Enhancing Emotion Recognition Using POS Tagging

Name: Pola Gnana Shekar
Roll No: 21CS10052
Course: Natural Language Processing (CS60075)
Assignment: 1

CONTENTS

Description:

This project explores how Part-of-Speech (POS) tagging can enhance emotion recognition in text. We'll build a custom POS tagger using the Viterbi algorithm and train it on a treebank corpus. Then, we'll develop a baseline emotion recognition model using TF-IDF features and a classifier (Naive Bayes or SVM). Finally, we'll integrate the POS tagger into a pipeline to enrich the features with grammatical information and compare the performance of this improved emotion recognition model to the baseline. The project aims to demonstrate the value of POS tagging in understanding the emotional context of text.

Task 1: POS Tagger Implementation

- **Description:**
  This task involved building a custom Part-of-Speech (POS) tagger from scratch. We leveraged the NLTK library's treebank corpus for training data. The core algorithm implemented is the Viterbi algorithm, a dynamic programming technique for efficiently finding the most probable sequence of POS tags for a given sentence.
- **Results:**
  We trained the custom POS tagger on the treebank dataset. As an example, the model successfully predicted the POS tags for the sentence "The love to be with him" as:
    - Sentence: The love to be with him
    - Predicted POS tags: ['DT', 'NN', 'TO', 'VB', 'IN', 'PRP']

  These predicted tags accurately reflect the grammatical roles of the words:
    - DT: Determiner (The)
    - NN: Noun (love)
    - TO: To preposition (to)
    - VB: Base verb (be)
    - IN: Preposition (with)
    - PRP: Personal pronoun (him)

These results demonstrate the functionality of the custom POS tagger in assigning appropriate POS tags to words in a sentence. The accuracy of the tagger on a larger dataset would require further evaluation.

Task 2: Vanilla Emotion Recognizer

❖ Description:

This task aimed to establish a baseline for emotion recognition using a simple approach that relies solely on TF-IDF features and classical machine learning models.

Key Steps:

1. **Data Preparation:**
   - Loaded the dair-ai/emotion dataset.
   - Split the dataset into training, validation, and test sets.
   - Extracted text data and corresponding emotion labels.
2. **Feature Engineering:**
   - Employed TF-IDF vectorization to transform text data into numerical features, capturing the importance of words based on their frequency.
3. **Model Training and Evaluation:**
   - Trained both Naive Bayes and SVM models on the TF-IDF features and labels.
   - Conducted hyperparameter tuning for max_features in the TF-IDF vectorizer to find the optimal setting.
   - Evaluated the models on the validation set to select the best performing model for each classifier.
   - Tested the final models on the test set and measured accuracy.
4. **Results Analysis:**
   - Analyzed the test accuracy for both Naive Bayes and SVM models.
   - Visualized classification reports and confusion matrices to understand model performance in detail.

❖ Results:

Dataset Statistics:

- Training samples: 16000
- Validation samples: 2000
- Test samples: 2000

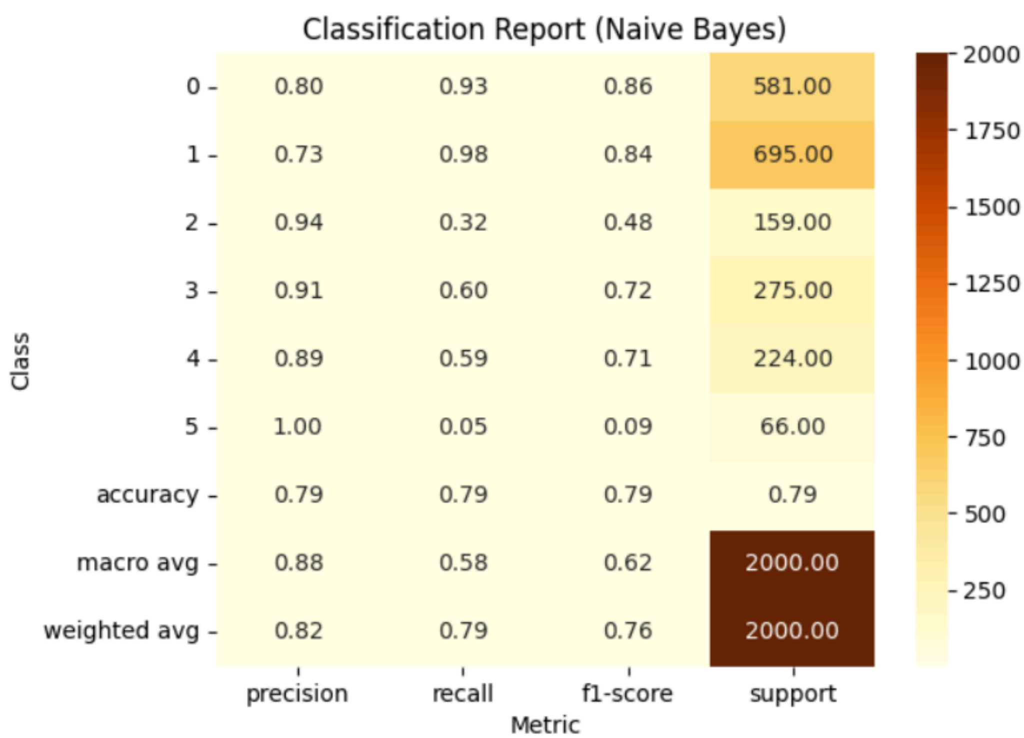
Hyperparameter Tuning:

- **Naive Bayes:**
  - Best max_features: 1250
  - Best validation accuracy: 0.7835
- **SVM:**
  - Best max_features: 5000
  - Best validation accuracy: 0.8875

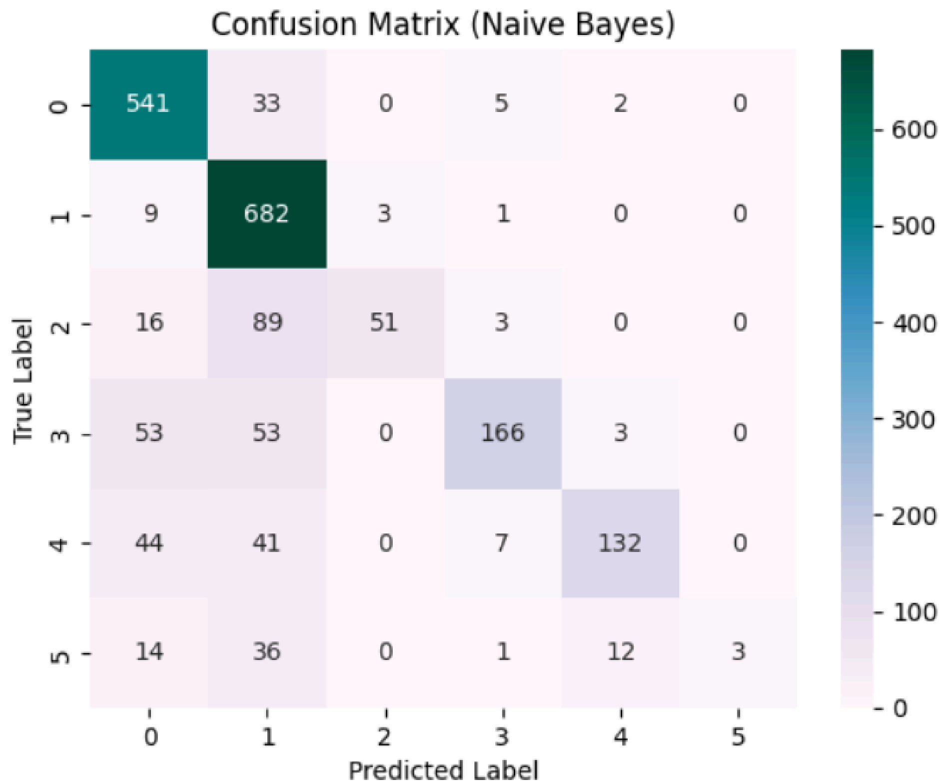
Test Accuracy:

- Naive Bayes: 0.7875
- SVM: 0.887

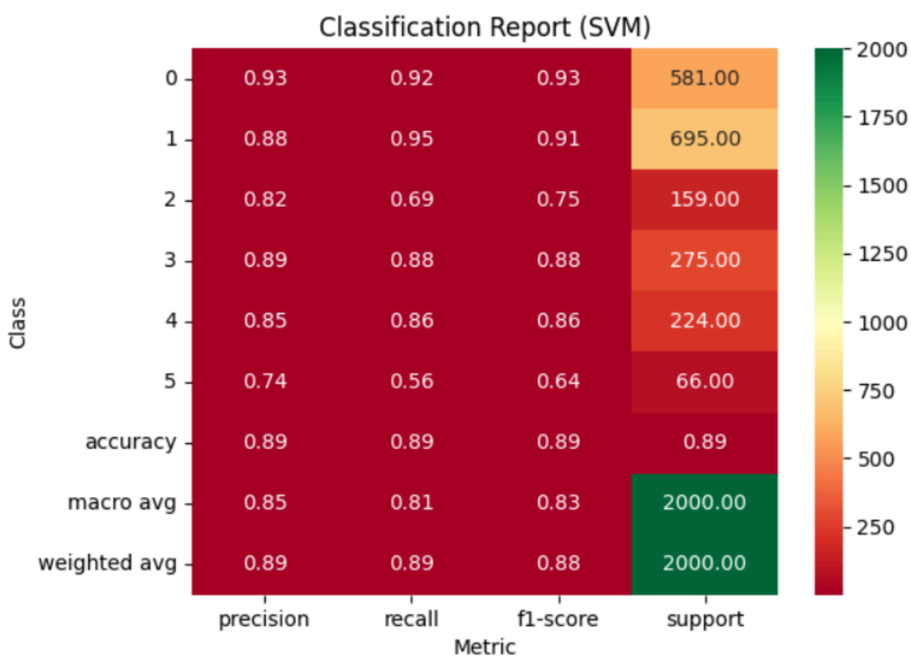
Classification Report (Naive bayes):



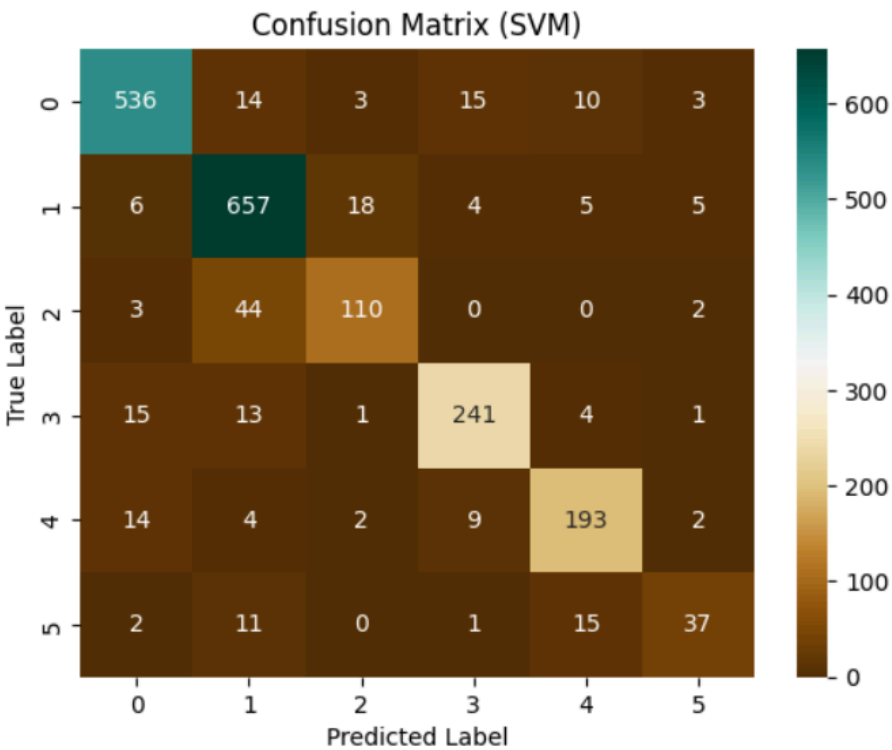
Confusion Matrix (Naive bayes):



Classification Report (SVM):



Confusion Matrix (SVM):



Task 3: Improved Emotion Recognizer

❖ **Description:**

This task aimed to enhance the emotion recognition model by incorporating Part-of-Speech (POS) tags as additional features. The rationale behind this lies in the belief that grammatical information can provide valuable context for understanding the emotional sentiment expressed in a text.

