 7.2 | 1/13/12 |
| 1h 46min | [Action, Dra | tt5013056 | Evacuation of Allied | The film alternates | 8.1 | 7/21/17 |
| 1h 33min | [Comedy, D | tt0104014/ | For a while now, beautiful 24-year-old D | | 5.3 | 2/21/92 |
| 1h 32min | [Drama, Th | tt0114142/ | The marriage of David Burgess, a senio | | 4 | 1/29/99 |

Figure 1: Snapshot of dataset – Movie Details

Likewise, the User Reviews dataset is previewed in Figure 2. Each movie_id has multiple rows of reviews. Similarly, each user_id has multiple rows of reviews for multiple movies. The review_text column contains the spoilers. The is_spoiler column is already present in the dataset which is a binary indicator whether the review is a spoiler or not.

| is_spoiler | movie_id | rating | review_date | review_summary | review_text | user_id |
|---|---|---|---|---|---|---|
| FALSE | tt0362227 | 8 | 10/4/04 | best airport movie | I usually avoid see | 19296 |
| FALSE | tt0116209 | 1 | 10/30/04 | Too much film, too lit | If I had to choose c | 150359 |
| TRUE | tt1136608 | 5 | 2/25/10 | I don't know..... | I'm still on the fenc | 69459 |
| FALSE | tt1259571 | 3 | 11/21/09 | A huge drop in qualit | I don't understand | 240380 |
| FALSE | tt1454468 | 8 | 10/21/13 | Pray that you never ( | I've been writing re | 186772 |

Figure 2: Snapshot of dataset – Reviews

## METHODOLOGY

The movie_id column, common to both datasets, is used to merge the two datasets into one. After merging, the dataset has multiple instances of each movie_id and other selected variables (shown in Figure 3).

In the merged dataset, 70% of the reviews don't contain any spoilers, while the other 30% did. Few movies have below one hundred reviews, while some have more than a thousand. To combat this bias, a stratified sampling method is used. Movies with more than 300 reviews are filtered into the dataset. Once the final dataset is prepared, there are 839 unique movies with over 95,000 reviews. Plot and review summaries are removed as they did not contain spoilers and do not contribute in identifying them.

| movie_id | duration | genre/0 | rating_x | release_date | plot_synopsis | is_spoiler | rating_y | review_date | review_text | user_id |
|---|---|---|---|---|---|---|---|---|---|---|
| tt0031381 | 3h 58min | Drama | 8.2 | 1/17/40 | y at his Twelve Oa | 0 | 10 | 2-Feb-17 | nded.#KiduMovie | ur34756529 |
| tt0032138 | 1h 42min | Adventure | 8.1 | 8/25/39 | able to reach her | 0 | 10 | 27-Feb-08 | ill also! Have fun! | ur18604480 |
| tt0032138 | 1h 42min | Adventure | 8.1 | 8/25/39 | able to reach her | 0 | 6 | 21-May-99 | ink you hear me k | ur0264325 |
| tt0031381 | 3h 58min | Drama | 8.2 | 1/17/40 | y at his Twelve Oa | 0 | 10 | 25-Nov-00 | ruer than us now. | ur0943394 |
| tt0017136 | 2h 33min | Drama | 8.3 | 3/13/27 | the year 2026, in | 0 | 2 | 8-May-11 | ay, it's pretty bad. | ur1408548 |
| tt0032138 | 1h 42min | Adventure | 8.1 | 8/25/39 | able to reach her | 1 | 10 | 1-Jul-06 | colors of Oz oppo | ur9752397 |
| tt0031381 | 3h 58min | Drama | 8.2 | 1/17/40 | y at his Twelve Oa | 0 | 10 | 13-Aug-06 | o match this film. | ur2245791 |
| tt0017136 | 2h 33min | Drama | 8.3 | 3/13/27 | the year 2026, in | 0 | 10 | 11-Mar-06 | be disappointed. | ur4186388 |
| tt0031381 | 3h 58min | Drama | 8.2 | 1/17/40 | y at his Twelve Oa | 0 | 10 | 21-Jul-02 | charm of the film. | ur1807320 |
| tt0031381 | 3h 58min | Drama | 8.2 | 1/17/40 | y at his Twelve Oa | 0 | 10 | 16-Dec-06 | ick by his brother | ur8206701 |

