CS571 AI LAB 09

29th November 2022

Ishita Singh 1901CS27
Kavya Goyal 1901CS30
Priyanka Sachan 1901CS43

Team Code: 1901cs30_1901cs27_1901cs43
Collab Link: cs30_cs27_cs43_AI09

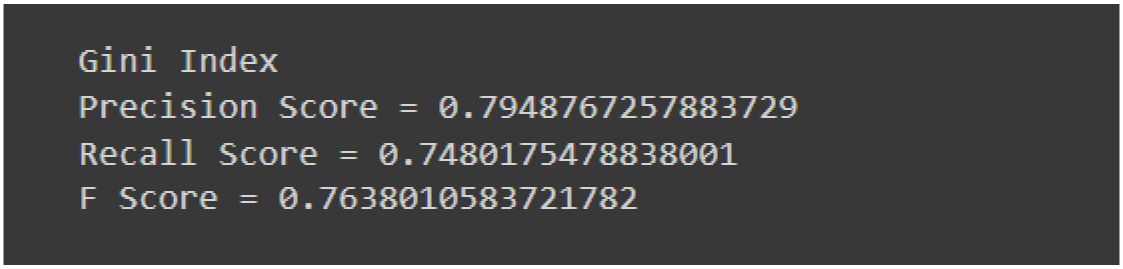
Objective

Our aim was to classify a dataset based on questions, depending on what type of answer it expects.
We made use of the Gini Index, Misclassification Error and Cross Entropy to aid with our final work, and made comparisons based on them. A study of how individual features affect the results was also made.

Precision= True Positives/ (True Positives + False Positives)
Recall= True Positives/ (True Positives + False Negatives)
F Score= (2*Precision*Recall)/ (Precision + Recall)

Result and Evaluation

1. Report the 10-fold cross-validation results in terms of precision, recall, and F-score.



```
Gini Index
Precision Score = 0.7948767257883729
Recall Score = 0.7480175478838001
F Score = 0.7638010583721782
```

Similarly, we trained for Entropy and Misclassification.

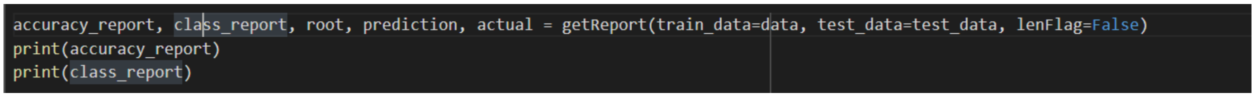
2. Report results of feature ablation study and state which feature has contributed most towards correctly predicting a particular class.

We have used *question length*, *POS Tagging* of sentences and its *unigram*, *bigram* and *trigram* tokens.

a. Length of Question

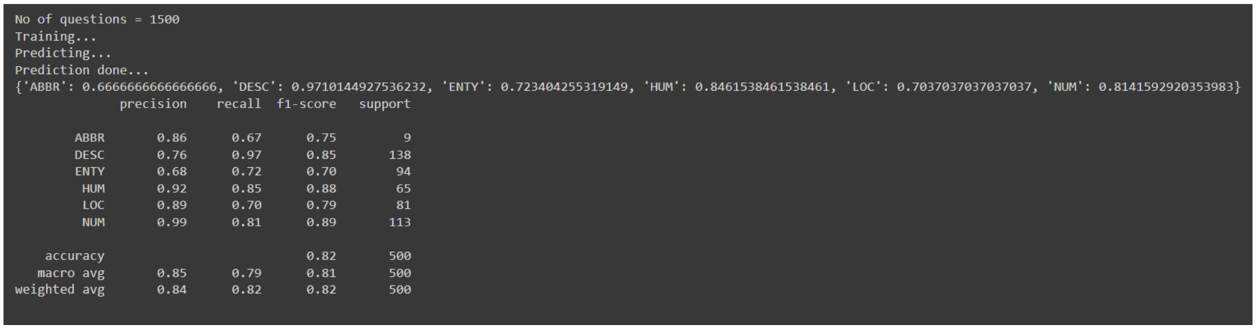
This means that the *length of the question* is not taken as the feature and all the remaining features remain intact. It is found over all the three indexes and the following results were obtained

