ABSTRACT

This research aims to anticipate consumer satisfaction in the fashion sector by analyzing product ratings from reviews using a mix of machine learning and deep learning methods. The customer feedback dataset, taken from Kaggle (Misra *et al.*, 2018) is extensively preprocessed via tokenization and vectorization. Techniques such as SMOTE, focused loss, and class weights are applied to address the class imbalance problem. This preparation is essential for improving the performance of predictive models designed for the specific purpose of rating prediction.

The technique evaluates many models, including traditional algorithms such as Logistic Regression, Naive Bayes, and Gradient Boosting, as well as advanced neural network architectures like CNN and LSTM. These models are evaluated in binary and multi-class classification scenarios to distinguish between positive and negative customer attitudes and forecast individual rating categories. Integrating techniques to tackle class imbalance during the training phase of neural network models is crucial for altering the model's sensitivity to minority classes, thereby guaranteeing equitable representation of prediction outcomes.

The model's performance is thoroughly evaluated using cross-validation and a wide range of metrics such as accuracy, precision, recall, and F1-score. Visual aids such as confusion matrices and classification reports provide detailed insights into the accuracy of each model in predicting customer ratings. The comprehensive comparative analysis demonstrates the importance of integrating textual and numerical input and the impact of imbalance handling techniques on improving model robustness. The study provides practical recommendations for utilizing customer reviews to predict ratings more accurately. It also suggests exploring multi-class classification methods and developing new techniques to enhance model generalization and effectiveness in future research.

1. BACKGROUND

In the rapidly evolving landscape of the fashion industry, the fashion industry's reliance on customer satisfaction to guide product improvement and market trends has underscored the importance of accurately analyzing consumer feedback. In the present world on online shopping and reviews, the ability to accurately interpret and predict customer sentiments from textual reviews is paramount. Current sentiment analysis methodologies predominantly rely on multi-class classification (Mukhin, Avksentieva, and Krotov, 2024., Gürcan, 2018). Traditional models often struggle with the biased distribution of ratings, leading to a propensity for predicting majority classes at the expense of minority ones, thus skewing the accuracy of predictions towards more frequently occurring ratings (Li and Liu, 2023). This limitation highlights the need for a more sophisticated system capable of overcoming these challenges.

By initially employing binary classification to effectively segregate positive from negative sentiments, and subsequently delving into the specifics of individual rating predictions through multi-class classification, this proposed methodology aims to provide a granular analysis of customer feedback. This dual-phase approach is designed to circumvent the inherent limitations of binary classification and tackle the class imbalance prevalent in multi-class scenarios using Textual analysis (Lei, Qian and Zhao, 2016). The proposed system, augmented by the integration of advanced deep learning models such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs), along with techniques to address class imbalance including Synthetic Minority Over-sampling Technique (SMOTE), focal loss, and class weights, represents a comprehensive solution to the identified challenges (Pane *et al.*, 2022; Habbat, Anoun, and Hassouni, 2023). These strategies collectively enhance the system's ability to:

1. Achieve a higher granularity in prediction, providing detailed insights into customer preferences.
2. Effectively manage class imbalance, ensuring equitable representation of all rating classes in the predictive model.
3. Improve model generalization, ensuring reliability and robustness across varied datasets.
4. Enhance predictive accuracy, offering valuable insights for product development and strategic decision-making.

2. DATA PRE-PROCESSING

The dataset from Rent the Runway, encompassing user experiences, item details, and demographics, is scrutinized for data preprocessing. Initial analysis identifies both textual and numerical data across 192,544 records, refined to 192,355 unique entries through duplicate removal. Data normalization includes converting ‘Weight’ to integers after removing ‘lbs’ and transforming ‘Height’ into centimetres for uniformity. Categorical columns such as ‘Fit’, ‘Body Type’, ‘Category’, and ‘Rented For’ are converted to categorical types to enhance memory efficiency, with ‘Review Date’ also formatted to datetime. Missing data in key columns like ‘bust size’, ‘weight’, ‘body type’, and others are addressed with the omission of rows lacking ‘rating’, ‘rented for’, and ‘height’ due to their critical role in prediction and negligible proportion. Missing values in ‘Bust Size’, ‘Weight (lbs)’, and ‘Body Type’ are imputed based on logical relationships and group means to maintain dataset integrity. Outlier management targets deviations in ‘Item Id’, ‘Height’, ‘Weight’, ‘Size’, and ‘Age’ with specific thresholds to exclude improbable data points, affecting less than 4% of the dataset. This targeted removal, supported by statistical analysis and visualized through box plots, ensures the exclusion of outliers likely to distort model training without indiscriminate data loss.

