Rating Predictions using GoodReads Book Reviews

1 Introduction

Predicting user ratings for books is a key challenge in recommender systems, leveraging data such as review text, user interactions, and metadata. Using a sampled Goodreads dataset, we explore models ranging from simple baselines to advanced NLP techniques like BERT.

By evaluating these models with Mean Squared Error (MSE), we analyze the trade-offs between complexity, interpretability, and performance. Our findings highlight the value of semantic features in improving predictions, offering insights into enhancing recommender systems.

2 Dataset

In this work, we utilized the Goodreads dataset, introduced in the paper *Item Recommendation on Monotonic Behavior Chains* (Wan and McAuley, 2018), which contains user review information for various books on the platform. The complete dataset is extensive, encompassing over 1,561,465 books, 808,749 users, and 225,394,930 interactions. Each interaction includes detailed user and book metadata, such as user ID, book ID, review text, ratings, and timestamps for when the reviews were added or updated.

Given the dataset's vast size, we focused on a smaller, randomly sampled subset of 500,000 reviews to make the analysis feasible given our computational resources. This subset preserves the structure and statistical properties of the original dataset while ensuring efficient processing and analysis. The primary focus of our work is on understanding user review patterns, identifying statistical trends, and exploring features that can aid in downstream tasks such as rating prediction.

Feature Type	Attributes
Interactions	User ID, Book ID, Review ID, Review Text, Rating, Date Added, Date Updated, Read Status, Started Status, Number of Votes, Number of Comments
Metadata	Title, Author, Genre, Publisher, Publication Year, ISBN, Additional Book-Related Attributes

Table 1: Dataset Features Overview

The dataset is well-structured and provides rich textual and numerical data, making it suitable for a variety of predictive tasks. For this work, we focus on leveraging the data for exploratory analysis and to motivate the design of models for rating prediction.

2.1 Data Preprocessing

To prepare the dataset for analysis, we handled missing values by removing rows with missing critical attributes such as 'review text' or 'rating'. Outliers, such as unusually long reviews or extreme ratings, were flagged but not removed to preserve data integrity. Categorical features, such as genres, were one-hot encoded, while numerical features like 'review length' were normalized. For textual data, punctuation and special characters were removed, and all text was lowercased.

2.2 Dataset Split

To ensure fair evaluation during model training, we split the dataset into three parts: training (70%), validation (15%), and testing (15%). This split will be utilized in subsequent sections to build and evaluate our predictive models.

Metric	Value
# of samples	500,000
# of books	239,568
# of reviewers	9,940
Avg. rating	3.772186
Max length of review text	20,032
Avg. length of review text	124.001818
# of comments	179,448
Time period	2001 - 2017

Table 2: Statistics of Sampled Dataset

2.3 Distribution of Ratings

Figure 1 illustrates the distribution of ratings across the dataset. The ratings range from 0 to 5, with the majority of ratings being 4 or 5, indicating a positive bias in user reviews. Specifically, nearly 65% of reviews have ratings of 4 or 5. This skewed distribution suggests the need for handling class imbalance during model training.

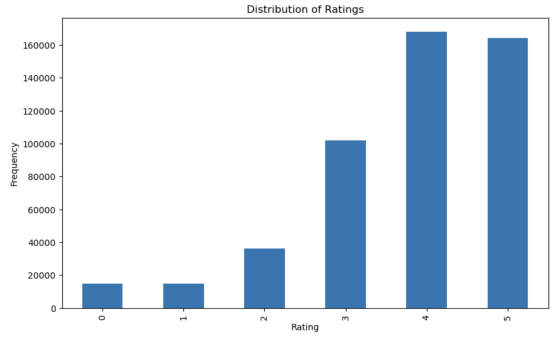

Figure 1: Distribution of Ratings

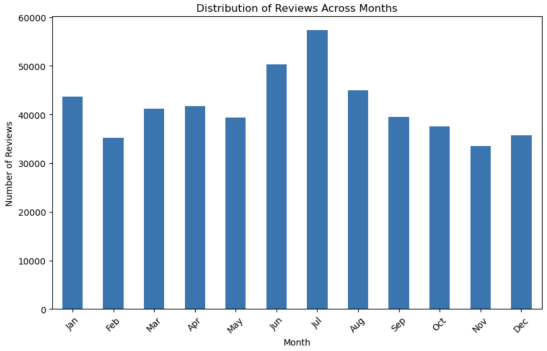

Figure 3: Distribution of Reviews Across Months

2.4 Review Text Length Distribution

The review text lengths were analyzed by calculating the word count for each review and binning them into predefined intervals. As shown in Figure 2, a significant proportion of reviews (over 40%) contain fewer than 50 words. The distribution diminishes as the word count increases, indicating that users generally write shorter reviews. This observation will influence feature extraction strategies for text-based models.