Function Call



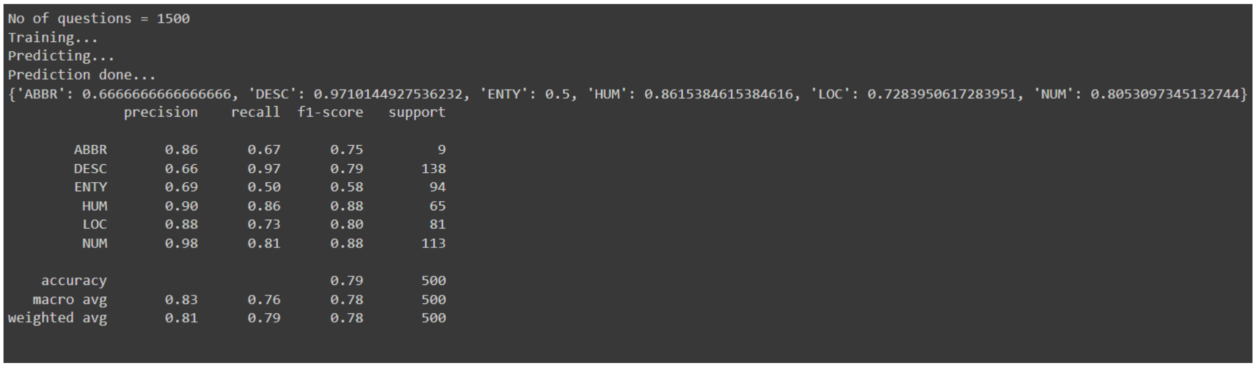
```
accuracy_report, class_report, root, prediction, actual = getReport(train_data=data, test_data=test_data, lenFlag=False)
print(accuracy_report)
print(class_report)
```

Gini Index Result



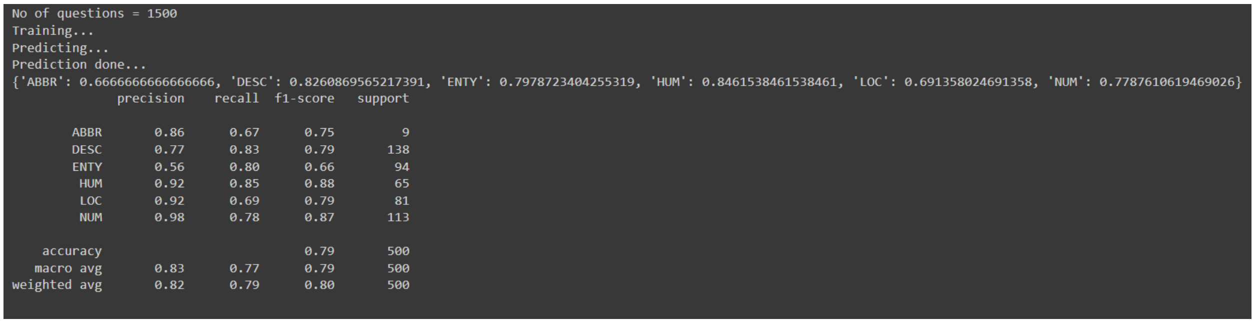
```
No of questions = 1500
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9710144927536232, 'ENTY': 0.723404255319149, 'HUM': 0.8461538461538461, 'LOC': 0.7037037037037037, 'NUM': 0.8141592920353983}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.76      0.97      0.85       138
        ENTY       0.68      0.72      0.70        94
         HUM       0.92      0.85      0.88        65
         LOC       0.89      0.70      0.79        81
         NUM       0.99      0.81      0.89       113

    accuracy                           0.82       500
   macro avg       0.85      0.79      0.81       500
weighted avg       0.84      0.82      0.82       500
```

Cross-Entropy Result



```
No of questions = 1500
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9710144927536232, 'ENTY': 0.5, 'HUM': 0.8615384615384616, 'LOC': 0.7283950617283951, 'NUM': 0.8053097345132744}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.66      0.97      0.79       138
        ENTY       0.69      0.50      0.58        94
         HUM       0.90      0.86      0.88        65
         LOC       0.88      0.73      0.80        81
         NUM       0.98      0.81      0.88       113

    accuracy                           0.79       500
   macro avg       0.83      0.76      0.78       500
weighted avg       0.81      0.79      0.78       500
```