3. EXPLORATORY DATA ANALYSIS

This data visualization analysis provides a concise overview of customer demographics and satisfaction for a fashion rental service. The distributions of height, weight, and age suggest a primary customer base in their 20s and 30s, which is further supported by the density plots. The body type distribution, as seen in the pie charts, and rental purposes indicate a preference for wedding and formal occasions, with hourglass and athletic being common body types. The ratings depicted in violin plots, predominantly skew higher, indicating overall customer satisfaction. Heatmaps, Scatter plots and line plots reveal a bias towards higher ratings, especially 10, over lower ratings such as 2, 4, and 6 (Fig 1 and Fig 2). This highlights the need for class imbalance correction during model training to ensure fair representation across the spectrum of customer feedback.

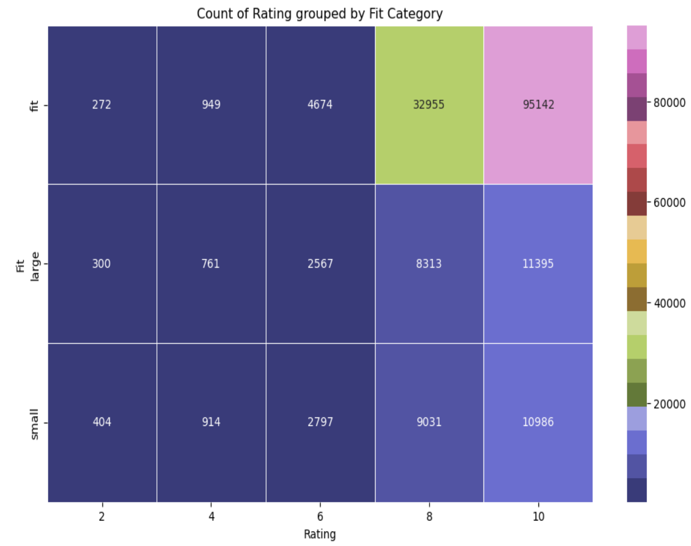

Fig 1. Distribution of Fit- Rating

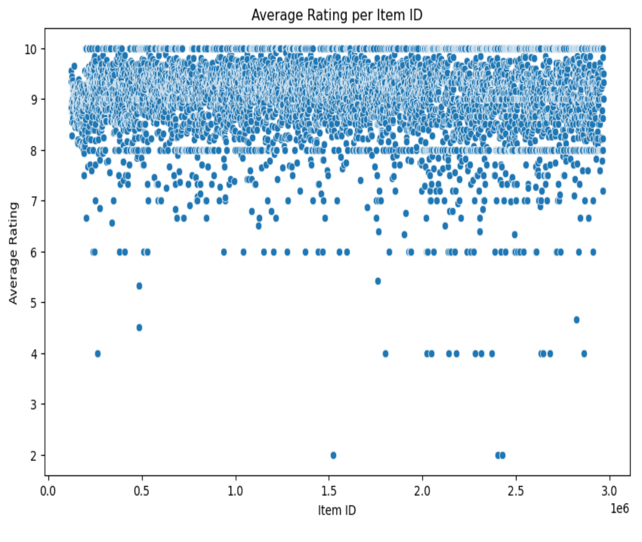

Fig 2. Rating Distribution

4. FEATURE SELECTION

In preparing data for rating predictions, the correlation matrix is used. A ‘Days_Since_Review’ feature is created to provide temporal insights, with scatterplots illustrating the rating distribution over time. Textual reviews –’Review_Text’ and ‘Review_Summary’ are merged into a single feature, ‘Combined_Text’, and processed via TF-IDF vectorization to emphasize relevant terms. Categorical features were label-encoded to numerical formats, and numeric features normalized for model compatibility. ‘User_Id’ and ‘Item_Id’ underwent frequency encoding to quantify user-item dynamics. Dimensionality reduction on TF-IDF text features was performed with Truncated SVD, streamlining the feature set. A final correlation analysis incorporating all processed features guided the selection and preparation of key predictive variables, such as ‘Days_Since_Review’, ‘Height(cm)’, ‘Body_Type’, ‘Item_Id’, and the condensed textual data, ‘Combined_Text’ (Fig 3). This strategic feature curation aims to enhance the predictive accuracy of the subsequent modeling phase.