Figure 3: Final dataset used for modeling and analysis

A crucial step in a Natural Language Processing (NLP) project is text cleaning and preprocessing. This involves keeping only relevant words by removing the stop words from the data based on Python's English dictionary. In addition to these, punctuation marks, special characters, and extra blanks are removed. Next, text is converted to lower-case. Stemming is applied to get the root word which eliminates the worry regarding tense. Snowball Stemmer is used for this purpose. Lastly, a split function is used to split words with a single space to avoid discrepancies in reading the text columns. These cleaning steps are applied on both the plot synopsis and review text columns.

## EXPLORATORY ANALYSIS

Initial exploratory analysis is performed to understand the influence of the input variables on the target variable: is_spoiler. Some of the results worth noting are following:

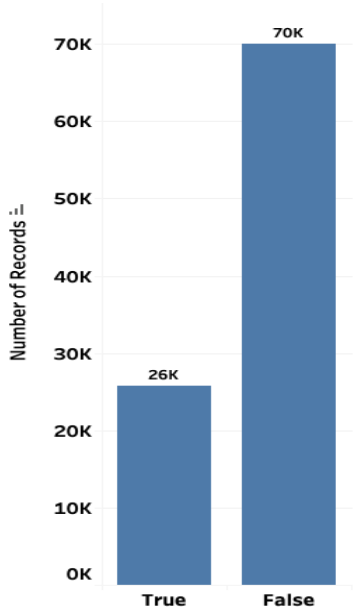


Fig 4: Distribution of spoilers and non-spoilers' reviews in the dataset

The dataset has 95k+ rows which are divided into 70k non-spoiler and 26k spoiler reviews.