Approach:

- ❖ **POS Tagging:**
  - ➢ The previously trained POS tagger from Task 1 was utilized to annotate each word in the dataset with its corresponding POS tag.
- ❖ **Feature Integration:**
  - ➢ A pipeline was constructed to combine TF-IDF features extracted from the original text with TF-IDF features extracted from the POS-tagged text. This integration aimed to capture both lexical

and grammatical information for a more comprehensive representation of the text.

- **Model Training and Evaluation:**
  - The same classifiers used in Task 2 (Naive Bayes and SVM) were employed again, but this time trained on the enriched feature representation.
  - The models were evaluated on the validation and test sets to assess their performance with the inclusion of POS tags.

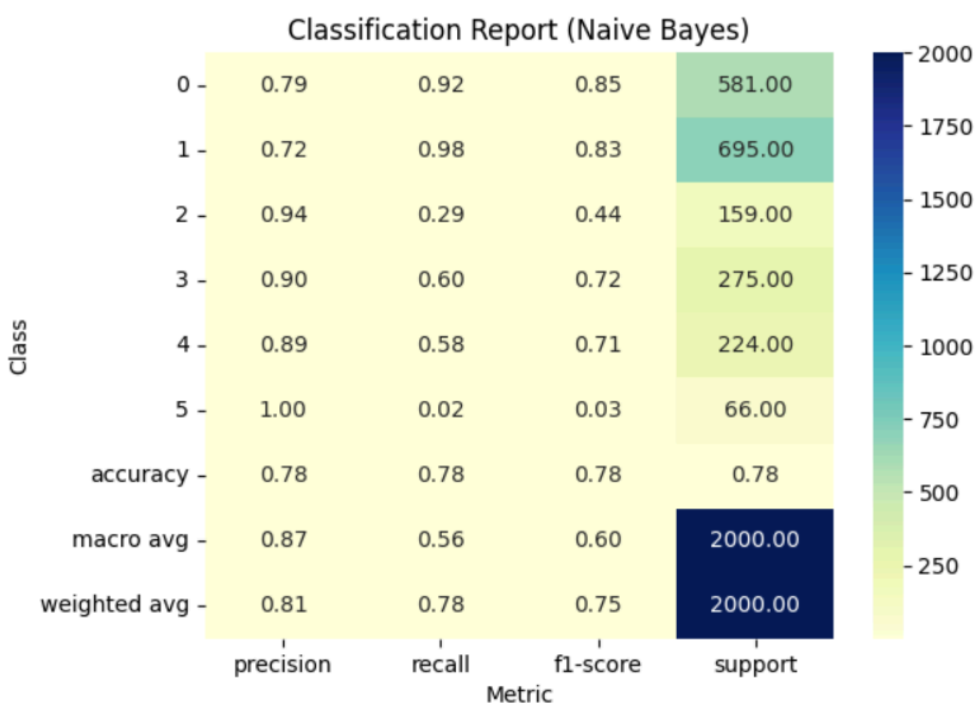
Results:

The SVM model consistently outperformed Naive Bayes in this task, even with the added POS tag features.

