



# The Book Review App

Amit Shah  
Chaitali Kamble  
Jinesh Dhruv

# Motivation

- ▶ Regular Purchases at Amazon
- ▶ Reviews, the dominating reason for convincing someone to buy a product.
- ▶ Sentiment Analysis
- ▶ The potential of Naïve Bayes in Text Analysis
- ▶ 2016 Local Consumer Survey:
  - ▶ 84% people trust online reviews.
  - ▶ 90% consumers read less than 10 reviews before forming an opinion.
  - ▶ 54% people visit website after reading positive reviews.

Source: <https://www.brightlocal.com/learn/local-consumer-review-survey/>

# Problem Definition

- ▶ Reviews important for product and company survival.
- ▶ Inconsistency amongst user understanding about Rating reviews vs their impression about the product, so User rating cannot be the sole factor to determine positive/negative review.
- ▶ *Need a way to categorize a review as a positive/negative review based on the actual review contents.*
- ▶ Most users do not read more than 10 reviews before making an opinion.
- ▶ *Need a way to rank and display the most helpful reviews to the user.*

# About the dataset

- ▶ Raw Dataset:
  - ▶ 8,89,401 reviews (rows) and 9 attributes.
- ▶ Data cleaning and sampling:
  - ▶ Sampled data to reduce to 50,000 reviews, for the scope of this project.
  - ▶ Thresholding: at least 100 reviews for each book.
- ▶ Final dataset:
  - ▶ 35,148 reviews of 78 unique books, and 5 key attributes.
  - ▶ Created Ground Truth: Assigning review with 4 star or more as positive, others negative.

# Key Issues

- ▶ Naïve Bayes did not perform very well on raw reviews.
- ▶ Determining a review as top helpful.
- ▶ Android Application Development
- ▶ Front end design issues

# Solutions

- ▶ Naïve Bayes did not perform very well on raw reviews.
  - ▶ *Preprocessing reviews through Transformation, Punctuation & Stop word Removal, Tokenization, Stemming.*
- ▶ Determining a review as top helpful.
  - ▶ *Factors considered: Date, Review Text, Review Rating*
- ▶ Android Application Development
  - ▶ *New to Android, hosting client server mechanism.*
- ▶ Front end design issues
  - ▶ *Design decisions about the interface.*

# Our Book Review System

- ▶ Performed Sentiment analysis of Amazon book reviews and built a model to predict a new review as a positive review or a negative review.
- ▶ Additional functionality of a data-driven ranking algorithm to identify the top 10 reviews for a book.
- ▶ Two parts:
  1. Sentiment Analysis and Review Categorization:
    - ▶ Rank a review as positive/negative.
    - ▶ Naïve Bayes Algorithm
  2. Review Ranking:
    - ▶ Display only top 10 reviews to the user.
    - ▶ Custom ranking algorithm

# Algorithm - (1) Sentiment Analysis

- ▶ Objective: Be able to classify a new review as positive / negative.
- ▶ Naïve Bayes - very powerful for text analysis.
- ▶ Advantages:
  - ▶ Efficient and easy
  - ▶ Robust to irrelevant attributes
- ▶ Performed Naïve Bayes on raw data directly - poor results (67% Accuracy)
- ▶ Need of preprocessing.



# Algorithm - (1) Sentiment Analysis

## ► Preprocessing:

1. Transformation
2. Punctuation and Stop word Removal
3. Tokenization
4. Stemming using Porter's stemmer

## ► Spell check - Failed Attempt.

## ► Split data:

- Training data (70%)
- Testing data (30%)