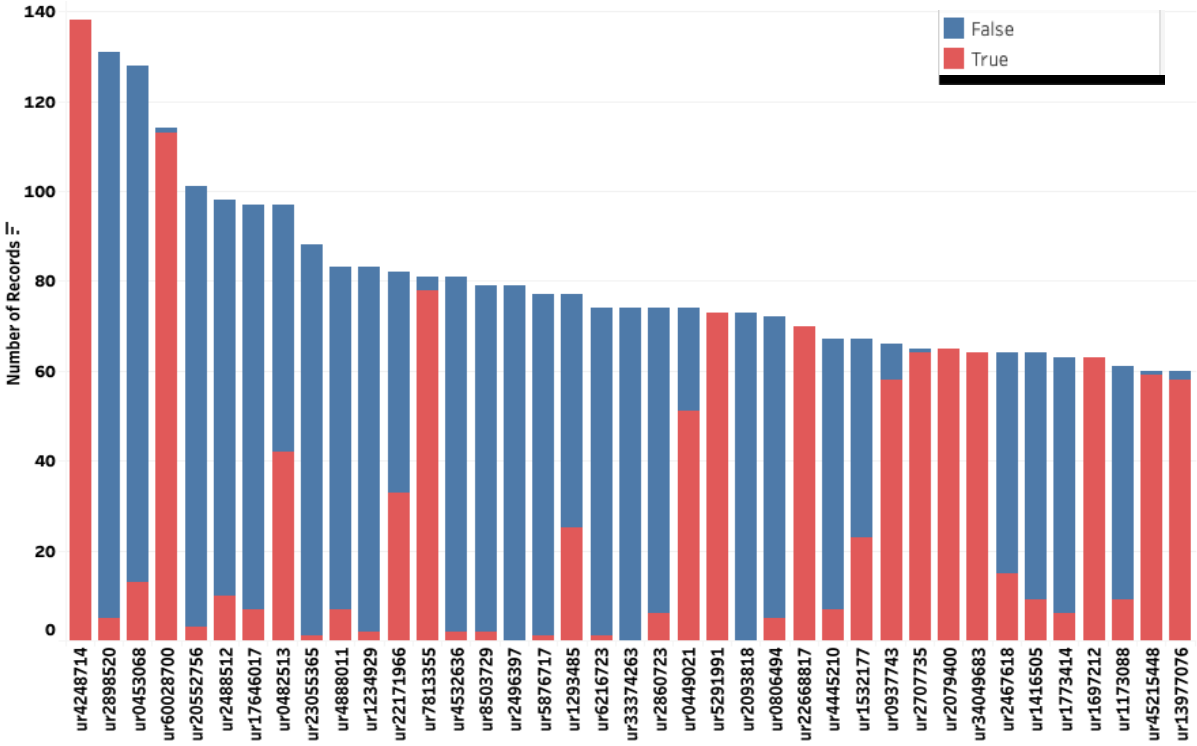


Figure 5: Reviewers who posted highest number of reviews

Figure 5 depicts that there are several reviewers such as 'ur4248714' and 'ur60028700' who are serial movie spoilers. The red color denotes reviews with spoilers, while the blue represents reviews with no spoilers. It is interesting to note that the reviewer with the highest number of reviews in the dataset, has spoiled movies in all his reviews.

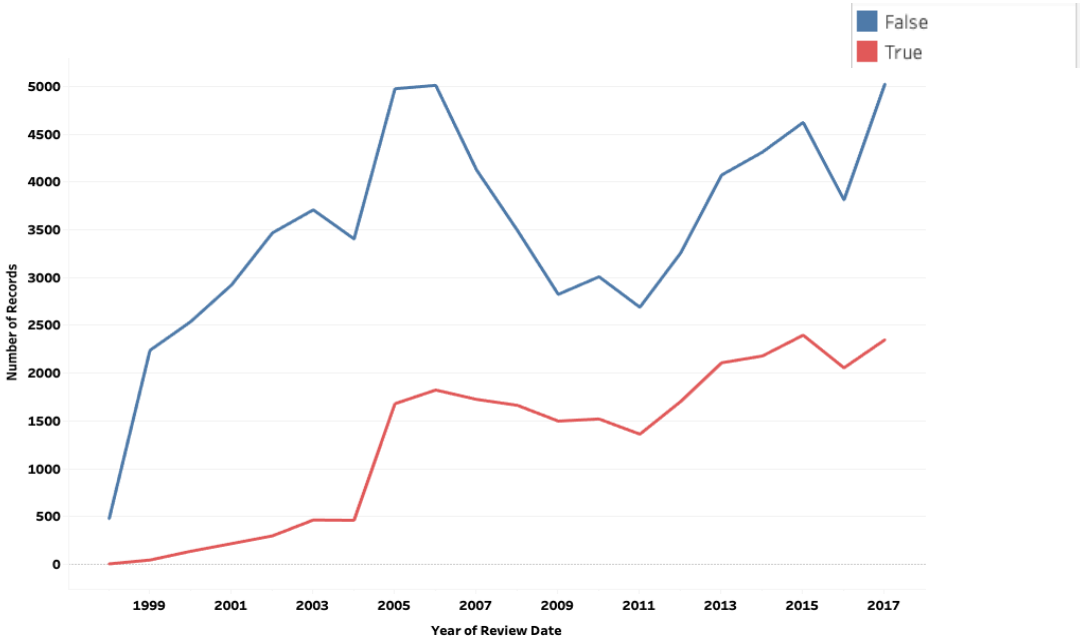


Figure 6: Increasing trend of spoiler reviews over time

Figure 6 shows increasing trend in the number if reviews that contain spoilers over the years. This re-iterates the increasing need for spoiler detection in future movie reviews.

