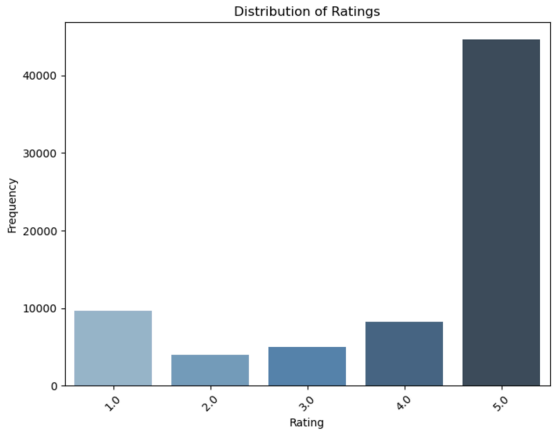
Sentiment Analysis for Magazine Subscription Ratings Utilizing Multiple Approaches

For this study, we selected a dataset of **Amazon reviews in 2023**[1], specifically focusing on the **magazine_subscription** category. With over 500 million total reviews available on Amazon, narrowing the scope to this category provided a more manageable and domain-specific subset for analysis. Within this category, we further reduced the dataset to a **random subset of 30,000 reviews** for our modeling phase, ensuring computational feasibility while maintaining a sufficiently large sample size to train robust machine learning models.



1. Identify Dataset to Study

Basic Statistics and Properties

The full dataset of **magazine_subscription** reviews comprises **71,497 records**, each representing a single review. Key features include:

- **Rating**: Numerical scores from 1.0 to 5.0.
- **Review text**: Written feedback from users about their experiences.
- **Metadata**: Fields such as the review title, associated images, user ID, timestamp, and the product's ASIN.
- **Helpful votes**: The number of users who found a review helpful.
- **Verified purchase**: Whether the review is from a verified Amazon purchase.

Key findings from the exploratory data analysis include the following:

Ratings Distribution

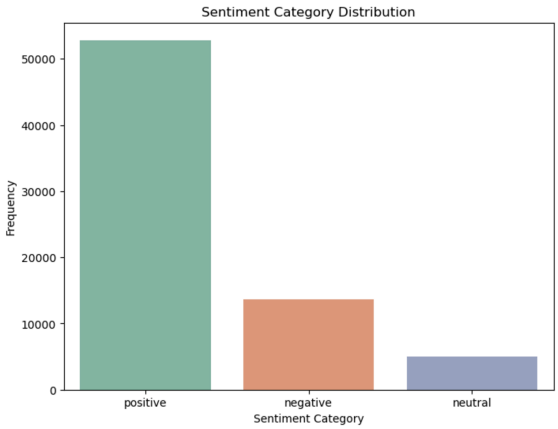
The dataset contains ratings distributed as follows:

- **5.0 stars**: 44,620 reviews (62.4%)
- **4.0 stars**: 8,206 reviews (11.5%)
- **3.0 stars**: 5,033 reviews (7.0%)
- **2.0 stars**: 3,953 reviews (5.5%)
- **1.0 star**: 9,685 reviews (13.6%)

Sentiment Category Distribution

To further analyze the reviews, we categorized them into **positive**, **negative**, and **neutral** sentiment based on their star ratings:

- **Positive (4–5 stars)**: 52,826 reviews (73.9%)
- **Negative (1–2 stars)**: 13,638 reviews (19.1%)
- **Neutral (3 stars)**: 5,033 reviews (7.0%)



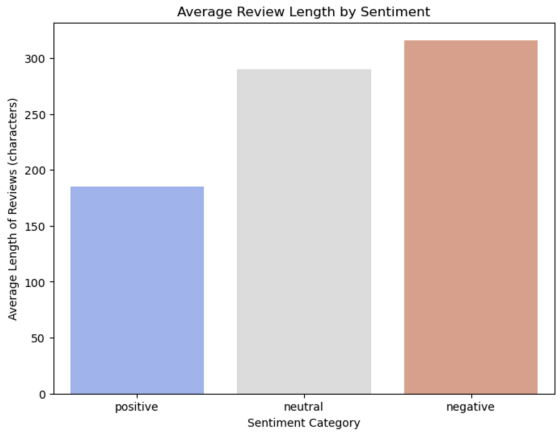
Review Length

The average length of review text across all reviews is **217.17 characters**. Breaking it down by sentiment:

- **Positive reviews**: Average length of **184.73 characters**.
- **Neutral reviews**: Average length of **290.11 characters**.

