Uncovering Hidden Bias: Rating-Sentiment Discrepancies in Consumer Reviews

Sanah Sarin & Serena Wong
University of Waterloo, CS 680
s6sarin@uwaterloo.ca , s257wong@uwaterloo.ca

ABSTRACT
This study examines hidden biases in consumer reviews by analyzing discrepancies between numerical ratings and textual sentiment across business categories. We train multiple sentiment classification models, achieving accuracies of 62–66%, and use their outputs to quantify rating sentiment mismatches at both the review and category level. Our analysis reveals that certain cuisines and business types systematically deviate from expected sentiment patterns, indicating behavioral and contextual biases in user feedback. These findings provide insight into how review platforms may better detect inconsistent ratings and design fairer recommendation and moderation systems.

1 Problem Statement

Although star ratings are frequently used to assess restaurants, they frequently miss the subtleties that are conveyed in written reviews. User error, sarcasm, cultural differences, or different expectations can all cause mismatches between text sentiment and rating. Customers and businesses alike rely heavily on ratings, and these discrepancies damage their credibility. The gap between text and rating is understudied in current sentiment analysis techniques, which usually concentrate on either one separately. By creating models that identify inconsistencies and reveal the underlying themes that lead to rating–review discrepancies, this project tackles this issue.

2 Introduction

Online reviews have become one of the most influential factors shaping consumer decision-making, particularly in the hospitality industry. Platforms such as Google, Yelp, and TripAdvisor provide numerical star ratings alongside written reviews, giving customers the space to describe their experiences in detail. However, these two forms of feedback do not always align. A reviewer may assign a low rating while expressing mostly positive sentiment, or provide a high rating despite outlining several complaints. Such inconsistencies raise questions about the reliability of ratings and complicate how businesses interpret customer feedback.

This project investigates mismatches by analyzing the relationship between textual sentiment and numerical star ratings in Yelp restaurant reviews. After constructing a cleaned subset of the dataset, we extract structured features such as cuisine type and business categories, and apply pretrained transformer models to derive both overall sentiment and aspect-based sentiment for *food*, *atmosphere* and *service*.

Using these processed inputs, we train both classical and deep learning models to predict star ratings from text and metadata. Classical approaches include Support Vector Machines and Logistic Regression with TF–IDF representations, while deep learning models incorporate CNN and LSTM architectures with embedding layers and engineered linguistic features. Across these methods, the models reach predictive accuracies in the 64-66% range. Comparing predicted ratings with true ratings enables us to identify reviews in which sentiment and rating diverge, highlighting systematic patterns across cuisines, restaurant types, and individual businesses.

Overall, this work combines natural language processing, machine learning, and linguistic feature engineering to better understand the gap between what consumers say and the ratings they assign. The insights contribute to improved recommendation systems, more reliable business intelligence, and a deeper understanding of consumer communication behaviour in online review environments.

3 Background

A number of studies have applied sentiment analysis and machine learning to predict user ratings across different domains. For example, Aralikatte et al. (2018) analyzed Android app reviews using features such as sentiment polarity, readability scores, review length, and punctuation. They tested both traditional machine learning and deep learning methods, finding that a Dependency-based Convolutional Neural Network achieved the highest accuracy of 92% on a mix of public and survey datasets. Similarly, Shrestha and Nasoz (2019) investigated Amazon product reviews, representing text as vectors and combining deep learning with Support Vector Machine (SVM) based sentiment classification. Their Gated Recurrent Unit (GRU) model achieved 81.3% accuracy. More recently, Akinlaja and Mosia (2021) examined Coursera course reviews, applying text preprocessing and vectorization before training a Convolutional Neural Network, which yielded 87% accuracy.

Across these studies, the general approach has been consistent: sentiment extraction from text reviews followed by rating prediction using machine learning or deep learning. Reported performance has generally ranged from 81% to 92% accuracy. However, these efforts have focused on product and educational domains. There remains a gap in extending these methods to restaurant reviews on platforms such as Google Reviews and Yelp, which are both widely used and linguistically distinct. Restaurant reviews tend to be shorter, more colloquial, and highly context-dependent (e.g., service speed, atmosphere, and regional food terms). These characteristics may challenge models designed for the more descriptive style of product or course reviews, making this an important area for further study.