Misclassification Result



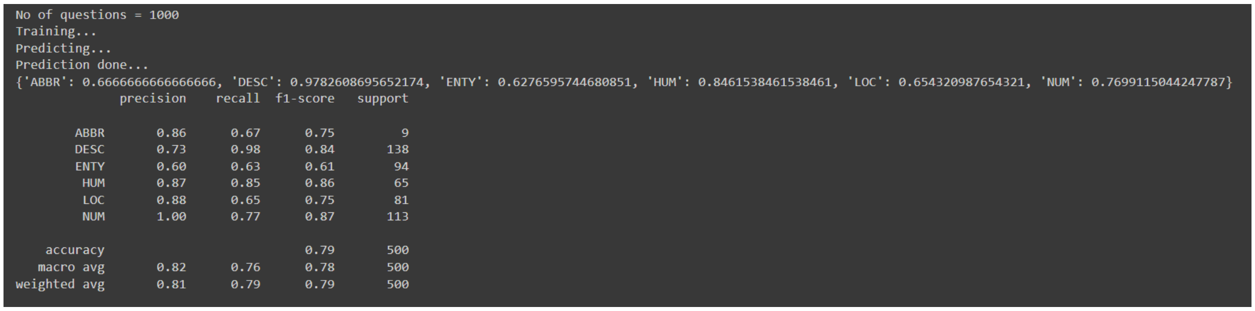
```
No of questions = 1500
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.8260869565217391, 'ENTY': 0.7978723404255319, 'HUM': 0.8461538461538461, 'LOC': 0.691358024691358, 'NUM': 0.7787610619469026}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.77      0.83      0.79       138
        ENTY       0.56      0.80      0.66        94
         HUM       0.92      0.85      0.88        65
         LOC       0.92      0.69      0.79        81
         NUM       0.98      0.78      0.87       113

    accuracy                           0.79       500
   macro avg       0.83      0.77      0.79       500
weighted avg       0.82      0.79      0.80       500
```

From above observations, we can conclude that *question length* does not affect the classification results much. We get nearly equal F1 scores both with and without the length feature on all three metrics - Gini, misclassification, and cross-entropy.

b. POS Tag

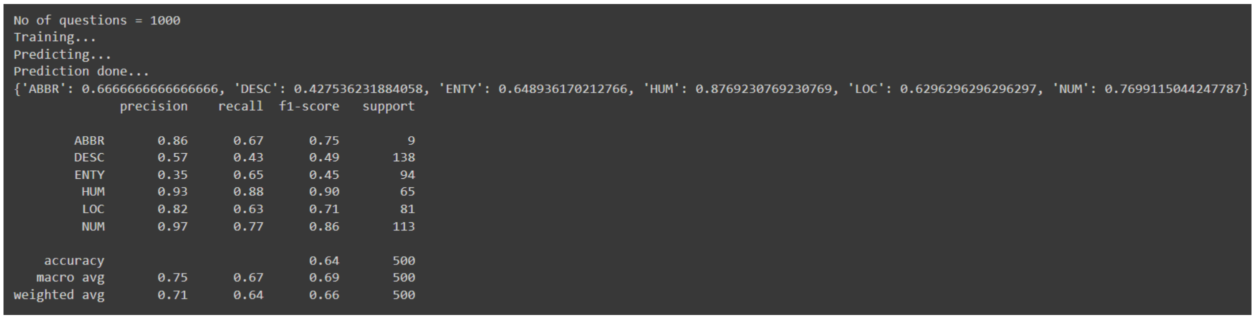
The following results were obtained after removing the POS tagging function

GINI Classification:



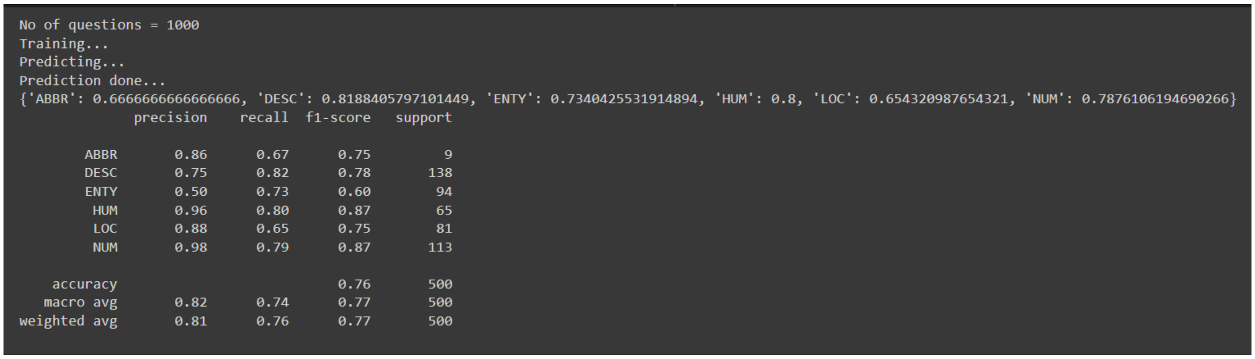
```
No of questions = 1000
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9782608695652174, 'ENTY': 0.6276595744680851, 'HUM': 0.8461538461538461, 'LOC': 0.654320987654321, 'NUM': 0.7699115044247787}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.73      0.98      0.84       138
        ENTY       0.60      0.63      0.61        94
         HUM       0.87      0.85      0.86        65
         LOC       0.88      0.65      0.75        81
         NUM       1.00      0.77      0.87       113

    accuracy                           0.79       500
   macro avg       0.82      0.76      0.78       500
weighted avg       0.81      0.79      0.79       500
```

Cross-Entropy:



```
No of questions = 1000
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.427536231884058, 'ENTY': 0.648936170212766, 'HUM': 0.8769230769230769, 'LOC': 0.6296296296296297, 'NUM': 0.7699115044247787}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.57      0.43      0.49       138
        ENTY       0.35      0.65      0.45        94
         HUM       0.93      0.88      0.90        65
         LOC       0.82      0.63      0.71        81
         NUM       0.97      0.77      0.86       113

    accuracy                           0.64       500
   macro avg       0.75      0.67      0.69       500
weighted avg       0.71      0.64      0.66       500
```

Misclassification:

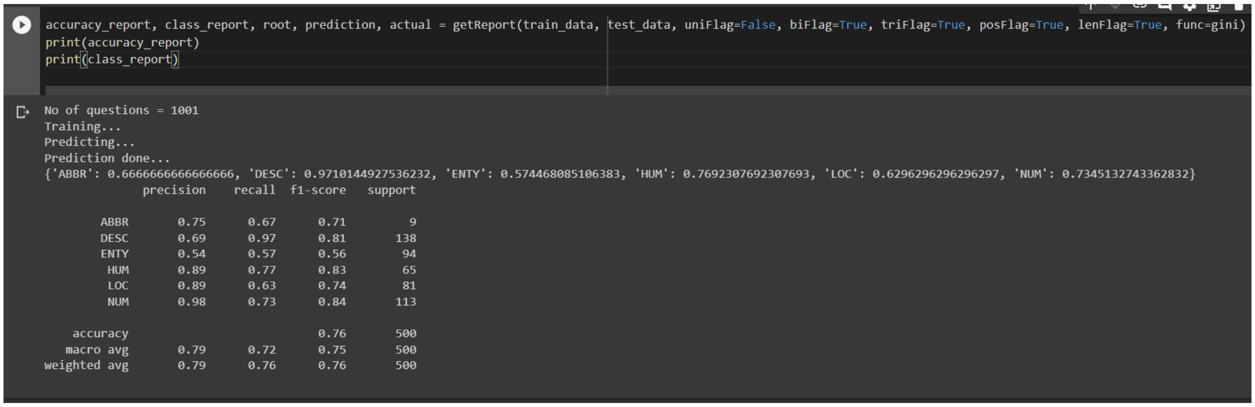


```
No of questions = 1000
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.8188405797101449, 'ENTY': 0.7340425531914894, 'HUM': 0.8, 'LOC': 0.654320987654321, 'NUM': 0.7876106194690266}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.75      0.82      0.78       138
        ENTY       0.50      0.73      0.60        94
         HUM       0.96      0.80      0.87        65
         LOC       0.88      0.65      0.75        81
         NUM       0.98      0.79      0.87       113

    accuracy                           0.76       500
   macro avg       0.82      0.74      0.77       500
weighted avg       0.81      0.76      0.77       500
```

From the result obtained, we can see that there is a significant decrease (~10%) in the overall accuracy and the F1 score of the test set when the *POS tag* feature is not used. Hence, we see that the POS tag is an important factor to classify the questions.