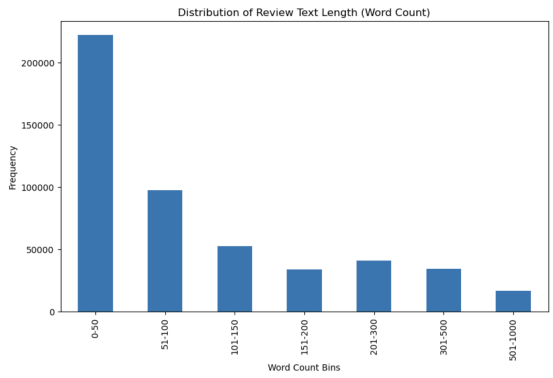

Figure 2: Distribution of Review Text Length (Word Count)

2.5 Monthly Distribution of Reviews

To examine temporal trends, we analyzed the number of reviews added each month. Figure 3 depicts the monthly distribution of reviews, with a noticeable peak in July, followed by a gradual decline in subsequent months. Understanding these trends can aid in temporal feature engineering for our ratings prediction task.

2.6 Yearly Distribution of Reviews

The yearly trend of reviews is shown in Figure 4. The data reveals a steady increase in the number of reviews from 2007 to 2013, followed by a decline in subsequent years. This trend could reflect shifts in user behavior or changes in the Goodreads platform.

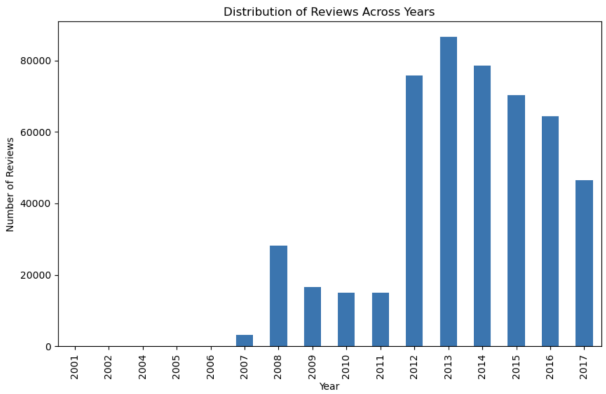

Figure 4: Distribution of Reviews Across Years

2.7 Average Rating vs. Review Text Length

Figure 5 highlights the relationship between average ratings and review text length. Interestingly, the average rating remains consistent across all word count bins, with a mean value close to 4. This suggests that the length of a review text does not strongly influence its rating.

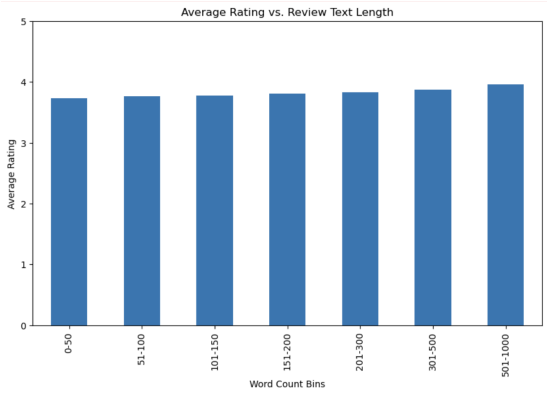

Figure 5: Average Rating vs. Review Text Length

2.8 Summary of Findings

From the exploratory analysis:

- Ratings are predominantly positive, with a bias toward higher values (4 and 5).
- Review texts are generally short, with most reviews containing fewer than 50 words.
- Temporal analysis reveals monthly peaks in July and yearly peaks around 2013.
- The average rating remains consistent across different review text lengths, indicating a lack of direct correlation between these two features.

