STATEMENT OF WORK (V2)

Arunpriya Gobinath || 100841238

Jugnu Sheffin || 100814025

Ramasubramanian Dharumaperumal || 100810343

Raj Dholakia || 100813041

Sourabh Potdar || 100844478

Product Manager: Marcos Bittencourt

TABLE OF CONTENTS

1. PROBLEM STATEMENT

Reviews and movies have gone hand in hand since the advent of multimedia. However, on the one hand the "good" reviews, do not reveal the plot of the literature being reviewed and on the other, there are several reviews on websites such as IMDb, TV Tropes, Twitter, and other social media platforms, etc. that tend to or completely reveal the main plot. A few people like to "spoil" the fun for others. Imagine reading a suspense thriller novel after knowing the end or the suspense. Would someone even care to read?

Most of the time, even when you are searching for a particular song or the artist's name of a name, you end up on a website that also contains movie reviews usually crowdsourced in the comments. It is necessary to identify these spoilers upfront when a website is opened and alert the user to decide whether he/she would like to continue visiting that website. The tricky part is that the spoilers could be written using different sentence formations and different words. And they could be embedded deep into a seemingly interesting, "good" movie review.

2. PROPOSED SOLUTION

The solution that we propose is to create a Google Chrome extension that can be easily downloaded and installed. This extension then implements ML algorithms to detect the possible presence of a spoiler. If a spoiler is found, it blurs the browser screen, alerts the user, and seeks the user's permission to proceed. We also plan to allow the user to choose the movie names whose spoilers he/she wants to avoid. The ML algorithm will be first taught to identify the spoilers based on the learning model taken from credible data sources (discussed in the next part).

3. DATA SOURCES

There are plenty of data sources for collecting the data on movie reviews. For this project, we focused on IMDb website and TV tropes for gathering reviews to detect spoilers. The data available is in different forms such as images, audio, and video. However, we centered our work solely on text data extracted from IMDb website and blocked movie list.

We are dealing with a **classification problem** (spoiler: YES/NO) solved through a **supervised learning** algorithm. Ideally, we want data that contains random text from different websites along with a label, which indicates whether the text extract contains spoilers for a particular movie. Therefore, we need labelled textual data. A good source of such texts would be movie reviews. Reviewers on IMDb label their review before posting.

Main Data Source: IMDb Movies (sourced from Kaggle)

If the product evolves to support TV shows and books, we will using TVtropes and Goodreads.

4. DATA ASSUMPTIONS, LIMITATIONS & CONSTRAINTS

4.1 DATA ASSUMPTIONS

1. Data has linear relationship only. Each review maps to only 1 movie.

2. Each of the training data points can be classified as either Spoiler or Non-Spoiler (+1,-1)

4.2 DATA LIMITATIONS & CONSTRAINTS

1. No multimedia processing – Spoiler Buster Algorithm will not process and filter any multimedia data, i.e., audio, video, images and gifs will not be processed for spoiler busting.

2. Language limitation – Reviews/Comments in other languages will not be processed.

3. Limited to movies – We will only be detecting and filtering spoiler comments pertaining to movies. TV shows, series, sports, and other entertainment shows will not be considered.

4. Limited to IMDb database – Movies with plots/entries in IMDb database will only be analyzed by the machine learning model.

5. Supports only Chrome browser – Spoiler Buster is a plugin which will only by available in chrome web store.

5. TEST PROCESS

There are two major parts to the Spoiler Block project: algorithm and extension. Effective testing techniques need to be used to evaluate the performance of the algorithm.

Testing the performance of the algorithm will define the expectations of the algorithm in the real-world.

5.1 DATA-SPLIT

The data will be split into the following sets.

NAME	PERCENTAGE OF DATA	USE
TRAIN	80%	This dataset will be used to train the algorithm.
VALIDATION	10%	This split will be used to test the performance of the algorithm.
TEST	10%	This split will be used for fine tuning the algorithm.

5.2 METRIC

Area under the ROC curve (AuC) will be used to measure and evaluate the performance of the algorithm accurately. AuC gives a good indication on the performance of a classification algorithm as a value of 0.5 defines random chance. Therefore, benchmarking the performance of the algorithm will be easier with the AuC metric.

5.3 EXPECTED AUC

Version 1 will form the **base model**, with an expected performance (AuC > 60%) that is better than random change (AuC = 50%).

Version 1	>60%
Version 2	>75%
Final	>90%

Finally, the performance of the algorithm will be manually tested by visiting the following websites:

- IMDb reviews

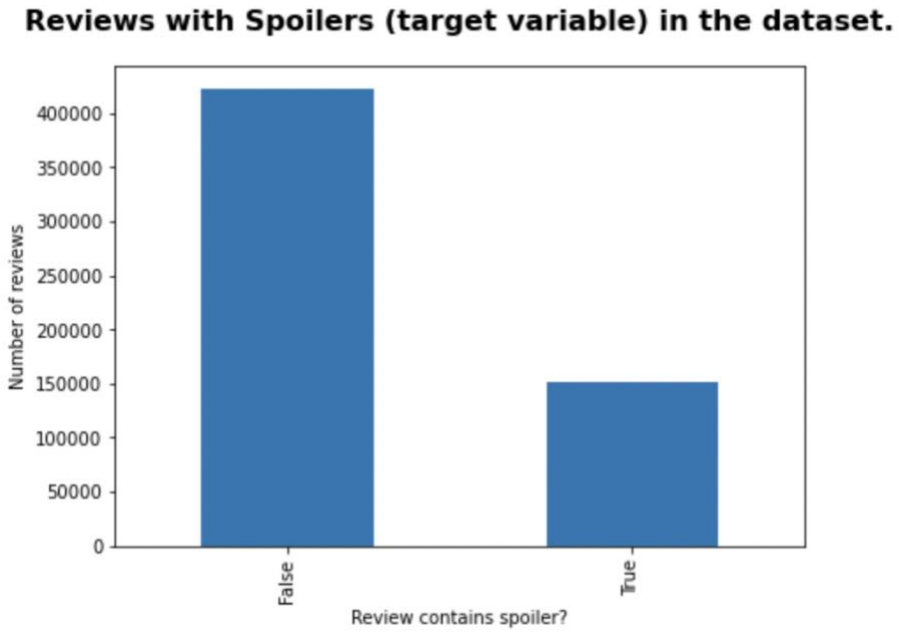
- Top 10 Google Search results for ‘Hulk’.
- Rotten Tomatoes

6. EXPLORATORY DATA ANALYSIS

*The following notebook contains the complete exploratory data analysis: “**1 - Exploratory Data Analysis**”.*

Some important points:

- There is a bias in the target variable, the dataset contains more samples of reviews that do not contain a spoiler, as shown in the figure below.



- There are 1,572 different movies reviewed by 263,407 users.
- A user has reviewed a maximum of 1303 movies, while ‘**Batman Begins**’ got the highest number of reviews at 4,845.

7. DATA PREPARATION

*The following notebook contains the complete data preparation: “**2 - Data Preparation**”.*

1. To begin with, we will only consider three (3) columns to train our base algorithm and dropped null values from these columns.

   is_spoiler : target variable (labels)

review_text : features

review_summary : features

2. To improve the speed of data processing, two datasets are created

   full_processed : contains processed data of all the samples in the dataset.

   sample_processed : contains 25,000 random processed reviews.

3. To clean text, punctuations, stop words, and any other characters that can be considered as noise in a text were removed. The words were then reduced into their stem from, example running to run. Also lowered the case of all characters in the text to maintain consistency.