- **Negative reviews**: Average length of **315.89 characters**.

Negative reviews tend to be longer and more detailed, indicating that dissatisfied customers may take more time to articulate their concerns.

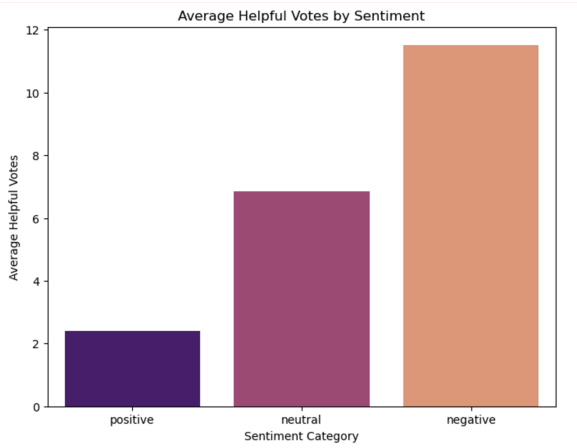


Helpful Votes

On average, reviews receive **4.45 helpful votes**, with significant variation by sentiment:

- **Positive reviews**: **2.40 helpful votes** on average.
- **Neutral reviews**: **6.86 helpful votes** on average.
- **Negative reviews**: **11.52 helpful votes** on average.

Negative reviews are more likely to resonate with other users, possibly due to their detailed nature and strong emotional content.



2. Identify a predictive task on this dataset

Logistics Regression based on these two vectors: TFIDF, Word2Vec
Use text length to predict ratings

3. Select/design an appropriate model

Models/Outcomes:
Linear Regression Model
We started with a simple and baseline model: Linear Regression. Linear regression predicts a continuous outcome, which was repurposed for sentiment classification by rounding predictions to the nearest integer and clipping them to the range of valid sentiment classes (0, 1, 2). The model was trained on the review length feature to minimize the mean squared error between predicted and true sentiment values. Predictions were evaluated using Mean Absolute Error (MAE) and accuracy. With this baseline model, we achieved an MAE of approximately 0.476 and an accuracy of approximately 66.87%. To achieve a higher accuracy in our predictions, we moved on to a Logistics Regression Model, which outperformed linear regression by leveraging its suitability for classification tasks.

Logistic Regression Model
Logistic regression is a probabilistic classification model optimized for discrete target variables. We trained the logistic regression model using the same feature set and labeled data. Logistic regression directly outputs probabilities for each class, which were converted into final predictions by selecting the class with the highest probability. The logistic regression model achieved an accuracy score of 73.78%, higher than the accuracy score of 66.87% achieved by the linear regression model. We believe this happened due to the fact that the logistic regression model is better suited for multi-class classification problems. The classification report can be found below:

```
Classification Report:
              precision    recall  f1-score   support

    Negative       0.35      0.02      0.04      2754
     Neutral       0.00      0.00      0.00       973
    Positive       0.74      0.99      0.85     10573

    accuracy                           0.74     14300
   macro avg       0.36      0.34      0.30     14300
weighted avg       0.62      0.74      0.64     14300
```

Despite the improvement, we felt that we could achieve an even greater accuracy score by building on these baselines models. Rather than simply looking at the text length, we began to take into account the words in the reviews.

