“Read Me Next”: Book Recommendations through User-Data Insights

Mohammad Hajjaj, Ahmad Diab
“Data Mining” Project

Motivation

Large body of reading material

Fast-paced life

What is the next "good" thing to read?

Goal

Develop a system,
'ReadMeNext',
able to predict the quality of a reading material and advice it's reading potential

Decisions are made based on user-generated data from GoodReads

Contributions

The problem of book recommendation was covered in literature

- From "**Author**" Perspective
- From "**Publisher**" Perspective
- From "**Book**" Perspective (i.e. *Title*, *Book Cover*)

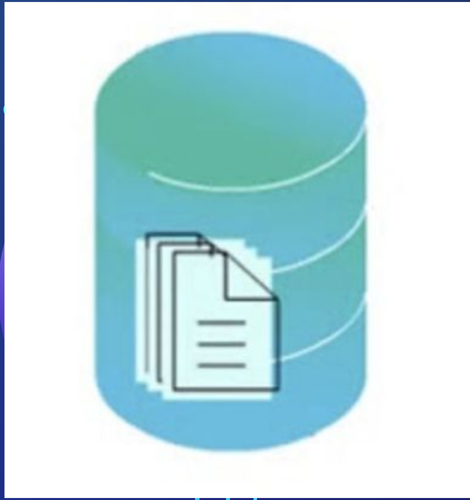
However, the user-generated data is an essential piece of information for individuals to make their decision of the next read
Was adapted in literature from a statistics point of view (i.e. number of reviews, number of comments)

Research Gap:
The use of ***Textual Content*** of reviews

Dataset

GoodReads website
Collected via **Kaggle**
Data Stats:

- 900`000 record
- Each representing a single review from a user, u_i, for a book, b_j
- Features include: user_id, book_id, review_rating, review_text, date_added, #votes, #comments



Data Transformation

Dataset as is does not help in answering our research question

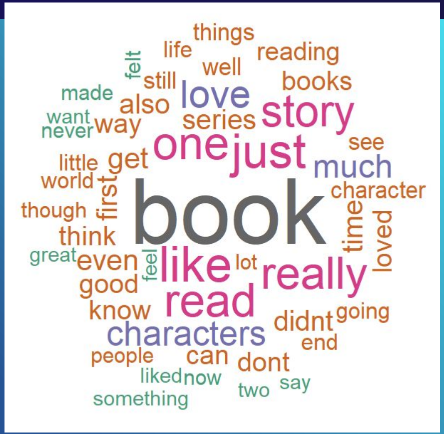
Need to transform based on unique books

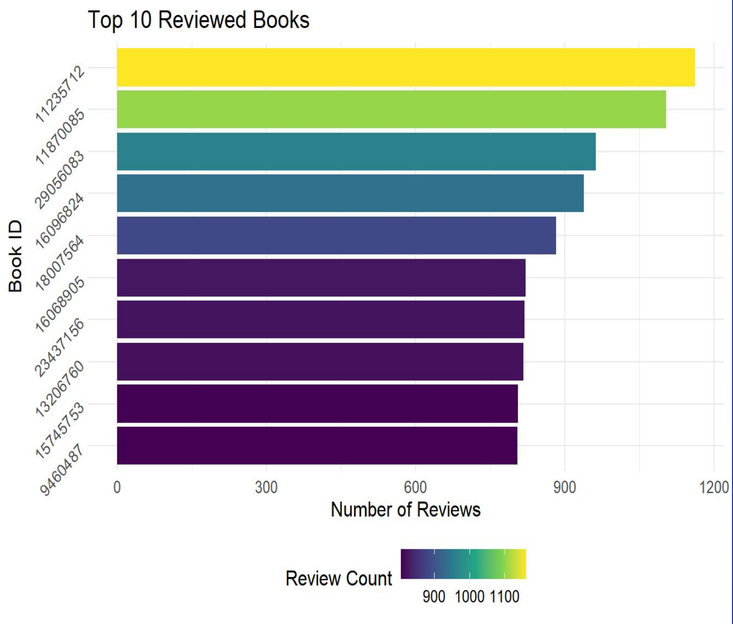
- 25k unique books
- Concatenate reviews for each book
- Average user-rating for each book
- Aggregate #comments, #votes, #reviews

