Machine Learning with Applications 23 (2026) 100849

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Towards an intelligent review helpfulness estimation: A novel dataset and machine learning framework

Rakibul Hassan *, Shubhashish Kar, Jorge Fonseca Cacho, Shaikh Arifuzzaman *

Department of Computer Science, University of Nevada Las Vegas, 4505 S Maryland Pkwy, Las Vegas, 89154, NV, USA

ARTICLE INFO

Keywords:
Review helpfulness prediction
Labeling reviews
KMeans clustering
Text classification

ABSTRACT

The rise of online rental platforms has led to an overwhelming amount of user-generated content, making it difficult for prospective consumers to discern which reviews are helpful. Existing approaches often rely on raw helpfulness votes, which are sparse, subjective, and temporally inconsistent. Also, there is lack of labeled dataset in the field of rental review usefulness prediction. This paper introduces a novel dataset of apartment reviews collected from online website and proposes an intelligent machine learning framework to predict the helpfulness of rental reviews. To address the challenge of obtaining reliable labels from sparse and subjective user votes, a scoring-based labeling strategy is developed that uses helpful vote count and timeliness. A diverse set of features including TF–IDF vectors, sentiment polarity, rating deviation, and review length are used to capture both textual and behavioral aspects of the reviews. Multiple classifiers, including Logistic Regression, Naive Bayes, and XGBoost, are systematically evaluated under 5-fold cross-validation, along with a rule-based and deep learning models.

Experimental results show that XGBoost consistently achieves the best overall performance with an accuracy of 0.71 and ROC-AUC of 0.75 when leveraging all features. This research makes three key contributions: (i) the first large-scale dataset for rental review, (ii) auto annotation technique that uses clustering approach with score from user votes and time since posted, and (iii) comprehensive evaluation pipeline spanning rule-based, traditional, and deep learning classifiers. Together, these advances establish a foundation for intelligent rental review helpfulness estimation, with broader implications for e-commerce, hospitality, and user-generated content analysis.

1. Introduction

In the digital era, online reviews have become pivotal in shaping consumer decisions across various sectors, including the rental housing market. According to a survey, 75% of renters examine a community's online reviews prior to making contact, and 64% acknowledge that these reviews contribute to their decision-making process (SatisFacts, 2025). Furthermore, 69% of renters rely on property ratings and reviews during their apartment search, with 84% indicating that such reviews influenced their leasing decisions (Apartments.com, 2025). Despite the importance of rental reviews in shaping prospective tenants' decisions, their perceived helpfulness remains far less studied than in e-commerce and product review contexts, motivating the need for this work.

Determining the helpfulness of a review remains a complex challenge, particularly for newly posted reviews. The elapsed time of any review is a crucial factor in review helpfulness assessment, as it directly influences a review's visibility and perceived usefulness. Newly posted reviews may not immediately garner sufficient votes to accurately reflect their utility, even if they contain critical insights about recent changes in property conditions, amenities, or management. This elapsed time in helpfulness voting can misrepresent the actual value of a review, leading potential renters to overlook pertinent and up-to-date information. Addressing this challenge requires a predictive framework that estimates helpfulness by leveraging textual and contextual features apart from the vote count. Unlike prior approaches, our study explicitly incorporates the temporal dynamics of review posting into the helpfulness labeling process, ensuring that newly posted yet potentially critical reviews are not undervalued.

Over the past decade, various machine learning and natural language processing (NLP) approaches have been applied to classify the helpfulness of online reviews. Traditional methods rely on supervised learning techniques, where models are trained using labeled datasets

* Corresponding authors.
E-mail addresses: hassar2@unlv.nevada.edu (R. Hassan), kars1@unlv.nevada.edu (S. Kar), jorge.fonsecacacho@unlv.edu (J.F. Cacho), shaikh.arifuzzaman@unlv.edu (S. Arifuzzaman).

https://doi.org/10.1016/j.mlwa.2026.100849
Received 26 July 2025; Received in revised form 6 January 2026; Accepted 17 January 2026
Available online 21 January 2026

Table 1
Prior studies on review helpfulness.

Studies	Datasets	Novel features	Techniques
Krishnamoorthy (2015)	Product reviews	Linguistic features, helpfulness votes	Linguistic category model
Kim et al. (2006)	Product reviews	Structural, lexical, syntactic features	SVM
Zeng et al. (2014)	Product reviews	Degree of detail	SVM
Li et al. (2020)	Product reviews	Source and content-based features	LSTM, CNN based DL models
Zheng et al. (2013)	Product reviews	Social features of reviewers, inherent nature of products	Semi-Supervised models

containing helpful and non-helpful reviews. However, most existing methodologies have focused on e-commerce and product reviews rather than rental property reviews. Unlike product reviews, rental reviews often contain highly subjective, experience-based narratives, making their classification more challenging. Additionally, there is a lack of labeled datasets specifically designed for rental review helpfulness prediction, which limits the effectiveness of supervised learning approaches. Furthermore, rental reviews include unique linguistic patterns, such as detailed descriptions of management responsiveness, maintenance issues, and lease-related policies, which differ significantly from traditional consumer product reviews. Table 1 demonstrates the existing studies in the field of review helpfulness prediction. To the best of our knowledge, this is the first attempt to construct a comprehensive dataset of rental property reviews for helpfulness prediction and to design a mathematically grounded labeling strategy that accounts for both time and voting behavior.

The contributions of this study are summarized as follows:

1. **Development of a Novel Dataset:** We compile a new dataset of online rental reviews from Apartments.com, encompassing various attributes which are apartment name, average rating for the apartment, rent and space information, geographical information, review title, review content, posted rating, review posting date, property manager, review response of the property manager, and helpfulness votes.
2. **Development of Mathematical Model for Labeling:** Recognizing the limitations of traditional labeling approaches, we propose a sophisticated labeling strategy that integrates helpful vote count, and time since posting. Proposed strategy, which uses weighted equation, aims to provide a more accurate representation of a review's helpfulness.
3. **Applying Learning Models for Helpfulness Prediction:** We apply rule-based classifier depending on various attributes. Also, we implement and evaluate various machine learning models including Logistic Regression, XGBoost and Naive Bayes on different combinations of feature sets. Further, we conduct experiments with LSTM and transformer-based-models to extract the latent language-based features to classify helpfulness leveraging the review content.

Through these contributions, we aim to enhance the predictive accuracy of review helpfulness in the rental housing market, thereby assisting prospective renters in making more informed decisions. The remainder of this paper is organized as follows: Section 2 reviews related work and highlights gaps in prior studies; Section 3 describes the methodology containing dataset development, feature extraction, labeling strategy and model selection; Section 4 presents the Experimental work and analysis of results; finally, Section 5 concludes with key findings and future research directions.

2. Background and related work

The study of review helpfulness has been widely explored in various domains, particularly in product reviews. However, to the best of our knowledge, no research has examined review helpfulness in the context of apartment reviews. This section discusses prior studies related to review helpfulness, focusing on key methodologies and findings that inform our research.

Numerous studies have explored automatic review helpfulness prediction primarily through content-based and contextual feature engineering. Content-based approaches, such as those using lexical, semantic, structural, and syntactic features, remain prevalent due to their generalizability across platforms (Hassan & Islam, 2020; Kim et al., 2006; Krishnamoorthy, 2015; Li et al., 2013; Martin & Pu, 2014; Yang et al., 2015). Contextual features—such as reviewer profiles (Cheng & Ho, 2015; Hu & Chen, 2016), social networks (Lu et al., 2010; Tang et al., 2013), metadata (Fang et al., 2016; Willemsen et al., 2011; Zhu et al., 2014), and less common elements like photos (Karimi & Wang, 2017; Yang et al., 2017), manager responses (Kwok & Xie, 2016), or travel distances (Lee et al., 2018)—have also been considered.

The textual content of the reviews serves as the fundamental feature for identifying their helpfulness. Kim et al. (2006) investigated various features for predicting review helpfulness and found that review length, unigrams, and product ratings were among the most influential indicators. Zeng et al. (2014) introduced a novel feature called the *degree of detail* which focuses on the overlap between the review and the corresponding product description. This metric effectively captured the richness of the review content and proved to be the most important feature in their experiments. Similarly, Liu et al. (2008) proposed a model for the helpfulness measurement of any review. They mentioned the three factors including writing style of the reviewer, timeliness of the review, and the reviewer's expertise to model the helpfulness. In addition, Hassan and Islam (2021) highlighted the importance of probabilistic sentiment scores in improving the detection of false reviews. Their proposed sentiment analysis model not only distinguishes between positive and negative reviews but also demonstrates how sentiment probability can improve the accuracy of deceptive review identification. Moreover, they found the term frequency–inverse document frequency (TF–IDF) as the most important feature for both sentiment analysis and fake review classification. Lee et al. (2021) highlighted the role of rating deviation in shaping review helpfulness. They found that a higher rating spread increases the trust of users in individual reviews, making detailed or extreme reviews more effective.

In addition, researchers have explored both supervised and unsupervised machine learning techniques to predict review helpfulness. Rathor et al. (2018) focused on the comparative study of different machine learning approaches, including Naive Bayes, Maximum Entropy, and Support Vector Machine, to analyze the sentiment of Amazon online reviews for products. Hendrawan et al. (2022) discussed the usefulness of Naive Bayes and XGBoost model on product review classification. Hassan and Islam (2019, 2020) used Naive Bayes and logistic regression model for effectively classifying fake online hotel reviews. From the perspective of deep learning, Li et al. (2020) demonstrated the three representative deep learning techniques including SRN, LSTM and CNN for classifying sentiment of online reviews. They also coined the term "readability" of the reviews to justify the accuracy of the better performing models. But only a few of the reviews are reliably labeled in the real-world huge datasets. Therefore, the supervised learning approaches do not always provide reliable results. Zheng et al. (2013) used semi-supervised ensemble learning technique to assess the usefulness of the online reviews.

Despite the extensive exploration of features and models for review helpfulness prediction, a major limitation in this domain remains the scarcity of publicly available, labeled datasets. Many studies rely on proprietary or platform-specific data (e.g., Amazon, Yelp), which limits

reproducibility and hinders generalization across domains (Alsmadi et al., 2020; Choi & Leon, 2023; Hassan & Islam, 2020; Jindal & Liu, 2007; Malik & Hussain, 2018; Pajola et al., 2023). Manual annotation by inspection is both time-consuming and subjective, motivating researchers to adopt other approaches. Ott et al. (2013) constructed a benchmark dataset for deception detection by generating deceptive reviews via Amazon Mechanical Turk. However, collecting very large scale data is not efficient using this method.