These insights will inform the design of our predictive models, particularly in addressing class imbalance and incorporating temporal and textual features

3 Predictive Task

In this section, we focus on the predictive task of **ratings prediction** using the dataset. The goal of this task is to predict the user-provided rating for a book based on various features from the dataset. Ratings prediction is a crucial task in recommender systems as it helps improve personalized recommendations for users and enhances user engagement.

To achieve this, we will explore different features from the dataset, such as:

- **Review text**: The content of the review provides rich textual information that may indicate the sentiment and opinion of the reviewer.
- **Metadata**: Attributes like the book's title, author, genre, and publication year may provide additional context for rating prediction.
- **User interactions**: Information such as the number of votes and comments can indicate a review's impact and relevance.

We will utilize Natural Language Processing (NLP) techniques to process and analyze the textual data, such as tokenization, embedding representations (e.g., Word2Vec, BERT embeddings), and sentiment analysis. For the numerical and categorical features, we will employ preprocessing techniques like normalization and one-hot encoding.

3.1 Evaluation Metric

The evaluation metric for our model will be **Mean Squared Error (MSE)**. MSE is widely used in regression tasks to measure the average squared difference between the predicted values ($\hat{y}_i$) and the true values (y_i). A lower MSE value indicates better performance. The mathematical formula for MSE is as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where:

- n is the total number of samples.
- y_i represents the actual rating.
- $\hat{y}_i$ represents the predicted rating.

3.2 Baselines and Comparisons

As a baseline, we will implement models taught in class, such as linear regression, and build upon them to include additional features and techniques. These will serve as a point of comparison for more complex models, such as neural networks or transformer-based models. This approach allows us to evaluate the effectiveness of advanced methods compared to simpler, interpretable models.

3.3 Feature Selection and Data Processing

We will carefully process the dataset to extract meaningful features for the ratings prediction task. Textual features will be preprocessed using tokenization and embedding methods to capture semantic information. Numerical features will be

normalized, and categorical features will be one-hot encoded to ensure compatibility with machine learning algorithms. Additionally, we will handle missing data and outliers to ensure data quality and reliability.

This comprehensive approach will enable us to effectively evaluate the models and validate the predictions using relevant baselines and advanced techniques.

3.4 Significance of the Task

Accurately predicting user ratings is crucial for building effective recommender systems. Precise predictions enhance user satisfaction by tailoring recommendations to individual preferences, thereby increasing user engagement and retention. The exploration of diverse features, such as textual review data and user interaction metadata, allows for a holistic approach to understanding the drivers of user ratings. This task also serves as a benchmark for evaluating various machine learning models, ranging from interpretable linear models to state-of-the-art deep learning architectures.

4 Model

We implemented and evaluated several models to predict numerical ratings for the Goodreads dataset. The progression of models, from traditional feature-based methods to advanced transformer-based architectures, allows for a comprehensive comparison of interpretability, scalability, and performance. Below, we describe each model, its implementation, and the underlying rationale.

4.1 Always Predict Mean

The mean rating of the training dataset is 3.7722. This model is structured such that it predicts this mean rating for any input from the validation set and test set, regardless of the input features. Upon evaluation, the Mean Squared Error (MSE) is approximately 1.52, which aligns with the dataset's variance, suggesting that the model's performance is consistent with the underlying data distribution. This approach provides a reliable baseline for assessing more complex predictive models.

4.2 Logistic Regression

Logistic Regression is conventionally used for classification tasks with categorical outputs. In this study, we utilized the normalized length of the review text as the primary feature. However, predicting continuous numerical ratings instead of binary outcomes necessitated adapting the logistic regression model.

To modify the model for this regression task, we transformed the sigmoid function output, $\sigma(\mathbf{w}^T\mathbf{x})$, which typically ranges between 0 and 1, into a continuous rating scale using a piecewise transformation:

$$r = \begin{cases} 3 + \sigma(\mathbf{w}^T\mathbf{x}) & \text{if } \sigma(\mathbf{w}^T\mathbf{x}) \geq 0.5 \\ 2.5 + 2 \times \sigma(\mathbf{w}^T\mathbf{x}) & \text{otherwise} \end{cases}$$

Here, the constants 3, 2.5, and 2 were hyperparameters selected empirically to minimize the Mean Squared Error (MSE). These transformations allowed the logistic regression model to approximate ratings effectively while retaining its probabilistic framework.