4 Methodology

4.1 Data

To analyze restaurant reviews and rating mismatches, we use predictive modelling techniques grounded in Natural Language Processing (NLP), machine learning, and deep learning. Our data originates from the publicly available Yelp Open Dataset, which contains approximately 6.9 million reviews from businesses across eight metropolitan areas in the United States and Canada. Although our initial plan included using both Yelp and Google Reviews, we ultimately restricted our analysis to Yelp due to its cleaner structure, comprehensive metadata, and more standardized review format. All data is available for academic and non-commercial use under the dataset license provided by the University of California San Diego and Yelp (2021).

Since the original dataset is extremely large, we applied a series of preprocessing and filtering steps, including removing non-English reviews, eliminating businesses with insufficient review volume, and extracting only the fields relevant to our models (text, star ratings, and business-level attributes). After preprocessing, our final working dataset consists of around 42,000 reviews, which we use consistently across all prediction tasks and mismatch analyses.

4.2 Data Cleaning and Processing Business Data

To prepare the business dataset for analysis, we performed a comprehensive cleaning and feature–engineering pipeline. All attribute dictionaries stored as strings in the raw CSV were safely parsed into Python objects, and inconsistent formats were normalized. Nested attributes such as `BusinessParking` and `Ambience` were flattened into structured features, including a unified binary `HasParking` variable and one-hot encoded ambience categories (e.g., `Ambience_classy`, `Ambience_casual`). Additionally, a robust parser was implemented to convert the comma-separated `categories` field into lists, followed by optimized one-hot encoding across all unique business categories. Boolean fields were standardized into binary 0/1 indicators, and the final cleaned feature matrix was exported for downstream modeling. This processing pipeline ensures that both structured and unstructured business metadata are transformed into a uniform, machine-learning–ready representation.

4.3 Data Cleaning and Processing Reviews Data

To prepare the reviews dataset for analysis, we applied a parallel cleaning and feature-engineering process. Punctuation-based indicators such as the presence of exclamation marks, question marks, and all-caps text were extracted as linguistic metadata. Overall sentiment labels were generated using the pretrained transformer *distilbert-base-uncased-finetuned-sst-2-english*, which assigns each review a positive, neutral, or negative classification. For aspect-based sentiment analysis, we employed the pretrained model *yangheng/distilbert-base-uncased-absa* to derive sentiment labels for three core restaurant aspects: food, service, and atmosphere. The resulting outputs provide a structured sentiment representation that complements the engineered metadata and supports downstream modeling.

We also constructed train-test partitions using a group cross-validation strategy. StratifiedGroupKFold was applied to preserve the overall distribution of star ratings while ensuring that reviews from the same business did not appear in both the training and the test sets. This approach prevents data leakage and yields evaluation splits that better reflect real-world deployment conditions.

4.4 Classic Machine Learning

4.4.1 Support Vector Machine Model

We used a linear Support Vector Machine (SVM) to predict star ratings from review text, leveraging TF–IDF vectors with unigrams and bigrams. A soft-margin formulation was employed to allow some misclassification, capturing the inherent ambiguity between adjacent ratings. The optimization problem is defined as:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0,$$

where $\mathbf{x}_i$ are TF–IDF features, y_i are star labels, ξ_i are slack variables, and $C = 1.0$ balances margin maximization with misclassification. Hard-margin SVM was unsuitable due to overlapping class boundaries. The model achieved 62% accuracy overall, with better performance on extreme ratings, and outputs were used for misclassification analysis to identify reviews and businesses with inconsistent ratings.