Hananto et al. (2022) proposed a text segmentation-based labeling strategy without relying on costly human annotation. Their method segments reviews into semantically coherent parts and fine-tunes a Latent Dirichlet Allocation (LDA) model for improved multi-topic clustering. Although most online reviews (Amazon, Yelp) are labeled using only helpfulness votes, Guo et al. (2021) labeled reviews with low votes by proposing an iterative Bayesian estimation method to generate more accurate helpfulness scores for training predictive models. However, most of the labeled datasets and labeling strategy are explored in sentiment analysis and product review domain. To the best of our knowledge, this is the first study in the field of rental review helpfulness prediction due to the lack of structured dataset in this field.

3. Methodology

The proposed methodology estimates the helpfulness score and subsequently classifies rental reviews through a structured pipeline, as illustrated in Fig. 1. We begin the process with data collection, where we gathered a large corpus of apartment reviews from Apartments.com using web-scraping. We then clean and preprocess the data to remove noise, correct formatting inconsistencies, and standardize textual content.

We design the methodology around an auto-annotation stage that programmatically assigns helpfulness labels to reviews using a composite scoring function based on temporal and engagement features. This stage computes scores with a technique that leverages the age and dynamics of reviews. We then apply K-means clustering to group reviews into distinct helpfulness categories. Next, we extract linguistic, semantic, and behavioral attributes from the review text and associated metadata. We analyze these features to assess their relevance and contribution to the classification task. Finally, we train and evaluate rule-based, machine learning, and deep learning models to predict the helpfulness labels, thereby completing the review analysis pipeline.

3.1. Data collection

We collected approximately **50,000 apartment reviews** from *Apartments.com*, representing a diverse set of residential properties across various cities and states in the United States. Each entry in the dataset includes both textual and structured information, enabling a comprehensive analysis of review characteristics and user feedback patterns.

To build the dataset, we employed a two-stage web-scraping approach. First, static content such as apartment names, ratings, and rent ranges was extracted using the BeautifulSoup library, which efficiently handles HTML parsing for structured elements. However, many parts of Apartments.com rely on JavaScript rendering, particularly user reviews and dynamic property details. To capture this information, we utilized Selenium, a browser automation framework capable of interacting with JavaScript-enabled webpages. Selenium allowed us to simulate real user behavior, such as scrolling, clicking "load more" buttons, and waiting for asynchronous elements to load, ensuring that reviews and associated metadata were collected completely. This combined approach provided a comprehensive dataset of both structured and unstructured information necessary for downstream analysis. Importantly, all data was collected in compliance with ethical research practices: only publicly available information was accessed, no user-identifiable details were retained, and scraping was conducted responsibly to avoid burdening the source website.

Table 2
Dataset attributes for apartment reviews.

Attribute name	Data type	Description
Apartment name	String	Name of the apartment complex
Rating	Float	Numerical rating provided by the reviewer (e.g., 1 to 5 stars)
Review	Text	The written review text describing the user's experience
Zip/City/State	String	Geographical identifiers for the property
Property manager	URL	URL referencing the management company or contact
Minimum rent/Maximum rent	Integer	Minimum and maximum rent prices listed for the property
Minimum Sqft/Maximum Sqft	Integer	Minimum and maximum unit sizes in square feet
Average rating	Float	Average rating of the apartment (used for feature engineering)
Helpfulness vote	Integer	Number of users who marked the review as helpful
Date	String	The date the review was posted

After performing initial data preparation, including basic filtering and quality checks, we reduced the dataset to approximately **40,000 reviews** suitable for further analysis. These reviews span a wide range of property types, geographic regions, and user sentiments. Table 2 shows the attributes and description of the final dataset.

This structured dataset provides a solid foundation for developing a review helpfulness estimation model using natural language processing and machine learning techniques. The next stage involves assigning helpfulness labels through a time-aware scoring function and clustering strategy.

3.2. Data cleaning and preprocessing

Effective data cleaning and preprocessing are crucial for building reliable models, especially when working with user-generated content that often includes inconsistencies, missing values, and noise. We performed several preprocessing steps to ensure the quality and integrity of the dataset used for helpfulness prediction.

3.2.1. Excluding recent reviews:

We excluded reviews posted within the last 75 days from the dataset. We determined this threshold by balancing the trade-off between minimizing data loss and allowing enough time for reviews to accumulate helpfulness feedback. The helpfulness vote system naturally favors older reviews that have had more time to receive votes. Including very recent reviews would introduce bias due to their limited exposure, often underestimating their actual helpfulness. By removing them, we ensured that each review had a fair opportunity to be evaluated by users, thereby making the helpfulness scores more meaningful. To support this decision, we analyzed the cumulative share of total helpfulness votes as a function of review age. As shown in Fig. 2, the majority of helpfulness feedback is concentrated in older reviews, while reviews posted within the last 75 days receive very few votes.

3.2.2. Handling missing average ratings

Some reviews lacked values for the average apartment rating. Since this feature is potentially useful in modeling the perceived quality of an apartment relative to individual review ratings, we imputed missing values using the average rating of other apartments within the same city. This localized imputation approach preserves city-level differences in rating behavior and avoids introducing global bias.

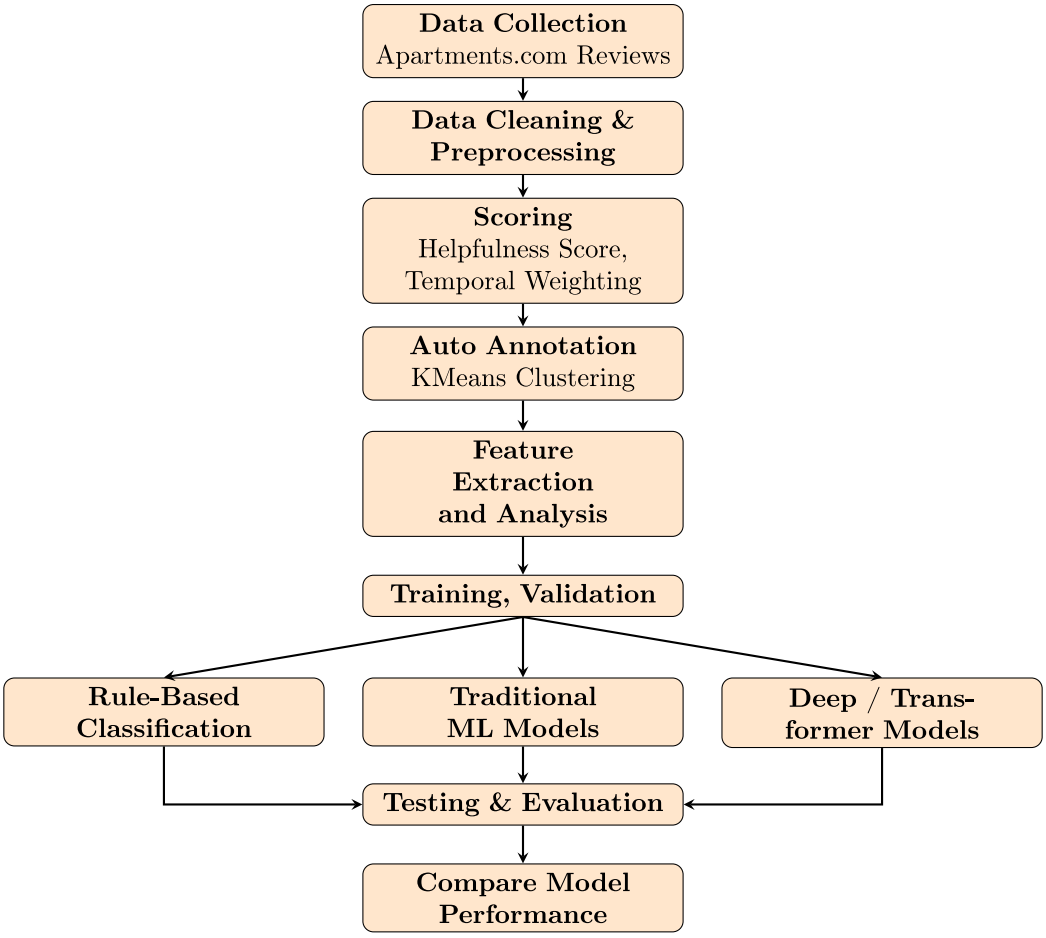


Fig. 1. Workflow diagram for review helpfulness classification with auto annotation.

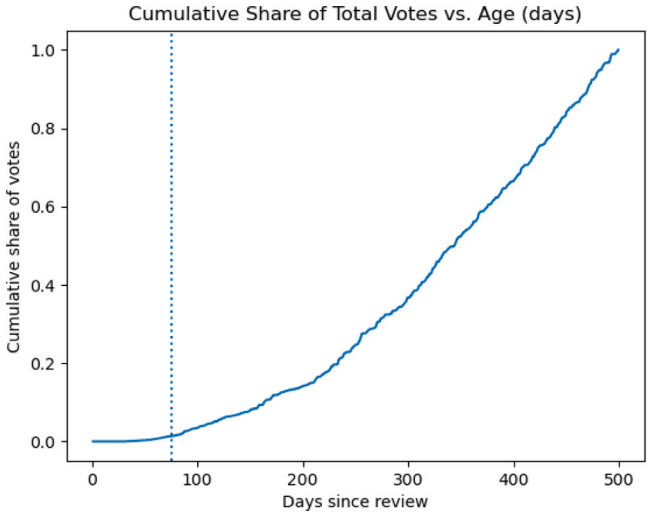


Fig. 2. Cumulative accumulation of helpfulness votes as a function of review age. Reviews newer than 75 days contribute very few votes, while older reviews account for the majority of helpfulness feedback.

3.2.3. Date field cleaning and standardization

The raw date entries in the dataset contained inconsistencies, such as varied abbreviations for month names (e.g., "Sept." vs. "Sep") and formatting artifacts (e.g., trailing periods). These inconsistencies interfered with automated date parsing and downstream feature extraction, such as calculating the number of days since a review was posted. We normalized the date formats to ensure consistency across all entries, enabling accurate time-based feature computation.

3.2.4. Removing entries with missing dates

We removed reviews without a valid posting date, since the date is a critical component in our labeling strategy and feature generation. Without it, we cannot determine a review's age, which directly affects the adjusted helpfulness score and temporal analysis. Including such entries would introduce noise and incomplete data into the model.

