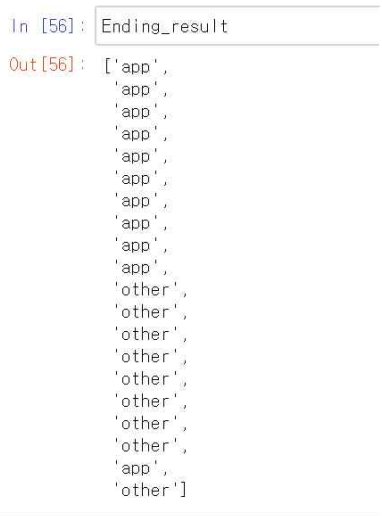
HW - Naive Bayes

2015147533 유현석

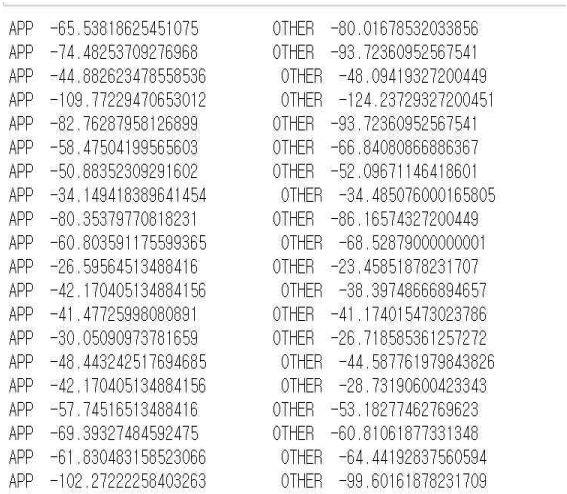
1. 예측 결과

아래의 그림과 같이 1~10번까지 APP class였던 것들에 대한 예측 결과는 10개 중에 10개 일치하였습니다. 즉 100%의 확률을 보여주었습니다.

11~20번까지의 OTHER class였던 것들에 대한 예측 결과는 10개 중에 9개가 일치한 90%의 확률을 보여주었습니다.



```
In [56]: Ending_result
Out[56]: ['app',
          'app',
          'app',
          'app',
          'app',
          'app',
          'app',
          'app',
          'app',
          'app',
          'other',
          'other',
          'other',
          'other',
          'other',
          'other',
          'other',
          'other',
          'app',
          'other']
```



```
APP  -65.53818625451075    OTHER  -80.01678532033856
APP  -74.48253709276968    OTHER  -93.72360952567541
APP  -44.882623478558536    OTHER  -48.09419327200449
APP  -109.77229470653012    OTHER  -124.23729327200451
APP  -82.76287958126899    OTHER  -93.72360952567541
APP  -58.47504199565603    OTHER  -66.84080866886367
APP  -50.88352309291602    OTHER  -52.09671146418601
APP  -34.149418389641454    OTHER  -34.485076000165805
APP  -80.35379770818231    OTHER  -86.16574327200449
APP  -60.803591175599365    OTHER  -68.52879000000001
APP  -26.59564513488416    OTHER  -23.45851878231707
APP  -42.170405134884156    OTHER  -38.39748666894657
APP  -41.47725998080891    OTHER  -41.174015473023786
APP  -30.05090973781659    OTHER  -26.718585361257272
APP  -48.443242517694685    OTHER  -44.587761979843826
APP  -42.170405134884156    OTHER  -28.73190600423343
APP  -57.74516513488416    OTHER  -53.18277462769623
APP  -69.39327484592475    OTHER  -60.81061877331348
APP  -61.830483158523066    OTHER  -64.44192837560594
APP  -102.27222258403263    OTHER  -99.60161878231709
```

또한 APP 토큰과 OTHER 토큰의 확률 값에 LN을 취한 후 그 합을 나타내었습니다. 이를 바탕으로 더 높은 값으로 예측을 할 수 있었습니다.