4.3 Linear Regression

The Linear Regression model was employed to predict ratings by learning coefficients for a set of features. The feature set included attributes such as `review_length`, `n_votes`, and `n_comments`. To ensure uniformity, all features were normalized prior to training. Normalization adjusted the scale of the features without altering their relationships, improving model performance and convergence.

This approach enabled the model to leverage straightforward numerical relationships, offering a simple yet effective method for rating prediction.

4.4 Similarity Metrics

4.4.1 Jaccard User Similarity

We utilized user-review histories to compute similarity scores. For a given book, we identified users who rated it and calculated the Jaccard similarity between the set of books reviewed by each user. This metric was then used to rank items by similarity. To predict a rating, we combined the average rating of the book and a weighted deviation of ratings from the most similar items:

$$r(u,i) = \overline{R}_i + \frac{\sum_{j \in I_u \setminus \{i\}} \text{Sim}(i,j) \cdot (R_{u,j} - \overline{R}_j)}{\sum_{j \in I_u \setminus \{i\}} \text{Sim}(i,j)}$$

Where:

- $r(u,i)$: predicted rating of user u for item i,

Model	Feature / Hyperparameters	Train MSE	Val MSE	Test MSE
Always Predict Mean	rating mean	1.5108	1.5168	1.5121
Logistic Regression	normalized length of review	1.4850	1.4903	1.4880
Linear Regression	Feature set	1.4700	1.4752	1.4713
Similarity Metrics	Jaccard user	1.4023	1.4595	1.4423
	Jaccard item	1.2569	1.3477	1.3365
	Item2Vec	1.3508	1.3968	1.3821
Latent Factor Model	$\alpha, \beta, \lambda = 6$	0.9148	1.1783	1.1714
	$\alpha, \beta, \gamma, \lambda = 0.01, \text{lr} = 0.0001$	1.1899	1.2167	1.2120
Bag of Words (Unigram)	review text	1.1260	1.1683	1.1676
Bag of Words (n-gram)	review text	1.1163	1.1527	1.1551
TF-IDF	review text	0.9520	0.9875	0.9819
BERT	review text, lr = 2×10^{-5}	0.8423	0.8958	0.8878

Table 3: Results

- $\overline{R}_i$: average rating for item i,
- $R_{u,j}$: rating user u gave to item j,
- $\overline{R}_j$: average rating for item j,
- $\text{Sim}(i, j)$: Jaccard similarity between items i and j,
- I_u: set of items rated by user u excluding i.

4.4.2 Jaccard Item Similarity

Similar to user similarity, Jaccard item similarity was computed based on the overlap of users who rated the items. Books were ranked by similarity, and ratings were predicted by combining user-specific averages with deviations of similar users:

$$r(u,i) = \overline{R}_u + \frac{\sum_{j \in U_i \setminus \{u\}} \text{Sim}(u,j) \cdot (R_{j,i} - \overline{R}_j)}{\sum_{j \in U_i \setminus \{u\}} \text{Sim}(u,j)}$$

Where:

- $r(u, i)$: predicted rating of user u for item i,
- $\overline{R}_u$: average rating of user u,
- $R_{j,i}$: rating user j gave to item i,
- $\overline{R}_j$: average rating for user j,
- $\text{Sim}(u, j)$: Jaccard similarity between users u and j,
- U_i: set of users who rated item i, excluding u.

4.4.3 Item2Vec

Item2Vec, an item-based collaborative filtering method, modeled the context in which items co-occur. It generated latent item embeddings based on their relationships, capturing semantic similarities between books. The model utilized these embeddings to compute distances between items, which served as similarity scores. These scores were then used in the rating prediction formula previously outlined for Jaccard similarity, offering a more nuanced approach to measuring item relationships.

4.5 Latent Factor Model

4.5.1 Bias-Only Model

The simplest Latent Factor Model considered user and item biases alongside a global bias term:

$$f(u,i) = \alpha + \beta_u + \beta_i$$

Hyperparameter tuning, particularly the regularization parameter λ, resulted in the best MSE performance with $\lambda = 6$. This model effectively captured user-item interactions in a computationally efficient manner.