3.2.5. Duplicate and irrelevant review removal

We identified and removed duplicate reviews to avoid overrepresenting certain user opinions, which could skew model learning. Additionally, non-informative or irrelevant reviews (e.g., empty reviews or those containing placeholder text) were filtered out to maintain a dataset rich in substantive user feedback.

3.2.6. Text normalization and tokenization

Review text was standardized through lowercasing, punctuation removal, and stopword elimination to reduce noise and improve textual consistency. We then tokenized the text into individual words to facilitate feature extraction and model input formatting. We also experimented with stemming and lemmatization techniques to determine

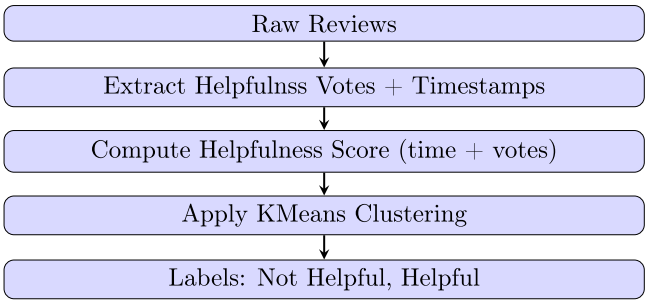


Fig. 3. Proposed approach for labeling dataset.

the most effective strategy for reducing words to their root forms while preserving semantic meaning.

Collectively, these preprocessing steps contributed to constructing a high-quality dataset suitable for feature engineering and modeling tasks. They helped mitigate biases, handle missing information thoughtfully, and ensure consistency in both structured and unstructured data components.

3.3. Labeling strategy: Temporal-adjusted score and K-means clustering

To develop an effective labeling strategy for helpfulness classification, we introduce a method based on a temporal-adjusted helpfulness score. This approach accounts for the impact of recency on the visibility and perceived helpfulness of a review, particularly addressing the issue that newer reviews may not have had sufficient time to accumulate helpful votes. Fig. 3 shows the proposed strategy for labeling dataset.

From the analysis, it is visible that the helpfulness votes in different cities for any particular apartment differ a lot. Some apartments in the big cities get a high number of people's views compared to the apartments in the small cities. Therefore, we score any review based on its apartment. This helps eliminate bias in scoring. The review that gets the maximum number of votes is used as the reference for calculating the scores of other reviews of the same apartment. The scoring interval is from 0 to 1.

The reviews that have been posted before the review with maximum helpfulness vote are scored based on the ratio of the review's vote and the maximum vote. The underlying reason is that the reviews posted before the review with maximum helpfulness vote should have at least as many votes as the maximum vote to get the highest score. We use Eq. (1) to calculate the score for each review.

$$s_r = \frac{v_r}{v_{max}} \tag{1}$$

Here, s_r represents the score of the review, v_r is the number of votes of review r, and v_{max} is the maximum number of votes received by any review among all the reviews for that apartment

The reviews that have been posted after the review with maximum helpfulness vote are scored based on the elapsed time compared to the review with the maximum vote. If the vote vs. elapsed time ratio for any review is higher than that of the review with maximum vote get score 1 out of 1.

$$\theta_r = \frac{v_r}{t_r} \tag{2}$$

$$\theta_{max} = \frac{v_{max}}{t_{max}} \tag{3}$$

$$s_r = \begin{cases} \frac{v_r \times t_{max}}{v_{max} \times t_r} & \theta_r < \theta_m \\ 1 & \theta_r \geq \theta_m \end{cases}$$

where:

- t_{max} is the elapsed time since the most-voted review has been posted (measured in number of days),
- t_r is the elapsed time for the review r,
- θ_r is the ratio of vote vs. elapsed time for the review,
- θ_{max} is the ratio of vote vs. elapsed time for the review with the maximum vote

This formulation emphasizes reviews that are both recent and moderately voted as potentially helpful, under the assumption that helpfulness scores for newer reviews are often underestimated due to limited exposure time. After computing this score for all reviews, we employed **K-Means clustering** with $k = 2$ to partition the dataset into two distinct classes:

1. **Not Helpful** – corresponding to the cluster with lower average score,
2. **Helpful** – corresponding to the cluster with higher average score.

This unsupervised clustering-based labeling technique allows for dynamic thresholding based on the distribution of helpfulness scores rather than relying on arbitrary vote cutoffs. It offers a scalable and temporally-aware framework for defining binary helpfulness labels, which are used in the subsequent classification models. Table 3 shows the number of samples for each label after clustering.

Table 3
Number of samples for each class.

Label value	Label meaning	Number of samples
0	Not helpful	20 509
1	Helpful	20 199

Although the labeling strategy is primarily used for offline dataset construction, it can be readily adapted for real-world scenarios in which new reviews arrive continuously. The system can be configured to operate in periodic batch cycles, recalculating $v_{\max}$ and $t_{\max}$ after each 75-day stabilization window. This procedure removes any reliance on future information, allows the model to incorporate newly accumulated votes, and ensures consistent labeling for both existing and incoming reviews.

3.4. Feature extraction

To construct a robust model for predicting review helpfulness, we extracted both textual and numerical features from the dataset. The features are designed to capture sentiment, linguistic patterns, behavioral signals, and contextual metadata. We organized the feature extraction process in an incremental fashion to assess the contribution of each feature set to model performance. Fig. 4 shows the overall process of our feature set generation.

3.4.1. Feature sets

We define four hierarchical feature sets, incrementally adding more informative attributes:

1. **Review Only:** Includes only the raw review text.
2. **Review and Rating Deviation:** Adds the difference between the individual rating and the average rating of the property.
3. **Review, Rating Deviation, and Sentiment:** Incorporates the sentiment polarity of the review text.
4. **Review, Rating Deviation, Sentiment, and Review Length:** Adds the number of words in each review as a proxy for verbosity or elaboration.

This stepwise expansion allowed us to evaluate the marginal value of each additional feature.

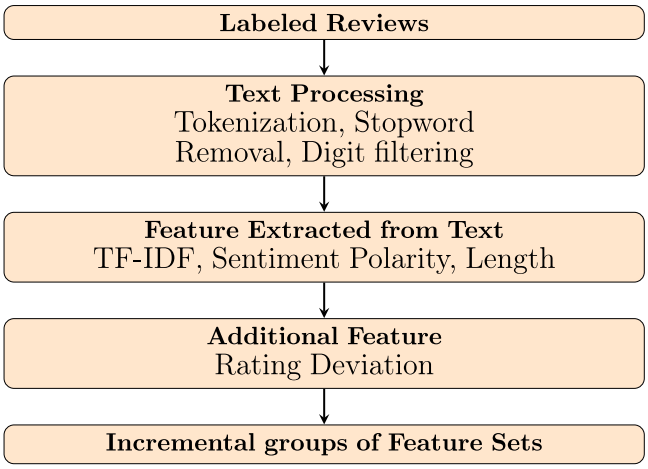


Fig. 4. Feature set generation process.

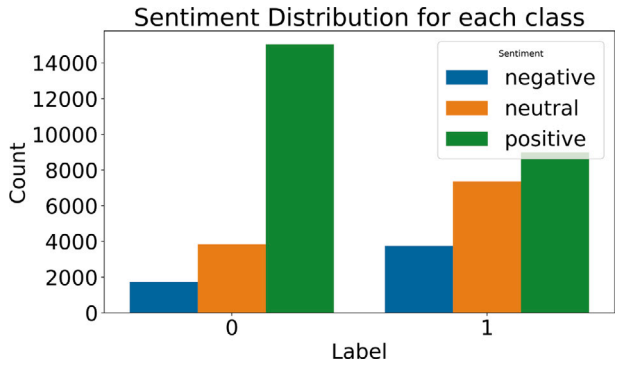


Fig. 5. Sentiment distribution by helpfulness label.

3.4.2. Textual feature engineering

We converted review text into numerical vectors using the TF–IDF (Term Frequency–Inverse Document Frequency) representation. We limited the number of features to 20,000 and used a custom preprocessing function that includes: Lowercasing, punctuation removal, Stopword removal, and Digit filtering.

TF–IDF enabled us to capture word importance across the corpus, which is particularly effective in sparse textual data such as online reviews.

3.4.3. Numerical feature engineering

We extracted the following numerical features from the structured data:

1. **Rating Deviation:** Calculated as the difference between a review's rating and the average rating of its apartment. This measures the extremity of a reviewer's opinion.
2. **Sentiment Score:** Obtained using TextBlob, providing a polarity score in the range [−1, 1]. This helps quantify the emotional tone of each review.
3. **Review Length:** Defined as the number of words in a review. Longer reviews may suggest more detailed and thus potentially more helpful content.

3.5. Feature analysis

To better understand the relationship between features and helpfulness labels, we performed exploratory data analysis on key attributes.

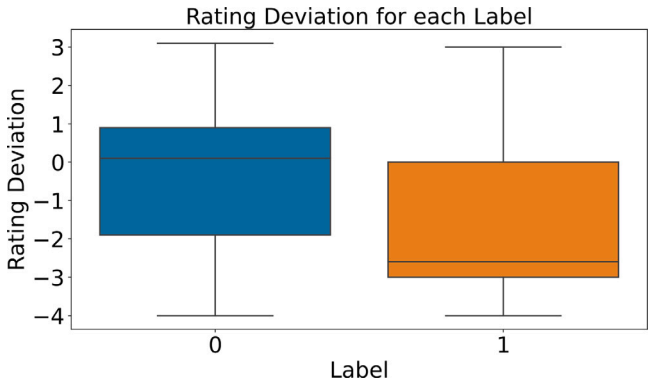


Fig. 6. Boxplot showing rating deviation for each label.

3.5.1. Sentiment distribution

Fig. 5 shows the distribution of the sentiment polarity for each class. Here, helpful reviews seem to be less polarized as the number of neutral reviews is higher in label 1. This indicates that the actual reviewers get both positive and negative experiences from different features of the apartments.

3.5.2. Rating deviation by label

Fig. 6 shows a box plot of rating deviation for the two classes generated by KMeans clustering, where label 0 represents not helpful reviews and label 1 represents helpful reviews. Reviews labeled as not helpful (label 0) have a median rating deviation of approximately 0.1, with the middle 50% of the data (interquartile range) falling between –1.9 and 0.9.

This indicates that these reviews tend to stay close to the average apartment rating. In contrast, helpful reviews (label 1) have a lower median of around –2.6, and their interquartile range spans from –3.0 to 0.0, showing a clear negative deviation from the average rating. This suggests that reviews which express stronger or more critical opinions, especially those that deviate negatively from the consensus, are more likely to be perceived as helpful by others

3.5.3. Review length distribution

Fig. 7 illustrates the length of reviews across helpfulness labels. Reviews in the "Helpful" category tend to be longer, suggesting that more elaborative reviews are associated with higher helpfulness perception.