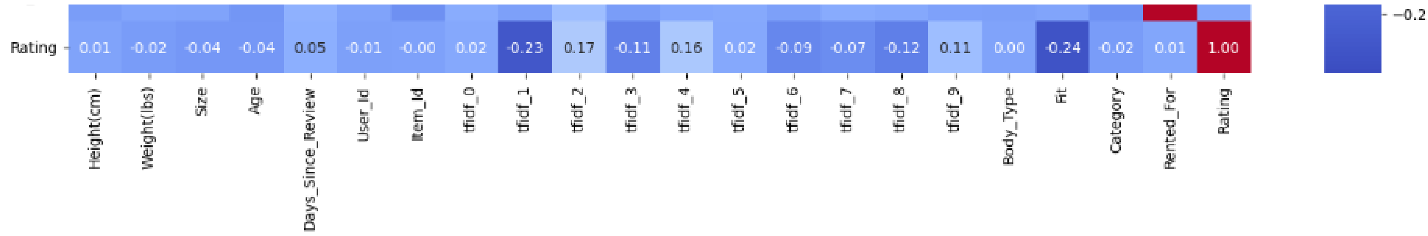

Fig 3. Correlation Matrix- Rating

5. TEXT FIELD PREPROCESSING

The textual analysis of ‘Review_Text’ and ‘Review_Summary’ incorporates several layers of preprocessing to optimize the data for sentiment analysis. Textual content is standardized through lowercasing and the removal of HTML tags and special characters. Essential to maintaining the context of customer feedback, negations and certain conjunctions are preserved while other stopwords are eliminated. Processed text is stored in ‘Processed_Review_Text’ and ‘Processed_Review_Summary’, with further processing via lemmatization to ensure words are reduced to their base or dictionary form. This approach, preferable to the more rudimentary stemming, ensures a context-aware condensation of words(Fig 4), crucial for accurate sentiment analysis,. The output, denoted with ‘_Lemmatize’ suffixes, captures the essence of customer sentiments more effectively for the subsequent modelling phase.

Review_Text	Processed_Review_Text_Lemmatize	Processed_Review_Text_Stem
The dress was much shorter than it appeared, too short for a wedding, so I wasn't able to wear it. Other reviews about the lace being a little tight on the shoulders is spot-on, but it would have been fine if the dress had been a little longer. Bummer, too, because it is such a pretty dress.	dress much shorter appeared short wedding not able wear review lace little tight shoulder spoton but would fine dress little longer bummer pretty dress	dress much shorter appear short wed not abl wear review lace littl tight shoulder spoton but would fine dress littl longer bummer pretti dress

Review_Summary	Processed_Review_Summary_Lemmatize	Processed_Review_Summary_Stem
Wasn't able to wear	not able wear	not abl wear

Fig 4. Comparison of Processed Reviews

6. MACHINE LEARNING MODEL 1: LOGISTIC REGRESSION (LR)

6.1 Summary of the Approach:

Chosen for its simplicity, LR assumes a linear relationship between independent variables and the log odds of ratings. Key to this model is its interpretability for binary and multiclass problems (Abramovich, Grinshtein and Levy, 2021).

6.2 Model Training and Evaluation:

For Features selected through correlation matrix, utilized TF-IDF for ‘Combined_Reviews’ text data transformation and MinMaxScaler and OneHotEncoder for numerical and categorical data normalization. Hyperparameters, regularization strength ‘C’ and optimization algorithm ‘solver’, were optimized via GridSearchCV, focusing on regularization and solver optimization. Evaluation metrics included accuracy, precision, recall, and F1 score, employing cross-validation.

6.3 Results and Discussion:

Achieved an accuracy of 71.87% using Code Logic 1(Fig 5), with compromised minority classes prediction due to class imbalance issue, which is addressed in Code Logic 2(Fig 6) by adjusting class weights, slightly reducing accuracy to 0.6916 but improving fairness across classes.

7. MACHINE LEARNING MODEL 2: LIGHT GRADIENT BOOSTING MACHINE (LGBM)

7.1 Summary of the Approach:

LGBM is an efficient and scalable implementation of gradient boosting framework that uses tree-based learning and is renowned for handling large data sets with ease (Taha *et al*., 2020). It's chosen for its ability to capture non-linearities and interactions between features.