TF-IDF and Word2Vec Features with Logistic Regression

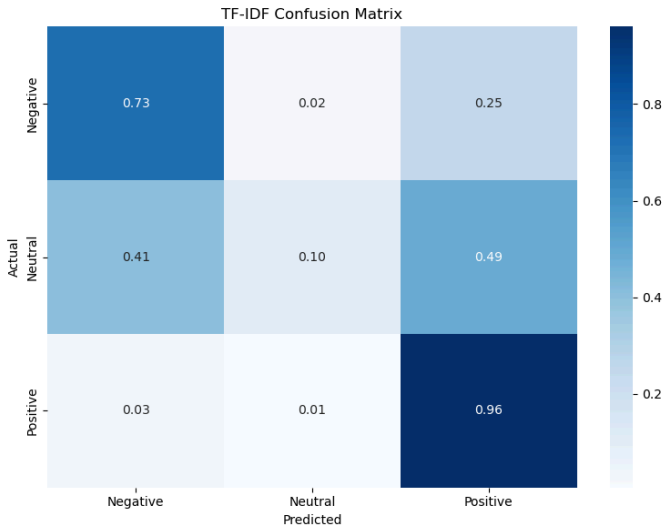
TF-IDF captures word importance by considering both term frequency within a review and the rarity of the term across all reviews. While TF-IDF captures frequency-based patterns, Word2Vec embeddings represent words with similar meanings close to each other in a vector space.

We started by using TF-IDF to transform reviews into numerical vectors. This representation enabled the logistic regression model to predict sentiment with an accuracy of **85.73%**. The classification report is as follows:

```
TF-IDF Classification Report:
              precision    recall  f1-score   support

    Negative       0.72      0.73      0.73      1148
     Neutral       0.45      0.10      0.17       414
    Positive       0.90      0.96      0.93      4438

    accuracy                           0.86      6000
   macro avg       0.69      0.60      0.61      6000
weighted avg       0.83      0.86      0.84      6000
```

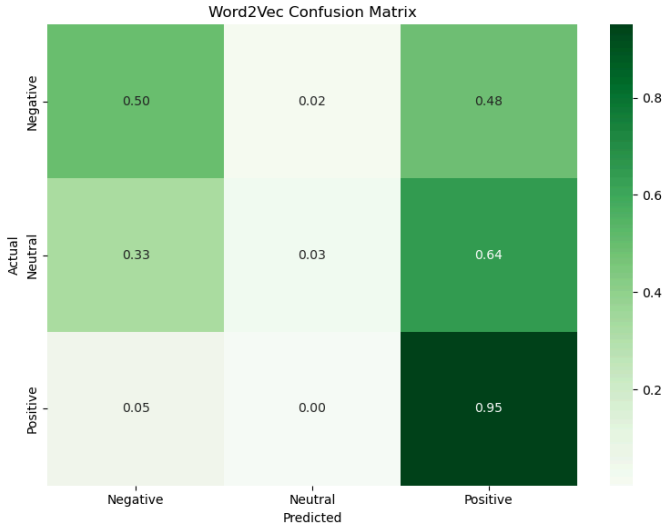


We then used Word2Vec embeddings By averaging the word2vec vectors, we generated document embeddings. The logistic regression model trained on these embeddings achieved an accuracy of **80.03%**, slightly lower than the TF-IDF approach. The classification report for the same is as follows:

```
Word2Vec Classification Report:
              precision    recall  f1-score   support

    Negative       0.61      0.49      0.55      1148
     Neutral       0.31      0.04      0.07       414
    Positive       0.84      0.95      0.89      4438

    accuracy                           0.80      6000
   macro avg       0.59      0.49      0.50      6000
weighted avg       0.76      0.80      0.77      6000
```

