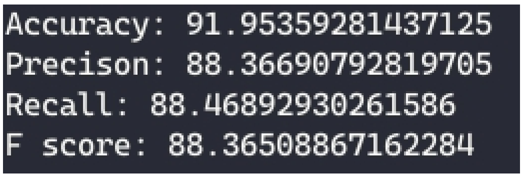
CS563: NLP - Mid Semester Assignment

Name: M. Maheeth Reddy	**Roll No**: 1801CS31	**Date**: 23-Feb-2022

Ans 1:

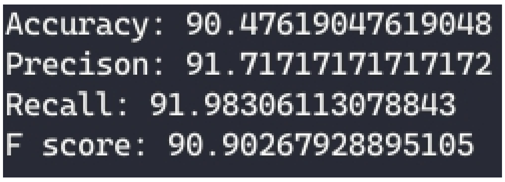
Task: To predict the actual intended characters in the test data and given paragraph
Code: midsem_q1.py
Test Data Directory: midsem_data/
Test Data Files: test_data.txt (in dataset), p1.txt (in question)
Running the code: **python3 midsem_q1.py** (scikit-learn must be installed)
Output Directory: output/
Output Files: output_test_data.txt, output_p1.txt
Evaluation metrics:

test_data.txt



```
Accuracy:  91.95359281437125
Precison:  88.36690792819705
Recall:  88.46892930261586
F score:  88.36508867162284
```

p1.txt

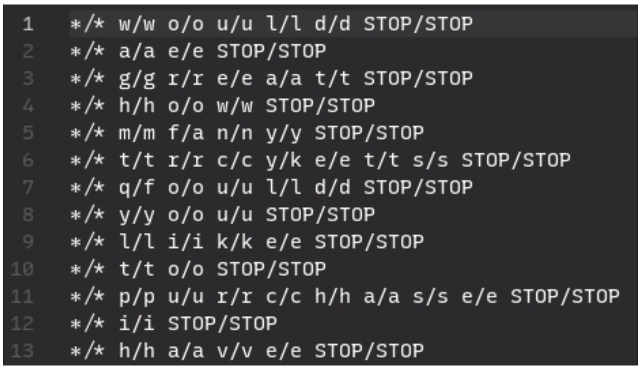


```
Accuracy:  90.47619047619048
Precison:  91.71717171717172
Recall:  91.98306113078843
F score:  90.90267928895105
```

Results: **On p1.txt**: 40% of the cases are correctly predicted
Observation: HMM easily corrects in the case where correct information is there and fails when x is to be detected.

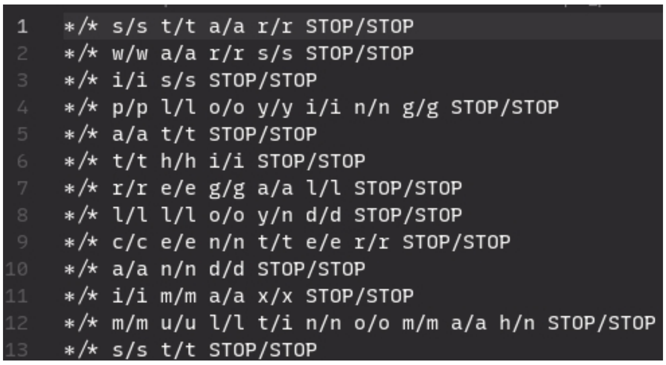
Output FIles & Screenshots:

output_test_data.txt



```
*/* w/w o/o u/u l/l d/d STOP/STOP
*/* a/a e/e STOP/STOP
*/* g/g r/r e/e a/a t/t STOP/STOP
*/* h/h o/o w/w STOP/STOP
*/* m/m f/a n/n y/y STOP/STOP
*/* t/t r/r c/c y/k e/e t/t s/s STOP/STOP
*/* q/f o/o u/u l/l d/d STOP/STOP
*/* y/y o/o u/u STOP/STOP
*/* l/l i/i k/k e/e STOP/STOP
*/* t/t o/o STOP/STOP
*/* p/p u/u r/r c/c h/h a/a s/s e/e STOP/STOP
*/* i/i STOP/STOP
*/* h/h a/a v/v e/e STOP/STOP
```