7.2 Model Training and Evaluation:

Applied similar preprocessing as LR. Hyperparameters like 'num_leaves', 'learning_rate', and 'n_estimators' were finely tuned in both code Logic 1 and 2, which is evaluated using Cross-Validation for comprehensive validation. Class imbalance was directly addressed within LGBM settings in code Logic 2.

7.3 Results and Discussion:

Initial accuracy of 0.7141 in code logic 1(Fig 5) dropped to 6367 in code logic 2(Fig 6) upon class weight adjustments, indicating a strategic focus on minority classes, which, while lowering overall accuracy, enhances model equity. Despite a lower overall accuracy, code logic 2's approach is chosen to promote a more balanced and fair predictive performance across all classes.

8. MACHINE LEARNING MODEL 3: MULTINOMIAL NAIVE BAYES (NB)

8.1 Summary of the Approach:

NB is a straightforward probabilistic classifier based on applying Bayes' theorem with the assumption of feature independence which is well-suited for text classification problems as evidenced by its frequent use in spam detection and document classification scenarios.

8.2 Model Training and Evaluation:

Followed the same preprocessing strategy as above mentioned models. The 'alpha' parameter is fine-tuned for optimal smoothing. Unlike other models, NB inherently addresses class probabilities without requiring explicit class weight adjustments as some class imbalance techniques like SMOTE are not directly applicable, as they could disrupt the underlying probability distribution. Evaluation remains same as above models.

8.3 Results and Discussion:

NB maintains a consistent accuracy of 0.6716 across both code logics(fig 5, Fig 6). Even though the overall accuracy is nearly equal to other two models, this model performs the least in predicting individual classes with 0.0 F1 score for minority classes 2 and 4. The decision to forgo SMOTE is due to its incompatibility with NB's probabilistic assumptions and its potential to inflate computational time. The model's strength lies in its speed and suitability for large-scale text analytics, despite the lower overall accuracy.

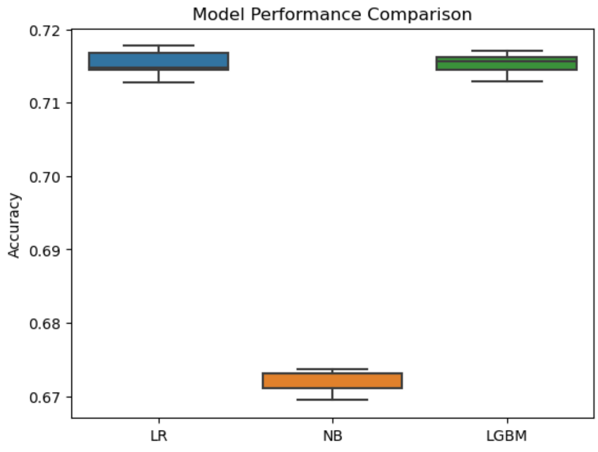

Fig 5. Accuracy of Traditional models(Code Logic 1)

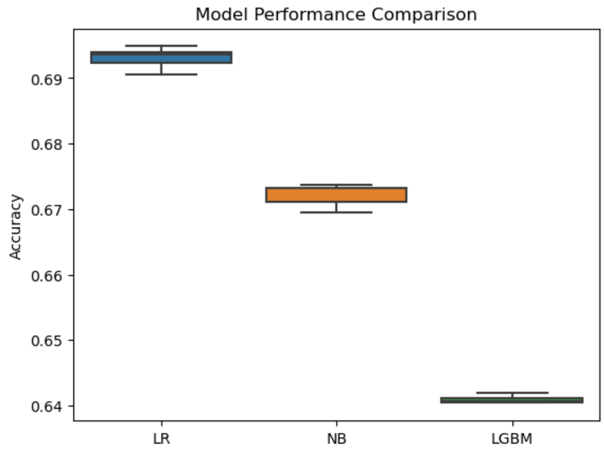

Fig 6. Accuracy of Traditional models(Code Logic 2)

9. DEEP LEARNING MODELS: CNN+LSTM

9.1 Summary of the Approach:

In addressing the text classification problem, Convolutional Neural Networks (CNN) and Long Short-Term Memory networks (LSTM) are implemented individually and in combination due to their unique strengths in feature extraction and sequence learning, respectively. CNNs are adept at capturing spatial hierarchies and local context within data, making them suitable for detecting patterns in text such as n-gram features. LSTMs excel in processing sequences with their ability to maintain long-term dependencies, beneficial for understanding the context in sentences. Combining these models leverages the spatial feature recognition of CNNs with the sequential context preservation of LSTMs, aiming to create a robust classifier for complex text data(Pane *et al*., 2022).