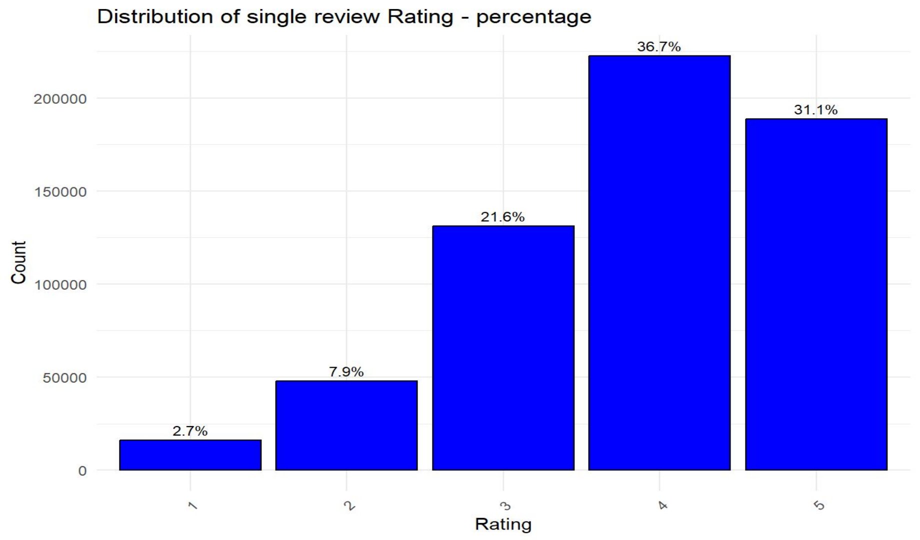
Data Pre-Processing

- Filtration:
  - Reviews with length <=10 characters of length (i.e. “ugh”, “what a book”)
  - Punctuations and special characters in each review
  - Book with <10 reviews
  - Reviews with less than 5 votes and 5 comments
- Text representation
  - RoBERTa large model
  - Each review is represented in 1024 vector
  - Reviews of a single book is averaged

Data Visualization



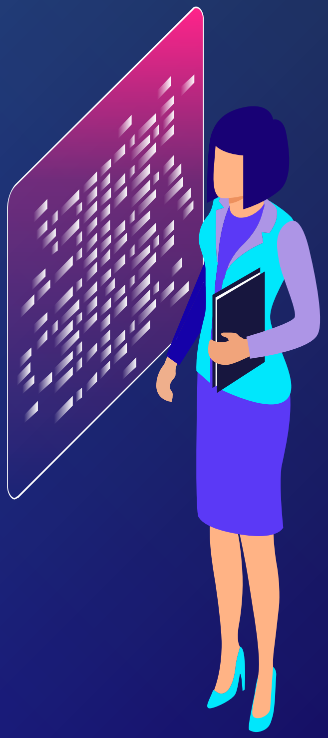




Challenges

What qualifies a book to be good?
Literature: Amazon best-sellers, private list of recommendations from GoodReads website
Our Approach: Review 10 lists of highly recommended books from GoodReads from different genres (100 books each, total of 1000 recommended books), manually inspect their rating, they have at least 3.7 user-rating

Decision: Consider 3.7 user-rating as a decision point that differentiate recommended books from others