3.6. Classification models

To assess the predictive power of our engineered features, we employed a **rule-based classifier**, a range of **traditional** supervised classifiers, and **transformer-based deep learning models**. Our goal was to establish robust baselines using interpretable and efficient algorithms before comparing their performance with deep learning models such as BERT. Fig. 8 shows the different classification approaches implemented to validate the usefulness of our proposed dataset and labeling technique.

Model selection

We experimented with the following widely-used classification models:

1. **Rule-Based Classifier:** This interpretable model uses manually derived heuristics, including review length, readability scores, and the presence of specific indicative words, to classify reviews as helpful or not. Thresholds for classification were determined empirically using training data percentiles. This approach provides a baseline and insight into simple textual patterns associated with helpfulness.

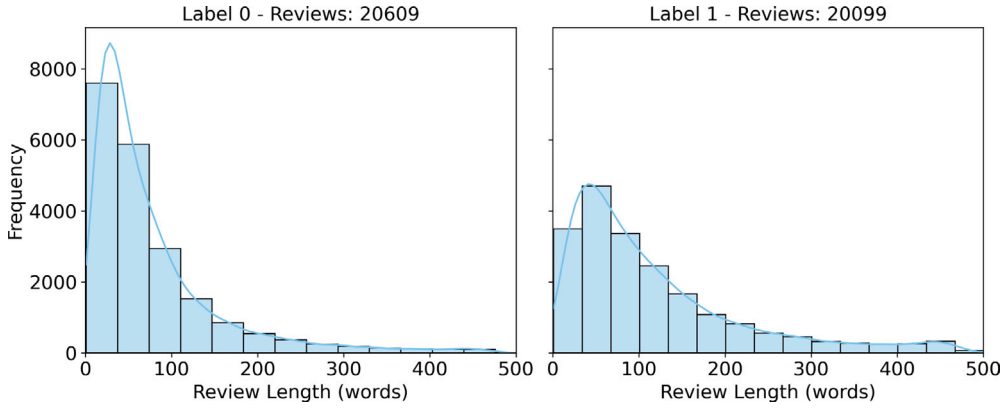


Fig. 7. Review length distribution by helpfulness label.

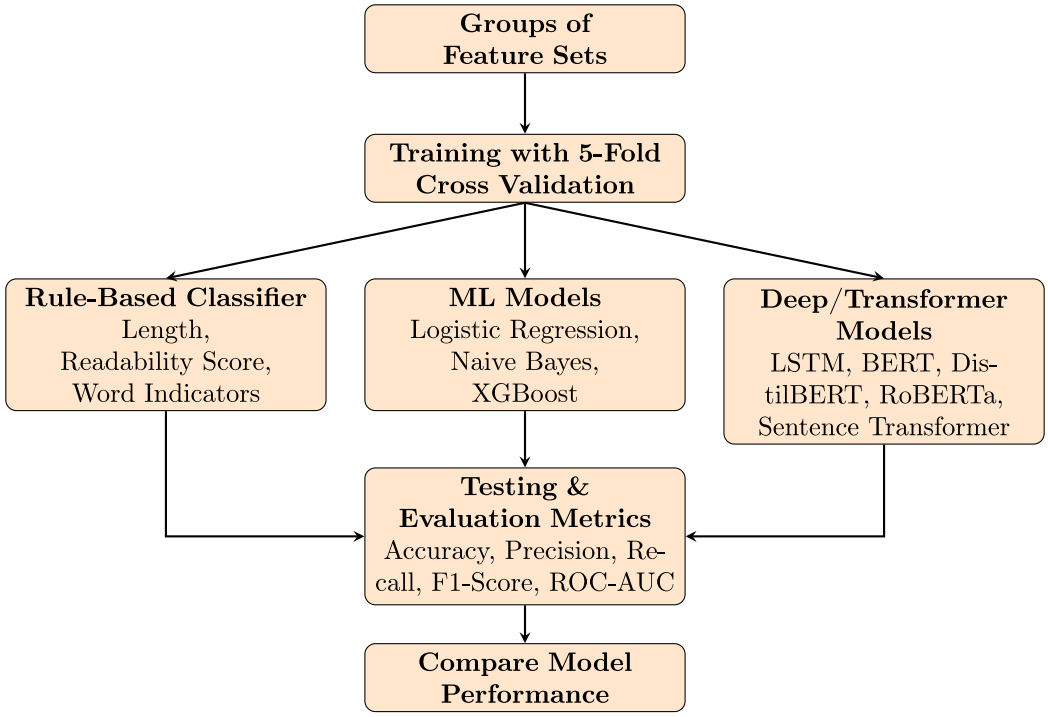


Fig. 8. Implemented classification methods for validation of dataset and labeling technique.

2. **Logistic Regression (LR):** A linear model suitable for binary classification. Chosen for its simplicity, interpretability, and effectiveness on high-dimensional data such as TF–IDF vectors.
3. **Multinomial Naive Bayes (MNB):** MNB models the conditional probability of each feature (word) given the class, assuming independence among features. For a review, it calculates the likelihood of the text under each class and predicts the class with the highest posterior probability. Laplace smoothing is applied to handle zero-frequency words. This approach is efficient for high-dimensional sparse data like TF–IDF and effectively captures the influence of word occurrence patterns on helpfulness prediction. Feature augmentation, such as adding numeric attributes (rating deviation, sentiment polarity), can be incorporated by discretizing or binning continuous values.
4. **Extreme Gradient Boosting (XGBoost):** XGBoost is an ensemble of decision trees trained sequentially using gradient boosting. Each tree corrects the errors of previous trees by minimizing a differentiable loss function, typically logistic loss for classification. Regularization techniques such as L1/L2 penalties, column subsampling, and tree pruning prevent overfitting. For review helpfulness prediction, XGBoost effectively integrates both sparse textual features (TF–IDF) and dense numerical features (rating deviation, sentiment score, review length), capturing non-linear relationships and feature interactions that simpler models cannot.
5. **Deep/Transformer-Based Models:** Sequence models such as LSTM, BERT, DistilBERT, and Sentence Transformers were used to capture semantic and contextual information from review text. For sentence transformers, embeddings were fed into a Support Vector Machine classifier. These models are particularly effective in learning nuanced language patterns that traditional models may miss.

Each model was trained and evaluated on four different feature sets, starting with only the review text and incrementally adding rating

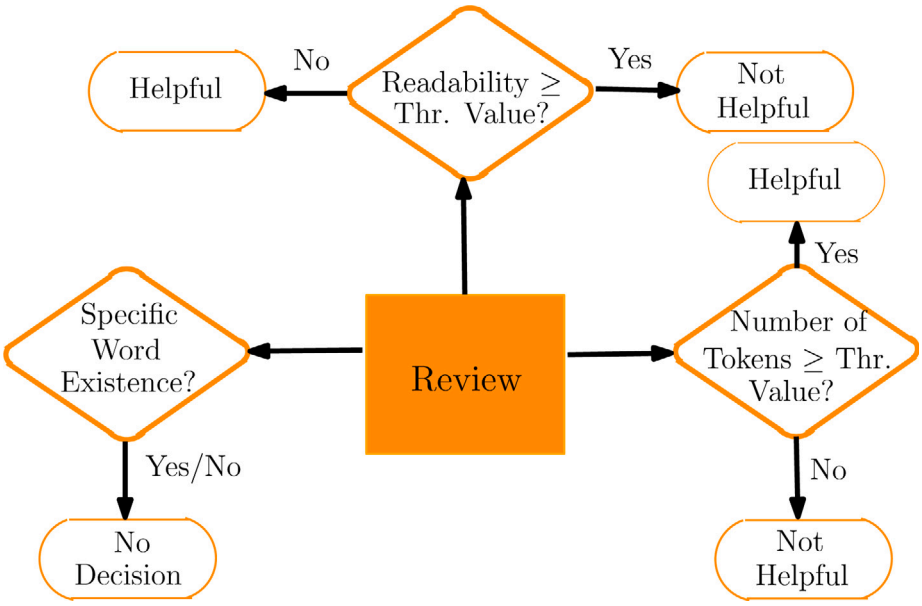


Fig. 9. Rule-based classifier for review helpfulness classification.

deviation, sentiment polarity, and review length. This setup allowed us to systematically analyze the impact of each feature on classification performance.

4. Experimental work and results

4.1. Experimental setup

To evaluate the performance of our proposed framework, we split the dataset into an **80:20 ratio**, where 80% of the data was used for training and 20% for testing. The experiments were conducted on a Computer with the following hardware specifications:

- **Processor:** 12th Gen Intel(R) Core(TM) i9-12900H @ 2.50 GHz
- **Installed RAM:** 16.0 GB
- **System Type:** 64-bit Windows 11 Operating System, x64-based Processor

4.2. Evaluation metrics

The performance of the classifiers was evaluated using standard metrics that quantify classification effectiveness which are accuracy, precision, recall, F1-score, the area under the Receiver Operating Characteristic Curve (ROC-AUC). Because the dataset is balanced, trivial baselines such as predicting the majority class achieve only 50% accuracy with negligible F1-score. The use of F1 in our evaluation provides an inherent comparison with such baselines. Vote-based heuristics were not considered as prediction baselines because vote counts are unavailable for new reviews, which are the primary focus of the helpfulness prediction task.

4.3. Performance evaluation of rule-based classifiers

This section presents the performance of the rule-based classifier based on the analysis of our dataset. We design some rule-based metrics based on the manual analysis of statistically significant random sample of our dataset (see Fig. 9).

Table 4
Classification performance metrics for rule-based classifier.

Metric	Accuracy	Precision	Recall	F1-Score
Review length	**0.60**	**0.61**	**0.60**	**0.60**
Readability	0.54	0.55	0.54	0.52

4.3.1. Rule-based metrics

Review Length: The review length, meaning the number of tokens in the review to get the helpfulness measurement of the particular review. We use 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th percentiles of review length as the partition of two classes. 60th percentile review length is better in terms of accuracy in the training dataset.

Readability Score: The readability score of any particular review is measured by the Flesch readability ease (Flesch & Gould, 1949) score. Similar to the review length, we run experiments for different percentiles and 30th percentile is better in comparison to others.

