Sentiment Analysis on McDonald's Reviews

Objective:

The primary objective of this project is to perform sentiment analysis on McDonald's customer reviews collected from across the US. The data is sourced from Kaggle. This analysis helps businesses like McDonald's understand customer satisfaction, identify areas for improvement (e.g., food quality, service), and make data-driven decisions. The project aims to create a model which accurately predicts the sentiment of a particular review.

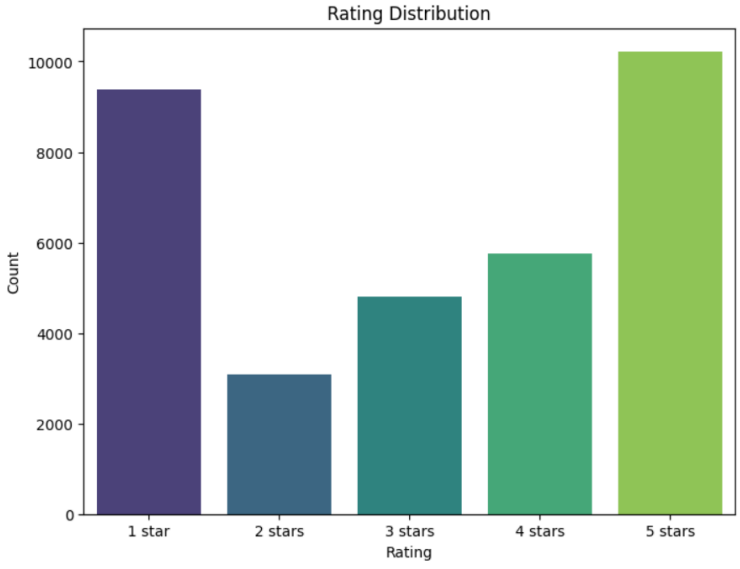
Data Description:

The dataset consists of customer reviews from McDonald's locations across the US, with approximately 33,000 rows. Key features include:

- **reviewer_id**: Unique identifier for each reviewer (anonymized)
- **store_name**: Name of the McDonald's store
- **category**: Category or type of the store
- **store_address**: Address of the store
- **latitude**: Latitude coordinate of the store's location
- **longitude**: Longitude coordinate of the store's location
- **rating_count**: Number of ratings/reviews for the store
- **review_time**: Timestamp of the review
- **review**: Textual content of the review
- **rating**: Rating provided by the reviewer

Exploratory Data Analysis (EDA):

Ratings Distribution:



Review Lengths:

Short texts (median: 11 words, 90th percentile: 40 words). This analysis helped set the max length for padding as 50.

Review Language:

Primarily English (76.94%, ~25,590), with non-English (~23%) filtered out for English-centric preprocessing.

Data Cleaning and Feature Engineering:

Special Characters/Emojis Removal:

Removed non-ASCII characters (e.g., "fries 😊 " → "fries").

Contractions Expansion:

Used a custom dictionary with regex (e.g., "don't" → "do not") to standardize text, as `contractions` library installation was challenging.

Lowercasing and Noise Removal

Converted to lowercase, removed URLs, normalized spaces.

Lemmatization and Stopwords:

Used spaCy for lemmatization and NLTK for English stopwords.

Empty Rows Handling:

Filtered ~18 empty sequences after tokenization to avoid invalid inputs.

Language Filtering:

Used “langdetect” to identify English reviews (76.94%, ~25,590), removing non-English (~23%) for English-centric models.

Numerical rating to sentiment conversion:

Converted the numerical 1-5 star rating to sentiment by labelling 4,5 stars as “positive”, 3 star as “neutral” and 1,2 star as “negative” sentiment. This gave the distribution below for sentiment labels

Positive (~42%, 11,066), Negative (~42%, 11,024), Neutral (~16%, 3,518).