9.2 Model Training and Evaluation:

Different Models and their combinations are tried with varying Feature selection, Hyper parameters and class Imbalance handling mechanisms. Additionally, the preprocessing on the Text features- ‘Combined_Reviews’, also varies to identify best performing method with its parameter values. All the below mentioned methods are evaluated based on accuracy, Precison, recall and F1-score over multiple number of epochs. The methods are listed as below.

9.2.1 Convolutional Neural Networks (CNN) - Code Logic 3

The CNN model was parameterized with a vocabulary size of 5000, embedding dimension of 50, and sequence length of 100. It included 128 filters for the convolutional layer and a kernel size of 5. Combination of Lemmatized Review Text and Review Summary is selected as feature, which is passed through Tokenizer to turn them into a numerical format for neural network to train on top of that. The evaluation metrics are set to the usual Classification Report attributes- Accuracy, Precision, F1 and recall.

9.2.2 Long Short-Term Memory Networks (LSTM) - Code Logic 4

The LSTM model follows a similar data preprocessing strategy as CNN. The LSTM layers, with 64 and 32 units respectively, aim to capture both short-term and long-term dependencies within the text. The Target field- ‘Rating’ is also encoded before training the model similar to CNN model. Evaluation metrics are same as the other model.

9.2.3 Combination of CNN and LSTM - Code Logic 5 & 6

The combined CNN and LSTM model in Code Logic 5 incorporates a convolutional layer before LSTM layers. The training involves optimizing convolutional filters, LSTM units, and dropout rates to manage the additional complexity of the combined model. The addition of dropout layers post-LSTM suggests a focus on reducing overfitting. While feature selection remains the same as above, GloVe embeddings is used for processing the Lemmatized Review fields, which provides a head start with embeddings that already understand some level of context and semantics. Early stopping is introduced to avoid overfitting and class Imbalance is handled by using class-weights. Similar setup is used for Code Logic 6 with exception of Features selected based on initial correlation matrix, which is processed similar to other traditional methods for Numeric and Categorical fields. Tokenizer is utilized for Combined_Reviews field.

9.2.4 Binary Classification on Higher and Lower Ratings - Code Logic 7 & 8

In contrast to all above methods based on multi-class classification, both Code Logic 7 and Code Logic 8, utilized Adam as the optimizer, with a learning rate of 0.001 in Code Logic 7 and 0.0001 in Code Logic 8. The training process incorporated early stopping, a regularization technique, to halt training when the validation loss ceased to decrease, preventing overfitting. In Code Logic 7, only Combined_Reviews is given as input whereas in Code Logic 8, additionally Height(cm), Body_type, Item_Id, Days_Since_Review are used as identified by correlation matrix.

In Code Logic 7, Focal Loss was implemented, directing the model’s attention to harder-to-classify examples and reducing the impact of easier ones. Code Logic 8 handled the imbalance through class weights, ensuring equitable representation of different rating classes. Dropout, a regularization technique, was incorporated in both models to prevent overfitting with different values. In terms of evaluation metrics, both models used accuracy as a primary metric to gauge overall performance. With these metrics, Binary classification is done to differentiate higher rating (8, 10) from lower ones (2, 4 6). Following this is further classification prediction for both the logics separately with difference in class imbalance technique, where SMOTE is used in Code Logic 7 to further differentiate among 2, 4 and 6. Class weights is used for other code – Code Logic 8.

9.3 Results and Discussion:

While the individual CNN and LSTM models yield 66% and 71% overall accuracy, they fall short in capturing the correct induvial class predictions. This shortfall is addressed by combining CNNs and LSTMs, which shows an improvement in recognizing textual nuances but still leaves room for improvement even after addressing class imbalance. Introducing binary classification helps streamline the problem by focusing on differentiating between distinct classes of ratings, resulting in a more specialized and potentially more accurate model. Code Logic 7 achieved a test accuracy of 93.60% for higher-lower rating classification, 57% for 2,4 and 6 and 73% for 8 and 10 prediction, while Code Logic 8 demonstrated a test accuracy of 93.56%, 64% and 72% for same purpose(Fig 7, 8, 9).

Even though, certain metrics of Logic 7, which uses SMOTE, Focal Loss etc, is marginally higher than Code Logic 8, after analysis data iterated through each epoch, it is seen that Code Logic 8 is better generalized and prone to Over-Fitting, hence making it to be considered better than Code Logic 7.