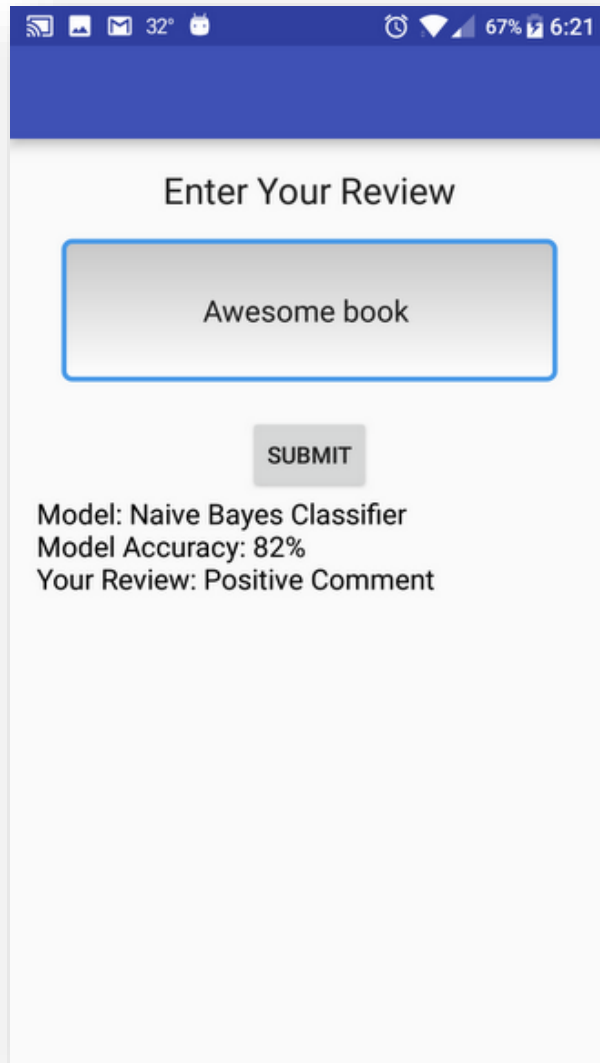
# Evaluation

- ▶ After training the system, evaluated performance on test data.
- ▶ Computed the accuracy by the formula

$$\text{Accuracy} = \frac{\text{Correctly predicted class}}{\text{Actual class}}$$

- ▶ Without preprocessing:
  - ▶ Accuracy 67%
  - ▶ Vocabulary of positive and negative words were not a true representation of category.
- ▶ After preprocessing:
  - ▶ Accuracy 82%
  - ▶ Not perfect, but working much better.

# Evaluation



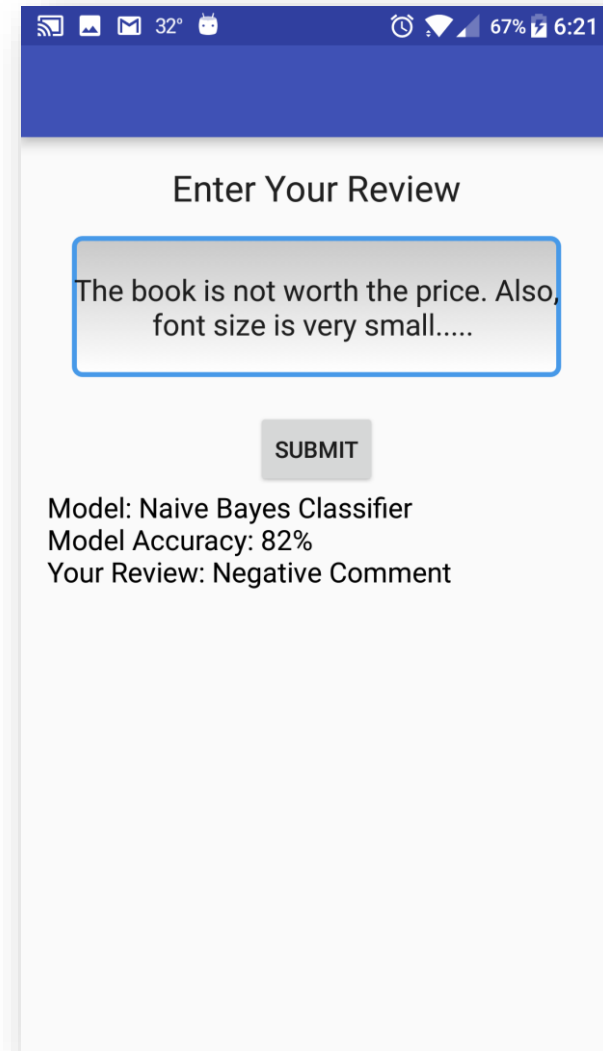
Enter Your Review

Awesome book

SUBMIT

Model: Naive Bayes Classifier  
Model Accuracy: 82%  
Your Review: Positive Comment