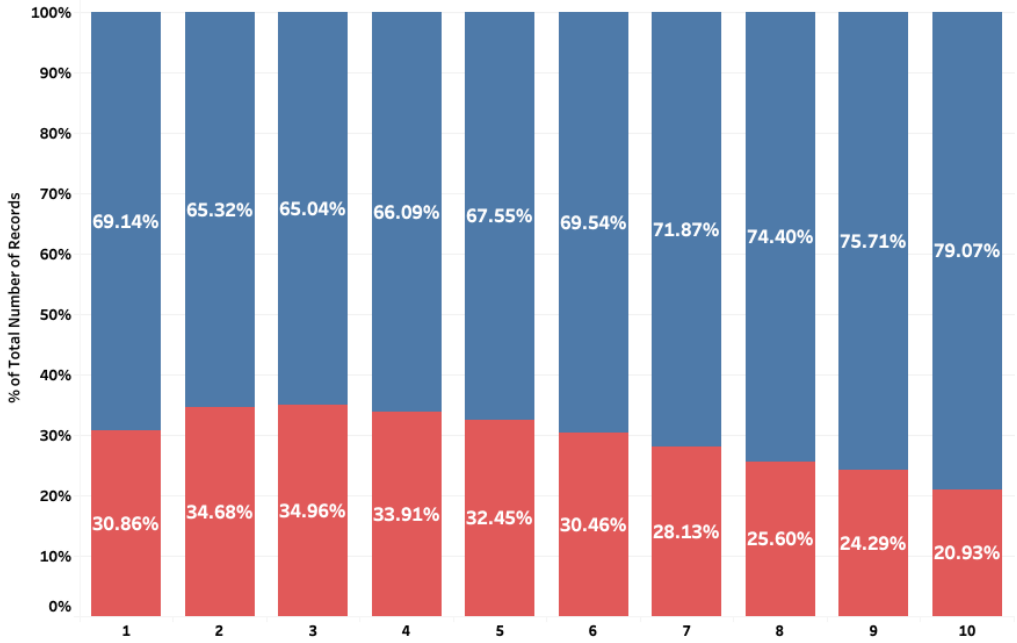


Figure 7: Gradual decrease in % of spoiler reviews as user rating increases

From Figure 7, it is safe to say that there is a gradual decrease in percentage of spoilers to non-spoilers in the reviews as the user rating increases from 3 to 10. This might imply that when people like a movie, they are less likely to mention spoilers.

