Project NLP: Automated Customer Reviews Classification

Collaborators:

Ginosca Alejandro Dávila
Natanael Santiago Morales

Ironhack Bootcamp:
Data Science and Machine Learning

Project Overview

Automated Customer Reviews Sentiment Classification

- **Goal:** Classify customer reviews from Amazon US Reviews as **Negative, Neutral, or Positive**.

- **Approach:** Compare **traditional machine learning models** with **BERT**.

Dataset

🔍 Dataset

- **Source:** Customer Reviews of Amazon Products (Kaggle)
- **Features Used:**
  - `reviews.text` (Main review text)
  - `reviews.title` (Review title)
  - `reviews.rating` (Star rating converted to sentiment labels)
- **Label Encoding:**
  - **1, 2, 3 → Negative (0)**
  - **4 → Neutral (1)**
  - **5 → Positive (2)**

Data Cleaning

Key Cleaning Steps

- Dropped irrelevant columns to remove unnecessary metadata.
- Removed duplicates based on the joint **review title & review text**.
- Checked for missing values – no missing values found in the dataset.
- Validated ratings to ensure all values were within the expected **1-5 range**.

Machine Learning Models Approach

Data Preprocessing for ML Models

1. Train-Test Split - Split dataset into 80% training, 20% test before any transformations to prevent data leakage
2. Label Encoding - Converted reviews.rating into sentiment labels:
   - 1, 2, 3 → Negative (0)
   - 4 → Neutral (1)
   - 5 → Positive (2)
3. Text Preprocessing - Applied the following transformations
   - Converted text to lowercase for consistency
   - Removed special characters, punctuation, and extra whitespace.
   - Removed stopwords to retain meaningful words.
   - Applied lemmatization to reduce words to their base form.
4. TF-IDF Vectorization - Converted cleaned and preprocessed text into numerical features using TF-IDF with:
   - Unigrams and Bigrams
   - Max features = 5000 to optimize performance
5. Final Representation - Combined TF-IDF features from both reviews.text and reviews.title to enrich the model's input.

Machine Learning Model Training

Trained and evaluated multiple ML models:

- **Naïve Bayes** (NB) – Baseline model for text classification.
- **Logistic Regression** – Strong linear model for text-based sentiment analysis.
- **Support Vector Machine (SVM)** – Effective in high-dimensional TF-IDF spaces like TF-IDF vectors.
- **Random Forest** – Ensemble model capturing non-linear patterns.
- **XGBoost** – Gradient boosting, good generalization.
- **LightGBM** – Efficient and high-performing boosting model.

Machine Learning Model Evaluation and Selection

- Metrics used: Accuracy, Precision, Recall, F1-score, Confusion Matrix

Model	Accuracy	Precision	Recall	F1-Score
Naïve Bayes	75.03%	71.42%	75.03%	69.10%
Logistic Regression	76.91%	74.34%	76.91%	72.33%
SVM	76.16%	72.23%	76.16%	72.02%
Random Forest	74.54%	69.45%	74.54%	67.51%
XGBoost	76.02%	73.45%	76.02%	70.73%
LightGBM	76.21%	72.98%	76.21%	72.83%

- Best Model: Logistic Regression (Accuracy: 76.91%, F1-score: 72.33%)
- Challenges

Per-Class Performance Across Models

- We evaluated the classification performance of each model for Negative, Neutral, and Positive sentiments using **Precision, Recall, and F1-Score**.

Model	Negative Precision	Negative Recall	Negative F1	Neutral Precision	Neutral Recall	Neutral F1	Positive Precision	Positive Recall	Positive F1
Naïve Bayes	85.00%	41.98%	56.20%	47.92%	9.90%	16.41%	75.59%	97.55%	85.18%
Logistic Regression	86.40%	53.33%	65.95%	55.61%	15.64%	24.41%	77.46%	96.93%	86.11%
SVM	75.32%	58.77%	66.02%	47.62%	15.78%	23.71%	78.33%	94.98%	85.85%
Random Forest	79.46%	43.95%	56.60%	43.04%	4.88%	8.76%	74.95%	97.89%	84.90%
XGBoost	79.83%	45.93%	58.31%	57.86%	13.20%	21.50%	76.63%	97.47%	85.80%
LightGBM	76.87%	55.80%	64.66%	49.49%	20.80%	29.29%	78.66%	94.18%	85.72%