4.4.2 Logistic Regression Model

We implemented a multinomial Logistic Regression model to predict star ratings using both textual and meta-features. Text data were transformed with TF–IDF vectorization (unigrams and bigrams, max 10,000 features), while additional features included categorical sentiment indicators, Boolean text markers (e.g., exclamations, questions, shouting), and numerical readability scores, all combined via a `ColumnTransformer`. The Logistic Regression objective minimizes the regularized negative log-likelihood:

$$\min_{\mathbf{w}} -\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbf{1}\{y_i = k\}\log\frac{\exp(\mathbf{w}_k^\top \mathbf{x}_i)}{\sum_{j=1}^{K}\exp(\mathbf{w}_j^\top \mathbf{x}_i)} + \frac{\lambda}{2}\sum_{k=1}^{K}\|\mathbf{w}_k\|^2,$$

where $\mathbf{x}_i$ represents the combined features, y_i the star label, K the number of classes, and $\lambda = 1/C$ the regularization term. The `saga` solver was used for efficient convergence with multi-class data. The model achieved an overall accuracy of 64%, outperforming the SVM baseline slightly, with stronger prediction on extreme ratings (1- and 5-star reviews). Outputs were saved to a misclassification DataFrame to analyze businesses with inconsistent reviews and identify patterns in rating-text mismatches.

4.5 Deep Learning

4.5.1 Text Convolutional Neural Network (TextCNN)

To predict review star ratings from textual data, we employed a Text Convolutional Neural Network (TextCNN), which is effective at capturing local n-gram patterns. The model first maps tokenized words to 100-dimensional embeddings, then applies three parallel convolutional layers with kernel sizes 3, 4, and 5 to detect tri-gram, four-gram, and five-gram features. Each convolution is followed by a second convolution and a Global Max Pooling layer to extract the most salient features. The pooled outputs are concatenated, passed through a dense layer with batch normalization and dropout, and finally combined with meta features (sentiment indicators, numerical readability scores, and Boolean markers) before the softmax output layer.

The dataset was split using Stratified Group K-Fold to prevent reviews from the same business appearing in both training and test sets. Text sequences were tokenized and padded to a maximum length of 150, while meta features were standardized or one-hot encoded. The model was trained using categorical cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\log\hat{y}_{ik},$$

where y_{ik} is the true one-hot label and $\hat{y}_{ik}$ is the predicted probability for class k.

The TextCNN achieved a test accuracy of 64%, with macro and weighted F1-scores of 58% and 64%, respectively, outperforming classical models by leveraging both textual patterns and meta-information. Misclassification analysis highlighted that mid-range star ratings were more frequently confused, reflecting the inherent ambiguity in review language.

4.5.2 *Deep Text Convolutional Neural Network (DeepTextCNN)*

Extending the original TextCNN, we developed a deeper convolutional model incorporating pretrained embeddings for enhanced feature extraction. Review text was tokenized with NLTK, and a vocabulary of the 10,000 most frequent tokens in the training set was created. Each review was padded or truncated to 200 tokens. The embedding layer used 100-dimensional GloVe vectors, which were trainable for domain-specific fine-tuning.

Metadata features were processed separately: categorical variables were one-hot encoded, numerical variables standardized, and all metadata combined into a single tensor.

The architecture has two parallel branches:

Text branch: A GloVe embedding layer feeds into parallel 1D convolutions with kernel sizes 3 and 5, each followed by batch normalization and ReLU activation. Max- and mean pooling are applied per convolution, and the pooled outputs are concatenated and projected through a fully connected layer.

Metadata branch: A two-layer multilayer perceptron (MLP) with ReLU activations and dropout processes the metadata features.

Outputs from both branches are concatenated and passed through a final dense block to predict the five-class star rating, combining local textual patterns with structural metadata.

Training: Cross-entropy loss was optimized using Adam with separate learning rates: 1×10^{-4} for embeddings and 1×10^{-3} for other parameters. A ReduceLROnPlateau scheduler halved the learning rate if validation loss stalled. Gradient accumulation (2 steps) and early stopping (patience 5, min improvement 1×10^{-4} were used. Training ran for up to 50 epochs with batch size 32.

Results: The best validation accuracy of 66.3% occurred at epoch 19, with weights saved as `best_cnn_model.pth`. Evaluation included accuracy, macro F1, precision, recall, a classification report, confusion matrix, and training curves.

4.5.3 *Deep Bi-directional Long Short-Term Memory Model (Bi-LSTM)*

A Deep Bi-directional LSTM model was constructed for the same task, using the same tokenization, vocabulary, and fixed sequence length. GloVe embeddings (100-dimensional) were frozen during training.

