

# 캡스톤디자인 과제 결과보고서

과제명 : 지능형 시스템 프로젝트 결과 보고서.

캡스톤디자인 과제 결과보고서를 붙임과 같이 제출합니다.

지도교수 : (인)

대표학생 : 송인용 (인)

참여인원 : 대표학생 송인용 외 1명



건양대학교 대학혁신지원사업단장

# 목 차

## 제1장. 과제내용 및 목표

제1절. 현황

제2절. 목적 및 필요성

제3절. 과제의 목표

제4절. 기대효과 및 활용방안

## 제2장. 개념설계 및 상세설계

제1절. 개념설계

제2절. 상세설계

## 제3장. 제작

제1절. 제작

## 제4장. 운용 및 실험

제1절. 운용 및 실험 요구조건

제2절. 운용 및 실험결과

## 제5장. 결론

제1절. 문제점 분석 및 처리결과

제2절. 기업연계과제 결과 기술

제3절. 총평

제4절. 부록(시제품 사진)

제1장. 과제내용 및 목표

제1절. 현황

캡스톤디자인 교과목 운영지원 사업 현황				
교과목명	지능형 시스템			
팀 명	고원준, 송인용 팀			
팀 주제	IT 서적을 통한 기술 유행 파악			
지출액	0원			
작품작동여부	<input type="checkbox"/> 작동			
특허.실용신안				
지도교수	소 속	융합IT	휴대전화	
	성 명	권택원	E-mail	
대표학생	소 속	고원준, 송인용 팀	휴대전화	
	성 명	송인용	E-mail	ttyy441@naver.com
참 여 학 생				
No.	소속학과	학년	학번	이름
1	융합 IT 학과	3		고원준
2				
3				
4				
5				
6				
7				
8				
9				
10				

## **제2절. 목적 및 필요성**

자연어 처리 기술을 이용해 가장 많은 관심을 받는 IT 서적을 분석하여 현재 가장 인기 있는 프로그래밍 언어나 기술을 알아보고자 함.

그해 가장 많이 팔리거나 언급되는 it 서적을 분석해봐도 어느 기술이 관심을 많이 받는 지 알 수 있을 것으로 생각하였고, 앞으로의 학습 방향과 진로를 결정하는 데도 도움이 될 것으로 생각하여 it 서적 데이터를 분석해 보기로 결정.

## **제3절. 과제의 목표**

자연어 처리 기술을 이용해 가장 많은 관심을 받는 IT 서적을 분석하여 현재 가장 인기 있는 프로그래밍 언어나 기술을 파악.

## **제4절. 기대효과 및 활용방안**

현재 인기 있는 언어나 기술을 파악하여 앞으로의 학습 방향과 진로를 결정하는 데에 필요한 지표로 삼음.

크롤링, 자연어 처리, 모델 분석 등의 경험 축적.

각 프로그래밍 언어가 어떤 기술과 연관이 있는지 파악

분기별, 연도별 프로그래밍 언어 순위 측정 가능

## 제2장. 개념설계 및 상세설계

### 제1절. 개념설계

#### 1. 이론적 배경

특정 사이트에서 데이터를 읽어와 전처리 후 모델을 만들어 데이터를 평가.

#### 2. 제품 스케치

'X'

#### 3. 전체일정

10월 10일~13일 : 주제 선정.

10월 14일~15일 : 세부적인 프로젝트 진행 방향 설정.

10월 16일 : 프로젝트 최종 결정 및 프로젝트 진행에 필요한 기술 파악.

10월 17일 : Packt 사이트의 데이터 크롤링.

10월 18일 : Amazon 사이트의 데이터 크롤링.

10월 19일 : Packt와 Amazon 사이트의 베스트셀러 데이터 크롤링.

10월 20일 : 프로젝트 중간발표.

10월 21일 : 결과 시각화 및 다중 라벨링 작업.