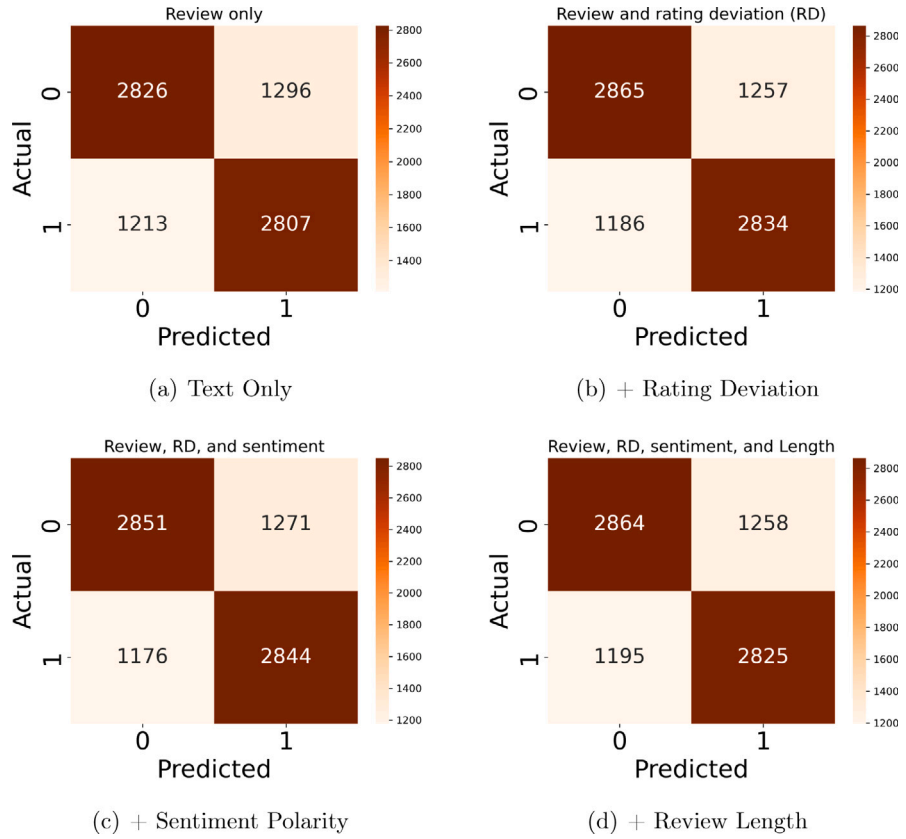
Specific Word Existence: In sentiment analysis of any review, there are certain words to indicate the sentiment. Different to sentiment analysis, there is no specific set of words which are the indicators of helpfulness.

Based on the length of the review, the classifier gets the threshold value from analyzing the training data and the performance in the validation data as follows (see Table 4):

4.4. Parameters tuning for classifiers

To ensure optimal performance of our classification models, we employed Grid Search with **5-fold Cross-Validation** on the training data to tune hyperparameters for each algorithm. This method exhaustively searches over specified parameter values and evaluates model performance using cross-validation, thus ensuring generalizability and robustness of the selected models. The search space for each model was defined based on prior studies and empirical testing, with attention to computational efficiency and predictive accuracy.

For Logistic Regression, we tuned the inverse regularization strength (`C`), penalty type, optimization algorithm, and maximum number of iterations. The best performance was obtained using: `C = 0.5`, `penalty = 'l2'`, `solver = 'lbfgs'`, and `max_iter = 500`.



(a) Text Only (b) + Rating Deviation

(c) + Sentiment Polarity (d) + Review Length

Fig. 10. Confusion matrices for logistic regression across different feature sets.

For the XGBoost classifier, we tuned critical hyperparameters that influence the depth and generalization ability of the ensemble trees: `n_estimators` = 100, `max_depth` = 7, `learning_rate` = 0.1, and `subsample` = 0.8.

These settings provide a good trade-off between model complexity and overfitting, particularly for medium-sized tabular datasets with high-dimensional textual features.

For the Naive Bayes classifier, we focused on smoothing and class prior learning:`alpha` = 0.5 (Laplace smoothing) and `fit_prior` = `False` (assumes equal class priors)

The optimal configurations identified were subsequently used in the final model training and testing phases.

4.5. Performance evaluation of ML classifiers

This section presents a comparative evaluation of three supervised classifiers—Logistic Regression (LR), XGBoost (XGB), and Multinomial Naive Bayes (NB)—on the task of estimating the helpfulness of apartment reviews. The models were trained and tested using progressively enriched feature sets, starting from review text only, and incrementally incorporating rating deviation, sentiment polarity, and review length. The evaluation metrics include accuracy, F1-score, and the area under the ROC curve (AUC), providing a holistic view of each model's classification performance.

Fig. 10 presents the confusion matrices for Logistic Regression across the four feature configurations. As additional features are incorporated, we observe minor but consistent improvements in true positive (TP) and true negative (TN) rates, particularly when rating deviation and sentiment polarity are included. For instance, the TP count increases from 2807 to 2844 when sentiment is added. However, the final configuration with all features shows a slight trade-off—while TN remains high (2864), the TP drops slightly to 2825, indicating a marginal increase in false negatives. The addition of review length did not lead to further improvement, suggesting diminishing returns for LR beyond sentiment analysis.

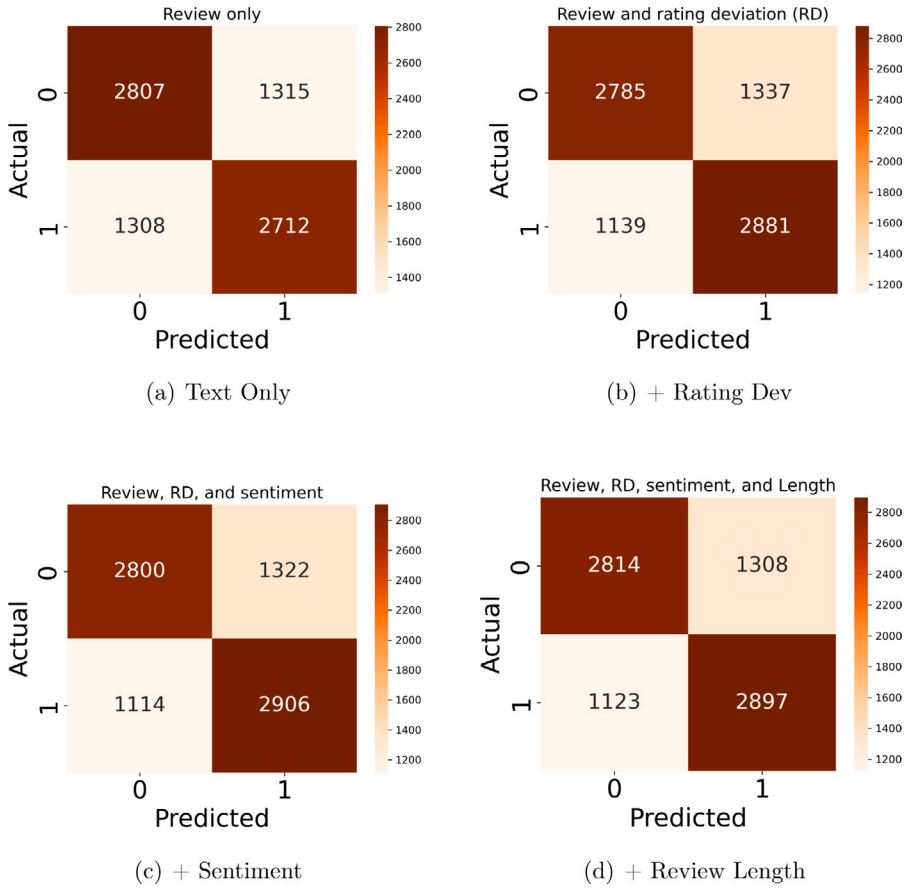
Fig. 11 shows the confusion matrices for XGBoost across the four feature configurations. Unlike LR, XGBoost exhibits significant gains with the addition of structured features. When only review text is used, the true positive (TP) count is relatively low at 2712. However, incorporating rating deviation increases TP to 2881, and the inclusion of sentiment polarity pushes it to a peak of 2906—the highest TP count across all models and configurations. While the final configuration with review length shows a slight dip in TP (2897), true negatives (TN) steadily rise, reaching 2814. This suggests that XGBoost benefits from both text and numerical features, particularly sentiment, with only marginal trade-offs when additional features are introduced.

Fig. 12 presents the confusion matrices for Naive Bayes. Compared to the other models, NB shows relatively stable but stagnant performance across all feature sets. The TP count starts at 2790 and fluctuates minimally, ending at 2770 even after all features are added. Similarly, TN sees only a modest increase from 2809 to 2856. The inclusion of rating deviation, sentiment, and review length offers little to no benefit, indicating that NB is limited in its ability to exploit numerical features due to its strong independence assumptions and simplistic modeling.

Table 5 shows the performance matrices for each classifiers with different feature sets.

4.5.1. Performance with review text only

Using only the vectorized review text (TF–IDF feature), all three models, LR, XGB, and NB, achieved an identical accuracy of 0.69. The F1 scores for each model matched their accuracy, reflecting balanced precision and recall. ROC-AUC values were also similar, with LR and NB reaching 0.74, and XGB close behind at 0.73. These findings underscore the strong predictive capacity of textual features alone, establishing a solid baseline for subsequent enhancements.



(a) Text Only (b) + Rating Dev

(c) + Sentiment (d) + Review Length

Fig. 11. Confusion matrices for XGBoost classifier across different feature sets.

Table 5
Performance comparison of classifiers across feature sets.

Model	Feature set	Accuracy	Precision	Recall	F1-Score	ROC-AUC
LR	Review text only	0.69	0.69	0.69	0.69	0.74
	+ Rating deviation	0.70	0.70	0.70	0.70	0.74
	+ Rating deviation + Sentiment	0.70	0.70	0.70	0.70	0.74
	+ Rating deviation + Sentiment + Review length	0.70	0.70	0.70	0.70	0.74
XGBoost	Review text only	0.69	0.69	0.69	0.69	0.73
	+ Rating deviation	0.70	0.70	0.70	0.70	0.75
	+ Rating deviation + Sentiment	0.70	0.71	0.71	0.71	**0.75**
	+ Rating deviation + Sentiment + Review length	0.71	0.71	0.71	0.71	**0.75**
Naive Bayes	Review text only	0.69	0.69	0.69	0.69	0.74
	+ Rating deviation	0.69	0.69	0.69	0.69	0.73
	+ Rating deviation + Sentiment	0.69	0.69	0.69	0.69	0.73
	+ Rating deviation + Sentiment + Review length	0.69	0.69	0.69	0.69	0.73

4.5.2. Adding rating deviation

Introducing the rating deviation feature, defined as the difference between the user-provided rating and the average property rating, yielded modest improvements. LR and XGB both increased in accuracy to 0.70, with corresponding ROC-AUC values rising to 0.74 and 0.75, respectively. The F1 scores tracked these improvements. However, NB remained at 0.69 accuracy and 0.73 AUC, unchanged from the previous configuration. This limited responsiveness highlights NB's reduced ability to model feature interactions between sparse text and dense numeric values.

