# NLP Assignment 1 Report
Anwesha Das - 21CS30007

A. Observations (also included in the ipynb file) :

Task 1:

- Total number of sentences in treebank corpus: 3914
- Total number of unique POS tags used: 46
- Training dataset size: 16000
- Validation dataset size: 2000
- Testing dataset size: 2000

Task 2:

- Optimal parameters for the model without POS tagging (found using validation data): SVM_C = 10, TF-IDF_max_features = 5000
- Best validation accuracy for Model 1 without POS tagging): 0.714
- Accuracy of Model 1 (without POS tagging): 0.85650

Task 3:

- Optimal parameters for the model with POS tagging (found using validation data): SVM_C = 10, TF-IDF_max_features = 5000
- Best validation accuracy for Model 1 without POS tagging): 0.7015
- Accuracy of Model 2 (with POS tagging): 0.86050

B. Comparing the performance of the POS-tag-enhanced model with the baseline model:
Support Vector Classifier from sklearn.svm is the baseline model (Model 1).

Model 1 is enhanced by encoding POS tags in its input dataset song with the regular input, where POS tags are generated using the Viterbi algorithm (Model 2)

In comparison with Model 1, Model 2 has a greater overall accuracy of 0.004, even when its best validation accuracy is less than that of Model 1 by 0.0085.

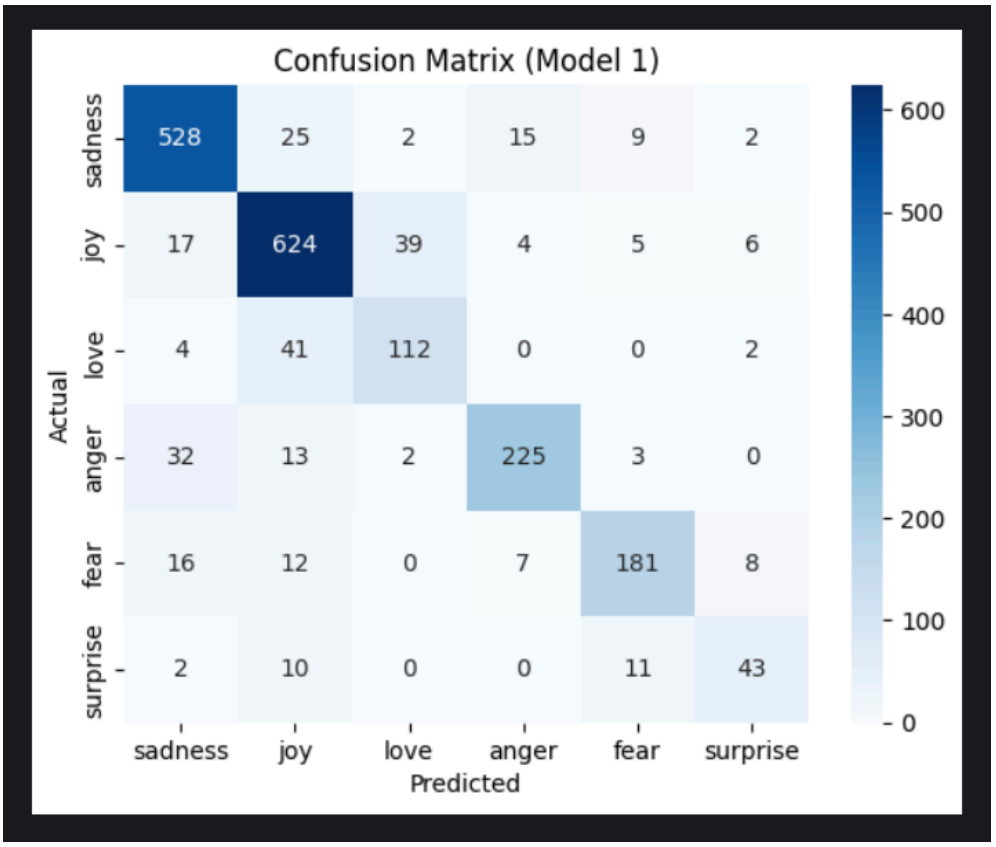
C.1 Classification Reports and Confusion Matrix

1. Model 1 (SVM Classifier Without POS Tags):

Classification Report 1

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.91 | 0.89 | 581 |
| 1 | 0.86 | 0.90 | 0.88 | 695 |
| 2 | 0.72 | 0.70 | 0.71 | 159 |
| 3 | 0.90 | 0.82 | 0.86 | 275 |