c. Unigram/ Bigram/ Trigram

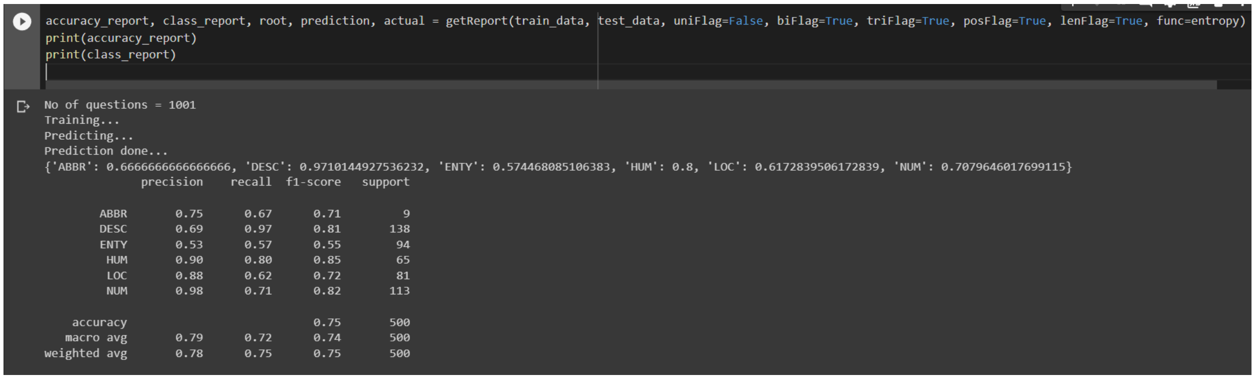
GINI Index



```
accuracy_report, class_report, root, prediction, actual = getReport(train_data, test_data, uniFlag=False, biFlag=True, triFlag=True, posFlag=True, lenFlag=True, func=gini)
print(accuracy_report)
print(class_report)
```

```
No of questions = 1001
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9710144927536232, 'ENTY': 0.574468085106383, 'HUM': 0.7692307692307693, 'LOC': 0.6296296296296297, 'NUM': 0.7345132743362832}
              precision    recall  f1-score   support

        ABBR       0.75      0.67      0.71         9
        DESC       0.69      0.97      0.81       138
        ENTY       0.54      0.57      0.56        94
         HUM       0.89      0.77      0.83        65
         LOC       0.89      0.63      0.74        81
         NUM       0.98      0.73      0.84       113

    accuracy                           0.76       500
   macro avg       0.79      0.72      0.75       500
weighted avg       0.79      0.76      0.76       500
```

Cross-Entropy:



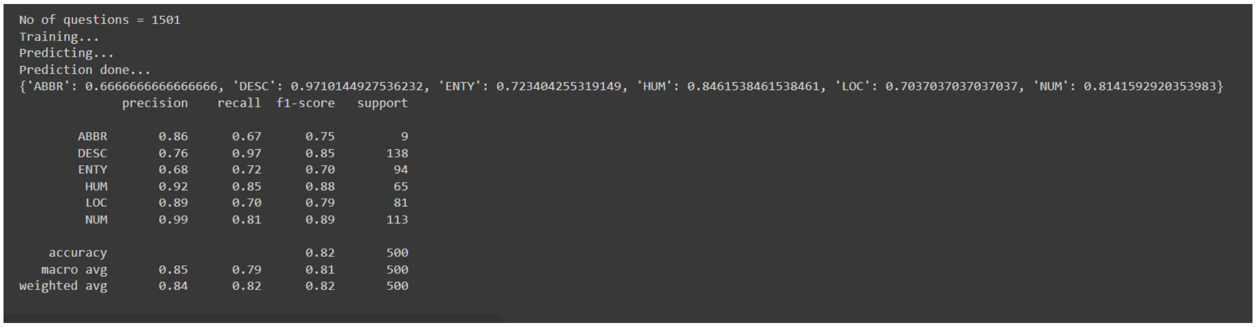
```
accuracy_report, class_report, root, prediction, actual = getReport(train_data, test_data, uniFlag=False, biFlag=True, triFlag=True, posFlag=True, lenFlag=True, func=entropy)
print(accuracy_report)
print(class_report)
```

```
No of questions = 1001
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9710144927536232, 'ENTY': 0.574468085106383, 'HUM': 0.8, 'LOC': 0.6172839506172839, 'NUM': 0.7079646017699115}
              precision    recall  f1-score   support

        ABBR       0.75      0.67      0.71         9
        DESC       0.69      0.97      0.81       138
        ENTY       0.53      0.57      0.55        94
         HUM       0.90      0.80      0.85        65
         LOC       0.88      0.62      0.72        81
         NUM       0.98      0.71      0.82       113

    accuracy                           0.75       500
   macro avg       0.79      0.72      0.74       500
weighted avg       0.78      0.75      0.75       500
```

Similarly the training was done and reports generated for Misclassification with unigram flag set as False, and also individually without bigram and trigram features.

