Healthcare Review Sentiment Analysis Report ~ Sailee Prashant Allyadwar

Objective:

To classify healthcare review texts into positive, neutral, or negative sentiments using Natural Language Processing and Machine Learning.

Dataset:

- Source: healthcare_reviews.csv
- Fields: Review_Text, Rating
- Ratings converted to Sentiment (Positive: 4-5, Neutral: 3, Negative: 1-2)

Steps:

1. Data Preprocessing

- Remove null reviews from the dataset.
- Clean text using regular expressions: lowercase, remove special characters and numbers.
- Tokenize the cleaned text into individual words.
- Remove stopwords using NLTK to keep only meaningful words.

2. Text Analysis

- Count and display the most frequent words.
- Create word clouds and bar plots to visualize common terms and rating distributions.

3. Sentiment Mapping

- Convert numeric ratings to sentiment labels:
    - Ratings 1–2 → Negative
    - Rating 3 → Neutral
    - Ratings 4–5 → Positive

4. Feature Extraction

- Use TF-IDF Vectorizer to convert text into numerical form.
- Include both unigrams and bigrams (`ngram_range=(1, 2)`).

5. Build Model Pipeline

- Create a pipeline that first applies TF-IDF, then trains a Naive Bayes classifier.

6. Define Hyperparameter Grid

- Test different values for:
  - `max_df` and `min_df` in TF-IDF
  - `alpha` and `fit_prior` in Naive Bayes

7. Grid Search with Cross-Validation

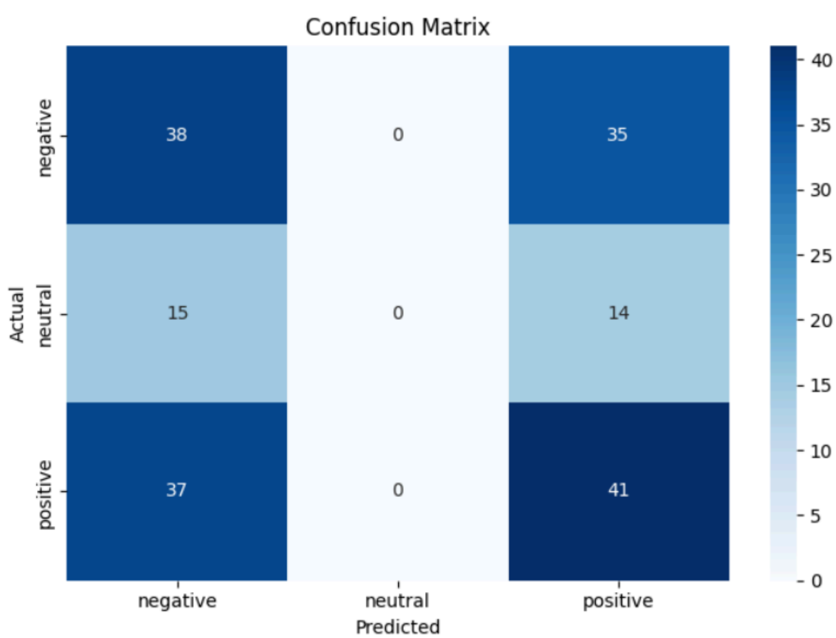
- Use GridSearchCV with 5-fold cross-validation.
- Automatically find the best combination of parameters.
- Fit the model using the best parameters on training data.

8. Save the Final Model

- Save the trained model using `joblib` as `sentiment_model.pkl` for future use or deployment.

Model Evaluation Summary

1. Before SMOTE – Naive Bayes Evaluation



ROC Curve:

- Class 0 (Negative): AUC = 0.51
- Class 1 (Neutral): AUC = 0.48
- Class 2 (Positive): AUC = 0.51
- These values are close to 0.5, meaning the model is no better than random guessing.

Confusion Matrix:

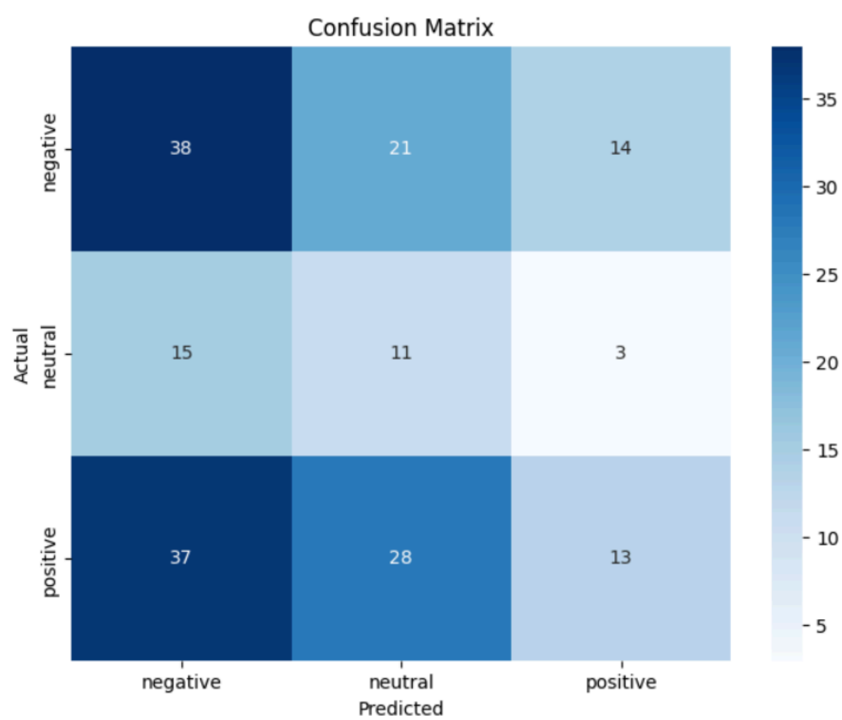
- The neutral class is never predicted.
- This is a common issue with imbalanced data where the model favors majority classes (positive and negative).

2. After Applying SMOTE (Neural Network)

Why SMOTE?

- You applied SMOTE (Synthetic Minority Oversampling Technique) to balance the classes by generating synthetic examples of the minority (neutral) class.

Confusion Matrix After SMOTE:



- Now all classes (negative, neutral, positive) are predicted.
- But accuracy dropped. This is expected because:
  - SMOTE increases recall for minority class but may reduce precision and overall performance.
  - Neural networks are sensitive to noise, and SMOTE may introduce synthetic samples that don't generalize well.

Key Insights

1. Original data imbalance led to poor neutral class predictions.
2. Naive Bayes performed better overall, even if it ignored the neutral class, due to simplicity and robustness.
3. Neural network + SMOTE increased class coverage, but at the cost of accuracy and reliability.
4. Best results might come from combining SMOTE with a simpler classifier (like Logistic Regression or Random Forest with tuning).
5. You could also try class weighting in the neural network instead of SMOTE.