Metadata preprocessing mirrored the CNN approach.

Text branch: Two-layer bidirectional LSTM (hidden size 256, dropout 0.3) processes the embeddings. Forward and backward last hidden states are concatenated and passed through a fully connected layer (128 units, ReLU, dropout 0.5) to capture sequential patterns.

Metadata branch: A two-layer MLP with batch normalization, ReLU, and dropout (hidden size 64) processes metadata.

Combined branch: Outputs from text and metadata branches are concatenated and passed through an additional dense block (128 units, ReLU, dropout) before the final five-class output layer.

Training: Cross-entropy loss with class weights addressed class imbalance. Adam optimizer with weight decay 1×10^{-5} and ReduceLROnPlateau (factor 0.7) managed optimization. Gradient accumulation (2 steps) and early stopping (patience 7, lowest validation loss) were applied.

Results: Validation accuracy plateaued at 63%, with weights saved as `best_model.pth`. Evaluation included accuracy, macro F1, precision, recall, classification report, confusion matrix, and training curves.

4.6 Misclassification Analysis

4.6.1 *Categorical Misclassification Analysis*

To analyze the Deep Learning (TextCNN) model's errors, we computed a *misclassification rate* for each business:

$$\text{Misclassification Rate} = \frac{\text{Total Misclassified Reviews}}{\text{Total Reviews Tested}}$$

Binary features (cuisine, ambiance, service attributes) were cleaned, filtered for sufficient data ($\geq$ 20 reviews), and merged with business-level misclassification rates. Weighted misclassification rates were calculated per feature to identify patterns of model confusion. Key trends were visualized using pie and bar charts, and the top misclassified businesses were examined to highlight characteristics associated with higher error rates.

4.6.2 *Grade Level and Aspect Misclassification Analysis*

To examine which review-level factors contribute to model errors, we merged the DeepTextCNN misclassification data with cleaned review metadata (aspect sentiment and readability). Sentiment fields (food, service, atmosphere, overall) were standardized to

integer values in $\{-1, 0, 1\}$, and misclassification labels were converted to binary indicators. Only aligned (business_id, review text) pairs were retained to ensure a clean one-to-one match.

We then analyzed how each aspect sentiment relates to misclassification. Correlation coefficients and logistic regression coefficients were computed to quantify the strength and direction of each feature's influence. Misclassification rates were also calculated within each sentiment level to identify patterns such as whether negative service sentiment or mixed aspect profiles produced more errors.

To explore non-linear interactions, reviews were grouped by full sentiment combinations across all aspects. For each combination, we computed total reviews, total misclassification, and the corresponding misclassification rate. This helped surface high-risk sentiment patterns, for example, cases where overall sentiment was positive but service or atmosphere sentiment was negative.

Finally, visualization tools (histograms, grouped bar charts, and combination-level plots) were used to compare error rates across sentiment categories and grade levels. These summaries highlight which review characteristics are most associated with the DeepTextCNN model's misclassifications and point toward ares where the model struggles to capture nuanced or internally conflicting sentiment patterns.

5 Results and Analysis

5.0.1 Comparison of Classical Machine Learning Methods

Table 1 summarizes the performance of the classical machine learning models, namely Support Vector Machine (SVM) and Logistic Regression (LR), on the star rating prediction task. Both models were evaluated on the same stratified group test split, ensuring that reviews from the same business did not appear in both training and testing sets.

Model	Accuracy	Macro F1	Weighted F1	Notes
SVM (Text Only)	0.62	0.54	0.61	Linear kernel, TF-IDF unigrams + bigrams
Logistic Regression (Text + Meta)	0.64	0.56	0.63	TF-IDF + categorical, Boolean, numerical features

Table 1: Performance of SVM and Logistic Regression

Overall, both models achieve moderate accuracy, with Logistic Regression slightly outperforming SVM due to its ability to leverage additional meta-features such as sentiment scores, Boolean indicators, and readability metrics. Precision and recall vary considerably across star classes, with mid-range ratings (2 and 3 stars) showing notably lower F1-scores. This reflects the inherent difficulty of distinguishing subtle differences in sentiment and the skewed distribution of ratings in the dataset, where extreme ratings (1 and 5 stars) dominate.