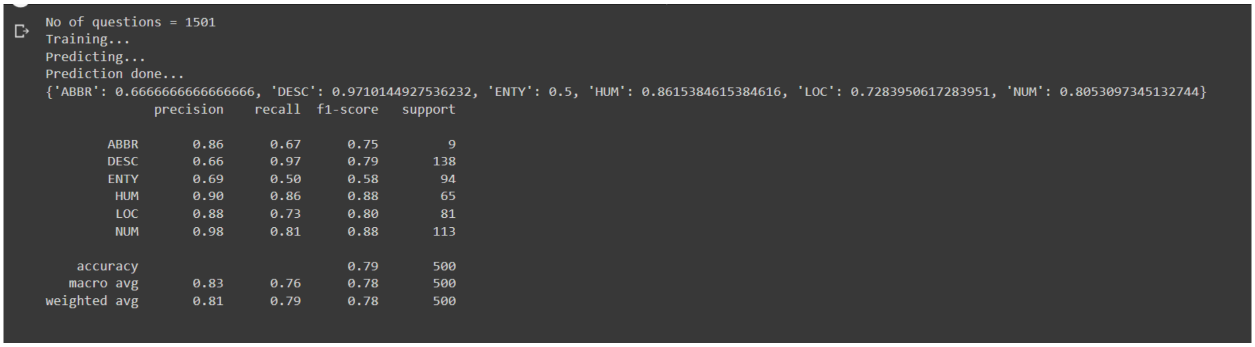
3. Report precision, recall, and F-score measures on test sets using models based on the gini index, mis-classification error and cross-entropy.

Results by using the GINI Index



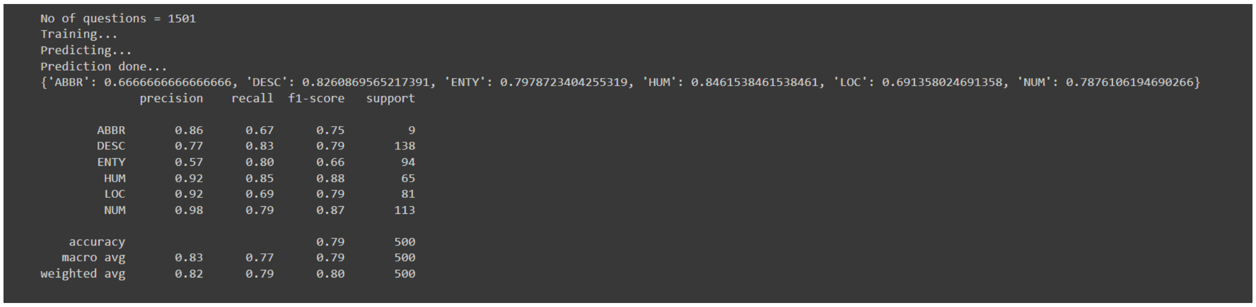
```
No of questions = 1501
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9710144927536232, 'ENTY': 0.723404255319149, 'HUM': 0.8461538461538461, 'LOC': 0.7037037037037037, 'NUM': 0.8141592920353983}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.76      0.97      0.85       138
        ENTY       0.68      0.72      0.70        94
         HUM       0.92      0.85      0.88        65
         LOC       0.89      0.70      0.79        81
         NUM       0.99      0.81      0.89       113

    accuracy                           0.82       500
   macro avg       0.85      0.79      0.81       500
weighted avg       0.84      0.82      0.82       500
```