Re-Labelling 3-Star Reviews:

Later when implementing models, observed that some of the assigning all 3 stars to neutral might be wrong, as the true sentiment from the review text might be different. So used VADER to re-label neutrals ~53% positive, ~18% negative, ~28% neutral, reducing neutral class to ~993.

Model Fitting and Results:

Implemented several models, starting with baselines and moving to deep learning:

Model 1: Logistic Regression (Baseline):

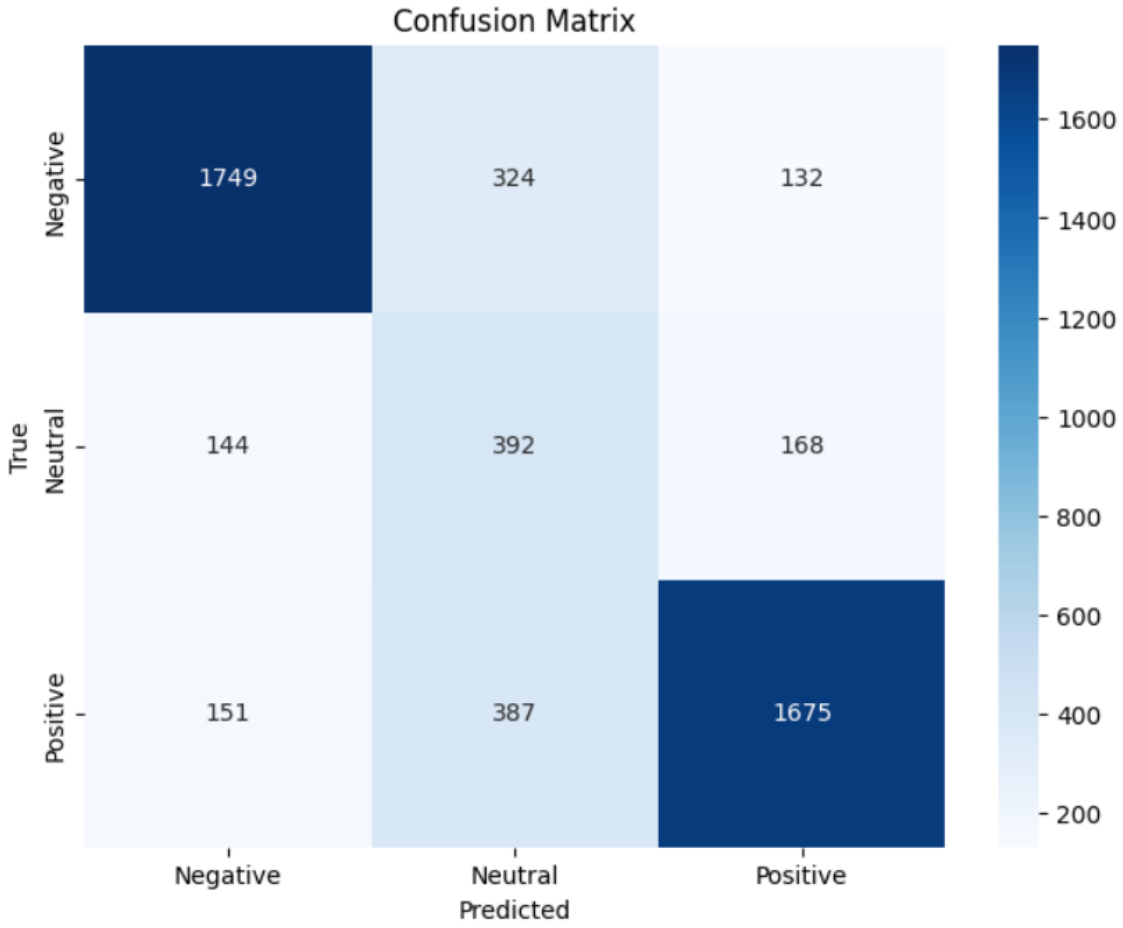
Used “TfidfVectorizer(max_features=5000)”, capturing word importance.

Tuned ‘max_iter=1000’, ‘class_weight='balanced''.

