Course Name: Machine Learning for Trading (CS 7646)

Project Name: Project 1C

Student Name: Aishvarya Krishnan

Metric used in Part 2

I used the percentage of correct pairing of documents in the result (good-good and bad-bad) of the total number of documents as the metric to measure correctness of the classification method.

⇨ Overall score = (no. of good-good and bad-bad pairs in result/total no. of documents) * 100

Number of good-good or bad-bad pairs is calculated from the name of the documents.

Improvement in Part 3

When I analyzed the values obtained for tf-idf and vector(cos), I observed that the values are non-linear. I thought of normalizing the values to converge at a smaller range to obtain more accurate results. I improved the method to calculate idf to improve the accuracy of classification. To normalize the values obtained from tf-idf, I modified the formula to calculate idf as follows:

idf = 1 + log_e(N/count)

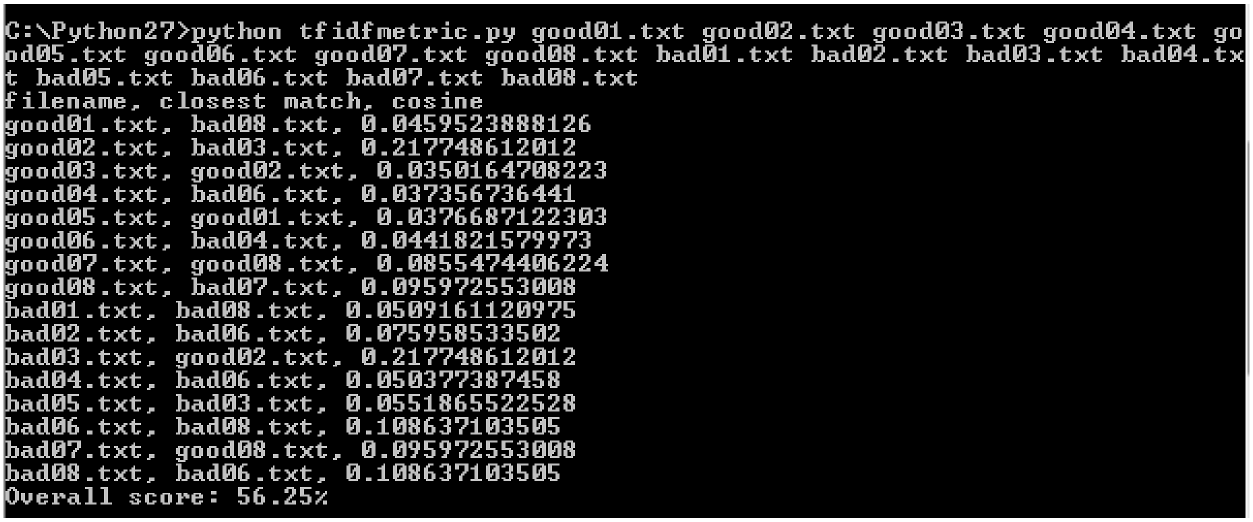
Where, N -> number of documents

count -> number of documents a term t appears in

Comparion and Assessment of Part 1 method and Part 3 method

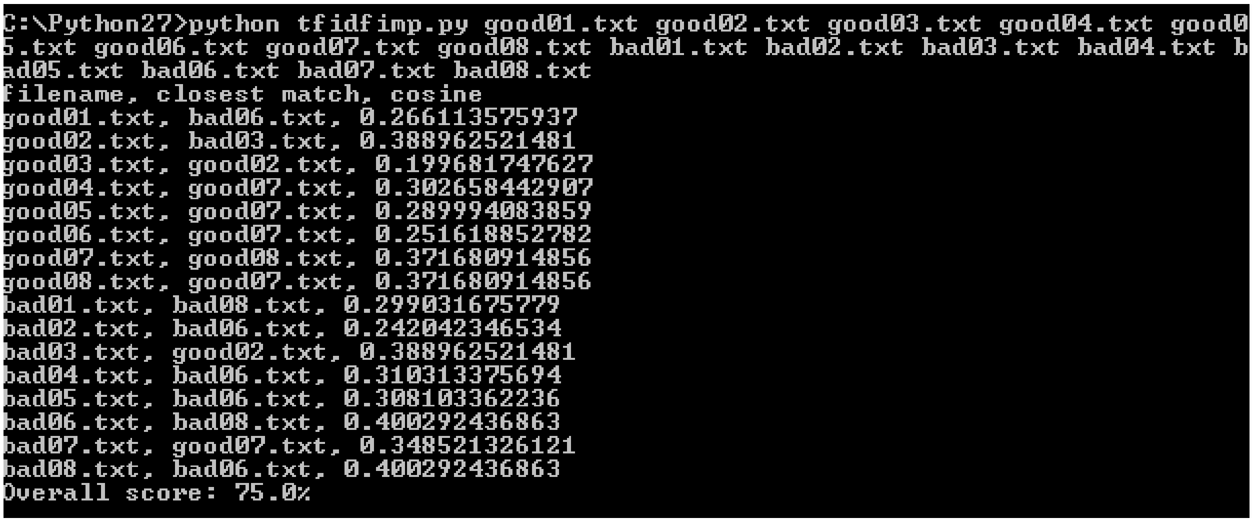
When we compare Part 1 and Part 3 methods, we see that the change in idf calculation to normalize the values has increased the accuracy from 56.25% to 75%. Hence, in distance-based classification, normalization of values is a key to achieving better results.

Output of Part 2



```
C:\Python27>python tfidfmetric.py good01.txt good02.txt good03.txt good04.txt go
od05.txt good06.txt good07.txt good08.txt bad01.txt bad02.txt bad03.txt bad04.tx