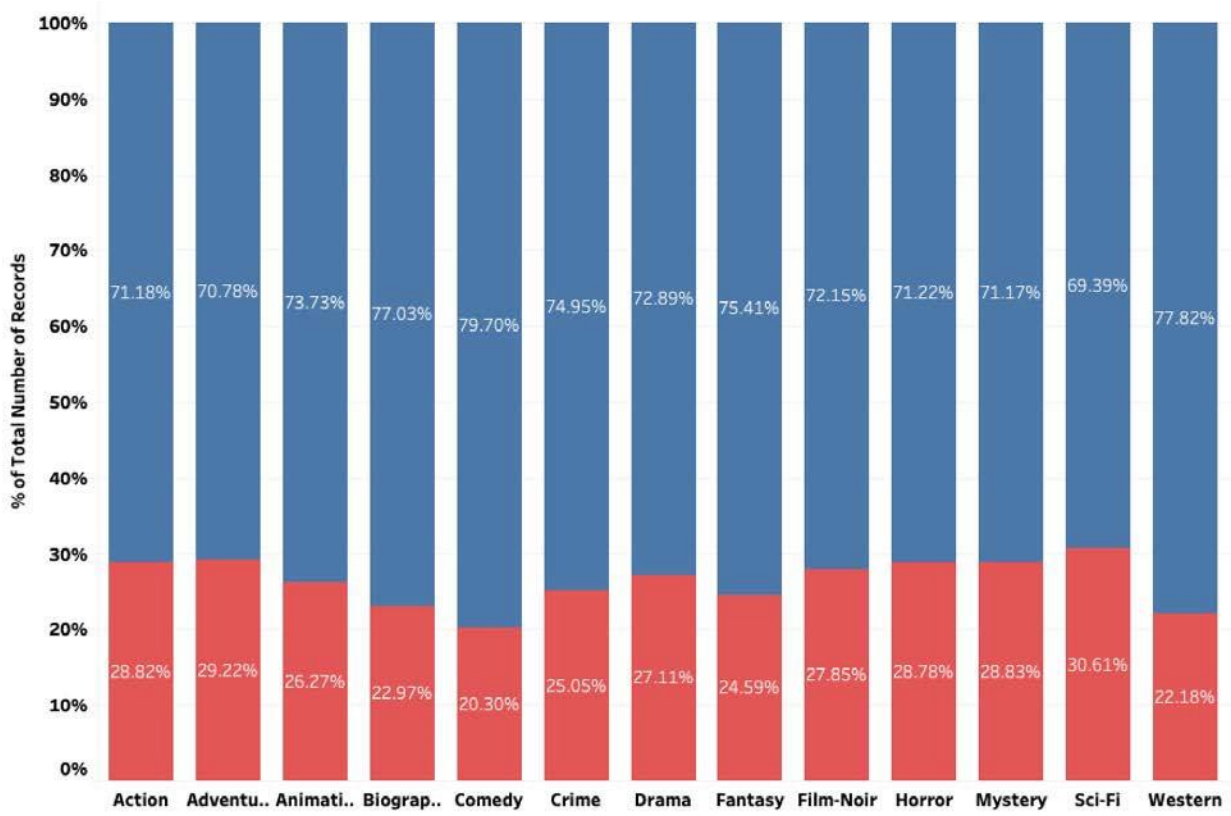


Figure 8: Spoilers in reviews with respect to movie genre

We can see that the highest spoiler ratio is in the genre of Sci-Fi, followed by adventure, mystery, action and horror.

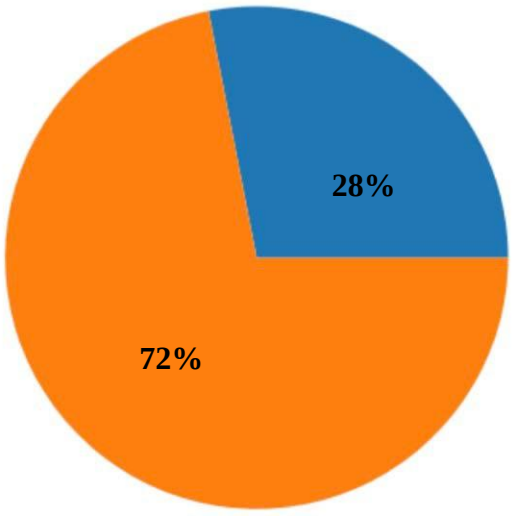


Fig 9: Spoiler reviews which contain the word 'spoiler'

Figure 9 represents only the spoiler reviews with the orange section indicating the percentage of spoiler reviews which actually have the word 'spoiler' in it. As mentioned in the introduction, it is a good practice to tag a review as a spoiler if it contains one, but the chart proves that not all reviewers practice this courtesy.