The relatively low overall accuracy can be attributed to several factors: (i) high subjectivity in review language, (ii) overlapping vocabulary across adjacent star ratings, and (iii) the coarse granularity of 5-class star labels, which increases the likelihood of near-miss predictions (e.g., predicting 4 stars instead of 5). Despite these challenges, the results indicate that meta-features contribute meaningfully to predictive performance, while text alone captures general sentiment trends effectively.

5.0.2 TextCNN Model Results

The TextCNN model achieved a test accuracy of 64%, with macro and weighted F1-scores of 58% and 64%, respectively. Performance was strongest for extreme ratings (1 and 5 stars), reflecting clearer sentiment cues, while mid-range ratings (2–4 stars) were more frequently misclassified, likely due to more ambiguous language in reviews. By combining convolutional layers for textual n-gram patterns with meta features, the model captured both local and contextual information, resulting in a modest improvement over classical methods. Overall, the results demonstrate the effectiveness of the model while highlighting the challenges of predicting subjective mid-range ratings.

Table 2: Classification Report of TextCNN

Star Rating	**Precision**	**Recall**	**F1-score**
1	0.70	0.80	0.75
2	0.45	0.36	0.40
3	0.49	0.30	0.37
4	0.51	0.47	0.49
5	0.74	0.84	0.79
Weighted Average	0.64	0.64	0.64

5.0.3 *DeepTextCNN Model Results*

The DeepTextCNN model achieved a test accuracy of 66%, with macro and weighted F1-scores of 58.6% and 65.1%, respectively. As with TextCNN, the performance was strongest for extreme ratings (1 and 5 stars), where sentiment cues are clearer, while mid-range ratings (2-4 stars) were more frequently misclassified. This pattern may be influenced by the uneven distribution of star ratings in the dataset, which is skewed toward the extremes, as well as the inherently ambiguous language in mid-range reviews. By incorporating a GloVe-initialized embedding layer and a deeper architecture combining convolutions with a multilayer perceptron for metadata features, the model achieved an improvement over the initial TextCNN.

Table 3: Classification Report of DeepTextCNN

Star Rating	**Precision**	**Recall**	**F1-score**
1	0.738	0.778	0.758
2	0.452	0.407	0.428
3	0.532	0.355	0.426
4	0.538	0.496	0.516
5	0.756	0.864	0.806
Weighted Average	0.648	0.663	0.651

Analysis of the confusion patterns revealed that adjacent mid-range classes, particularly 2 vs 3 stars and 3 vs 4 stars, were the most common sources of error. These misclassifications suggest that the model captures strong sentiment signals effectively but struggles when reviews express mixed or neutral sentiment. Overall, these results demonstrate the effectiveness of integrating pretrained embeddings and deeper network structures while highlighting the persistent challenge of accurately predicting subjective mid-range ratings.

5.0.4 *Bi-LSTM Results*

The Bi-LSTM model achieved a test accuracy of 63%, with macro and weighted F1-scores of 57.9% and 58.7%, respectively. As with the CNN models, performance was strongest for extreme ratings (1 and 5 stars), where sentiment cues are clearer and reviews tend to contain more explicit positive or negative language. In contrast, mid-range ratings (2-4 stars) were more frequently misclassified, reflecting the inherent ambiguity of reviewer language and the subtle distinctions between adjacent star ratings. This patten is also likely influenced by the uneven distribution of star ratings in the dataset, which is skewed toward the extreme classes.

Table 4: Classification Report of Bi-LSTM

Star Rating	**Precision**	**Recall**	**F1-score**
1	0.751	0.733	0.742
2	0.405	0.567	0.472
3	0.451	0.370	0.406
4	0.499	0.515	0.507
5	0.787	0.753	0.769
Weighted Average	0.639	0.631	0.633

Analysis of the confusion patterns revealed the same trend for sources of error as DeepTextCNN. These misclassifications suggest that the model captures strong sentiment signals effectively but struggles when reviews express mixed or neutral sentiment. Overall, the results indicate that the Bi-LSTM with GloVe embeddings can successfully leverage sequential textual patterns to identify clear positive or negative sentiment.