4.5.3. Inclusion of sentiment polarity

Adding sentiment polarity, computed using TextBlob (Loria, 2018), maintained the previous accuracy levels for LR and XGB at 0.70 and 0.71, respectively. ROC-AUC for LR remained stable at 0.74, while XGB showed no further improvement, staying at 0.75. Again, NB showed no change in performance. These results suggest that sentiment polarity can provide incremental benefit when combined with more expressive models like XGB but may not influence simpler models constrained by strong independence assumptions.

4.5.4. Adding review length

Finally, review length in tokens was incorporated to account for verbosity, a feature that is often correlated with perceived review helpfulness. XGBoost continued to lead, achieving its peak ROC-AUC of 0.75 and maintaining an accuracy of 0.71. Logistic Regression metrics remained stable across feature sets, and Naive Bayes showed no change, with accuracy and AUC staying at 0.69 and 0.73, respectively. In our experiments, adding review length did not provide additional predictive benefit beyond what the models might have already captured from the textual representations. This result does not imply that length is irrelevant to helpfulness; rather, it suggests that the models and feature

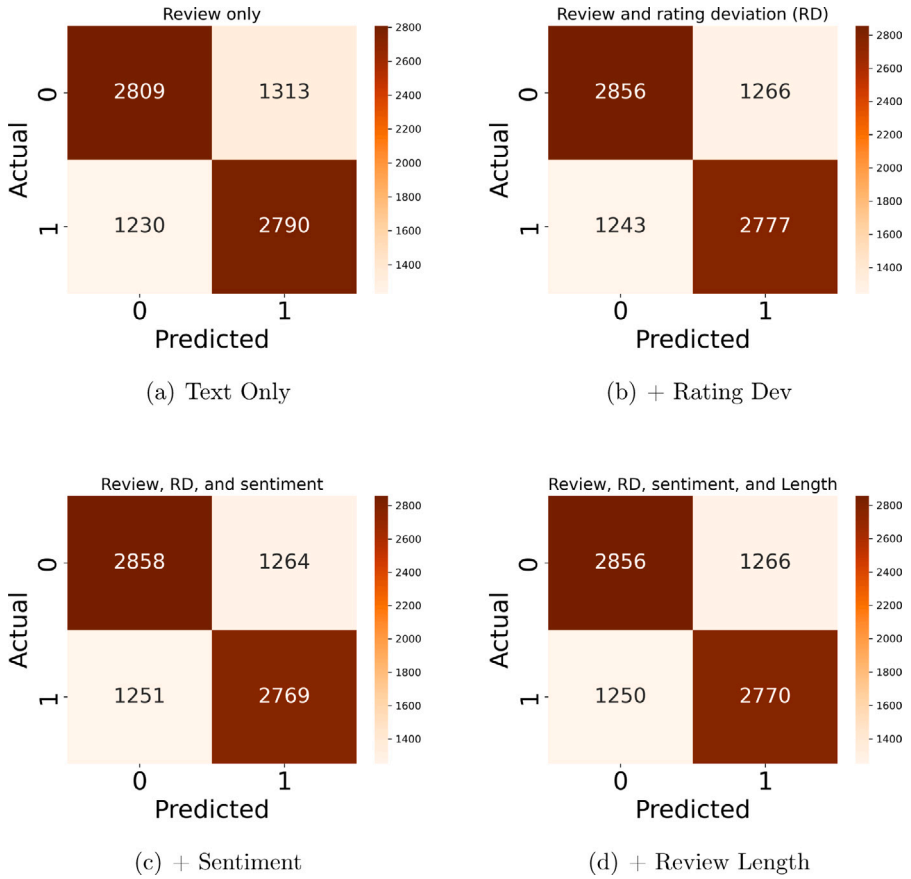


Fig. 12. Confusion matrices for Naive Bayes classifier across different feature sets.

space used here may already encode length-related cues, or may not be able to exploit this feature further in this setting.

4.5.5. Statistical assessment of added features

To determine whether the observed performance differences represented meaningful improvements, we applied McNemar's test on paired predictions. The full feature set in XGBoost demonstrated a statistically significant improvement over the text-only model ($\chi^2 = 5.56$, $p = 0.018$), indicating that the observed increase in accuracy reflects a consistent performance gain. For Logistic Regression, the inclusion of the rating deviation feature likewise resulted in a statistically significant improvement over the baseline text-only configuration ($\chi^2 = 7.49$, $p = 0.0062$).

Although the corresponding p-values are modest and the effect sizes are small, this behavior is expected given the balanced dataset and the subtle nature of helpfulness cues, where even incremental gains require correcting non-trivial borderline cases. In contrast, for Naive Bayes, incorporating additional features did not yield performance advantages, which is consistent with the model's strong dependence on unigram likelihoods and its limited ability to exploit non-textual features. Overall, these statistically supported results highlight the value of incorporating structured behavioral features alongside textual information, even when the observable gains are numerically modest.

4.6. ROC curve analysis

Figs. 13, 14, 15 show the ROC curves for the three models in the different feature sets. These curves reflect the classifiers' ability to balance sensitivity and specificity. Notably, XGBoost with the full feature set demonstrates a marked advantage, particularly in high-recall regions, consistent with its highest ROC-AUC of 0.75. Logistic Regression maintains stable curves across configurations, while Naive Bayes displays minimal deviation, reinforcing the numerical trends.

XGBoost consistently benefitted from each feature augmentation, particularly from numeric attributes like rating deviation and sentiment score due to its gradient boosting architecture. Logistic Regression demonstrated resilience and adaptability across all configurations. In contrast, Naive Bayes, though efficient and effective on pure text features, failed to exploit the added value of non-textual data, constrained by its simplifying assumptions.

4.7. Performance evaluation of deep learning and transformer based classifiers

This section is centered around the classification result of different transformer based sequence classification and deep learning classifiers. The representatives of the transformer-based models are BERT, Roberta, DistilBERT, Sentence-Transformer and we use LSTM based model to classify the helpfulness on the basis of the textual content of the reviews. In addition to the textual content of the reviews, we leverage additional attributes such as the spatial information of the reviews. For BERT, Roberta and DistilBERT, we use **bert-base-uncased**, **roberta-base**, and **distilbert-base-uncased** respectively. Furthermore, we experiment with some variants of Sentence Transformers including **all-mpnet-base-v2**, **multi-qa-mpnet-base-dot-v1**, and **all-distilroberta-v1**. Specifically, for the sentence transformer, we use the context embedding generated by the transformers and leverage Support Vector Machine for classification. Fig. 16(a) demonstrates the impact on accuracy with changing regularization parameter, C and different kernels for support vector machine. The tuning for regularization parameter withing a specific range is necessary to avoid model complexity leading to overfitting. It shows that SVC

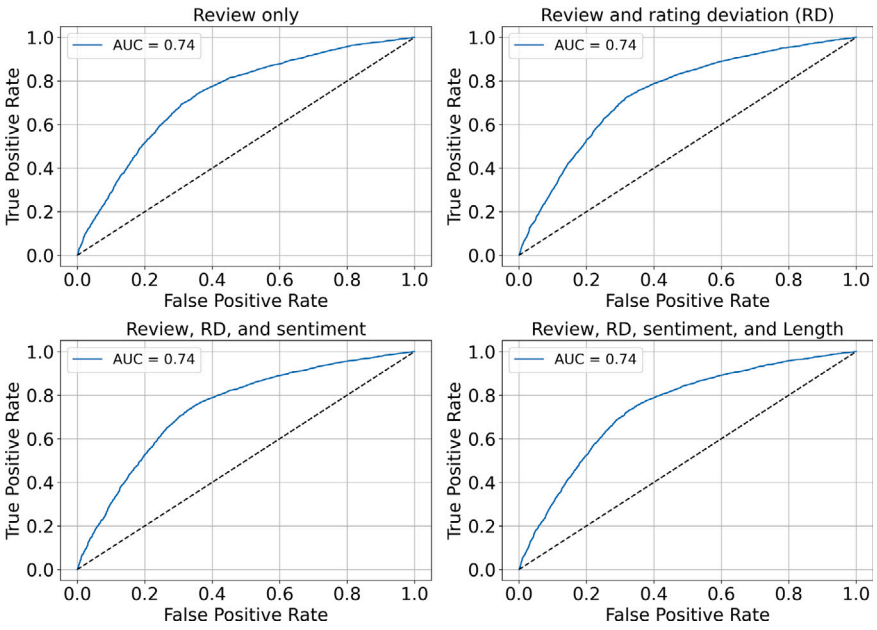


Fig. 13. Logistic regression classifier's ROC curves for different feature sets.

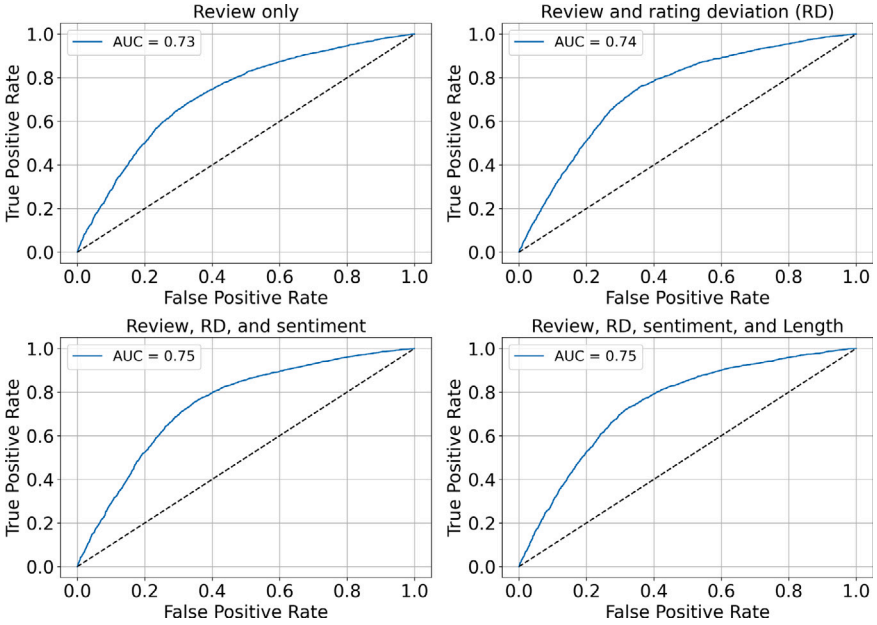


Fig. 14. XGBoost classifier's ROC curves for different feature sets.