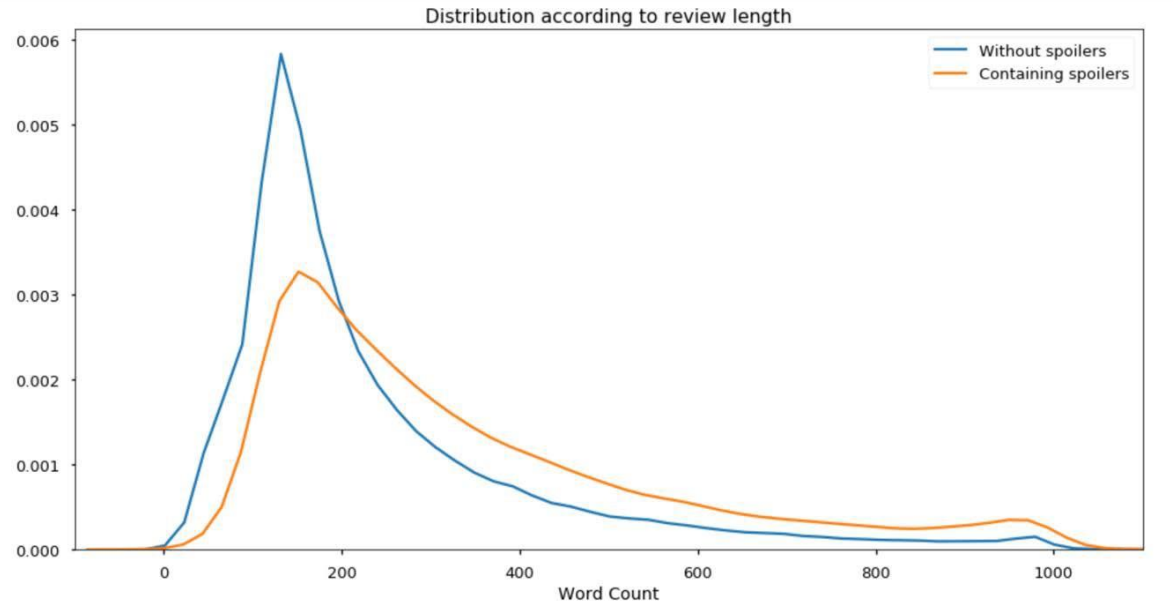


Figure 10: Word count for reviews

We can see that the number of words in a review are usually higher for spoiler reviews when compared to non-spoiler reviews.

## MODEL

The approach this paper attempts is to consider the movie plot synopsis and the review text to classify a review as a spoiler or not. Once the text preprocessing is done, the approach is to score the similarity between the plot and the reviews.

To maintain the sequence and essence of the plots and reviews, n-grams method is used to break them down into words in a continuous sequence. Words are split while cleaning and pre-processing the data, n-grams keeps the words in a sequence of 5 words in a bucket for this paper.

```
[('film', 'set', 'year', 'extraordinari', 'gothic'),
 ('set', 'year', 'extraordinari', 'gothic', 'skyscrap'),
 ('year', 'extraordinari', 'gothic', 'skyscrap', 'corpor'),
 ('extraordinari', 'gothic', 'skyscrap', 'corpor', 'citi'),
 ('gothic', 'skyscrap', 'corpor', 'citi', 'state')]
```

Figure 11: First 5 lines of a movie to portray how n-grams divided the sentences

From previous research work, it is proved that the movie climax/twist is usually mentioned near the end of the plot. Using this criterion, a variable 'position' is created using the index number allotted by Python. The higher the index number, the closer the word is to the end of the plot. From domain knowledge and further reading, another assumption is made: in a movie plot, a twist or a spoiler is mentioned only once. For example, if it is mentioned that Iron Man dies in Avengers: End Game, this line is mentioned only once in the movie plot. Based on the frequency of a word appearing in the plot, the 'frequency' variable is created. Less frequent words are given more importance than the words repeated numerous times.

Using these two variables, each movie plot is scored. Once the score is assigned, the top 20 words with the highest scores from each movie plot are chosen. Higher scores indicate higher position and lower frequency. Similarly, each review is scored. But as mentioned before, reviews are unique to a reviewer. Reviewers can mention spoilers anywhere in the review and they can repeat words as well, since there is no required format to a review. And therefore, 'position' and 'frequency' are not used to score the reviews. The reviews are scored based on matching words, i.e., how many words in the review matched with the top 20 words extracted from the movie plot. Finally, a dataset with the movie_id, user_id, a binary variable to identify whether it is a spoiler or not and a score for each review is created.

```
[(('result', 'climact', 'scene', 'joh', 'fredersen'), 8.63791196249539),
 (('joh', 'fredersen', 'watch', 'terror', 'son'), 8.667322177936192),
 (('climact', 'scene', 'joh', 'fredersen', 'watch'), 8.667497761311958),
 (('maria', 'freder', 'return', 'street', 'freder'), 8.690007550085157),
 (('scene', 'joh', 'fredersen', 'watch', 'terror'), 8.697083560128526),
 (('fredersen', 'watch', 'terror', 'son', 'struggl'), 8.726581567257213),
 (('terror', 'son', 'struggl', 'rotwang', 'cathedr'), 8.773883728688569),
 (('watch', 'terror', 'son', 'struggl', 'rotwang'), 8.77397152037645),
 (('son', 'struggl', 'rotwang', 'cathedr', 'roof'), 8.803469527505136),
 (('street', 'freder', 'unit', 'fredersen', 'grot'), 8.885589872350884),
 (('freder', 'unit', 'fredersen', 'grot', 'fulfil'), 8.915175671167454),
 (('return', 'street', 'freder', 'unit', 'fredersen'), 8.91535125454322),
 (('struggl', 'rotwang', 'cathedr', 'roof', 'rotwang'), 8.916141379734166),
 (('rotwang', 'cathedr', 'roof', 'rotwang', 'fall'), 8.945727178550735),
 (('cathedr', 'roof', 'rotwang', 'fall', 'death'), 8.951574104963742),
 (('freder', 'return', 'street', 'freder', 'unit'), 9.14689305216582),
 (('unit', 'fredersen', 'grot', 'fulfil', 'role'), 9.17028075781785),
 (('fredersen', 'grot', 'fulfil', 'role', 'mediat'), 9.199866556634419),
 (('grot', 'fulfil', 'role', 'mediat', 'copi'), 9.466841079486594),
 (('fulfil', 'role', 'mediat', 'copi', 'wikipedia'), 9.555774059312064)]
```