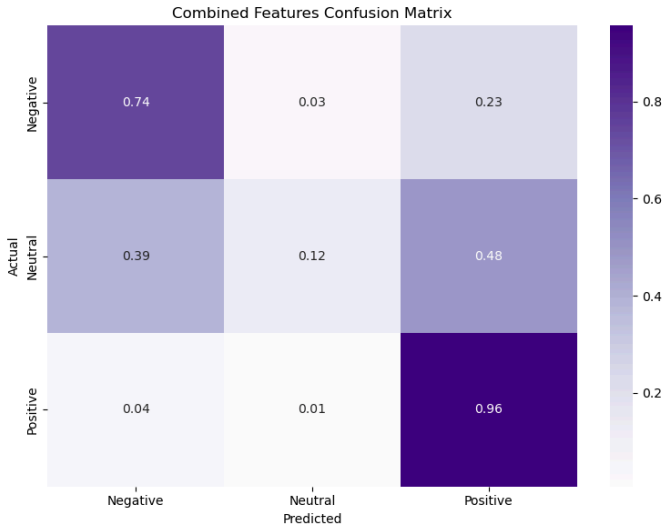


After that, we combined the TF-IDF and Word2Vec features and evaluated the logistic regression model with these combined features. This approach achieved an

accuracy of **85.77%**, showing marginal improvement over the TF-IDF model.

Combined Features Classification Report:

	precision	recall	f1-score	support
Negative	0.72	0.74	0.73	1148
Neutral	0.44	0.12	0.19	414
Positive	0.90	0.96	0.93	4438
accuracy			0.86	6000
macro avg	0.69	0.61	0.62	6000
weighted avg	0.84	0.86	0.84	6000



Lastly, we scaled the Word2Vec embeddings, used a weighted combination of features (70% TF-IDF, 30% Word2Vec), and tuned logistic regression hyperparameters using GridSearchCV. This optimized model achieved an accuracy of **85.83%**, the highest among all approaches.

```
Best Parameters: {'C': 10, 'max_iter': 200, 'solver': 'liblinear'}
Optimized Logistic Regression Classification Report:
              precision    recall  f1-score   support

    Negative       0.72      0.74      0.73      1148
     Neutral       0.43      0.16      0.24       414
    Positive       0.91      0.95      0.93      4438

    accuracy                           0.86      6000
   macro avg       0.69      0.62      0.63      6000
weighted avg       0.84      0.86      0.84      6000
```

Despite this limitation, the TF-IDF-based model provided a robust baseline for text-based sentiment classification.

Linear regression achieved a Mean Absolute Error (MAE) of **0.476** and an accuracy of approximately **66.87%**. This served as a foundational baseline but fell short of the performance needed to handle the complexities of sentiment classification effectively. Recognizing its limitations, we transitioned to a more suitable model for classification tasks: logistic regression.

Logistic Regression Model

Logistic regression is inherently designed for discrete target variables, making it better suited for tasks like sentiment classification. Unlike linear regression, which predicts continuous outputs, logistic regression estimates the probability of each sentiment class. These probabilities are then converted into class predictions by selecting the label with the highest probability. This probabilistic framework allows logistic regression to model non-linear decision boundaries and effectively handle multi-class problems.

Using logistic regression, we achieved an improved accuracy of **73.78%**, significantly outperforming the linear regression model. This improvement can be attributed to the model's ability to optimize for classification tasks and its robustness in separating classes. However, despite this improvement, the model still relied on simple features like review length, which limited its ability to fully capture the linguistic nuances within the data.

To address these shortcomings, we shifted focus to textual features that could better represent the underlying sentiment of reviews. This is where feature extraction techniques like **TF-IDF** and **Word2Vec** came into play.

Why TF-IDF and Word2Vec Improve Performance

Human language is inherently complex, and sentiment often depends on nuanced patterns and context within the text. Models relying solely on superficial features, such as review length, fail to capture these deeper relationships. By introducing **TF-IDF** and **Word2Vec**, we leveraged advanced feature extraction methods that encode the linguistic and semantic richness of the reviews.

- **TF-IDF (Term Frequency-Inverse Document Frequency):** TF-IDF captures word importance by considering both term frequency within a review and the rarity of the term across all

reviews. This allows the model to focus on sentiment-laden words or phrases (e.g., “excellent,” “poor”) while ignoring overly common, non-informative words (e.g., “the,” “and”). By representing reviews as vectors of unigrams, bigrams, and trigrams, TF-IDF captures both individual terms and their contextual usage, significantly enhancing the model’s ability to distinguish between sentiments.

