

테스트 계획



김영호



테스트 LIST

테스트 1 : 유사도 분석1

테스트 2 : 유사도 분석2

테스트 3 : 문서 보고서 생성



테스트 1

유사도 분석 - 비교할 문서 제공



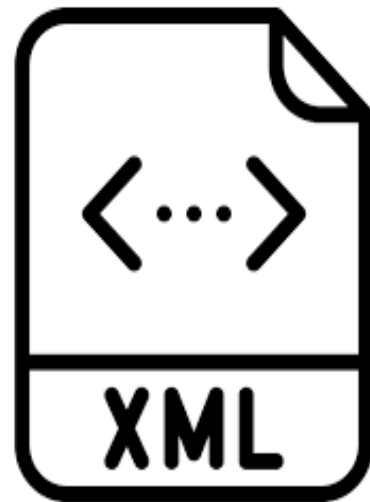
문서 제공

비교할 문서 A와
비교될 문서 B의
데이터화 및 데이터
베이스 저장

A와 B의 텍스트
유사도 분석 - cos
model

A와 B의 이미지
유사도 분석

이미지와 텍스트
유사도 결과를 각각
제공 및 견해 추가



장 유사한 단어 100개

```
pd.DataFrame(model.wv.most_similar(df.iloc[0]['단어'], topn=100), columns=['단어', df.iloc[0]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[1]['단어'], topn=100), columns=['단어', df.iloc[1]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[2]['단어'], topn=100), columns=['단어', df.iloc[2]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[3]['단어'], topn=100), columns=['단어', df.iloc[3]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[4]['단어'], topn=100), columns=['단어', df.iloc[4]['단어']])
[df.set_index(['단어']) for df in [df1, df2, df3, df4, df5]]
pd.concat(dfs, join='inner', axis=1).reset_index()
head(10)
```

어	지리산	주왕산	오대산	북한산	무등산
행	0.993371	0.991652	0.993379	0.989478	0.990786
내	0.992279	0.989850	0.990963	0.990785	0.984577
연	0.991686	0.990178	0.992029	0.989620	0.987261
태	0.991578	0.991713	0.991425	0.989564	0.990859
립공원	0.991563	0.990256	0.991777	0.990065	0.990014
행	0.991324	0.991155	0.990788	0.987760	0.987495
전	0.991318	0.990381	0.988500	0.987981	0.988239
원	0.991224	0.991438	0.990089	0.987834	0.986611
곡	0.991077	0.990552	0.991274	0.988901	0.988866
기	0.990900	0.988718	0.992161	0.988860	0.985771

요약

A와 B의 문서에서 유사한 단어들을 각각의 확률표로 정리하여 이를 파일 (csv, xml ...)로 저장해 제공



테스트 2

유사도 분석 - 비교할 문서 미제공



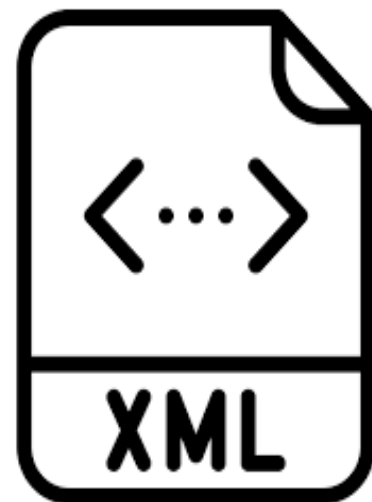
문서 미제공

비교될 문서 B 탐색 -
AI HUB

비교할 문서 A와
비교될 문서 B의
데이터화 및 데이터
베이스 저장

A와 B의 텍스트
유사도 분석 - cos
model

A와 B의 이미지
유사도 분석



장 유사한 단어 100개

```
pd.DataFrame(model.wv.most_similar(df.iloc[0]['단어'], topn=100), columns=['단어', df.iloc[0]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[1]['단어'], topn=100), columns=['단어', df.iloc[1]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[2]['단어'], topn=100), columns=['단어', df.iloc[2]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[3]['단어'], topn=100), columns=['단어', df.iloc[3]['단어']])
pd.DataFrame(model.wv.most_similar(df.iloc[4]['단어'], topn=100), columns=['단어', df.iloc[4]['단어']])
[df.set_index(['단어']) for df in [df1, df2, df3, df4, df5]]
pd.concat(dfs, join='inner', axis=1).reset_index()
head(10)
```

어	지리산	주왕산	오대산	북한산	무등산
행	0.993371	0.991652	0.993379	0.989478	0.990786
내	0.992279	0.989850	0.990963	0.990785	0.984577
연	0.991686	0.990178	0.992029	0.989620	0.987261
태	0.991578	0.991713	0.991425	0.989564	0.990859
립공원	0.991563	0.990256	0.991777	0.990065	0.990014
행	0.991324	0.991155	0.990788	0.987760	0.987495
전	0.991318	0.990381	0.988500	0.987981	0.988239
원	0.991224	0.991438	0.990089	0.987834	0.986611
곡	0.991077	0.990552	0.991274	0.988901	0.988866
기	0.990900	0.988718	0.992161	0.988860	0.985771

요약

A와 B의 문서에서 유사한 단어들을 각각의 확률표로 정리하여 이를 파일 (csv, xml ...)로 저장해 제공



테스트 3

문서 보고서 생성



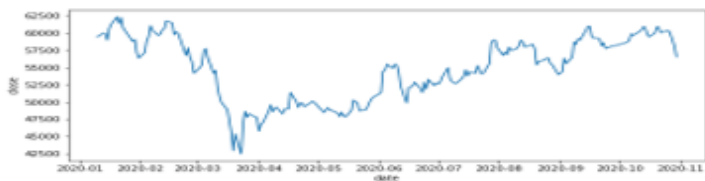
1

주식 보고서

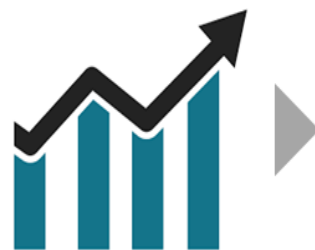
보고서 작성일 : 20201101

2

삼성전자, 56600원에 거래 마감



	date	close	open	high	low	volume
1	2020-01-16 00:00:00	56400	55000	56000	56400	262770428
2	2020-01-24 00:00:00	56100	56000	56200	56100	20524664
3	2020-01-28 00:00:00	56000	56000	56100	56000	14375882
4	2020-01-27 00:00:00	56000	56000	56100	56000	17283116
5	2020-01-24 00:00:00	56000	56000	56100	56000	12483726
6	2020-01-23 00:00:00	56000	56000	56100	56000	8227227
7	2020-01-22 00:00:00	56000	56000	56100	56000	14784095
8	2020-01-21 00:00:00	56000	56000	56100	56000	15781457
9	2020-01-20 00:00:00	56000	56000	56100	56000	18928115
10	2020-01-19 00:00:00	56000	56000	56100	56000	14474905



- 주식데이터 가져오기



- 분석 보고서 만들기



- 로그인
- 메일 전송

요약

지금까지의 데이터를 바탕으로 보고서를 제작하여 사용자에게 제공

+



○



●



감사합니다.