Figure 12: Example of a movie scores based on frequency and position

The next approach is to use a network map (Figure 13), which visualizes the relationship of words to each other within the text. Each node/vertex in this graph is a word and the lines connecting a pair of nodes (words) denotes that they occur together or one after another in

a sentence. Using the network mapping in 2D space, two other variables are calculated − centrality score between words and page rank score, which takes into account the Term Frequency-Inverse Document Frequency (tf-idf) structure of a document. The centrality of a word in a word network indicates the degree to which the word is related to other words in the document, on an average. The page rank score tells which are the most and least frequent words within each review. So, for each review, two more columns are added to the dataset.

From each review, the top 10 words with the highest and lowest page rank score are chosen, which also accounts for the centrality score. This means that the words with the highest page rank along with the words they are most related to were chosen in the top 10 words for each review. Since 10 words are chosen and each given their own column, our dataset now has 20 new columns − 10 for the highest page rank score and 10 for the lowest page rank score for each review.

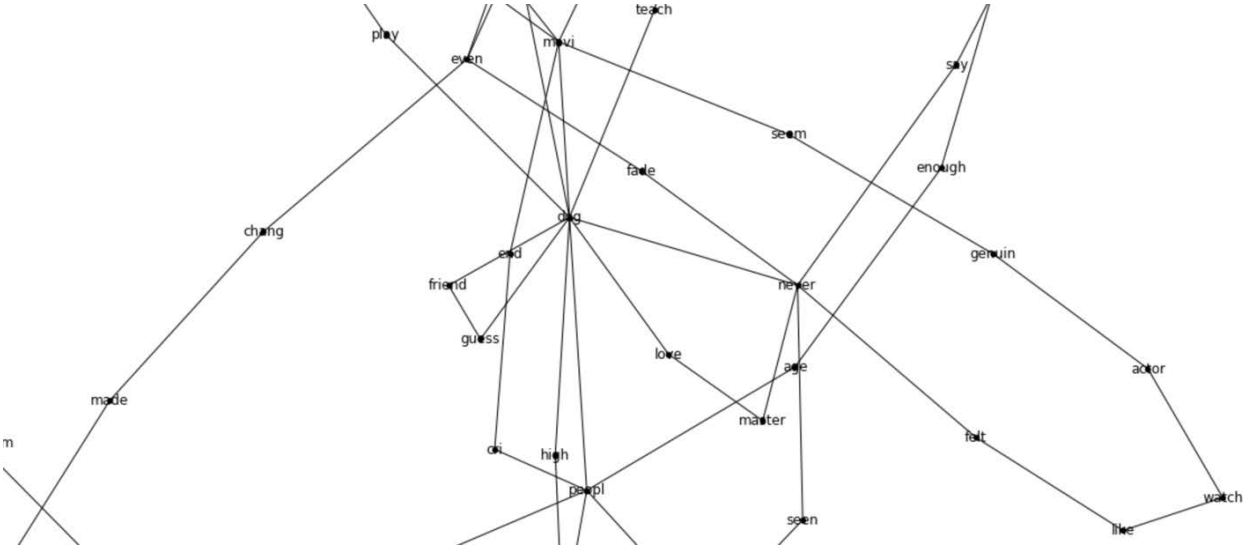


Figure 13: Example of a network graph being built for each movie and review

The target variable 'is_spoiler' is assigned either 0 or a 1 for modeling purposes. Since the dataset has over 96k rows and with addition of the new features for modeling leading to more than 25 columns, scoring is done separately for movie plots and reviews.