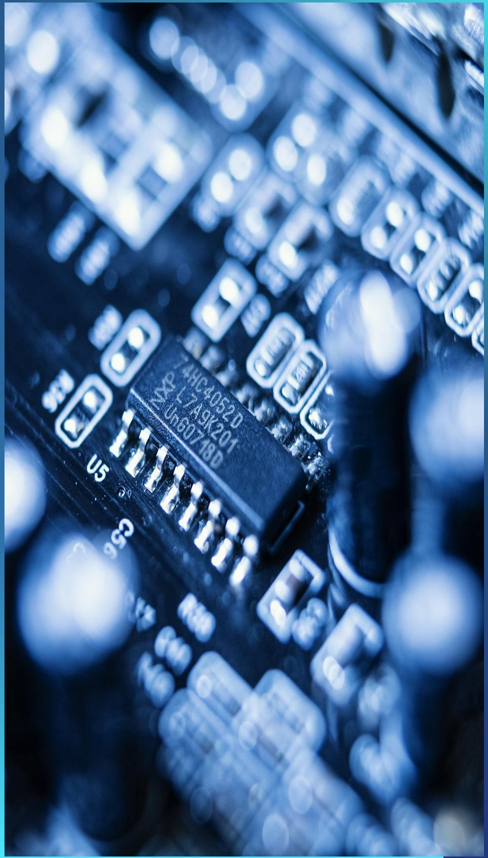
Feature Engineering

Goal: Recommendation based on user-generated data

Features:

1. Comments (2): total_number_of_comment, average_number_of_comments
2. Votes (2): total, average
3. Reviews (2): total_number_of_comment, average_number_of_comments
4. Text representation (1024): Average of all comments representation, generated from RoBERTa model

Total Features: **1030** (= 2 + 2 + 2 + 1024) features



Methods

Inspired by literature, we deployed several machine learning models, including:

- Naive Bayes
- Generalized Linear Model (glm)
- Support Vector Machine (SVM) - Radial Kernel
- Decision Tree (rpart)
- Random Forest
- K Nearest Neighbour (KNN)

Evaluation Metrics

To test our performance clearly, we used an array of common evaluation metrics, such as:

- **Accuracy**
- **Precision**
- **Recall**
- **F-1 score**
- **AUC**
- **ROC**

The training was performed using 5-fold Cross Validation

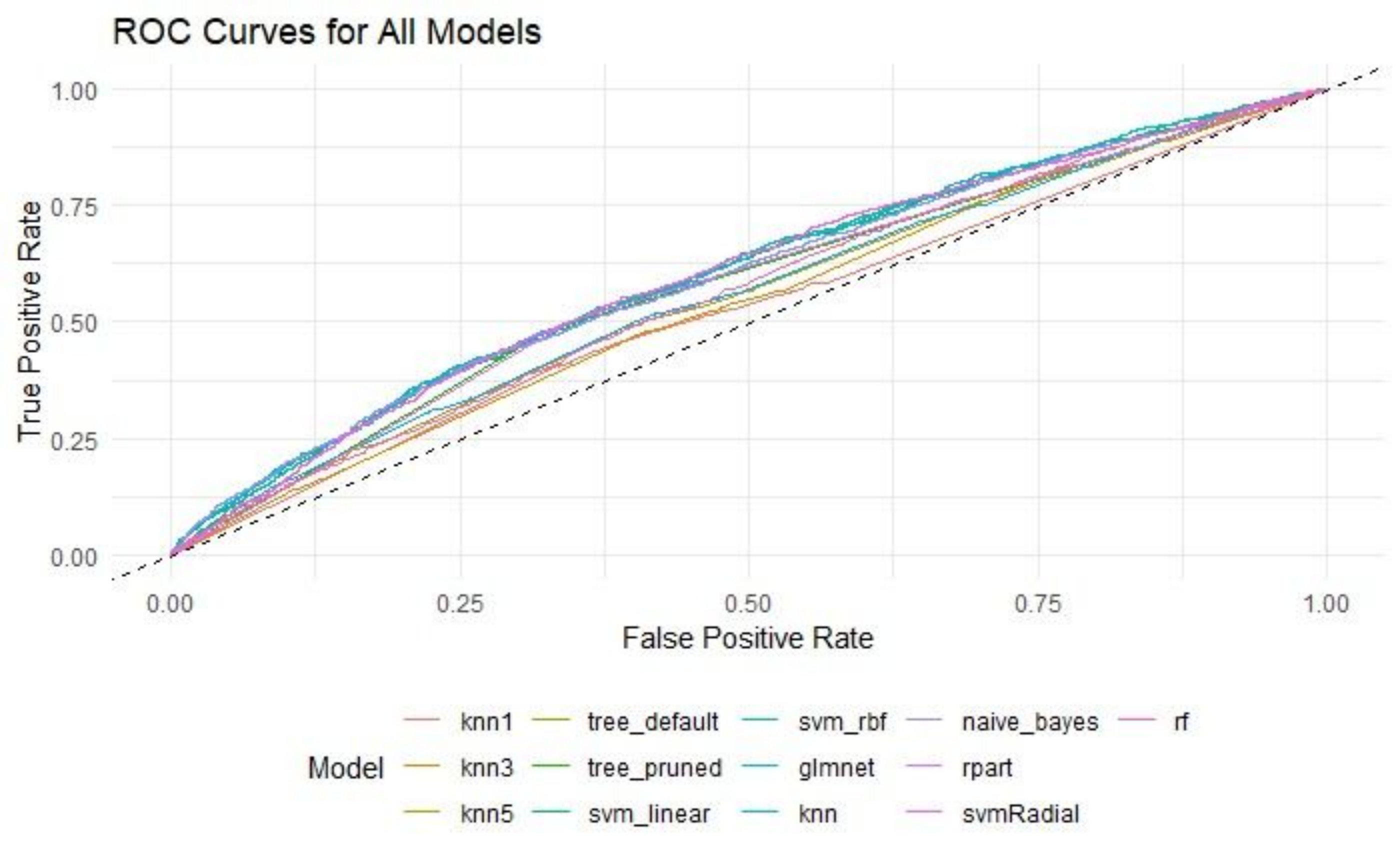
Results

	Accuracy	Precision	Recall	F-1
glm	0.761	0.7581	0.723	***0.795***
Naive Bayes	0.6959	0.751	0.753	0.752
SVM	0.7901	0.78	0.8	***0.81***
Decision Tree	0.701	0.791	0.547	0.647
Random Forest	0.6232	0.6	0.81	0.694
knn	0.5251	0.576	0.521	0.605

Results - Without Text Rep.

	Accuracy	Precision	Recall	F-1
glm	0.592	0.6	0.7576	**0.67**
Naive Bayes	0.5914	0.601	0.7522	**0.6683**
SVM	0.5837	0.6175	0.6288	0.623
Decision Tree	0.585	0.6132	0.6546	0.6332
Random Forest	0.55	0.5855	0.608	0.623
knn	0.5446	0.5855	0.608	0.5969

Models Variants - Smaller Data (20%)



Future Works

- Models optimizations
    - Due to time restrictions and time required for each run, our results are based on the default parameters of all models
    - We plan to run Grid Search-Style and study the difference in performance
    - Results will be included in the final report
- Feature Reduction
    - Text representation is dominant in our feature engineering (1024 vs 6)
    - Argument: Full text representation is needed to capture the semantics of the reviews
    - Counter-Argument: Not all (1024) features are equally important
    - Approach: Test PCA to reduce dimensionality and study its effect on performance
- Sentiment Analysis
    - Although it is captured by the semantic of the text
    - It is worth trying to include its influence on results

Future Works

- Models optimizations
  - Due to time restrictions and time required for each run, our results are based on the default parameters of all models
  - We plan to run Grid Search-Style and study the difference in performance
  - Results will be included in the final report
- Feature Reduction
  - Text representation is dominant in our feature engineering (1024 vs 6)
  - Argument: Full text representation is needed to capture the semantics of the reviews
  - Counter-Argument: Not all (1024) features are equally important
  - Approach: Test PCA to reduce dimensionality and study its effect on performance
- Sentiment Analysis
  - Although it is captured by the semantic of the text
  - It is worth trying to include its influence on results

Future-Future Work

- More literature review
    - The innovation is based on textual user-generated data (i.e. reviews)
    - Based on our literature review, no previous works have done it
    - The idea is worth publishing after more comprehensive literature review
- Results comparison to literature
    - The “good” book decision in our work differs from literature
    - Can we find a common ground that guarantees fairness in comparison across different works?

Takeaways

Old problem, different perspective

Combining more features (i.e. author and book info) can improve our results

Methodology is as important as final results

Thank you!