2. 코딩과정

2-1 각각의 엑셀 파일을 읽은 후 소문자로 바꾼 후 특수 문자를 빈칸으로 변경하기!!

2-2 빈칸을 기준으로 단어로 쪼갠 후 4자리 이상의 단어들만 남긴다. 그 남은 단어들의 각각의 개수를 구해준다.

2-3 개수를 바탕으로 DataFrame을 만든 후 +1를 해주고 LN을 취해준다. 그러면 APP과 OTHER에 대한 토큰들과 LN값이 완성된다.

```
65]: new_df
```

65]:

	[blog]	using	nullmailer	mandrill	your	ubuntu	linux	server	outboud	mail	...	resets	payment	notifications	r
num	2.000000	11.000000	2.000000	94.000000	11.000000	3.000000	2.000000	3.000000	2.000000	4.000000	...	1.000000	1.000000	1.000000	
num+1	3.000000	12.000000	3.000000	95.000000	12.000000	4.000000	3.000000	4.000000	3.000000	5.000000	...	2.000000	2.000000	2.000000	
percent	0.001245	0.004979	0.001245	0.039419	0.004979	0.001660	0.001245	0.001660	0.001245	0.002075	...	0.000830	0.000830	0.000830	
LN	-6.688770	-5.302475	-6.688770	-3.233505	-5.302475	-6.401088	-6.688770	-6.401088	-6.688770	-6.177944	...	-7.094235	-7.094235	-7.094235	

4 rows × 823 columns

<APP에 대한 DataFrame>

	donde	esta	remontada	mandrill	.@katie_phd	alternate	'reproachful	mandrill'	cover	@davidquammen's	...	getting	mood	
num	1.000000	2.000000	1.000000	90.000000	1.000000	1.000000	1.000000	1.000000	2.000000	1.000000	...	1.000000	1.000000	
num+1	2.000000	3.000000	2.000000	91.000000	2.000000	2.000000	2.000000	2.000000	3.000000	2.000000	...	2.000000	2.000000	
percent	0.000988	0.001481	0.000988	0.044938	0.000988	0.000988	0.000988	0.000988	0.001481	0.000988	...	0.000988	0.000988	
LN	-6.920178	-6.514713	-6.920178	-3.102465	-6.920178	-6.920178	-6.920178	-6.920178	-6.514713	-6.920178	...	-6.920178	-6.920178	

4 rows × 805 columns

<OTHER에 대한 DataFrame>

2-4 예측을 해야 하는 파일을 읽어준 후 위와 마찬가지로 각각 단어로 나누어준다. 그 후 3자리 이하 단어는 버리고 3자리 이상의 단어만을 반복문을 통해 구한다.

2-5 남은 단어들 중 각각 APP과 OTHER 토큰들 중 일치하는 토큰이 있다면 그 단어를 가져오고 없다면 1를 넣어준다.

2-6 일치하는 단어들은 해당하는 LN값을 더해주고 1로 바뀌어진 단어는 해당 APP과 OTHER의 토큰들의 합에 +1 한 총합의 역수를 취해준다. 그 후 LN을 취해준 후 아까의 값들에 더해준다.

```
In [67]: for i in range (len(result_app_predi)):
             for j in range (len(result_app_predi[i])):
                 if result_app_predi[i][j]==1:
                     app_predi=app_predi+(-7.78738)
                 else:
                     app_predi=app_predi+new_df[result_app_predi[i][j]][3]
             for k in range (len(result_other_predi[i])):
                 if result_other_predi[i][k]==1:
                     other_predi=other_predi+(-7.61431)
                 else:
                     other_predi=other_predi+new2_df[result_other_predi[i][k]][3]
             if app_predi>other_predi:
                 Ending_result.append("app")
             else:
                 Ending_result.append("other")
             app_predi=0
             other_predi=0
```

2-7 APP의 LN값과 OTHER의 LN값을 비교해준 후 예측된 결과를 출력한다.