5.0.5 *Misclassification Analysis Results*

We analyzed the TextCNN model's errors across cuisines and individual businesses. Table 3 summarizes misclassification rates by cuisine group. Japanese (43.6%), Vietnamese (42.1%), and Thai (40.3%) cuisines exhibit the highest misclassification rates, indicating that reviews for these cuisines are harder to classify correctly. In contrast, Caribbean cuisine has the lowest rate (25.5%). These patterns may reflect limited training data, diverse menu items, or ambiguous review language.

Table 5: Misclassification Rate By Cuisine

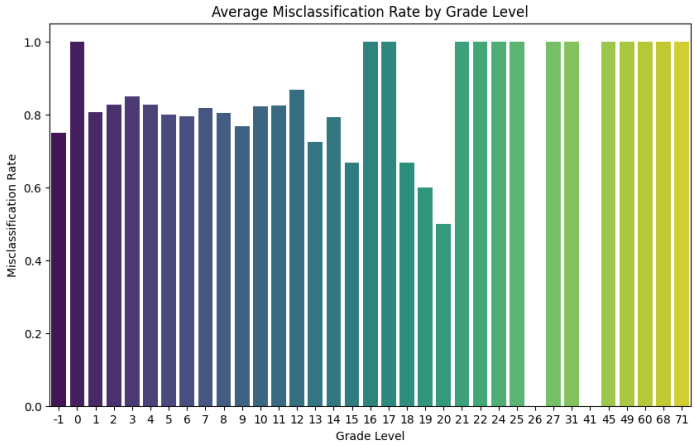
Cuisine	Total Reviews	Misclassified	Misclassification Rate
Japanese	307	134	43.6%
Vietnamese	221	93	42.1%
Thai	605	244	40.3%
Italian	896	342	38.2%
American	8279	3180	38.4%
Caribbean	51	13	25.5%

5.0.6 *Grade Level Misclassification Analysis Results*

We further examined whether the readability of a review influenced the DeepTextCNN model's ability to predict star ratings correctly. Each review's grade level (rounded Flesch-Kincaid score) was paired with its misclassification outcome, and average error rates were computed per grade level.

Several grade-levels, particularly *very low* (0) and *very high* (45, 49, 60, 68, 71), show misclassification rates of 100%, though these groups contain very few reviews. These extremes likely reflect outliers in writing complexity rather than consistent model weaknesses. More populated grade levels still show elevated error rates: reviews at grade levels 3-12 have misclassification rates between roughly $78-87\%$, suggested the model struggles most with reviews written in short, simple sentences or those with mid-range complexity.

Figure 1: Misclassification Rate By Grade Level



In contrast, a few grade levels (e.g. 26 and 41) exhibit 0% misclassification, again due to very small sample sizes. Overall, the results indicate that readability alone does not drive performance issues, but mid-complexity writing seems to introduce ambiguity that makes star-rating prediction harder.

5.0.7 *Aspect Misclassification Analysis Results*

To understand why the model misclassifies certain reviews, we examined three levels of diagnostic information: (1) misclassification rates for each individual aspect-sentiment pair, (2) how strongly each aspect correlates with misclassification, and (3) misclassification rates for specific combinations of aspects within a review.

1. Per-aspect misclassification rates.

Across individual sentiment categories, the model struggles the most when aspects express *negative* sentiment. Food, atmosphere, service, and overall sentiment all show misclassification rates above 84% when label as negative. Neutral and positive sentiments perform slightly better but still display high error rates (typically $77-80\%$). This suggests the model is less reliable when reviews describe mixed or negative experiences, possibly due to broader language variability or less consistent phrasing.

Table 5: Misclassification Rate By Aspect-based Sentiment

Aspect	Sentiment	Misclassification Rate
Food	Negative	0.859
Atmosphere	Negative	0.856
Service	Negative	0.848
Overall	Negative	0.847
Food	Neutral	0.824
Service	Positive	0.796
Overall	Positive	0.795
Atmosphere	Positive	0.795
Food	Positive	0.792
Atmosphere	Neutral	0.772
Service	Neutral	0.772

2. Correlation and logistic regression coefficients.

The correlation analysis shows relatively small absolute values across aspects ($\approx 0.06-0.07$), indicating no single aspect is dominant, though food sentiment displays the strongest absolute correlation. Logistic regression coefficients are also small, but their signs reinforce the same conclusion: aspects with negative polarity have slightly stronger associations with misclassification. Overall, misclassification is diffuse rather than driven by one specific feature.