Results by using the Cross-Entropy



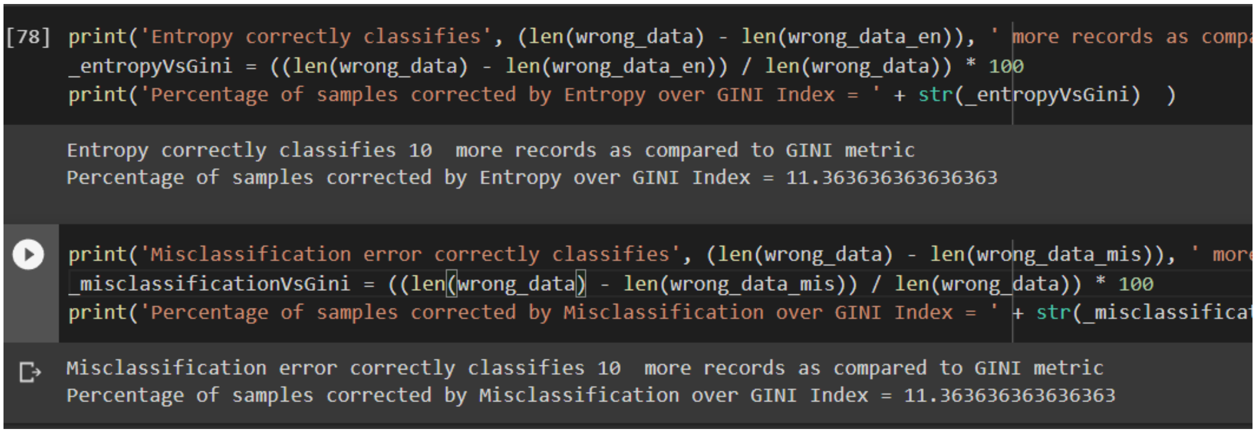
```
No of questions = 1501
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.9710144927536232, 'ENTY': 0.5, 'HUM': 0.8615384615384616, 'LOC': 0.7283950617283951, 'NUM': 0.8053097345132744}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.66      0.97      0.79       138
        ENTY       0.69      0.50      0.58        94
         HUM       0.90      0.86      0.88        65
         LOC       0.88      0.73      0.80        81
         NUM       0.98      0.81      0.88       113

    accuracy                           0.79       500
   macro avg       0.83      0.76      0.78       500
weighted avg       0.81      0.79      0.78       500
```

Results by using Misclassification



```
No of questions = 1501
Training...
Predicting...
Prediction done...
{'ABBR': 0.6666666666666666, 'DESC': 0.8260869565217391, 'ENTY': 0.7978723404255319, 'HUM': 0.8461538461538461, 'LOC': 0.691358024691358, 'NUM': 0.7876106194690266}
              precision    recall  f1-score   support

        ABBR       0.86      0.67      0.75         9
        DESC       0.77      0.83      0.79       138
        ENTY       0.57      0.80      0.66        94
         HUM       0.92      0.85      0.88        65
         LOC       0.92      0.69      0.79        81
         NUM       0.98      0.79      0.87       113

    accuracy                           0.79       500
   macro avg       0.83      0.77      0.79       500
weighted avg       0.82      0.79      0.80       500
```

4. Show whether errors propagated by one model are corrected by other models or not. If yes, then report how many percent of samples are corrected.



```
[78] print('Entropy correctly classifies', (len(wrong_data) - len(wrong_data_en)), ' more records as compa
     _entropyVsGini = ((len(wrong_data) - len(wrong_data_en)) / len(wrong_data)) * 100
     print('Percentage of samples corrected by Entropy over GINI Index = ' + str(_entropyVsGini)  )

     Entropy correctly classifies 10  more records as compared to GINI metric
     Percentage of samples corrected by Entropy over GINI Index = 11.363636363636363
```

```
print('Misclassification error correctly classifies', (len(wrong_data) - len(wrong_data_mis)), ' more
_misclassificationVsGini = ((len(wrong_data) - len(wrong_data_mis)) / len(wrong_data)) * 100
print('Percentage of samples corrected by Misclassification over GINI Index = ' + str(_misclassifica

Misclassification error correctly classifies 10  more records as compared to GINI metric
Percentage of samples corrected by Misclassification over GINI Index = 11.363636363636363
```

As we can see here, the number of incorrectly classified records by Entropy, Gini Index and Misclassification are different. Therefore, the errors propagated by one model are corrected in other models.

Execution

1. Open the collab link
2. Add the sample data

Train Data: train_ai09.txt
Test Data: trec_ai09.txt
(Attached)

3. Run all
4. Google Collab was used for training and testing because of resource limitations locally.