Per-Class Performance Across Models

- **Logistic Regression performs best overall**, achieving **high precision and recall balance** across all sentiment classes.
- **XGBoost shows strong Neutral Precision (57.86%)**, making it slightly better at distinguishing Neutral reviews than other models.
- **LightGBM provides better recall for Negative (55.80%) and Neutral (20.80%) sentiments compared to Random Forest.**
- **All models struggle with Neutral classification**, but XGBoost and Logistic Regression handle it slightly better.
- **Naïve Bayes, despite being a simple model, still provides a competitive baseline.**

Best Machine Learning Model: Logistic Regression

After training and evaluating multiple models, **Logistic Regression** emerged as the best-performing ML model for sentiment classification.

Why Logistic Regression?

- **Highest Accuracy:** 76.91%
- **Balanced Precision & Recall:** Best tradeoff between false positives and false negatives
- **Strong F1-Score:** 72.33%, outperforming other ML models in overall performance
- **Computationally Efficient:** Faster training and inference time compared to ensemble models
- **Consistent Across Classes:** Performs well on Negative, Neutral, and Positive sentiments
-

Next Steps & Future Improvements

1. Hyperparameter Tuning
   - Optimize Logistic Regression and other ML models to improve accuracy and F1- Score
   - Fine-tune regularization parameters to reduce misclassifications.
2. Feature Engineering
   - Explore n-grams, word embeddings, or sentiment lexicons to enhance model inputs.
   - Identify the most influential words contributing to classification decisions.
3. Model Deployment
   - Convert the best ML model into a deployable API for real-time sentiment analysis.
   - Integrate with customer feedback platforms for automated review insights.
4. Deep Learning Integration
   - Evaluate BERT's performance against ML models for sentiment classification.
   - Fine-tune BERT to handle complex sentence structures and improve Neutral sentiment detection.
5. Address Class Imbalance
   - Implement resampling techniques to improve classification of Neutral sentiment.
   - Adjust loss functions to reduce bias toward positive reviews.

Final Goal: Build a robust, scalable, and interpretable sentiment analysis model for real-word applications.

Transformer Approach

Data pre-processing

- Remaining columns:
    - text and rating
- Combined reviews.title and reviews.text
    - (title) text
    - Removed tags and encodings
- Three ratings: negative, neutral, positive
    - Later encoded
- Hugging Face Dataset

	text	rating
0	(Kindle) This product so far has not disappoin...	positive
1	(very fast) great for beginner or experienced ...	positive
2	(Beginner tablet for our 9 year old son.) Inex...	positive
3	(Good!!!) I've had my Fire HD 8 two weeks now ...	neutral
4	(Fantastic Tablet for kids) I bought this for ...	positive

Pre-processing

- Selected BERT model
  - “bert-based-uncased”
  - "nlptown/bert-base-multilingual-uncased-sentiment"
- Tokenized
- Reformat HuggingFace Dataset to use with PyTorch