Aspect	Correlation	Logistic Regression Coefficient	Absolute Correlation
Food	-0.074	-0.178	0.074
Atmosphere	-0.067	-0.064	0.067
Service	-0.0593	0.0108	0.0593
Overall	-0.0592	-0.019	0.0592

Table 5: Correlation and Logistic Regression Coefficient By Aspect

3. Aspect-combination patterns.

When analyzing combinations of all three aspects and overall sentiment, the highest misclassification rates (1.00) occur in

combinations where multiple aspects carry negative sentiment. Many of these categories have only a handful of examples, but the pattern remains consistent in larger groups as well: combinations such as `(-1, -1, -1, -1)` and `(1, 1, 1, 1)` have extremely high misclassification rates ($0.80 - 0.88$) even with thousands of samples.

This suggests the model is challenged by reviews that express uniformly strong sentiment, whether highly positive or highly negative, likely because the surface linguistic cues resemble exaggerated or informal writing patterns that resemble opposing classes.

6 Future Direction

Future work could focus on identifying causal drivers of misclassification rather than relying solely on correlations, allowing us for a deeper understanding of why certain cuisines and businesses consistently confuse the model. Expanding the dataset beyond the USA would improve generalizability and reduce regional bias. Incorporating more advanced language models or contextual embeddings may also help capture nuanced mid-range reviews, which remain the hardest to classify. Finally, conducting more granular error analyses, such as examining review length, sentiment polarity shifts, or conflicting user opinions, could provide clearer insights into model weaknesses and guide targeted improvements.

7 Limitations

This analysis is limited by several factors. First, misclassification highlights correlations rather than causation, making it difficult to determine the underlying reasons for errors. Second, the dataset includes only reviews from the USA, limiting generalizability to other regions like Canada. Finally, textual data is inherently noisy and ambiguous, which contributes to the higher misclassification rates observed for mid-range star ratings.

8 Conclusion

This project examined the underexplored issue of discrepancies between textual sentiment and star ratings in restaurant reviews, which can undermine the reliability of online ratings for both consumers and businesses. By leveraging sentiment, structural and metadata features within both traditional and deep learning models, we observed comparable performance across approaches, with the DeepTextCNN achieving the best accuracy score of 66%. While this performance was lower than anticipated, it highlights the inherent unpredictability of human behaviour in rating decisions. Despite this, our analysis successfully identified and quantified mismatches between sentiment and star ratings, revealing patterns across cuisines, grade levels, and aspect-based sentiment. These patterns suggest that errors may be influenced by limited training data, ambiguous language, and informal writing styles. Overall, no strong correlations were found between misclassification rates and specific cuisines, grade levels, or aspect sentiments. The study is limited by its dataset scope (restricted to the USA), the challenges of analyzing textual data, and the variability of human ratings, but it provides a structured framework for understanding and quantifying rating-sentiment inconsistencies in online reviews.

REFERENCES

[1] Akinlaja, O., & Mosia, M. (2021). *Using Deep Learning and Sentiment Analysis to Identify Mismatches between Online Courses' Reviews and Ratings*. 2021 3rd International Multidisciplinary Information Technology and Engineering Conference (IMITEC), 1–6. https://doi.org/10.1109/IMITEC52926.2021.9714655

[2] Aralikatte, R., Sridhara, G., Gantayat, N., & Mani, S. (2018). *Fault in your stars: An analysis of Android app reviews*. Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, 57–66. https://doi.org/10.1145/3152494.3152500

[3] Shrestha, N., & Nasoz, F. (2019). *Deep Learning Sentiment Analysis of Amazon.com Reviews and Ratings*. International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), 8(1). https://doi.org/10.48550/arXiv.1904.04096

[4] Yelp. (2021). *Yelp Open Dataset.* Yelp Inc. https://business.yelp.com/data/resources/open-dataset/