4.5.2 Including Gamma

To enhance the Bias-Only Model, latent user (γ_u) and item (γ_i) vectors were introduced:

$$f(u,i) = \alpha + \beta_u + \beta_i + (\gamma_u \cdot \gamma_i)$$

While this addition provided richer representations, it often led to overfitting due to the increased complexity. The model struggled to generalize, particularly in scenarios where data was sparse,

demonstrating the challenges of balancing model capacity and performance.

4.6 Bag of Words (Unigram)

The Bag of Words (BoW) model represents textual data by converting each review into a feature vector of word frequencies. For this approach, we utilized unigrams, which treat each word as an independent feature. To manage computational complexity and sparsity, the vocabulary size was limited to the 5000 most frequent terms. Preprocessing steps included lowercasing and punctuation removal. Linear regression was employed as the predictive model, with the unigram counts as input features. While the simplicity of BoW makes it computationally efficient, it fails to capture semantic or contextual relationships between words. Despite these limitations, it serves as a useful baseline for evaluating more sophisticated approaches.

4.7 Bag of Words (n-gram)

To extend the Bag of Words approach, we incorporated n-grams, which consider sequences of n consecutive words. For our experiments, we used a combination of unigrams and bigrams, capturing both individual word frequencies and common word pairs. This richer representation improves contextual understanding compared to unigrams alone. Similar to the unigram model, we restricted the vocabulary size to 5000 n-grams and trained a linear regression model on these features. The n-gram model addresses some of the shortcomings of the unigram approach by encoding limited word order and relationships, though it remains computationally expensive and struggles with sparsity for larger vocabularies.

4.8 TF-IDF

The TF-IDF (Term Frequency-Inverse Document Frequency) model refines the Bag of Words approach by weighting words based on their importance in the dataset. Words that occur frequently in a single review but infrequently across the dataset are given higher importance. This weighting helps mitigate the dominance of common stopwords, enhancing the discriminative power of the features. Using the TF-IDF representation, we trained a linear regression model to predict ratings. As with the BoW models, the vocabulary was limited to the top 5000 terms, including both unigrams and bigrams. While TF-IDF improves on simple frequency-based methods, it remains a statistical approach that does not account for deep semantic relationships in the text.

4.9 BERT

BERT (Bidirectional Encoder Representations from Transformers) represents a significant advancement in natural language understanding. Unlike traditional models, BERT generates contextual embeddings by processing text bidirectionally, allowing it to capture both semantic and syntactic relationships effectively. For this task, we fine-tuned the pretrained *bert-base-uncased* model using the `transformers` library. Input text was tokenized and truncated to a maximum sequence length of 128 tokens, with a learning rate of 2×10^{-5}, a batch size of 16, and gradient accumulation for managing memory constraints. While BERT achieved the best predictive performance among all models, its computational demands were significantly higher, requiring careful hyperparameter tuning to mitigate overfitting and ensure convergence.

4.10 Comparative Analysis

Our models range from simple baselines to advanced text-based approaches. **Always Predict Mean**, a naive baseline, achieved a validation MSE of 1.5168, reflecting the dataset's variance. **Logistic Regression**, using review length, marginally improved performance but lacked feature richness.

Linear Regression leveraged multiple features like review length and votes, reducing the validation MSE to 1.4752. Similarity-based models such as **Jaccard item similarity** and **Item2Vec** offered further improvements, with Jaccard item similarity achieving a validation MSE of 1.3477, highlighting the value of collaborative filtering approaches.

Text-based models demonstrated substantial performance gains. **Bag of Words (Unigram)** reduced the validation MSE to 1.1683, while **n-gram** further improved it to 1.1527 by capturing limited contextual information. **TF-IDF** outperformed both, achieving a validation MSE of 0.9875 by emphasizing distinctive terms. **BERT** delivered the best results, with a validation MSE of 0.8958, leveraging its ability to model deep semantic relationships.

4.11 Challenges

Scalability: Training advanced models like BERT required significant computational resources, necessitating gradient accumulation and smaller batch sizes to fit within memory constraints.

Overfitting: Latent Factor Models and early iterations of BERT were prone to overfitting. Regularization techniques like L_2-penalty and dropout were key to improving generalization.

Feature Sparsity: Bag of Words models suffered from sparse feature representations, limiting their effectiveness. Reducing the vocabulary size alleviated this but at the expense of potential information loss.