Results:

Accuracy: 0.75, Precision: 0.78, Recall: 0.75, F1: 0.76

Confusion Matrix:



Model 2: SVM (LinearSVC):

Used the same TfIDF for feature extraction.

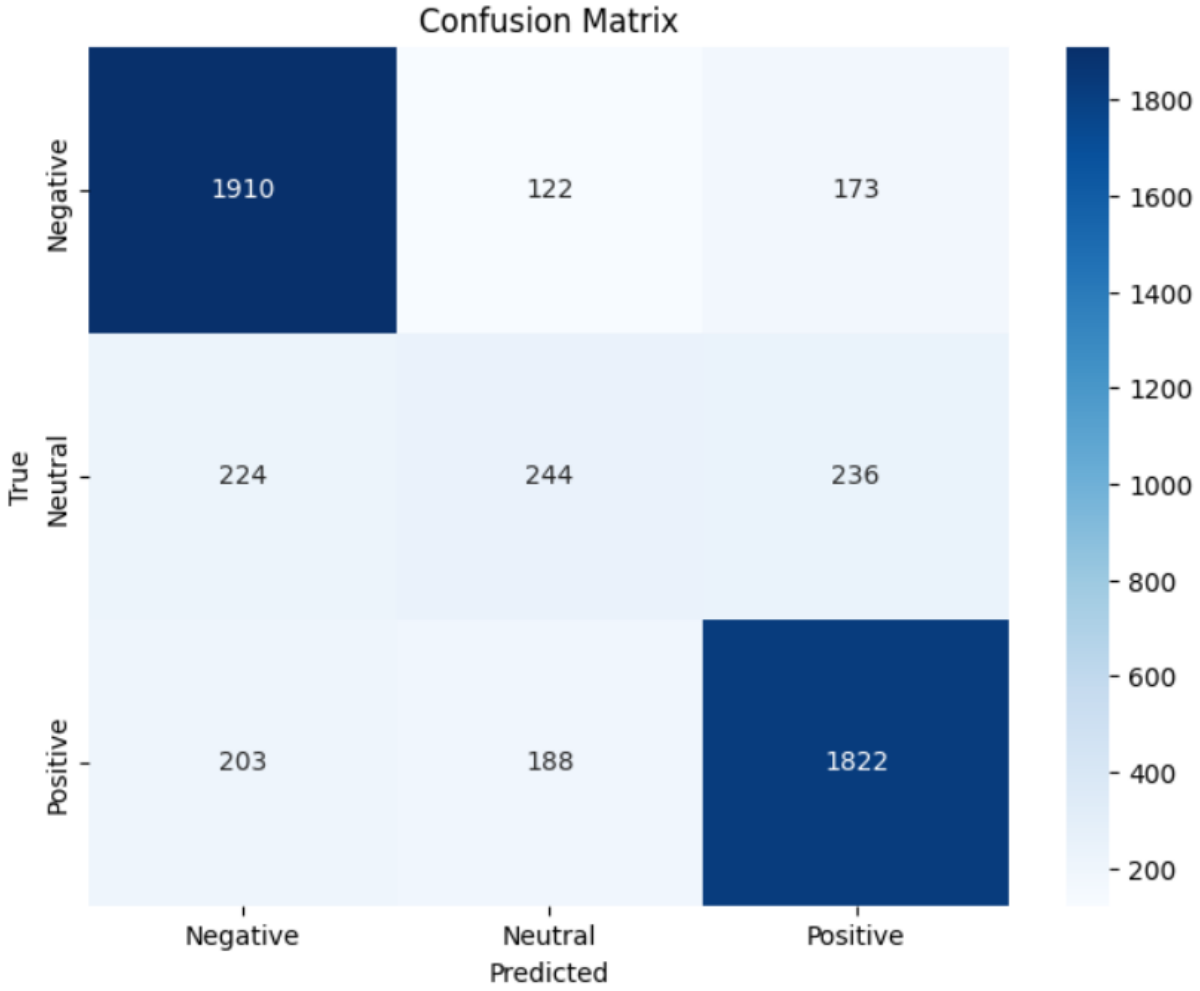
Tuned the “c” value and “max_iter” using GridSearchCV

Results:

Best params: {'C': 0.1, 'max_iter': 1000}

SVM - Accuracy: 0.78, Precision: 0.77, Recall: 0.78, F1: 0.77

Confusion Matrix:



Model 3: LSTM:

Implemented LSTM with Tokenizer to extract the features.