- **Word2Vec:** While TF-IDF captures frequency-based patterns, Word2Vec embeddings add a layer of semantic understanding by mapping words into a dense vector space. These embeddings represent words with similar meanings close to each other in the vector space (e.g., “good” and “great”). By averaging the word embeddings for all words in a review, we generated document-level representations that encode contextual relationships and sentiment-related nuances.

Both methods provide representations that go beyond basic features like length, allowing logistic regression to model the intricate relationships between text and sentiment. As we moved to these feature sets, we observed significant improvements in accuracy, confirming their effectiveness.

4. Describe related literature

Describe literature related to the problem you are studying. If you are using an existing dataset, where did it come from and how was it used? What other similar datasets have been studied in the past and how? What are the state-of-the-art methods currently employed to study this type of data? Are the conclusions from existing work similar to or different from your own findings?

Origin and Use of Dataset
The dataset used for this analysis is part of the Amazon Reviews Dataset 2023[1], curated by McAuley Lab. This dataset represents a large-scale collection of user reviews and product metadata, providing a comprehensive resource for exploring consumer behavior, sentiment analysis, and product recommendation systems. Similar datasets have been employed in studies that leverage textual reviews to predict customer satisfaction, identify product sentiment trends, or classify feedback into sentiment categories. For example, datasets like Amazon Reviews and Yelp Reviews have frequently been used in natural language processing (NLP) tasks to evaluate the effectiveness of feature engineering and machine learning models in sentiment classification.
This dataset has been used in the past for these purposes:

1. **Binary Sentiment Classification:**
    - The dataset was utilized to train a sentiment classification model distinguishing between positive and negative sentiments, the fine-tuned model achieved a high performance.
2. **Title Generation for Reviews:**
    - Another application involved generating succinct titles for reviews based on their content, user ratings, and product metadata. For instance:
        - Input: A 1-star review of a floor mat product that described safety hazards and poor quality.
        - Output: "These mats slip, fold, bunch, and roll around your car floor. AVOID."
    - This task demonstrates the utility of the dataset in summarizing textual information for better content understanding and presentation.
3. **Recommendation System Enhancement:**
    - Models such as **BLaIR-roberta-large** were pre-trained using this dataset to improve item recommendations and retrieval scenarios. BLaIR specializes in bridging item metadata with language context. The dataset enabled training on 570 million reviews and 48 million items from 33 categories.

By employing the dataset for tasks ranging from sentiment classification to advanced recommendation system training, researchers have illustrated this dataset’s

flexibility and importance in modern NLP and recommendation research applications.

Similar Datasets Studied in the Past

Sentiment analysis tasks often involve use of datasets that pair textual reviews with ratings or sentiment labels. The dataset used for our analysis — comprising 70,000 customer reviews(later trimmed to 30,000 reviews) categorized into Positive, Neutral, and Negative sentiments based on numerical ratings—shares structural similarities with datasets used in prior research papers.

Twitter Airline Sentiment Dataset

The Twitter Airline Sentiment Dataset (Acosta et al.)[2] contains over 14,000 tweets about U.S. airlines, labeled as Positive, Neutral, or Negative. While smaller than the dataset in our report (Amazon Reviews Dataset 2023), it presents challenges such as:

- Imbalanced sentiment distribution: Negative sentiments dominate the dataset, akin to how Neutral sentiments are underrepresented in our dataset.
- Short text length: Tweets are limited to 280 characters, making them more concise and contextually ambiguous than the longer reviews in our dataset.

This dataset was analyzed using Word2Vec embeddings combined with machine learning classifiers like Logistic Regression, Support Vector Machines, and Naive Bayes. The Word2Vec model, particularly in its skip-gram implementation, was shown to capture semantic relationships effectively. However, its performance was hindered by the brevity of tweets, which provided limited context. Logistic Regression with Word2Vec achieved respectable accuracy (~72%) but struggled to differentiate Neutral from Positive sentiments—a problem also observed with our sentiment analysis when using Word2Vec embeddings alone.

