# Goodreads Dataset: Exploratory Data Analysis and Spoiler Detection Model

Micol Clement, Rhea Sablani, Lanxue Zha, Xile Zhang

01

# Dataset Overview

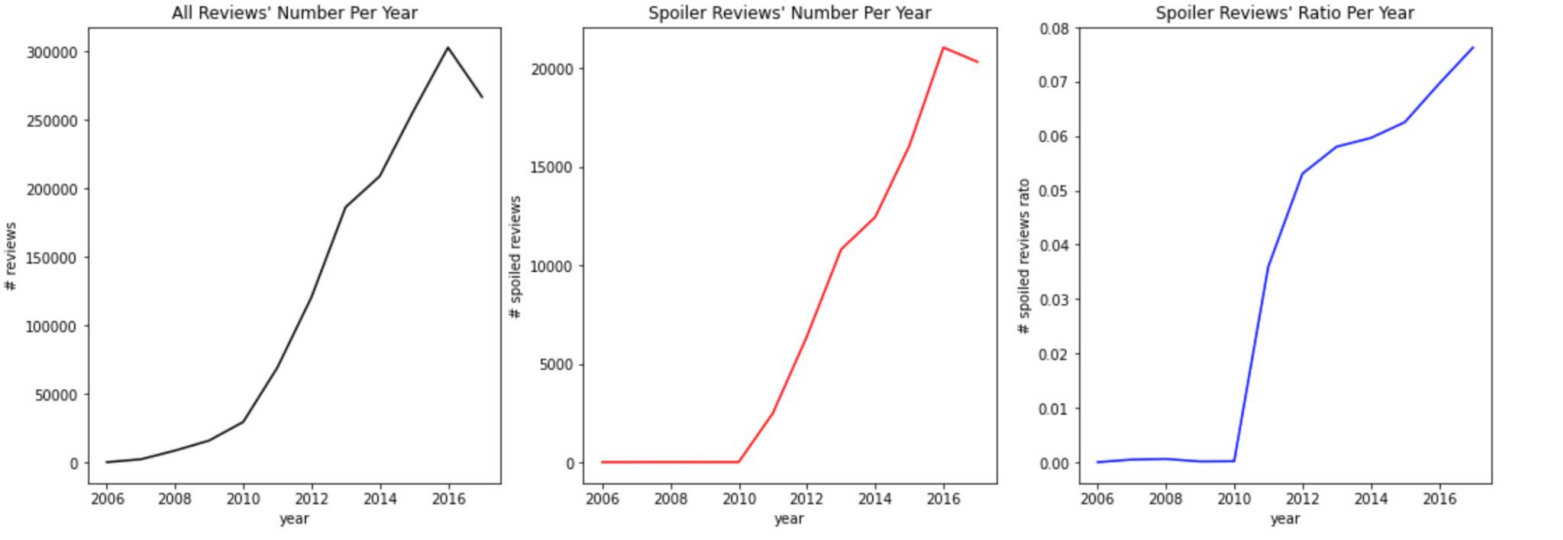
# Goodreads Dataset

- **Goodreads**
  - social cataloging website
  - users can sign up, register and rate books, write reviews
- Dataset  is extract of concentrated reviews in English
- Over 1.3M English book reviews across 25K books and 19K users where each book/user has at least one associated spoiler review
- Columns: review_id, review, book_id, true/false for spoiler, rating, timestamp, user_id

https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/reviews?authuser=0

# Data Cleaning

- JSON → CSV
- Renaming columns
  - Class: has spoiler = 1; no spoiler = 0
- Took subset of data because GoodReads didn't support spoiler tag until 2011



02

# Project Goals

# Objectives

- Explore if there are relationships between the different features and whether or not a review contained a spoiler about a book
- Create machine learning models for predicting whether or not a review contains a spoiler
- Evaluate models with the following metrics: train time, accuracy, precision, recall, F1 score, and AUC