Used Softmax activation function and “categorical_crossentropy” as loss function.

Results:

LSTM - Accuracy: 0.76

Weighted - Precision: 0.78, Recall: 0.76, F1: 0.77

Macro - Precision: 0.69, Recall: 0.70, F1: 0.69

Per-Class Metrics:

Negative: Precision=0.85, Recall=0.82, F1=0.83 (Support: 2205)

Neutral: Precision=0.39, Recall=0.47, F1=0.42 (Support: 702)

Positive: Precision=0.83, Recall=0.80, F1=0.81 (Support: 2212)

Model4: BERT:

Transformer model with Hugging Face.

Fine-tuned “nlptown/bert-base-multilingual-uncased-sentiment” with Optuna (`num_train_epochs`, `batch_size`, `learning_rate`).

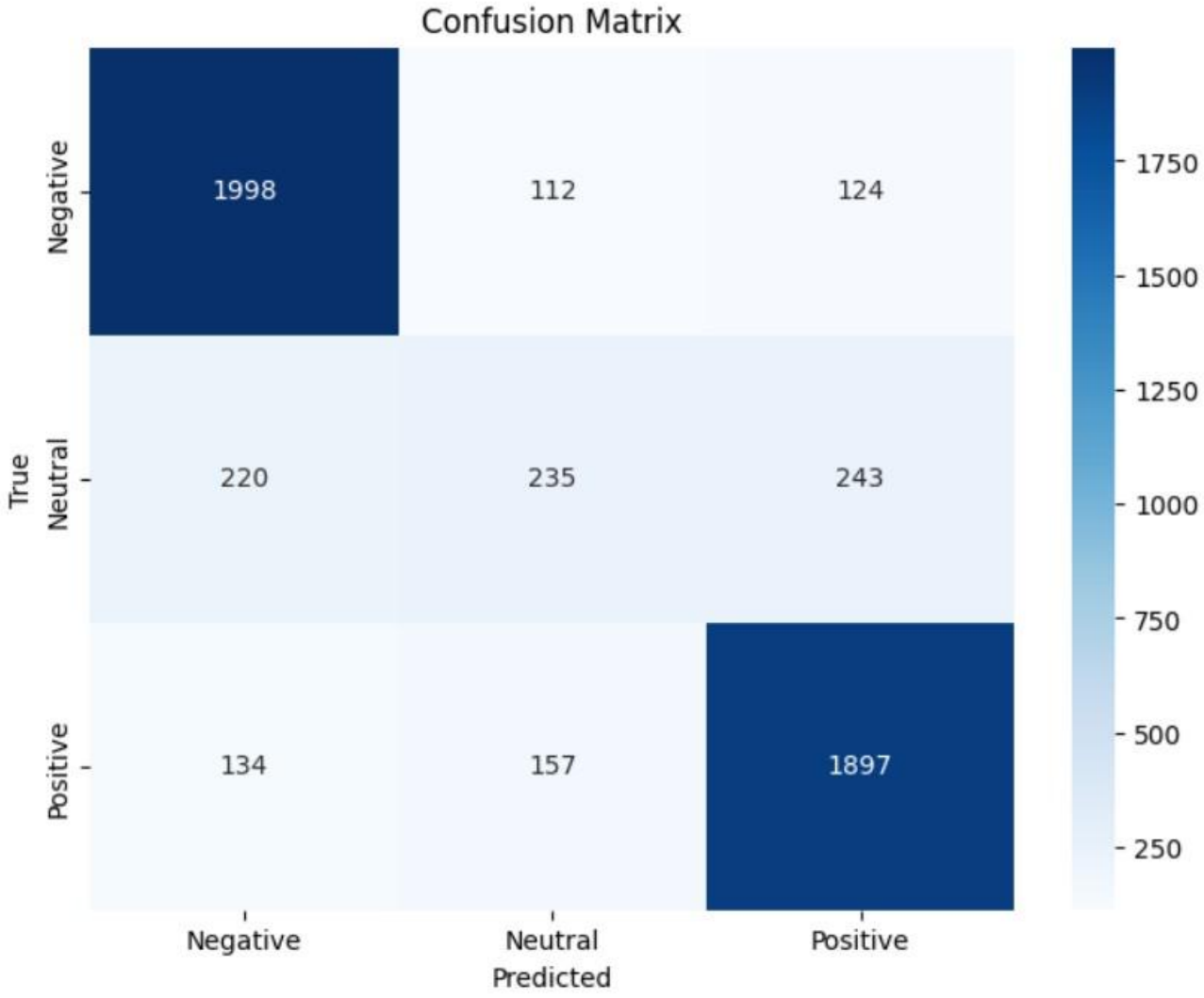
Results:

BERT - Eval Loss: 0.53, Accuracy: 0.81

Weighted - Precision: 0.79, Recall: 0.81, F1: 0.80

Macro - Precision: 0.72, Recall: 0.70, F1: 0.70

Confusion Matrix:



Model 5: BERT on Re-labelled Data:

Retrained BERT after relabelling the 3 star reviews as discussed in Feature Engineering Section

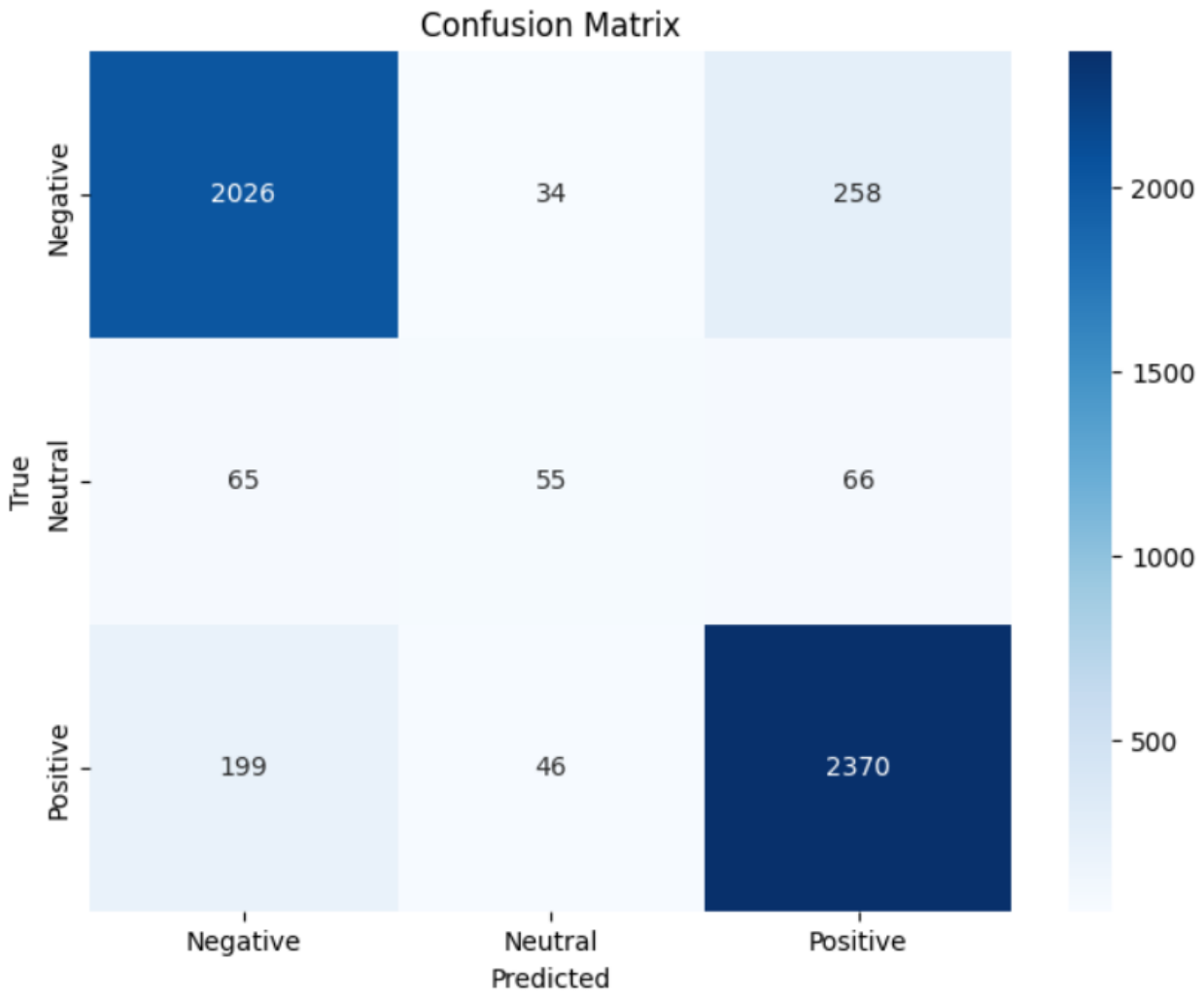
Results:

BERT - Eval Loss: 0.39, Accuracy: 0.87

Weighted - Precision: 0.86, Recall: 0.87, F1: 0.87

Macro - Precision: 0.72, Recall: 0.69, F1: 0.70

Confusion Matrix



BERT on re-labelled data improved weighted F1 (0.80 → 0.87) but macro F1 remained 0.70 due to neutral class challenges.

Final Model Selection:

The final model selected is BERT (fine-tuned `nlptown/bert-base-multilingual-uncased-sentiment` with 3 classes):

Performance: Highest weighted F1 (0.87) and accuracy (0.87), outperforming Logistic Regression (weighted F1 ~0.80) on the dataset's distribution.

Contextual Understanding: Captures nuances in short reviews (median: 11 words) better than non-transformer models like LSTM (weighted F1 0.77).

Multilingual Capability: Handles any residual non-English snippets, though dataset is filtered to have only English reviews.

Re-Labelling Benefit: Improved with VADER re-labeling, reducing neutral ambiguity.

Drawbacks Addressed: Neutral F1 (0.39) is low, but weighted metrics prioritize majority classes, aligning with real-world review distribution.

Confusion Matrix Insights: Strong on positive (2370 correct) and negative (2026 correct), but neutral (55 correct) is weak with high misclassifications (131 false negatives, 269 false positives).

Conclusions:

- This project successfully implemented sentiment analysis on McDonald's reviews, identifying positive/negative sentiment well but struggling with neutral reviews due to their ambiguity.
- Re-labelling with VADER revealed that 3-star reviews often lean positive/negative, improving weighted F1 but not macro F1.
- BERT is the best model for its contextual strength, but the neutral class remains challenging, likely due to data limitations (short reviews, imbalance).
- Overall, the models provide valuable insights for McDonald's to focus on customer pain points.

Next Steps:

Data Expansion: Collect more neutral reviews or use augmentation (e.g., EDA) to balance classes.

Advanced Models: Experiment with Grok-4 or Llama-3.1 for potential F1 boosts to 0.85+.

Neutral Sub-Model: Build a separate binary model for neutral detection to improve the classification of neutral class.

Monitoring: Retrain periodically with new data to adapt to changing sentiment trends.