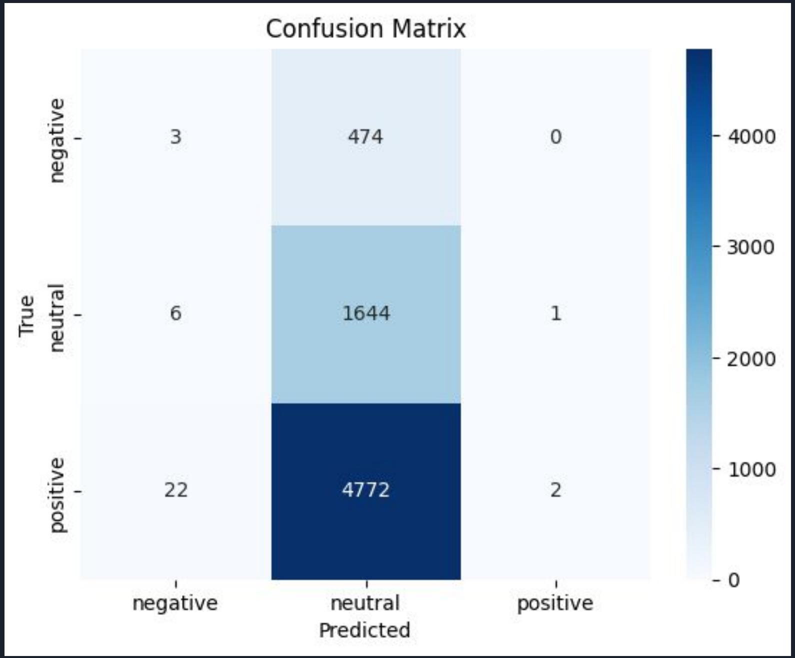
```python
# [3. Model & Tokenizer] -----------------------------------------
model_name = "bert-base-uncased"  # For English reviews_hugging
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(
    model_name,
    num_labels=3,
    id2label={i: label for i, label in enumerate(le.classes_)}
)

```

BERT based uncased

Accuracy: 0.2382
Macro F1: 0.1325

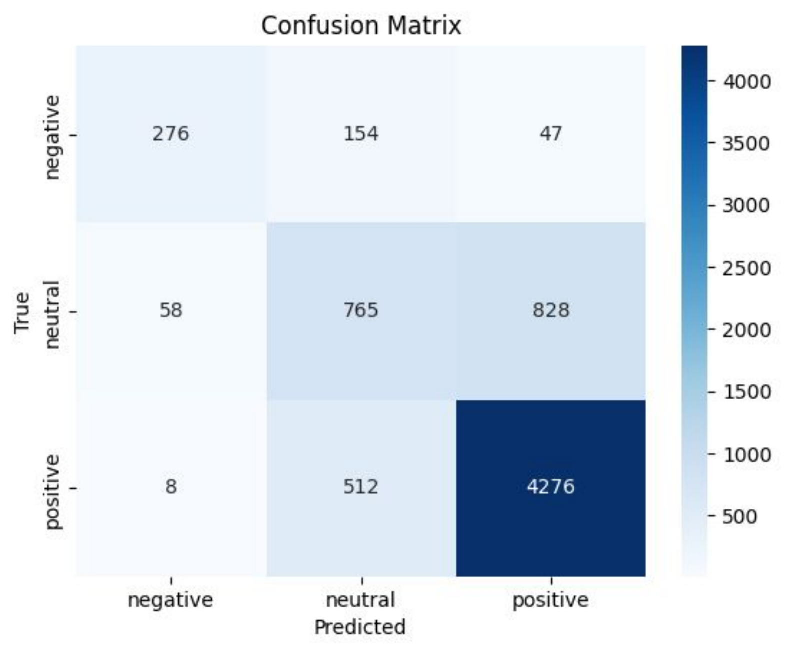
Classification Report:

	precision	recall	f1-score	support
negative	0.10	0.01	0.01	477
neutral	0.24	1.00	0.38	1651
positive	0.67	0.00	0.00	4796
accuracy			0.24	6924
macro avg	0.33	0.33	0.13	6924
weighted avg	0.53	0.24	0.09	6924