# 03

# Exploratory Data Analysis

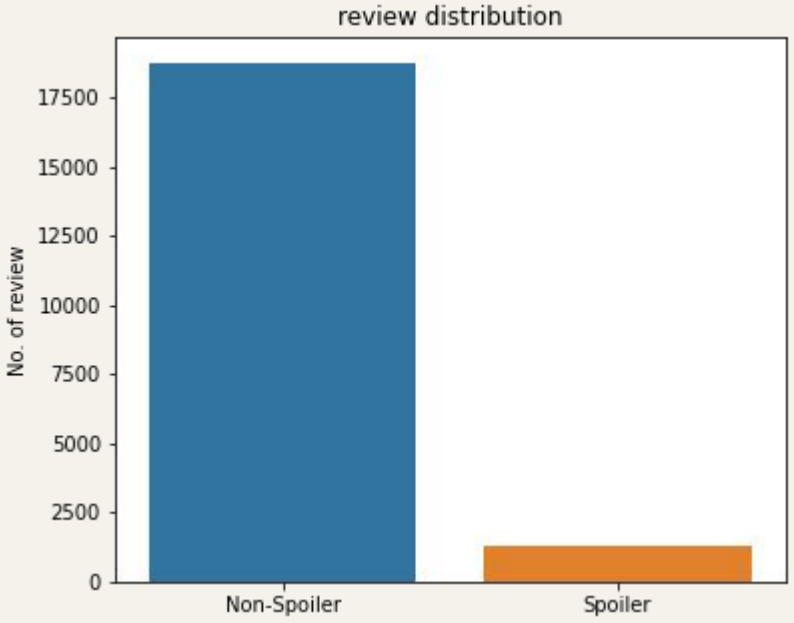
# Ex. Review & Review Distribution

- Spoiler review example

  “Come on? So Victor is the good guy and the hero and Eli is the bad villain? I'm sorry. I don't buy it. They both did bad things. While this does have a good build up for a sequel (fingers crossed) when Eli breaks out of jail, I don't think that was a good ending. Isn't the idea that good and evil are fluid? I was surprised the athame was connected to the thing that killed Cas's father. I was very surprised by that. I was also surprised by the whole last 50 pages or so, it was NOT what I expected”