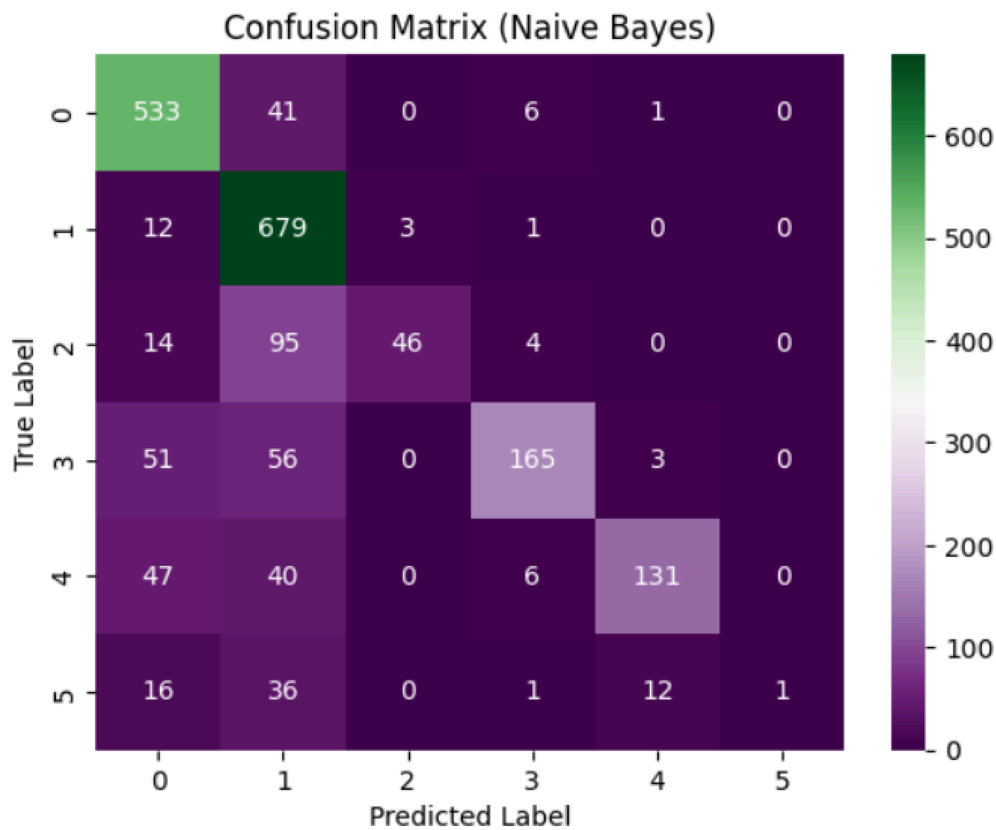
Test Accuracy:

- Naive Bayes: 0.7775
- SVM: 0.8665

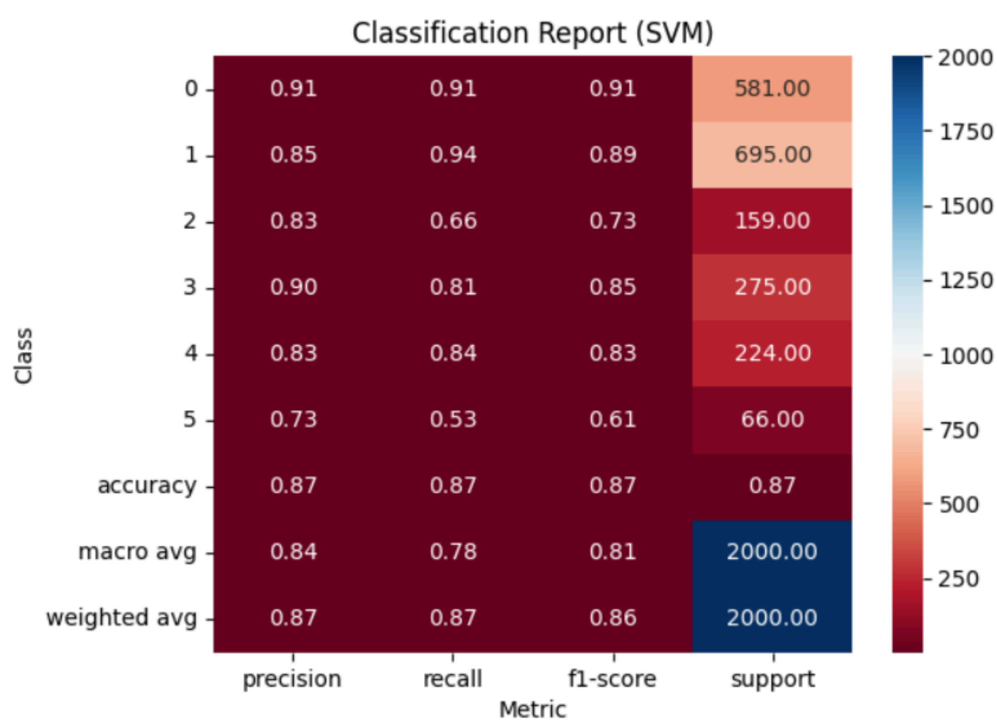
Classification Report (Naive bayes):