This screenshot shows a mobile application interface for submitting a review. At the top, there's a status bar with icons for signal, battery, and time (6:21). Below that is a blue header bar. The main content area has a title "Enter Your Review" and a text input field containing "Awesome book". A "SUBMIT" button is located below the input field. At the bottom, the model information is displayed: "Model: Naive Bayes Classifier", "Model Accuracy: 82%", and "Your Review: Positive Comment".



Enter Your Review

The book is not worth the price. Also,  
font size is very small.....

SUBMIT

Model: Naive Bayes Classifier  
Model Accuracy: 82%  
Your Review: Negative Comment

This screenshot shows the same mobile application interface as the previous one, but with a negative review. The text input field now contains "The book is not worth the price. Also, font size is very small.....". The "SUBMIT" button remains. The model information at the bottom is the same, but the final line now reads "Your Review: Negative Comment".

# Algorithm - (2) Review Ranking

- ▶ Review important for user to make purchase decision.
- ▶ Local Survey Stats:
  - ▶ 90% consumers read less than 10 reviews before forming an opinion.
- ▶ Order of review display matters.
- ▶ What factors should determine the order ?
  - ▶ Is only date enough?
  - ▶ ***NO!***

# Algorithm - (2) Review Ranking

- ▶ Analysis of reviews, to determine key factors.
- ▶ Assigned each factor weight, to signify importance.
- ▶ Factors (Total Weight: 75):
  - ▶ Date (Max Weight: 35)
  - ▶ Review Text (Max Weight: 25) - length is important
  - ▶ Rating (Max Weight: 15)
- ▶ **Top 10 Reviews** with maximum total weight will be displayed.

# Algorithm - (2) Review Ranking

- ▶ Example:
- ▶ Consider below reviews, which is better?
  - ▶ “Good.” Date: 08/01/2017 Rating: 4 stars
  - ▶ “*This is a great book, its very interesting and the scene is portrayed in a beautiful way.*” Date: 02/15/2012. Rating: 5 stars

# Algorithm - (2) Review Ranking

- ▶ Example:
- ▶ Consider below reviews, which is better?
  - ▶ “Good.” Date: 08/01/2017 Rating: 4 stars
  - ▶ “*This is a great book, its very interesting and the scene is portrayed in a beautiful way.*” Date: 02/15/2012 Rating: 5 stars

## Factors:

**Date**

**Review Text**

**Rating**

# System Architecture

- ▶ The application resides on the Android phone and interacts with a back-end server to retrieve data.
- ▶ The application involves:
- ▶ 1. **A back-end server:** (Java TCP Socket Programming)
  - ▶ This server performs two core functionalities.
    - a) Build model on training data of Naïve Bayes and predict a new review as positive / negative.
    - b) Rank and display top 10 reviews to the user.
- ▶ 2. **Front-end application:** (Android)
  - ▶ User can search a specific book by ID and see its reviews.
  - ▶ User can also post a new review.



# Limitations

- ▶ The model results are not perfect, it does predict incorrectly sometimes
- ▶ Naïve Bayes does not consider bigrams properly.
- ▶ Review Ranking functionality is based on very limited factors.

# Conclusion and Future Work.

- ▶ Built a portable android application to help user find and post reviews of a book.
- ▶ The User Interface can be improvised to add more functionalities and improvise the design
- ▶ Learnt usage of Naïve Bayes for sentiment analysis and the importance of preprocessing.
- ▶ The results of Naïve Bayes can be enhanced further by adding bigrams to the model.
- ▶ Developed a custom ranking algorithm to identify top reviews, this functionality can be enhanced to include more factors and assign weight more efficiently while ranking reviews.