Table 6
Performance comparison of The DL and transformer-based classifiers.

Model	Feature set	Accuracy	Precision	Recall	F1-Score	ROC-AUC
BERT (bert-base-uncased)	Review text only	0.69	0.69	0.69	**0.69**	0.71
	+ Additional features	0.68	0.68	0.68	0.68	0.71
Roberta (roberta-base)	Review text only	0.69	0.69	0.69	0.69	0.71
DistilBERT (distilbert-base-uncased)	Review text only	0.51	0.25	0.50	0.34	0.51
Sen. Trans. (all-mpnet-base-v2)	Review text only	**0.70**	0.69	**0.70**	**0.69**	**0.74**
Sen. Trans. (multi-qa-mpnet-base-dot-v1)	Review text only	**0.70**	**0.70**	0.69	**0.69**	**0.74**
Sen. Trans. (all-distilroberta-v1)	Review text only	**0.70**	0.69	**0.70**	**0.69**	**0.74**
LSTM	Review text only	0.67	0.68	0.67	0.67	0.70

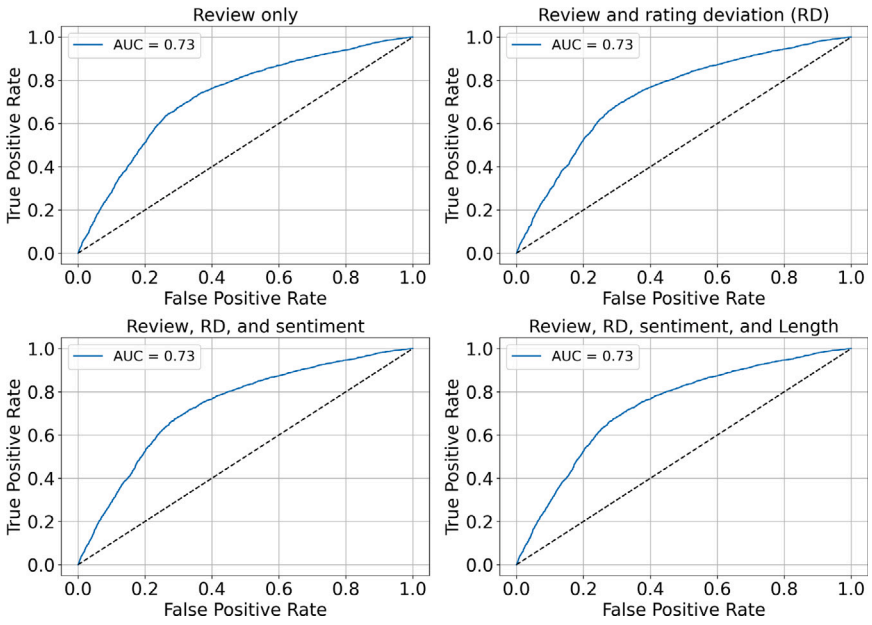
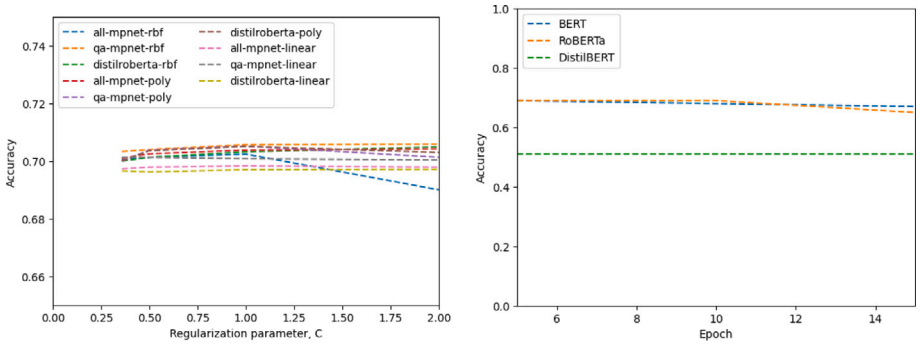


Fig. 15. Naive Bayes classifier's ROC curves for different feature sets.



(a) Accuracy vs. SVC configuration (b) Accuracy vs. number of epochs.

Fig. 16. Accuracy of transformer based models with respect to different SVC configurations and number of epochs.

with regularization parameter, C = 1, we get better accuracy (0.7059) for **multi-qa-mpnet-base-dot-v1** sentence transformer embedding. In addition to the default transformer settings, we conducted an extended series of fine-tuning experiments to ensure that the transformer-based classifiers were properly optimized for the review helpfulness prediction task. For the BERT, RoBERTa, and DistilBERT models, we explored multiple configurations of learning rate, batch size, warmup steps, weight decay, and training epochs (5, 10, and 20). Fig. 16(b) exhibits the impact of the number of epochs on the accuracy. In case of the transformer based models, we get better accuracy at 10 epochs for the corresponding models considering training time and dataset size. Although the additional tuning stabilized model convergence, the overall performance improvements remained modest, suggesting that the limited and subtle helpfulness cues present in short renter reviews may not be fully captured by general-purpose pretrained transformers. In You et al. (2019), it shows that the increased batch size with the appropriate learning rate tends to improve the model's performance. Considering the increased memory footprint for GPU, the batch size is set to 16 and the weight decay is set to 0.01. For the DistilBERT model (Sanh et al., 2019), the accuracy is reduced because of architectural simplicity and it is less suited for deep contextual understanding. Though it is good for less training time, the accuracy is lower compared to the other models.

According to Table 6, among all the classifiers, the sentence transformer outperforms all other classifiers in terms of accuracy. Fig. 17 demonstrates the ROC-AUC for different transformer based models making inference depending on textual reviews.

4.8. Model performance insights

Our experimental analysis reveals distinct strengths and limitations for each classification approach. The rule-based classifier is highly interpretable and requires no training, offering insights into simple textual patterns such as review length and readability; however, it achieves limited predictive performance (accuracy up to 0.60) and cannot capture complex feature interactions. Logistic Regression demonstrates stability across feature augmentations, is effective with high-dimensional TF–IDF vectors, and is resilient to overfitting, yet it cannot model non-linear interactions and shows diminishing returns when numeric or contextual features are added. Multinomial Naive Bayes is computationally efficient and performs well with sparse textual data, making it a strong baseline for text classification, but its assumption of feature independence prevents it from leveraging numeric or correlated features, limiting improvement beyond an accuracy of 0.69. XGBoost captures non-linear relationships and feature interactions, benefits from incremental feature augmentation, and achieves the highest ROC-AUC

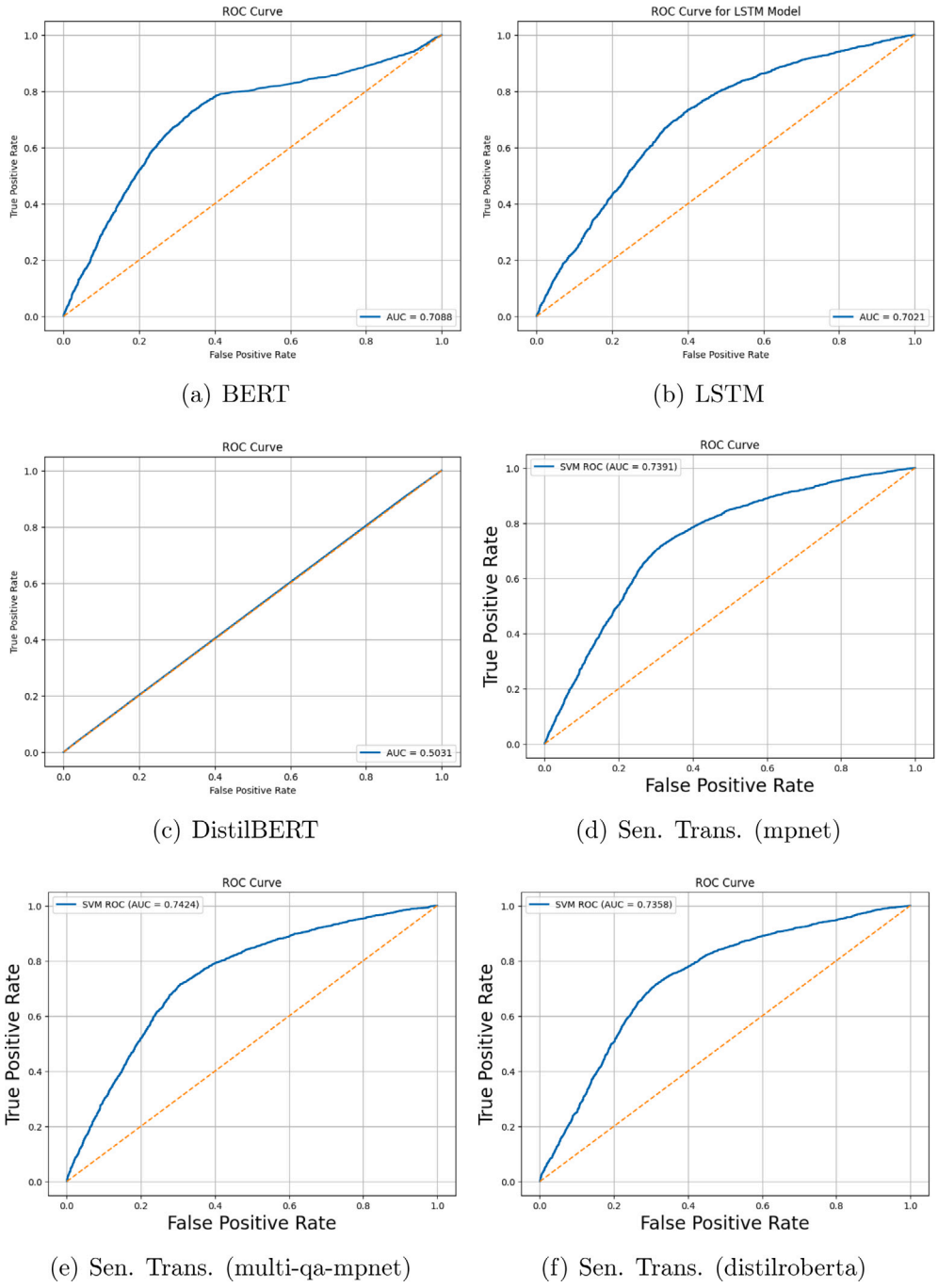


Fig. 17. ROC curves for different Transformer-based models' with text reviews.

of 0.75; its drawbacks include higher computational cost, reduced interpretability, and sensitivity to hyperparameter tuning. Deep learning and transformer-based models such as LSTM, BERT, DistilBERT, and Sentence Transformer excel at capturing rich semantic and contextual information from review text, with the Sentence Transformer achieving the highest accuracy of 0.70 among deep models. However, these models are computationally intensive, require careful fine-tuning, and some models, including DistilBERT, underperformed when using only review text (accuracy 0.51), while numeric and contextual features provided limited additional benefit. Overall, these insights highlight the trade-offs between interpretability, computational efficiency, and predictive performance across different model families.