Commuterline and Transjakarta Sentiment Dataset

Cahyani et al. studied sentiment classification using Indonesian-language datasets[3] collected from tweets about public transportation services. While these datasets focused on emotion classification (e.g., happy, angry, sad) rather than sentiment polarity, the methodology overlaps significantly with sentiment analysis tasks. These datasets included both short-form social media posts and sparse, emotion-rich language patterns, analogous to reviews expressing nuanced or mixed sentiments.

This research concluded the following:

- TF-IDF outperformed Word2Vec embeddings, especially on smaller datasets, as TF-IDF efficiently captured word importance and frequency patterns without requiring large corpora for training.
- Word2Vec embeddings demonstrated limitations in representing infrequent words or phrases, which are crucial for emotion and sentiment detection in specific contexts.
- Combining TF-IDF with other features significantly improved classification performance, particularly in distinguishing closely related categories.

Insights and Comparisons

Both the Twitter Airline and Commuter Line datasets align closely with our dataset regarding structure and sentiment categorization. However, our dataset's larger size and richer textual context provide more robust data for feature extraction and analysis. From these studies, we see that:

- Imbalanced datasets pose challenges for both traditional (TF-IDF) and semantic (Word2Vec) feature extraction methods, especially in classifying underrepresented classes like Neutral sentiment.
- Short text samples, such as tweets, limit the utility of Word2Vec embeddings, whereas longer texts offer richer semantic contexts, allowing Word2Vec to perform closer to its potential.
- Combining TF-IDF and Word2Vec features consistently demonstrates superior performance, as this hybrid approach balances the strengths of frequency-based and contextual representations. This aligns with the findings of our report,

where the combined model achieved the highest accuracy (86%) across all sentiment categories.

State-of-the-Art Methods
State-of-the-art methods for sentiment analysis include a combination of traditional machine learning and advanced deep learning techniques. Popular approaches involve **preprocessing** text with TF-IDF, which identifies important words and phrases, and Word2Vec, which captures semantic relationships between words. **Machine learning models** like Logistic Regression and SVM remain effective for simpler datasets, while transformer-based models like BERT dominate more complex tasks by understanding word context in depth. **Hybrid approaches** that combine TF-IDF and Word2Vec features often outperform standalone methods by balancing statistical and semantic insights. These methods align with findings from our analysis, where the combined features and Logistic Regression delivered strong performance.

5. Describe your results

TF-IDF Features with Logistic Regression

The first approach used TF-IDF to transform reviews into numerical vectors, capturing frequency-based patterns across unigrams, bigrams, and trigrams. This representation enabled the logistic regression model to focus on sentiment-laden terms and their contextual usage, resulting in an accuracy of **85.73%**. The classification report highlighted strong performance for Positive and Negative sentiments, with precision and recall exceeding **0.73** for these classes. However, Neutral sentiments remained challenging due to overlapping vocabulary with Positive reviews, achieving only **10% recall**. Despite this limitation, the TF-IDF-based model provided a robust baseline for text-based sentiment classification.

Word2Vec Features with Logistic Regression

Next, Word2Vec embeddings were employed to encode semantic relationships between words. By averaging the embeddings of all words in a review, we generated a compact vector representing the overall semantic content of the text. The logistic regression model trained on these embeddings achieved an accuracy of **80.02%**, slightly lower than the TF-IDF approach. While it excelled at capturing context for Positive sentiments (precision: **0.84**, recall: **0.95**), its performance for Neutral and Negative classes was weaker, with Neutral recall dropping to **4%**. These results underscored the challenge of using semantic embeddings for short, context-dependent reviews but highlighted their ability to model nuanced sentiment relationships.