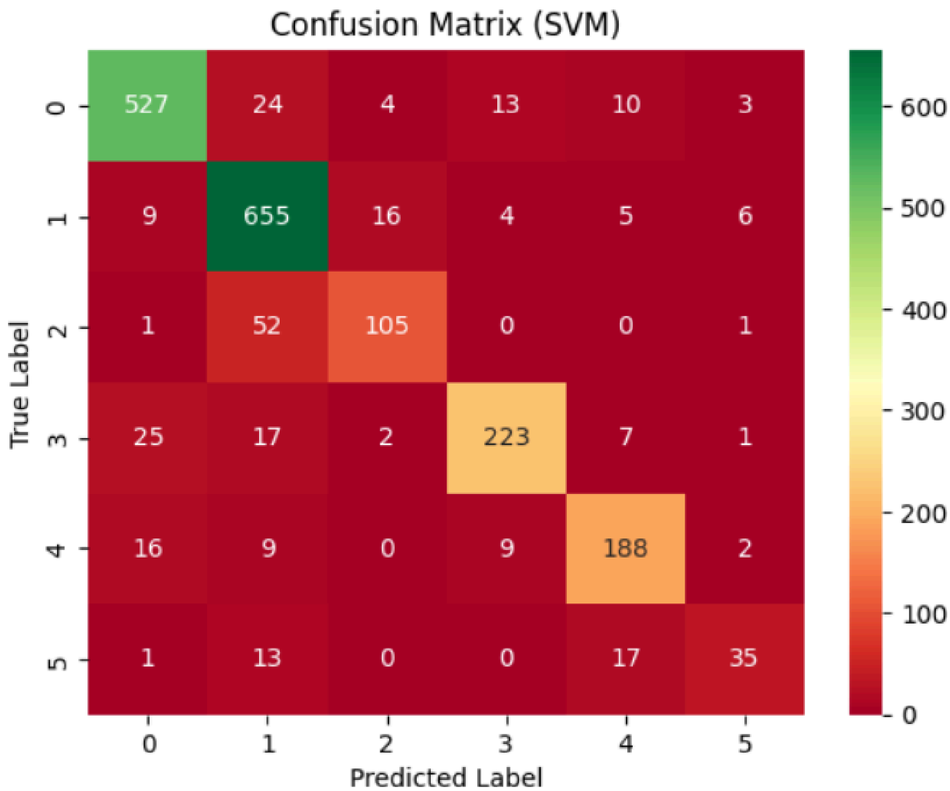
Confusion Matrix (Naive bayes):



Classification Report (SVM):



Confusion Matrix (SVM):



Comparison:

- **SVM:**
  - The SVM model's accuracy decreased slightly from 0.887 to 0.8665 after incorporating POS tags.
  - Precision and recall both decreased slightly, indicating a trade-off in performance.
- **Naive Bayes:**
  - Naive Bayes saw a slight decrease in accuracy from 0.7875 to 0.7775.
  - Precision improved, while recall decreased, suggesting a shift in the model's prediction behavior.

Overall:

While POS tags did improve performance for some metrics in both models, the overall impact was limited. In some cases, there was a trade-off between precision and recall.

This suggests that the chosen POS tagging approach or the way POS features were integrated might not have been the most optimal for this specific dataset or task.

Conclusion:

While POS tags can provide valuable contextual information, their impact on emotion recognition in this specific case was limited. Further research and experimentation are needed to fully explore the potential benefits of incorporating POS tags in this domain.