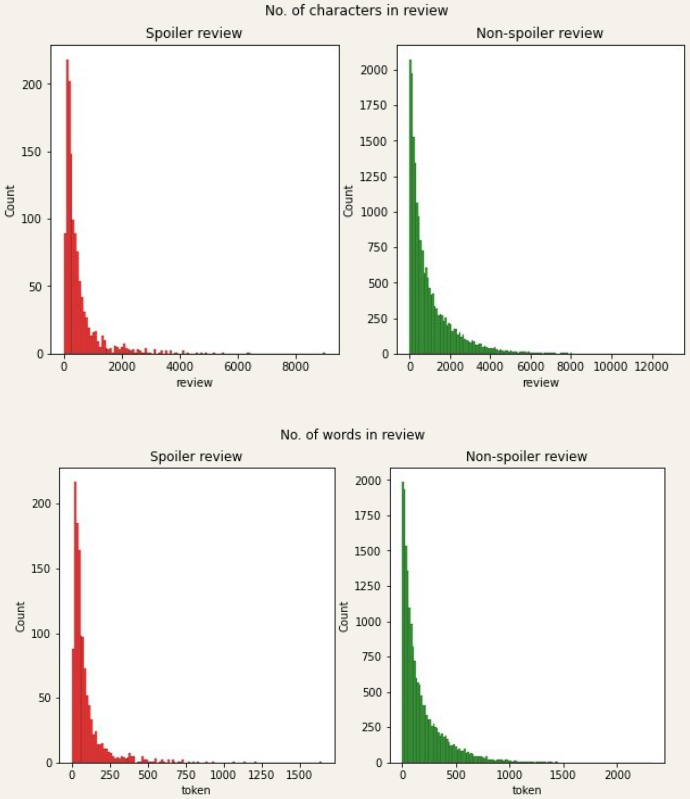
- Non-spoiler review example

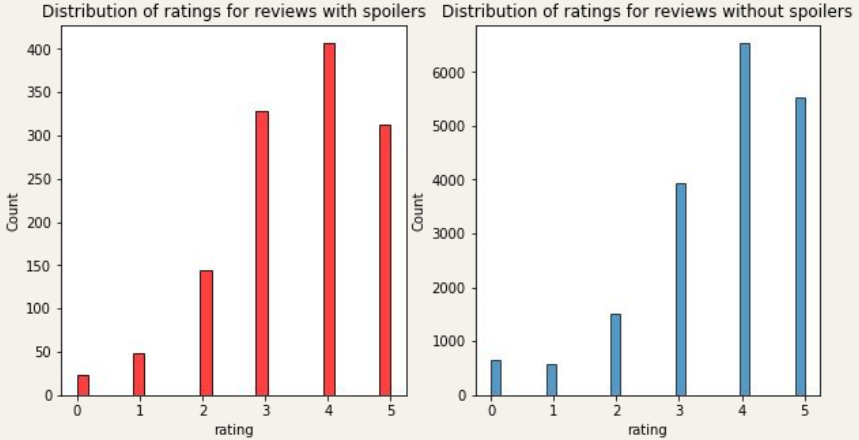
  This was an amusing read... romantic, spicy, and humorous. The battling wars between Evil and Carter were genius, had me chuckling all along. Wonderful love-hate romcom, with some steam added.




Imbalanced data
93.7% of dataset has non-spoiler reviews

# Spoiler vs Non-Spoiler Reviews




Minimal differences between spoiler reviews and non-spoiler reviews with regards to the distribution of # characters in review, # words in review, and rating given

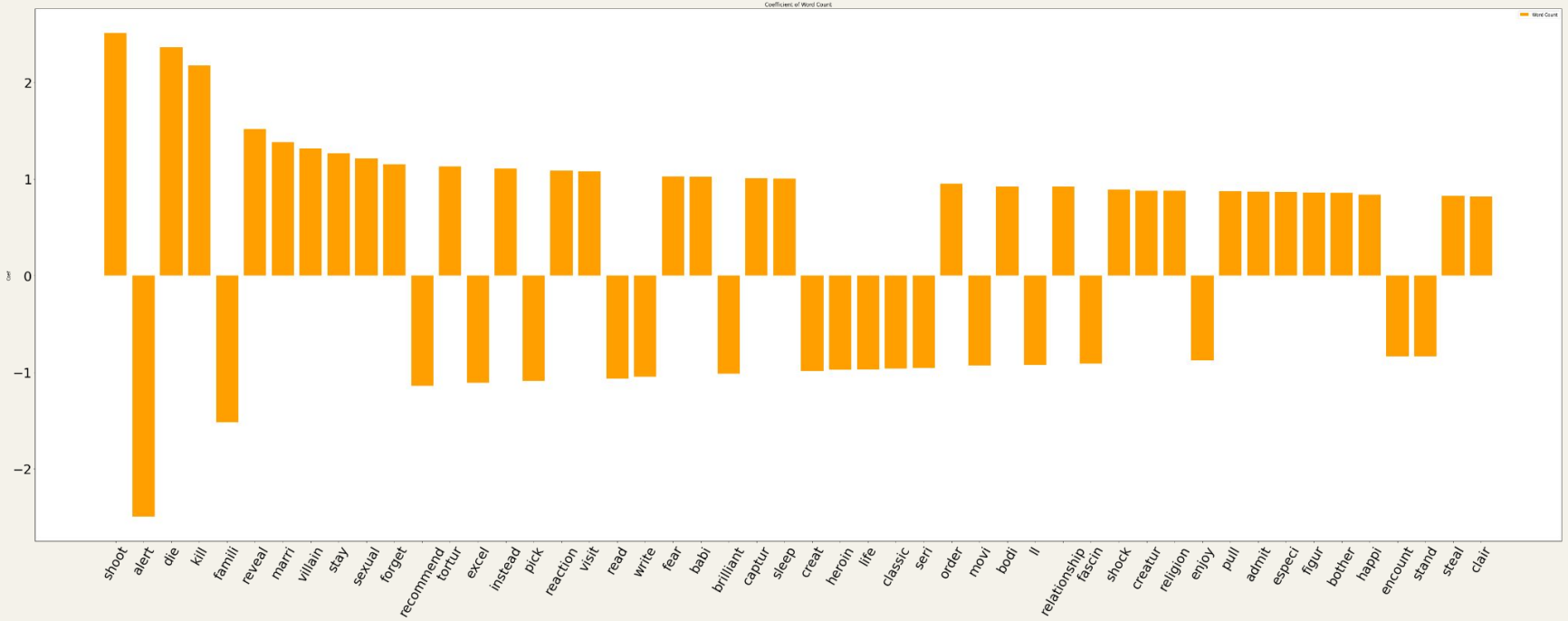
04

# Spoiler Detection Model

# Background & Preprocessing

- Common algorithms for spoiler detection include SVM and LSTM
- Goal: predict whether or not review is detected as spoiler
- Subset of data (year > 2011)
- Preprocessing
  - Removed stop words, numbers, non-alphabetic characters
  - Stem and lemmatize text
  - Encode data with bag of words