To include the Mean Squared Error (MSE) of the combined model (TF-IDF with Word2Vec) in the description, we can insert it into the appropriate section as follows:

Combined Features with Logistic Regression

To combine the strengths of TF-IDF and Word2Vec, we concatenated their representations into a single feature set. This approach achieved an accuracy of **85.67%**, nearly matching the TF-IDF model. The Mean Squared Error (MSE) for this combined model was calculated as **0.359**, reflecting the small average squared difference between predicted and true sentiment values. The combined model maintained strong performance for Positive and Negative classes while slightly improving Neutral sentiment classification (recall: **11%**). Despite the marginal improvement, this approach demonstrated the potential of integrating frequency-based and semantic features to capture diverse aspects of sentiment.

Weighted Combined Features with Hyperparameter Tuning

Finally, we scaled Word2Vec embeddings, applied weighing (80% TF-IDF, 20% Word2Vec), and tuned logistic regression hyperparameters using GridSearchCV. This optimized model achieved an accuracy of **85.83%**, the highest among all approaches. The model demonstrated noticeable improvements in Neutral sentiment classification (recall: **16%**, precision: **43%**), addressing one of the key challenges identified in earlier models. This result highlights the importance of feature integration and parameter tuning in enhancing model performance.

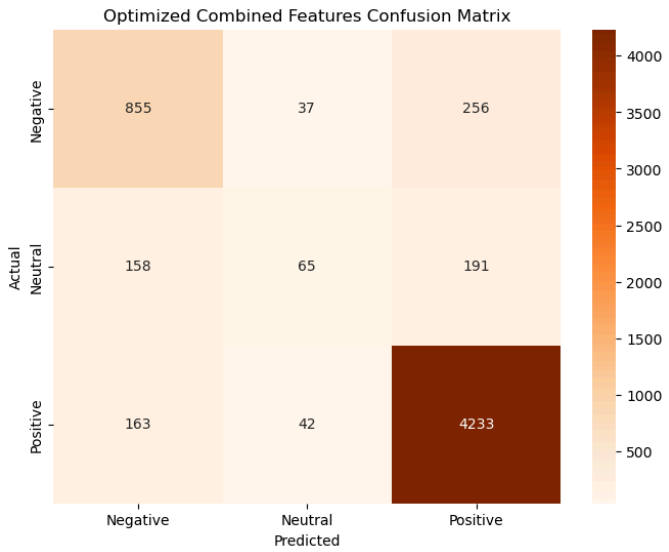
Conclusion

The transition from linear regression to logistic regression, coupled with the integration of TF-IDF and Word2Vec features, demonstrates the power of combining classification-optimized models with advanced feature extraction techniques. Logistic regression, with its probabilistic foundation and ability to model non-linear boundaries, proved effective for this multi-class classification problem. By incorporating diverse textual features, the models progressively improved in capturing nuanced sentiment relationships, culminating in a robust and balanced solution for sentiment analysis.

```
Best Parameters: {'C': 10, 'max_iter': 200, 'solver': 'liblinear'}
Optimized Logistic Regression Classification Report:
              precision    recall  f1-score   support

    Negative       0.72      0.74      0.73      1148
     Neutral       0.43      0.16      0.24       414
    Positive       0.91      0.95      0.93      4438

    accuracy                           0.86      6000
   macro avg       0.69      0.62      0.63      6000
weighted avg       0.84      0.86      0.84      6000
```



References

[1] Hou, Y., Li, J., He, Z., Yan, A., Chen, X., & McAuley, J. (2024). Bridging language and items for retrieval and recommendation. arXiv preprint arXiv:2403.03952.

[2] Acosta, J., Lamaute, N., Luo, M., Finkelstein, E., & Cotoranu, A. (2017). Sentiment Analysis of Twitter Messages Using Word2Vec.

[3] Cahyani, D., & Patasik, I. (2021). Performance comparison of TF-IDF and Word2Vec models for emotion text classification. *Bulletin of Electrical Engineering and Informatics, 10*(5), 2780-2788. doi:https://doi.org/10.11591/eei.v10i5.3157