Using these top 10 and bottom 10 columns from page rank score and centrality score, plots and reviews are compared for word match scores. That is out of these 20 words how many words are common between the two columns (movie-review pair) which led to the creation of a new column called − word_match_score. Once the position, frequency, word matches, centrality score and page rank score features are extracted, the dataset was ready for

modeling. This dataset still consists of 70% spoiler and 30% non-spoilers. For our modeling purpose, a balanced dataset is created with 50% spoilers and 50% non-spoilers for training data. Sampling is applied only on the training dataset keeping in mind that in real life the dataset won't exhibit a 50-50 ratio. Once the balanced training dataset is created, K-Nearest-Neighbors Classifier, MLPClassifier and Random Forest Classifier are applied to build the model. Next, the model is trained using the training dataset. Once the model is built, it is fitted on testing data.

## RESULTS

Since the objective of this paper is to detect spoilers, a sensitivity metric is used to measure the reliability of the model. Sensitivity is calculated by finding the ratio of the True Positives to the total number of the (True Positives + False Negatives). Using the f1 score as the metric to choose the best model, the Random Forest Classifier with 'Gini' criterion emerged as the champion model with a f1 score of 0.69. The sensitivity of the model is pegged at 85.6% and the specificity (True Negatives / (True Negatives + False Positives)) at 72.1%.

## CONCLUSION

The exploratory analysis proved the need to detect movie spoilers in future. Using NLP techniques combined with domain knowledge, this paper was able to detect spoilers with ~86% accuracy. Features like 'position' of the word, 'frequency' of words, similarity score between a movie plot and a review were used in the model to avoid the use of any past movie or reviewer data. With the help of this model, the entertainment industry can detect if a review contains a spoiler or not. Either the reviewer can be prompted that he/she is mentioning a spoiler and therefore, it must be tagged as a 'spoiler', or the reader of the review can be made aware of the fact that the review features a spoiler.

## FUTURE SCOPE

Word2Vec, an alternate approach revolves around the concept of word matches between the plot and the review. However, given the volume of the dataset and the memory required, the CPU is not able to process the dataset. The number of unique words in movie plot are over 78k. That is, a 78k+ dimension matrix is being created with values of 0/1 which is difficult to handle. If the computational resources are available, the Word2Vec approach might help to get a better f1 score for similar applications. Also, although this model performs well in detecting a spoiler, it is not very good with detecting a non-spoiler. With a specificity value of 72%, this model is misclassifying many non-spoilers. In subsequent research, the aim would be to increase the specificity of the model as well and to include Word2Vec method to check if they give better results for spoiler detection.

## REFERENCES

[1] Mengting Wan, Rishabh Misra, Ndapa Nakashole and Julain McAuley. 2013. "Spoiler Alert: Machine Learning Approaches to Detect Social Media Posts with Revelatory Information." Florence, Italy: Association for Computational Linguistics.

[2] Jordan Boyd-Graber, Kimberly Glasgow, and Jackie Sauter Zajac. 2019. "Fine-Grained Spoiler Detection from Large-Scale Review Corpora." *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2605–2610. Florence, Italy: Association for Computational Linguistics. Montreal, Quebec, Canada: ASIST

[3] IMDB Spoiler Dataset - https://www.kaggle.com/rmisra/imdb-spoiler-dataset

[4] Sungho Jeon, Sungchul Kim, and Hwanjo Yu. 2013. "Spoiler detection in TV program tweets.". South Korea: Department of Computer Science and Engineering, POSTECH

## ACKNOWLEDGMENTS

I would like to thank my friend, Shubhodeep Chakraborty, Graduate Student from RWTH Aachen University for his immense help and his time for guiding me throughout the research.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Sreejita Biswas
Master of Science in Business Analytics
Oklahoma State University
sbiswas@okstate.edu
https://www.linkedin.com/in/sreebiswas/

Goutam Chakraborty
SAS Professor of Marketing Analytics
Director of Master of Science in Business Analytics
Oklahoma State University
Goutam.chakraborty@okstate.edu