t bad05.txt bad06.txt bad07.txt bad08.txt
filename, closest match, cosine
good01.txt, bad08.txt, 0.0459523888126
good02.txt, bad03.txt, 0.217748612012
good03.txt, good02.txt, 0.0350164708223
good04.txt, bad06.txt, 0.037356736441
good05.txt, good01.txt, 0.0376687122303
good06.txt, bad04.txt, 0.0441821579973
good07.txt, good08.txt, 0.0855474406224
good08.txt, bad07.txt, 0.095972553008
bad01.txt, bad08.txt, 0.0509161120975
bad02.txt, bad06.txt, 0.075958533502
bad03.txt, good02.txt, 0.217748612012
bad04.txt, bad06.txt, 0.050377387458
bad05.txt, bad03.txt, 0.0551865522528
bad06.txt, bad08.txt, 0.108637103505
bad07.txt, good08.txt, 0.095972553008
bad08.txt, bad06.txt, 0.108637103505
Overall score: 56.25%
```

Output of Part 3



```
C:\Python27>python tfidfimp.py good01.txt good02.txt good03.txt good04.txt good0
5.txt good06.txt good07.txt good08.txt bad01.txt bad02.txt bad03.txt bad04.txt b
ad05.txt bad06.txt bad07.txt bad08.txt
filename, closest match, cosine
good01.txt, bad06.txt, 0.266113575937
good02.txt, bad03.txt, 0.388962521481
good03.txt, good02.txt, 0.199681747627
good04.txt, good07.txt, 0.302658442907
good05.txt, good07.txt, 0.289994083859
good06.txt, good07.txt, 0.251618852782
good07.txt, good08.txt, 0.371680914856
good08.txt, good07.txt, 0.371680914856
bad01.txt, bad08.txt, 0.299031675779
bad02.txt, bad06.txt, 0.242042346534
bad03.txt, good02.txt, 0.388962521481
bad04.txt, bad06.txt, 0.310313375694
bad05.txt, bad06.txt, 0.308103362236
bad06.txt, bad08.txt, 0.400292436863
bad07.txt, good07.txt, 0.348521326121
bad08.txt, bad06.txt, 0.400292436863
Overall score: 75.0%
```