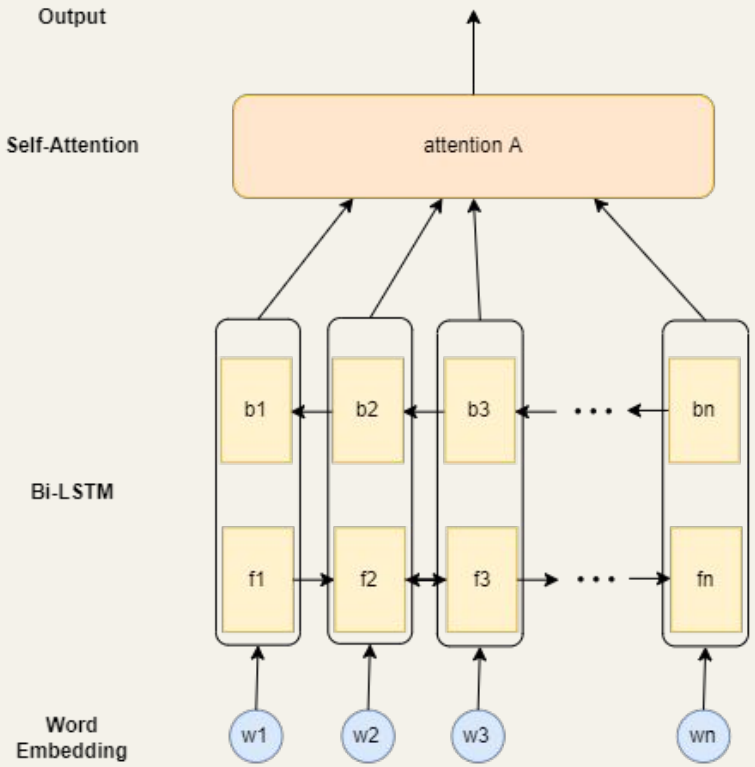
# What has SVM learnt from data




| | shoot | alert | die | kill | family |
|---|---|---|---|---|---|
| **coefficient** | 2.509725 | -2.495762 | 2.362985 | 2.173229 | -1.518742 |

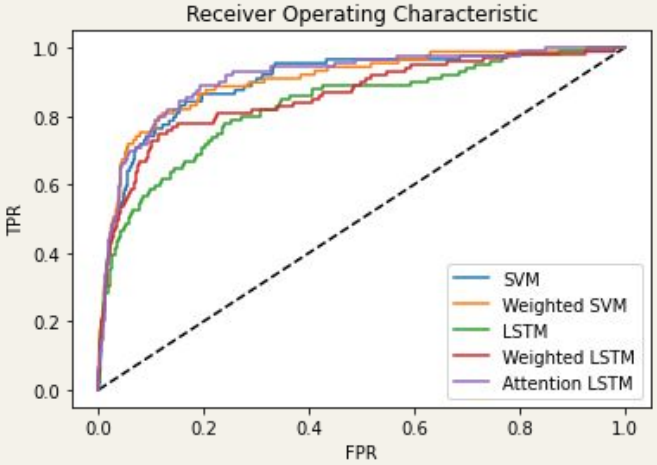
# LSTM-Attention Model :

- Encoding using the Tokenizer of tensorflow.
- Initial embedding weights from the pretrained GloVe embedding



# Results

| | Baseline SVM | Balanced SVM | LSTM | Balanced LSTM | Attention LSTM |
|---|---|---|---|---|---|
| **Train time (s)** | 36.4 | 84.7 | 52.53 | 9.4 | 41.62 |
| **Accuracy** | 0.955 | 0.88 | 0.958 | 0.907 | 0.887 |
| **Precision** | 0.4 | 0.241 | 0.941 | 0.296 | 0.259 |
| **Recall** | 0.0225 | 0.789 | 0.162 | 0.636 | 0.820 |
| **F1 score** | 0.0426 | 0.368 | 0.276 | 0.403 | 0.394 |
| **AUC** | 0.902 | 0.905 | 0.831 | 0.858 | 0.912 |



# Questions?