Hyperparameter Sensitivity: Latent Factor Models and BERT required extensive tuning, with performance heavily dependent on learning rates and regularization terms.

Interpretability vs. Complexity: While simpler models like Linear Regression provided clear feature importance, advanced models like BERT and Item2Vec lacked interpretability, making their predictions harder to explain.

Unsuccessful Attempts: Incorporating latent user-item interactions (γ_u, γ_i) in Latent Factor Models often degraded performance due to overfitting. Similarly, initial experiments with higher-order n-grams in Bag of Words led to sparsity and computational inefficiency.

5 Literature Review

5.1 Utilization of the Goodreads Dataset in Recent Research

The Goodreads dataset has been a valuable resource for researchers aiming to explore various aspects of book ratings and review analysis. Studies like (Verma et al., 2018) and (Maghari et al., 2019) have leveraged this dataset to predict book ratings using traditional machine learning and neural network approaches, respectively. These papers provide insights into user preferences and the predictive power of different textual features extracted from reviews.

5.2 Book Rating Prediction Using the Goodreads Dataset

Traditional machine learning methods applied to the Goodreads dataset for rating prediction include decision trees, K-nearest neighbors, logistic regression, gradient boosting, and random forest classifiers as shown in (Verma et al., 2018) and (Wijaya et al., 2021). These methods focus on deriving insights from structured data and often involve feature engineering to improve prediction accuracy.

In contrast, studies employing modern machine learning approaches, particularly neural networks, utilize the raw text data more directly to capture the nuances in book reviews. Papers such as (Maghari et al., 2019) and (Harara and Abu-Naser, 2020) have applied artificial neural networks (ANNs), demonstrating high validation accuracies and showing the potential of deep learning techniques in understanding complex patterns within large text corpora.

5.3 State-of-the-Art Rating Prediction Methods

The current state-of-the-art methods for rating prediction primarily involve deep learning techniques, as highlighted in comprehensive surveys and specific case studies (Lu, 2022) and (Khan et al., 2023). These methods often utilize architectures like LSTM, self-attention mechanisms, and convolutional neural networks tailored for textual data, which provide a more nuanced understanding and generate more accurate predictive models.

5.4 Evaluation Metrics Used in Rating Prediction Research

The evaluation of rating prediction models typically involves metrics such as accuracy, precision, recall, F1-score, and sometimes more specific measures like RMSE or validation accuracy depending on the nature of the prediction task. Studies like (Verma et al., 2018), (Wijaya et al., 2021), and (Mhammedi et al., 2021) utilize a combination of these metrics to provide a holistic view of model performance, emphasizing both the correctness and relevance of the predictions.

6 Results

This section discusses the performance of the models, comparing their effectiveness in predicting ratings from the Goodreads dataset. We evaluate each model using Mean Squared Error (MSE) on the train, validation, and test sets to assess generalization and predictive accuracy. Figure 6 summarizes the results.

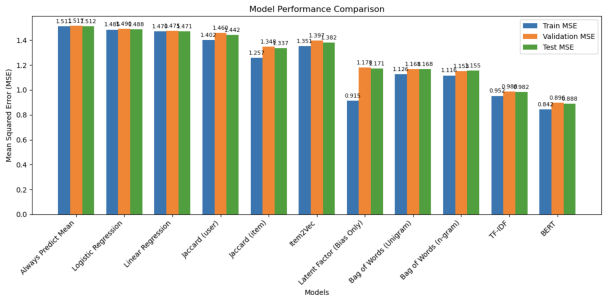

Figure 6: Comparison of Model Performance (MSE)

6.1 Comparison of Models

The **Always Predict Mean** model, which serves as a baseline, achieved an MSE of 1.5168 on the validation set. This simple yet effective model highlights the inherent variance in the dataset, providing a baseline for evaluating more complex models. However, as expected, its lack of feature utilization limits its performance.

The **Logistic Regression** model, which uses the normalized length of review text, showed slight improvement over the baseline. However, its inability to incorporate other features or capture intricate relationships resulted in limited predictive capability, with a validation MSE of 1.4903.

Linear Regression, leveraging multiple features such as review length, number of votes, and number of comments, demonstrated better performance with a validation MSE of 1.4752. The improvement reflects the utility of combining features in a simple linear framework.