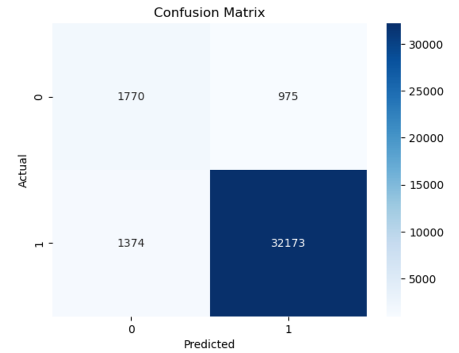

Fig 7. 0-1 prediction

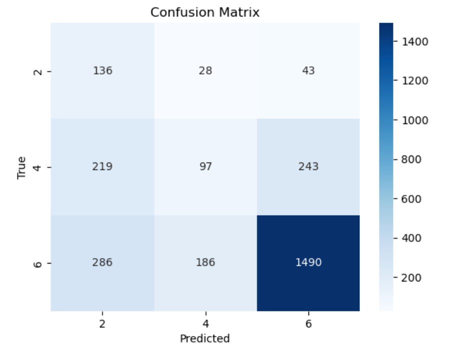

Fig 8. 2-4-6 prediction

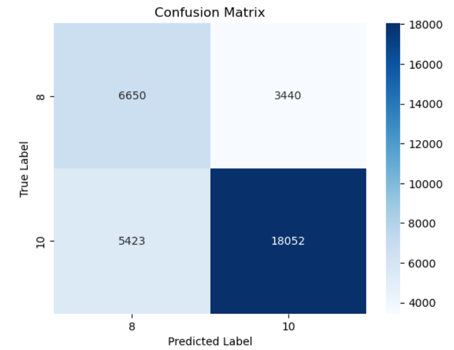

Fig 9. 8-10 prediction

10. EVALUATION STRATEGIES

For traditional algorithms, GridSearchCV is utilized to explore hyperparameter ranges, identifying the most effective configurations systematically. In deep learning, manual tuning determine optimal settings. Regularization techniques such as dropout and early stopping are paramount to mitigate overfitting and conserving computational resources. Cross-validation, segmenting data into multiple folds, ensures that models are tested against various subsets, providing a robust performance assessment and highlighting generalizability. Accuracy, precision, recall, and F1-score serve as principal evaluation metrics, offering a comprehensive view of model performance. Accuracy provides an overall performance assessment, while precision and recall are crucial for imbalanced datasets, focusing on the model's capability to accurately predict minority classes. The F1-score, harmonizing precision and recall, addresses the trade-offs between false positives and negatives.

11. RESULT

Traditional models (Fig 10) like Logistic Regression achieved an accuracy of 69.16%, demonstrating a reasonable distinction between different rating classes, as reflected by its class-wise precision, recall, and F1 scores. However, the Naive Bayes model, despite an overall accuracy of 67.16%, struggled with the minority classes, revealing the model's challenges with imbalanced data—a frequent scenario in text classification. LightGBM's accuracy stood at 63.67%, lower compared to its counterparts, possibly due to its handling of sparse text data and imbalanced classes.

On the other hand, deep learning models (Fig 11), particularly the combination of CNN and LSTM, stand out in their performance, with the binary classification model achieving a remarkable accuracy of 93.56% for distinguishing between higher and lower rating groups. This indicates the model's efficiency in capturing both local features through CNN and long-term dependencies through LSTM. Similarly, the sub level Binary-class CNN + LSTM model for differentiating between the higher group - 8 and 10 rating shows higher accuracy over traditional models for individual rating classes, demonstrating the deep learning approach's capability to understand and classify complex textual patterns.