output_p1.txt

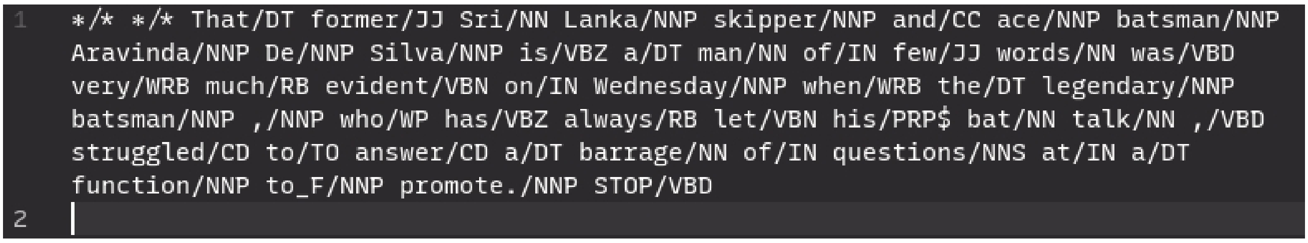


```
*/* s/s t/t a/a r/r STOP/STOP
*/* w/w a/a r/r s/s STOP/STOP
*/* i/i s/s STOP/STOP
*/* p/p l/l o/o y/y i/i n/n g/g STOP/STOP
*/* a/a t/t STOP/STOP
*/* t/t h/h i/i STOP/STOP
*/* r/r e/e g/g a/a l/l STOP/STOP
*/* l/l l/l o/o y/n d/d STOP/STOP
*/* c/c e/e n/n t/t e/e r/r STOP/STOP
*/* a/a n/n d/d STOP/STOP
*/* i/i m/m a/a x/x STOP/STOP
*/* m/m u/u l/l t/i n/n o/o m/m a/a h/n STOP/STOP
*/* s/s t/t STOP/STOP
```

Ans 2:

Part 2a

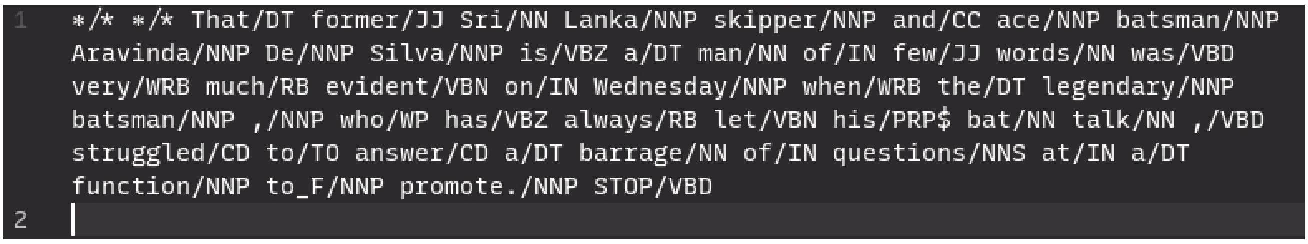
Task: To determine the PoS tagging sequence for the given sentence using the trained HMM model
Code: midsem_q2a.py
Test Data Directory: midsem_data/
Test Data Files: penn-data.json (in dataset)
Running the code: **python3 midsem_q2a.py**
Output Directory: output/
Output Files: output_q2a.txt



```
*/* */* That/DT former/JJ Sri/NN Lanka/NNP skipper/NNP and/CC ace/NNP batsman/NNP
Aravinda/NNP De/NNP Silva/NNP is/VBZ a/DT man/NN of/IN few/JJ words/NN was/VBD
very/WRB much/RB evident/VBN on/IN Wednesday/NNP when/WRB the/DT legendary/NNP
batsman/NNP ,/NNP who/WP has/VBZ always/RB let/VBN his/PRP$ bat/NN talk/NN ,/VBD
struggled/CD to/TO answer/CD a/DT barrage/NN of/IN questions/NNS at/IN a/DT
function/NNP to_F/NNP promote./NNP STOP/VBD
```

Part 2b

Task:
Code: midsem_q2b.py
Test Data Directory: midsem_data/
Test Data Files: penn-data.json (in dataset)
Running the code: **python3 midsem_q2b.py**
Output Directory: output/
Output Files: output_q2b.txt
The tagging will be the same as Part 2a:



```
*/* */* That/DT former/JJ Sri/NN Lanka/NNP skipper/NNP and/CC ace/NNP batsman/NNP
Aravinda/NNP De/NNP Silva/NNP is/VBZ a/DT man/NN of/IN few/JJ words/NN was/VBD
very/WRB much/RB evident/VBN on/IN Wednesday/NNP when/WRB the/DT legendary/NNP
batsman/NNP ,/NNP who/WP has/VBZ always/RB let/VBN his/PRP$ bat/NN talk/NN ,/VBD
struggled/CD to/TO answer/CD a/DT barrage/NN of/IN questions/NNS at/IN a/DT
function/NNP to_F/NNP promote./NNP STOP/VBD
```

Part 2c

Result: The tagging is the same in parts 2a and 2b.
Reason: Even if we consider the top 3 probabilities for each tag of each word, we obtain the same tagging as at each step because the tags which have less probability at the current step will also have less probability in the future.
As, the total probability is basically the product of all previous probabilites, if we have a tag with high probability through one path, that will continue till the end. This explains the results.