5. Conclusion and future work

In this study, we introduce a comprehensive framework for predicting the helpfulness of rental reviews using a novel dataset curated from Apartments.com. A key contribution of this work is the development of an interpretable labeling strategy to label into two categories based on the helpfulness vote count and timeliness of the review, avoiding biases introduced by subjective content-based assumptions.

We evaluated three categories of classifiers across multiple feature configurations. Results show that while the textual content (captured through TF–IDF) is highly informative, adding numerical features such as rating deviation and sentiment polarity incrementally improved the

models' F1-scores and ROC-AUC values. XGBoost outperforms other models slightly, achieving the highest accuracy and AUC, indicating its strength in handling heterogeneous feature sets. We also run extensive experiments leveraging LSTM and Transformer-based models to classify based on the review content. We also try to blend some exogenous attributes and review content for the Transformer-Based Models. However, all models showed relatively close performance, suggesting that our feature engineering and labeling methodology maintained consistency across algorithms.

In future research, the proposed framework can be extended by incorporating additional behavioral and contextual features such as user profile characteristics, review–response interactions, and apartment-level metadata. In recent times, reviews often contain audio and images in addition to textual content. Leveraging multimodal models to capture the relationships among text, audio, and visual cues with helpfulness is therefore a promising direction for research. The labeling strategy may also be refined by integrating user credibility metrics or temporal dynamics beyond simple recency. Furthermore, a hybrid labeling approach that combines automated clustering with limited human validation could enhance label quality and support the development of explainable AI methods. Such human–machine collaboration may provide deeper insights into why certain reviews are perceived as helpful and improve the robustness of future helpfulness prediction systems.

CRediT authorship contribution statement

Rakibul Hassan: Conceptualization, Dataset preparation, Methodology, Implementation, Formal analysis, Writing – original draft, Visualization. **Shubhashish Kar:** Dataset preparation, Methodology, Implementation, Investigation, Validation, Writing – original draft, Visualization. **Jorge Fonseca Cacho:** Resources, Data acquisition, Writing – review & editing. **Shaikh Arifuzzaman:** Supervision, Project administration, Writing – review & editing.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the author(s) used ChatGPT (OpenAI, GPT-4.5) to assist with language refinement and formatting suggestions. After using this tool, the author(s) thoroughly reviewed and edited the content to ensure accuracy, originality, and adherence to academic standards. The author(s) take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Alsmadi, A., AlZu'bi, S., Al-Ayyoub, M., & Jararweh, Y. (2020). Predicting helpfulness of online reviews. arXiv preprint arXiv:2008.10129.

Apartments. com (2025). Apartment features renters want most. https://www.apartments.com/blog/apartment-features-renters-want-most. (Accessed 3 April 2025).

Cheng, Y.-H., & Ho, H.-Y. (2015). Social influence's impact on reader perceptions of online reviews. *Journal of Business Research*, *68*(4), 883–887.

Choi, H. S., & Leon, S. (2023). When trust cues help helpfulness: investigating the effect of trust cues on online review helpfulness using big data survey based on the amazon platform. *Electronic Commerce Research*, 1–28.

Fang, B., Ye, Q., Kucukusta, D., & Law, R. (2016). Analysis of the perceived value of online tourism reviews: Influence of readability and reviewer characteristics. *Tourism Management*, *52*, 498–506.

Flesch, R. F., & Gould, A. J. (1949). The art of readable writing. (No Title).

Guo, X., Chen, G., Wang, C., Wei, Q., & Zhang, Z. (2021). Calibration of voting-based helpfulness measurement for online reviews: an iterative bayesian probability approach. *INFORMS Journal on Computing*, *33*(1), 246–261.

Hananto, V. R., Serdült, U., & Kryssanov, V. (2022). A text segmentation approach for automated annotation of online customer reviews, based on topic modeling. *Applied Sciences*, *12*(7), 3412.

Hassan, N., & Islam, M. S. (2019). Fake review detection using semi-supervised and supervised learning techniques. *Expert Systems with Applications*, *125*, 156–171.

Hassan, R., & Islam, M. R. (2020). A supervised machine learning approach to detect fake online reviews. In *2020 23rd international conference on computer and information technology* (pp. 1–6).

Hassan, R., & Islam, M. R. (2021). Impact of sentiment analysis in fake online review detection. In *2021 international conference on information and communication technology for sustainable development* (pp. 21–24).

Hendrawan, I. R., Utami, E., & Hartanto, A. D. (2022). Comparison of naïve bayes algorithm and XGBoost on local product review text classification. *Edumatic: Jurnal Pendidikan Informatika*, *6*(1), 143–149.

Hu, Y.-H., & Chen, K. (2016). Predicting hotel review helpfulness: The impact of review visibility, and interaction between hotel stars and review ratings. *International Journal of Information Management*, *36*(6), 929–944.

Jindal, N., & Liu, B. (2007). Analyzing and detecting review spam. In *Seventh IEEE international conference on data mining* (pp. 547–552).

Karimi, S., & Wang, F. (2017). Online review helpfulness: Impact of reviewer profile image. *Decision Support Systems*, *96*, 39–48.

Kim, S.-M., Pantel, P., Chklovski, T., & Pennacchiotti, M. (2006). Automatically assessing review helpfulness. In *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 423–430).

Krishnamoorthy, S. (2015). Linguistic features for review helpfulness prediction. *Expert Systems with Applications*, *42*(7), 3751–3759.

Kwok, L., & Xie, K. L. (2016). Factors contributing to the helpfulness of online hotel reviews: does manager response play a role? *International Journal of Contemporary Hospitality Management*, *28*(10), 2156–2177.

Lee, P.-J., Hu, Y.-H., & Lu, K.-T. (2018). Assessing the helpfulness of online hotel reviews: A classification-based approach. *Telematics and Informatics*, *35*(2), 436–445.

Lee, S., Lee, S., & Baek, H. (2021). Does the dispersion of online review ratings affect review helpfulness? *Computers in Human Behavior*, *117*, Article 106670.

Li, L., Goh, T.-T., & Jin, D. (2020). How textual quality of online reviews affect classification performance: a case of deep learning sentiment analysis. *Neural Computing and Applications*, *32*, 4387–4415.

Li, M., Huang, L., Tan, C.-H., & Wei, K.-K. (2013). Helpfulness of online product reviews as seen by consumers: Source and content features. *International Journal of Electronic Commerce*, *17*(4), 101–136.

Liu, Y., Huang, X., An, A., & Yu, X. (2008). Modeling and predicting the helpfulness of online reviews. In *2008 eighth IEEE international conference on data mining* (pp. 443–452). IEEE.

Loria, S. (2018). TextBlob: Simplified text processing. https://textblob.readthedocs.io/. Version 0.15.3.

Lu, Y., Tsaparas, P., Ntoulas, A., & Polanyi, L. (2010). Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on world wide web* (pp. 691–700).

Malik, M., & Hussain, A. (2018). An analysis of review content and reviewer variables that contribute to review helpfulness. *Information Processing & Management*, *54*(1), 88–104.

Martin, L., & Pu, P. (2014). Prediction of helpful reviews using emotions extraction. *vol. 28*, In *Proceedings of the AAAI conference on artificial intelligence*.

Ott, M., Cardie, C., & Hancock, J. T. (2013). Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies* (pp. 497–501).

Pajola, L., Chen, D., Conti, M., & Subrahmanian, V. (2023). A novel review helpfulness measure based on the user-review-item paradigm. *ACM Transactions on the Web*, *17*(4), 1–31.

Rathor, A. S., Agarwal, A., & Dimri, P. (2018). Comparative study of machine learning approaches for Amazon reviews. *Procedia Computer Science*, *132*, 1552–1561.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

SatisFacts (2025). 50 renter stats every multifamily professional needs to know. https://www.satisfacts.com/50-renter-stats-every-multifamily-professional-needs-to-know/. (Accessed 3 April 2025).

Tang, J., Gao, H., Hu, X., & Liu, H. (2013). Context-aware review helpfulness rating prediction. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 1–8).

Willemsen, L. M., Neijens, P. C., Bronner, F., & De Ridder, J. A. (2011). "Highly recommended!" the content characteristics and perceived usefulness of online consumer reviews. *Journal of Computer-Mediated Communication*, *17*(1), 19–38.

Yang, S.-B., Shin, S.-H., Joun, Y., & Koo, C. (2017). Exploring the comparative importance of online hotel reviews' heuristic attributes in review helpfulness: a conjoint analysis approach. *Journal of Travel & Tourism Marketing*, *34*(7), 963–985.

Yang, Y., Yan, Y., Qiu, M., & Bao, F. (2015). Semantic analysis and helpfulness prediction of text for online product reviews. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: short papers)* (pp. 38–44).

You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., Song, X., Demmel, J., Keutzer, K., & Hsieh, C.-J. (2019). Large batch optimization for deep learning: Training bert in 76 minutes. arXiv preprint arXiv:1904.00962.

Zeng, Y.-C., Ku, T., Wu, S.-H., Chen, L.-P., & Chen, G.-D. (2014). Modeling the helpful opinion mining of online consumer reviews as a classification problem. In *International journal of computational linguistics & Chinese language processing, volume 19, number 2, June 2014*.

Zheng, X., Zhu, S., & Lin, Z. (2013). Capturing the essence of word-of-mouth for social commerce: Assessing the quality of online e-commerce reviews by a semi-supervised approach. *Decision Support Systems*, *56*, 211–222.

Zhu, L., Yin, G., & He, W. (2014). Is this opinion leader's review useful? Peripheral cues for online review helpfulness. *Journal of Electronic Commerce Research*, *15*(4), 267.