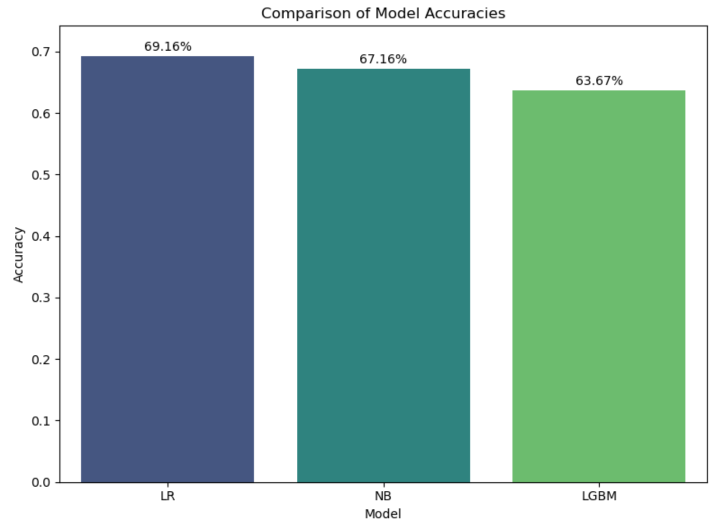

Fig 10. Comparison of Traditional models

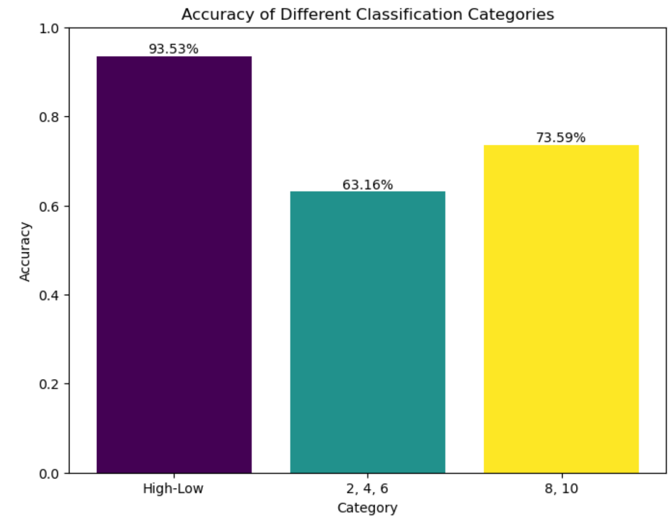

Fig 11. Comparison of Deep Learning models

However, deep learning's effectiveness in distinguishing individual classes within lower rating group (2, 4, 6) was only average. This is because of the complexity of textual sentiment analysis in reviews, which may not always align with the numerical rating given—a point underscored by the example(Fig 12) of a sample review , where rating is 2, even though the Review_Text and Review_summary is largely positive in tone. This inconsistency between textual sentiment and given ratings indicates the limitations of class imbalance techniques. Reviews with seemingly positive feedback may still be associated with lower ratings, potentially confounding the model's learning process. Since even with the usage of multiple class balancing techniques such as SMOTE, Class Weights, Focal loss, etc, the performance remains less when it comes to identifying lower most rating in the group as potentially, these reviewed when oversampled or under sampled create Noise rather than adding to the feature performance.

Therefore, deep learning models are proficient in capturing complex patterns in text and has a commendable performance when it comes to Binary classification of distinguish overall higher and lower Rating more effectively than multi- class prediction in case Rating prediction from consumer feedback Reviews.

Rating	Review_Text	Review_Summary
2	I am petite and ordered an 8R. I was so excited since this was my first rental with RTR. I tried it on but it was too tight. The 10 would have been better but the length was way too long! I ended up wearing my back-up dress. The color of the gown is awesome! Wish I could have worn this one. If you are a petite lady, order petite!	Beautiful color!

Fig 12. Sample of Lower rating(Rating 2)

Model Type	Model	Accuracy	Rating-wise		
			Precision	Recall	F1-Score
Traditional	Logistic Regression (LR)	69.16%	2: 0.16, 4: 0.19, 6: 0.32, 8: 0.54, 10: 0.81	2: 0.33, 4: 0.34, 6: 0.41, 8: 0.41, 10: 0.85	2: 0.21, 4: 0.24, 6: 0.36, 8: 0.46, 10: 0.83
Traditional	Naive Bayes (NB)	67.16%	2: 0.00, 4: 0.00, 6: 0.46, 8: 0.49, 10: 0.69	2: 0.00, 4: 0.00, 6: 0.02, 8: 0.15, 10: 0.97	2: 0.00, 4: 0.00, 6: 0.05, 8: 0.23, 10: 0.81
Traditional	LightGBM (LGBM)	63.67%	2: 0.15, 4: 0.17, 6: 0.25, 8: 0.46, 10: 0.84	2: 0.42, 4: 0.40, 6: 0.46, 8: 0.50, 10: 0.72	2: 0.22, 4: 0.24, 6: 0.33, 8: 0.48, 10: 0.78
Deep Learning	CNN + LSTM Binary Classification (Higher vs. Lower)	93.56%	Lower: 0.57, Higher: 0.97	Lower: 0.65, Higher: 0.96	Lower: 0.60, Higher: 0.96
Deep Learning	CNN + LSTM Multi-Class (Ratings 2, 4, 6)	63.16%	2: 0.21, 4: 0.31, 6: 0.84	2: 0.66, 4: 0.17, 6: 0.76	2: 0.32, 4: 0.22, 6: 0.80
Deep Learning	CNN + LSTM Binary Classification (Ratings 8, 10)	73.59%	8: 0.55, 10: 0.84	8: 0.66, 10: 0.77	8: 0.60, 10: 0.80