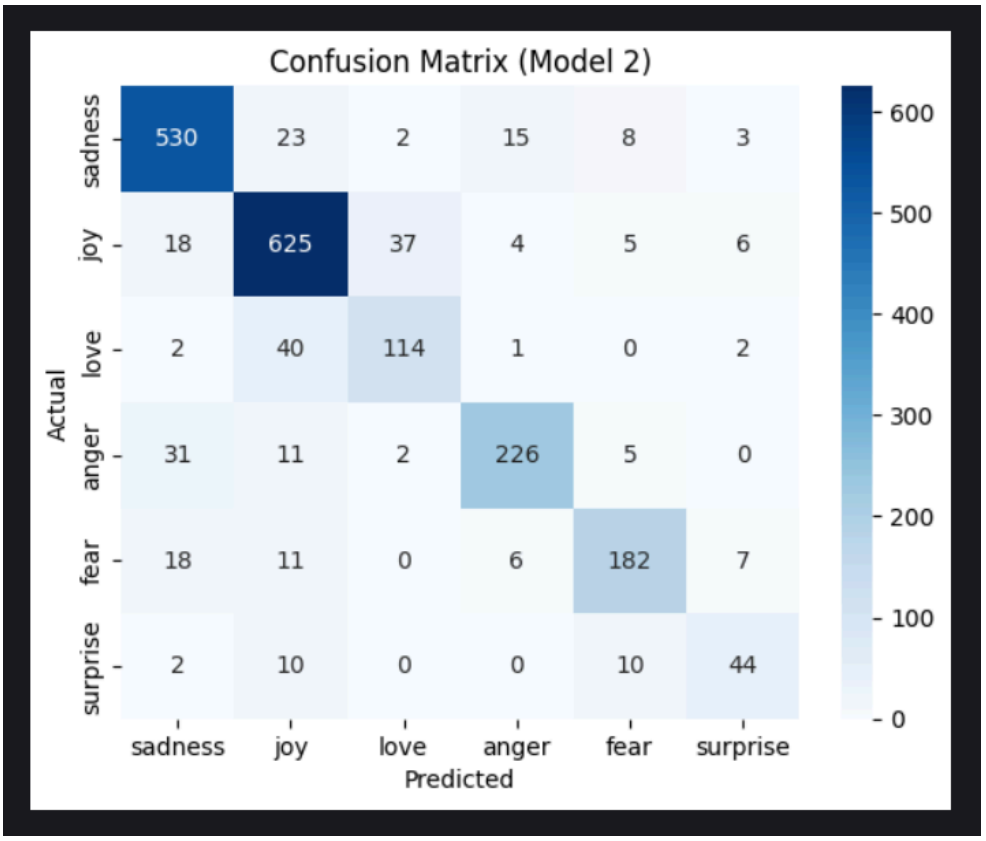
| | | | | |
|---|---|---|---|---|
| 4 | 0.87 | 0.81 | 0.84 | 224 |
| 5 | 0.70 | 0.65 | 0.68 | 66 |
| accuracy | | | 0.86 | 2000 |
| macro avg | 0.82 | 0.80 | 0.81 | 2000 |
| weighted avg | 0.86 | 0.86 | 0.86 | 2000 |




2. Model 2 (SVM Classifier With POS Tags):

Classification Report 2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.91 | 0.90 | 581 |
| 1 | 0.87 | 0.90 | 0.88 | 695 |
| 2 | 0.74 | 0.72 | 0.73 | 159 |
| 3 | 0.90 | 0.82 | 0.86 | 275 |
| 4 | 0.87 | 0.81 | 0.84 | 224 |
| 5 | 0.71 | 0.67 | 0.69 | 66 |
| accuracy | | | 0.86 | 2000 |
| macro avg | 0.83 | 0.80 | 0.82 | 2000 |
| weighted avg | 0.86 | 0.86 | 0.86 | 2000 |




D. Modifications and Strategies used:

- In the Viterbi algorithm, in case the probability of a word being of a particular tag is not found, use a minimal value for it (1e-8 or 10^-8)

- Used concatenation to integrate the pos tagged features with the sentence embeddings. Created two transformers for pos tagging and concatenation respectively, and utilized pipeline to concatenate each of the three datasets (train, validation, test) with their pos tagged versions.

- Tuned the following hyperparameters using the validation set: Regularisation parameter C of SVM, max_features of TF-IDF. Max_features has a choice of 5 alternatives (1000 to 5000 in steps of 1000) and C has a choice of 4 alternatives (0.1, 1, 10, 100). Used 5-fold cross-validation for each of 20 candidates, therefore a total of 100 fits (both for Model 1 and Model 2).

- Used pipelining to combine tfidf vectorization and hyperparameter tuning via grid search (both for Model 1 and Model 2).