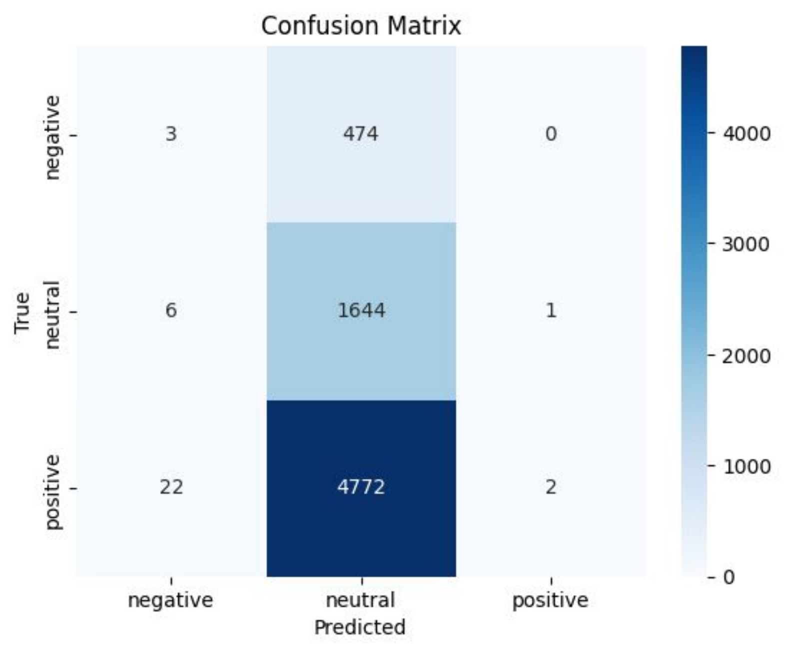


BERT based uncased
(“fine”-tuned)

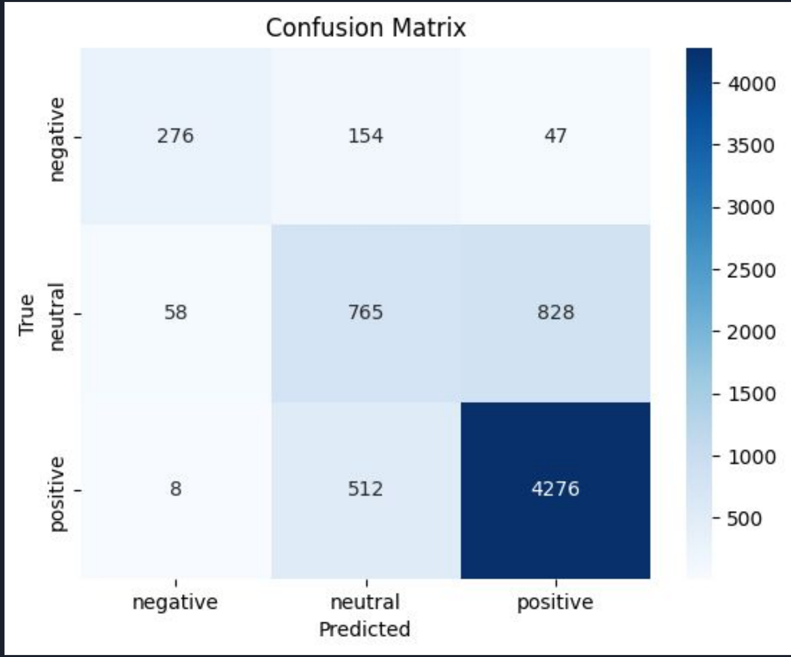
```
Accuracy: 0.7679
Macro F1: 0.6767

Classification Report:
              precision    recall  f1-score
support

    negative       0.81      0.58      0.67
477
     neutral       0.53      0.46      0.50
1651
    positive       0.83      0.89      0.86
4796

    accuracy                           0.77
6924
   macro avg       0.72      0.64      0.68
6924
weighted avg       0.76      0.77      0.76
6924
```