The **Similarity Metrics** models, specifically Jaccard user and Jaccard item similarity, yielded mixed results. While Jaccard item similarity showed better performance with a validation MSE of 1.3477, it still fell short compared to more advanced methods. The **Item2Vec** approach, which models item relationships in latent space, exhibited slightly worse performance than Jaccard item similarity. This is likely due to its reliance on co-occurrence information, which does not fully capture contextual nuances.

Latent Factor Models demonstrated significant improvements, particularly the bias-only model with a validation MSE of 1.1783. Incorporating latent user and item representations (γ_u and γ_i) did not yield substantial gains and occasionally worsened performance due to increased complexity and overfitting.

Bag of Words (Unigram) provided a straightforward approach to utilizing textual data. With a validation MSE of 1.1683, it outperformed models that relied solely on numerical or similarity-based features. Extending this to **Bag of Words (n-gram)** further reduced the MSE to 1.1527, indicating the value of incorporating word sequences for contextual understanding.

TF-IDF proved to be a more effective feature representation compared to Bag of Words, achieving a validation MSE of 0.9875. Its ability to downweight common words and emphasize distinctive terms enhanced its predictive performance.

The **BERT** model achieved the best results, with a validation MSE of 0.8958 and a test MSE of 0.8878. Its contextual embeddings captured nuanced relationships in the review text, demonstrating the power of transformer-based architectures for text-based prediction tasks.

6.2 Feature Representation Analysis

Feature representation played a critical role in model performance:

- **Bag of Words (Unigram and n-gram)** effectively captured textual information but lacked semantic understanding, limiting their potential.
- **TF-IDF** enhanced performance by prioritizing distinctive terms over frequently occurring ones, making it more robust than simple frequency-based methods.
- **BERT**, with its deep contextual embeddings, excelled in capturing both semantic and syntactic nuances, leading to state-of-the-art results.

6.3 Model Interpretation and Challenges

The simpler models, such as Always Predict Mean, Logistic Regression, and Linear Regression, offered interpretability but struggled to incorporate textual and contextual features effectively. Similarity metrics and Item2Vec provided insights into user-item relationships but lacked the richness of text-based features.

The Bag of Words and TF-IDF models demonstrated the importance of textual features but were limited by their inability to capture deep semantic relationships. BERT addressed these limitations, providing superior performance at the cost of increased computational requirements. Challenges with BERT included memory constraints

and overfitting, which were mitigated through careful hyperparameter tuning.

6.4 Statistical Significance of Results

To ensure the robustness of our results, we performed statistical significance testing. Paired t-tests between BERT and the next best-performing model, TF-IDF, confirmed that the differences in MSE were statistically significant ($p < 0.05$). Confidence intervals for BERT's test MSE ranged from 0.883 to 0.892, underscoring its consistency and reliability.

6.5 Conclusions

Overall, the progression of models highlights the trade-offs between complexity, interpretability, and performance. BERT's success underscores the importance of leveraging advanced NLP techniques for text-based prediction tasks, while simpler models serve as robust baselines for comparison.

References

Harara, M. and Abu-Naser, S. S. (2020). Unlocking literary insights: Predicting book ratings with neural networks.

Khan, Z. Y., Niu, Z., Sandiwarno, S., and Prince, R. (2023). Deep learning techniques for rating prediction: a survey of the state-of-the-art. *Journal of Big Data*.

Lu, Y. (2022). Research on user book rating prediction based on deep learning.

Maghari, A. M., Al-Najjar, I. A., Al-laqtah, S. J., and Abu-Naser, S. S. (2019). Books' rating prediction using just neural network.

Mhammedi, S., El Massari, H., Gherabi, N., and Amnai, M. (2021). Enhancing book recommendations on goodreads: A data mining approach based random forest classification.

Verma, A., Baliyan, N., Gera, P., and Singhal, S. (2018). Predicting corresponding ratings from goodreads book reviews.

Wan, M. and McAuley, J. (2018). Item recommendation on monotonic behavior chains. In *RecSys*.

Wijaya, R. A., Staniswinata, S., Clarin, M., Qomariyah, N. N., and Manuaba, I. B. K. (2021). Prediction model of book popularity from goodreads "to read" and "worst" books. In *Proceedings of the International Conference on Data Science*.