Table 1. Comparison of finalized model performance

12. CONCLUSION

In conclusion, this research across various machine learning and deep learning models reveals distinct performance characteristics. Among traditional models, Logistic Regression (LR) shows commendable performance, achieving a balance between simplicity and predictive power. However, the detailed understanding required for textual data, coupled with the natural imbalances within the classes, positions deep learning models, especially those employing a combination of CNNs and LSTMs, as more adept at addressing these challenges. It is evident that with deep learning models, particularly the combination of CNN and LSTM, demonstrates superior performance in handling the complexities of text classification tasks. This superiority can be attributed to their ability to learn high-level features from text data and effectively manage the sequence nature of language.

Based on the results, it is recommended to prioritize deep learning models for text classification challenges, especially when dealing with heterogeneous and extensive datasets. However, it is crucial to consider the computational resources and time required for training such models. For datasets with limited complexity or when computational efficiency is a priority, traditional models may still hold value. Future work could explore more sophisticated model architectures, such as Transformer-based models, BERT models, which have shown promise in various natural language processing tasks. Investigating alternative text representation techniques, beyond tokenization and embeddings, could yield further insights into model performance improvements. Additionally, expanding the scope to include unsupervised or semi-supervised learning approaches might offer novel perspectives, particularly in scenarios where labelled data is scarce or expensive to obtain.

REFERENCES

Abramovich, F., Grinshtein, V. and Levy, T., (2021). ‘Multiclass classification by sparse multinomial logistic regression’, *IEEE Transactions on Information Theory*, 67(7), pp.4637-4646

Gürcan, F. (2018). ‘Multi-Class Classification of Turkish Texts with Machine Learning Algorithms’, *2nd International Symposium on Multidisciplinary Studies and Innovative* Technologies (ISMSIT), Ankara, Turkey, 1-5. https://doi.org/10.1109/ISMSIT.2018.8567307.

Habbat, N., Anoun, H., & Hassouni, L. (2023). ‘Combination of GRU and CNN Deep Learning Models for Sentiment Analysis on French Customer Reviews Using XLNet Model’, *IEEE Engineering Management Review*, 51(1), 41-51. https://doi.org/10.1109/EMR.2022.3208818.

Lei, X., Qian, X., & Zhao, G. (2016). ‘Rating Prediction Based on Social Sentiment From Textual Reviews’, *IEEE Transactions on Multimedia*, 18(9), 1910-1921. https://doi.org/10.1109/TMM.2016.2575738.

Li, P., & Liu, H. (2023). ‘Binary Decomposition for Multi-Class Classification Problems: Development and Applications’, *International Conference on Machine Learning and Cybernetics (ICMLC*), Adelaide, Australia, 452-457. https://doi.org/10.1109/ICMLC58545.2023.10327973.

Misra, Rishabh, Mengting Wan, and Julian McAuley. (2018) ‘Decomposing fit semantics for product size recommendation in metric spaces’, *In Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 422-426.

Misra, Rishabh and Jigyasa Grover. (2021) ‘Sculpting Data for ML: The first act of Machine Learning’, ISBN 9798585463570.

Mukhin, I., Avksentieva, E., & Krotov, A. (2024). ‘Sentiment analysis: A review and comparative analysis of existing approaches’, *Proceedings of the 2023 6th International Conference on Machine Learning and Natural Language Processing (MLNLP ‘23). Association for Computing Machinery, New York, NY, USA,* 199–203. https://doi.org/10.1145/3639479.3639520.

Pane, S.F., Ramdan, J., Putrada, A.G., Fauzan, M.N., Awangga, R.M., & Alamsyah, N. (2022). ‘A Hybrid CNN-LSTM Model With Word-Emoji Embedding For Improving The Twitter Sentiment Analysis on Indonesia’s PPKM Policy’, *6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, Indonesia*, 51-56. https://doi.org/10.1109/ICITISEE57756.2022.10057720.

Taha, A.A. and Malebary, S.J., (2020). An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access, 8, pp.25579-25587.