BERT based uncased



Base model

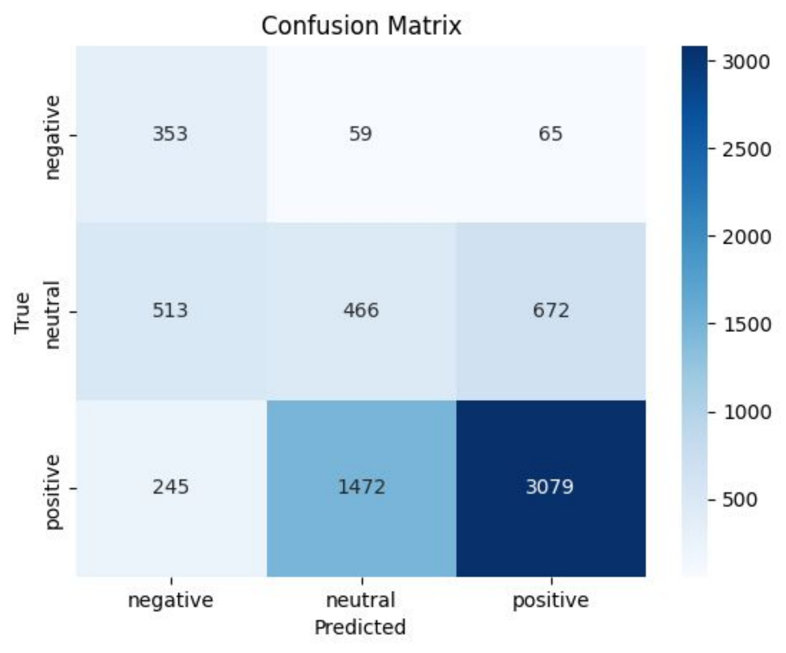


“fine”-tuned

BERT base multilingual uncased sentiment

nlptown/bert-base-multilingual-uncased-sentiment

```
Accuracy: 0.5630
Macro F1: 0.4717
              precision    recall  f1-score   support

    negative       0.32      0.74      0.44       477
     neutral       0.23      0.28      0.26      1651
    positive       0.81      0.64      0.72      4796

    accuracy                           0.56      6924
   macro avg       0.45      0.55      0.47      6924
weighted avg       0.64      0.56      0.59      6924
```

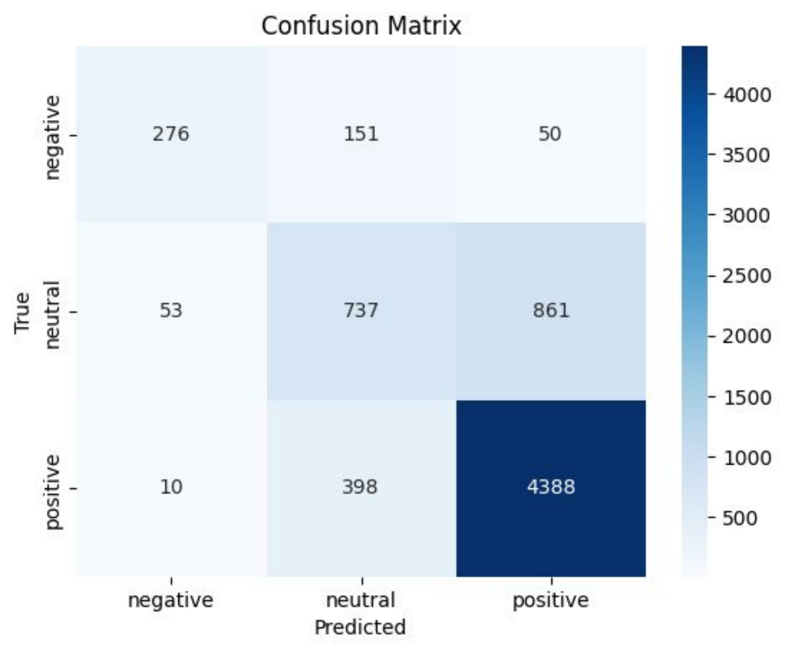


BERT base multilingual uncased sentiment ("fine"-tuned)

nlptown/bert-base-multilingual-uncased-sentiment

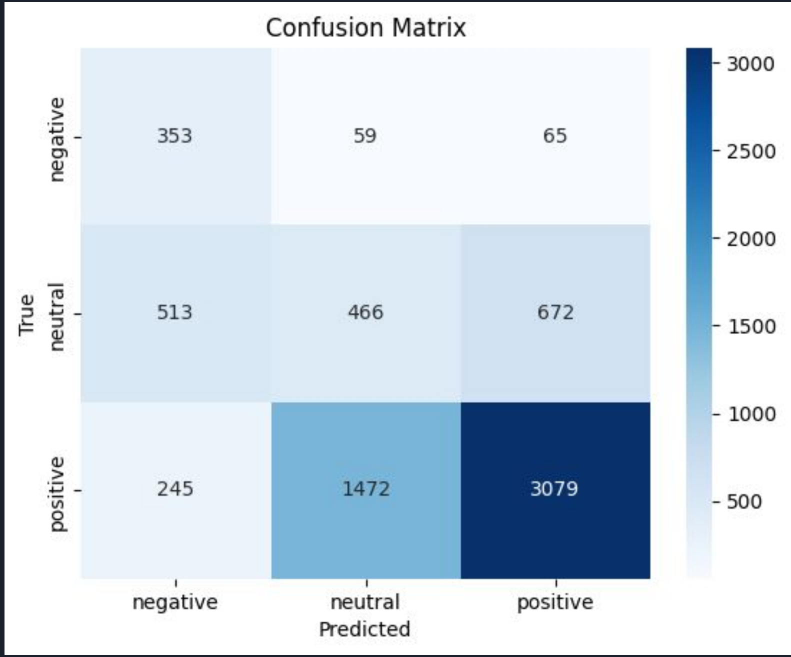
Accuracy: 0.7800
Macro F1: 0.6826

	precision	recall	f1-score	support
negative	0.81	0.58	0.68	477
neutral	0.57	0.45	0.50	1651
positive	0.83	0.91	0.87	4796
accuracy			0.78	6924
macro avg	0.74	0.65	0.68	6924
weighted avg	0.77	0.78	0.77	6924

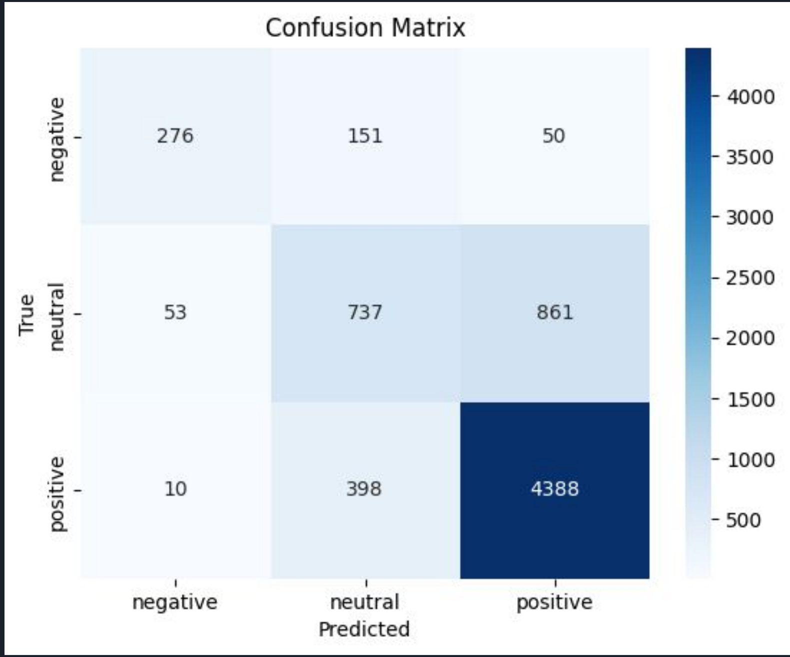


BERT base multilingual uncased sentiment ("fine"-tuned)

nlptown/bert-base-multilingual-uncased-sentiment

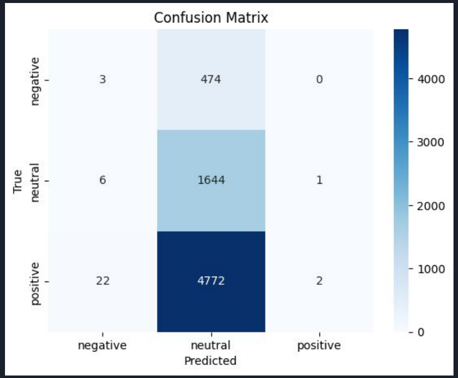


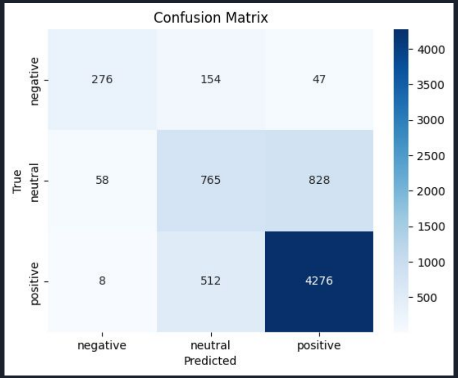
Base model



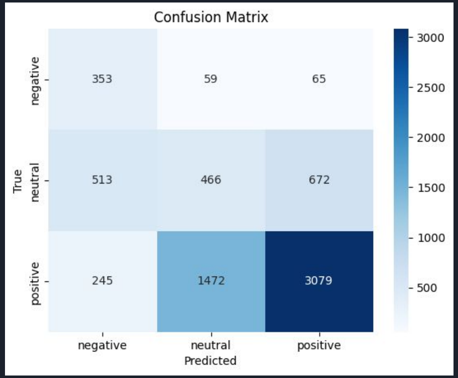
"fine"-tuned

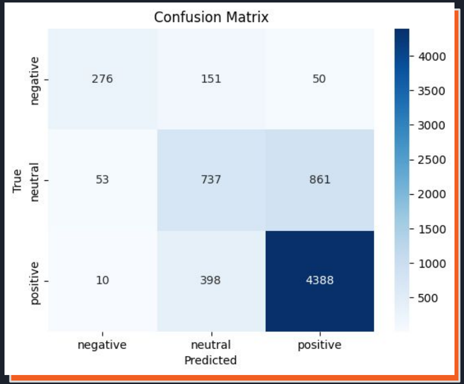
bert-base-uncased





nlptown/bert-base-multilingual
- uncased-sentiment





Base model

“fine”-tuned

The best model was the fine-tuned `nlptown BERT base uncased model`, with an accuracy of 0.78 and a macro-F1 of 0.68.

Comparison: Logistic Regression (Best ML Model) vs. BERT (Best Deep Learning Model)

Overall Performance Metrics

Model	Accuracy	Macro F1-Score
Logistic Regression (ML)	76.91%	72.33%
BERT (Deep Learning)	78.00%	68.26%

Accuracy: BERT performs slightly better than Logistic Regression (**78.00% vs. 76.91%**).

Macro F1-Score: Logistic Regression has a higher macro F1-score (**72.33% vs. 68.26%**), which suggests that it maintains a better balance across all three sentiment classes.

Per-Class Performance Comparison

Model	Negative Precision	Negative Recall	Negative F1	Neutral Precision	Neutral Recall	Neutral F1	Positive Precision	Positive Recall	Positive F1
Logistic Regression	86.40%	53.33%	**65.95%**	55.61%	15.64%	**24.41%**	77.46%	96.93%	**86.11%**
BERT	81.00%	**58.00%**	68.00%	57.00%	**45.00%**	50.00%	**83.00%**	**91.00%**	87.00%

Key Observations:

- BERT has higher accuracy, indicating it classifies overall sentiment slightly better.
- BERT significantly improves recall for the Neutral class (45.00% vs. 15.64%), meaning it correctly identifies more Neutral reviews.
- Logistic Regression maintains stronger overall balance, with a higher macro F1-score (72.33% vs. 68.26%), meaning it provides more consistent performance across all sentiment classes.
- BERT performs better for Positive sentiment, while Logistic Regression does better in Negative sentiment classification.

Conclusion

- In terms of accuracy, BERT performs slightly better.
- If balanced classification across all classes is the priority, Logistic Regression performs better.
- BERT significantly improves Neutral classification, which was the main challenge in ML models.
- BERT is expected to generalize better to more complex sentences, but it is computationally heavier.

Final Choice: BERT is a better choice overall, especially due to its ability to improve Neutral class recall, which was a major issue in ML models.

THANK YOU!

IRON
HACK