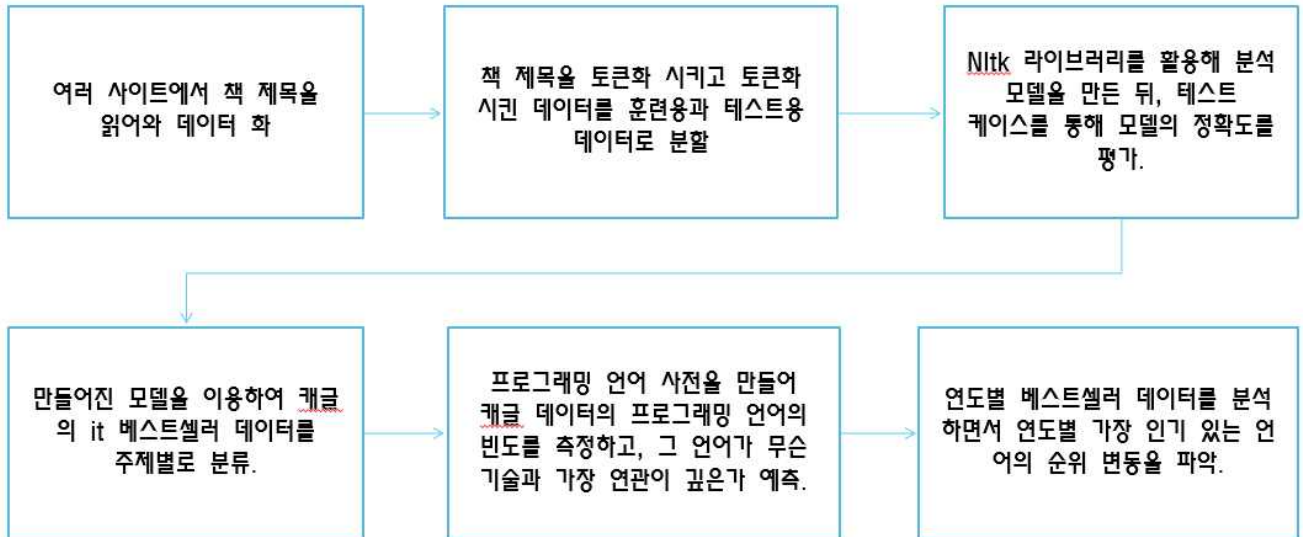
10월 22일 : 시연 영상 제작.

10월 23일 : 최종 프로젝트 발표.

10월 24일 : 캡스톤디자인 보고서 제출.

## 제2절. 상세설계

### 1. 상세설계도면 및 부품리스트



### 2. 도면 또는 부품 변경내역

연도별 인기상품 데이터 수집의 어려움으로 인해 연도별 가장 인기 있는 언어 순위에서 올해 가장 인기 있는 언어 순위로 변경.

## 제3장. 제 작

### 제1절. 제작

#### 1. 제작과정

Selenium을 이용하여 크롤링으로 서적 데이터 수집.

- > 토큰 생성기를 통한 수집 데이터 토큰화 및 구분자 제거.
- > TF-Df를 이용한 단어의 중요도 평가.
- > N그램을 이용하여 1~2개의 토큰 병합. ex) Big Data 등.
- > 불용어 및 특성값 제거.
- > 선형 판별 분석을 통한 모델 성능 평가.
- > 성능 향상을 위한 다중 라벨링 적용.

#### 2. 제작의 문제점 및 처리결과

크롤링 과정에서 동적 데이터 수집 불가 문제 > Selenium을 사용하여 해결.

모델 성능 측정 결과 로지스틱 모델이 나이브베이즈 모델보다 더 높은 성능을 내기에 나  
이브베이즈 알고리즘을 사용함

Big Data 등 2개 이상의 단어로 이루어진 단어들을 오직 한개의 토큰으로만 처리하면 정  
보의 손실이 일어나므로 N그램 기술을 사용하여 최대 2개의 단어로 이루어진 토큰을 사  
용함











## 제4장. 운용 및 시험

### 제1절. 운용 및 시험 요구조건













Selenium, NLTK 등의 라이브러리 설치.  
데이터 분석 환경을 위해 아나콘다 설치  
서적 데이터.

### 제2절. 운용 및 시험결과

Packt 서적 데이터



 0_AllBookName.xlsx	2020-10-20 오후 11:15	Microsoft Excel 워...	158KB
 1_Business_And_Other.xlsx	2020-10-20 오후 10:59	Microsoft Excel 워...	18KB
 2_Cloud_And_Networking.xlsx	2020-10-20 오후 11:02	Microsoft Excel 워...	32KB
 3_Data.xlsx	2020-10-20 오후 11:04	Microsoft Excel 워...	31KB
 4_GaneDevelopment.xlsx	2020-10-20 오후 11:05	Microsoft Excel 워...	16KB
 5_IoT_And_Hardware.xlsx	2020-10-20 오후 11:06	Microsoft Excel 워...	11KB
 6_Mobile.xlsx	2020-10-20 오후 11:07	Microsoft Excel 워...	16KB
 7_Programming.xlsx	2020-10-20 오후 11:10	Microsoft Excel 워...	31KB
 8_Tech.xlsx	2020-10-20 오후 11:12	Microsoft Excel 워...	31KB
 9_WebDevelopment.xlsx	2020-10-20 오후 11:15	Microsoft Excel 워...	31KB

Amazon 서적 데이터

 0_total.xlsx	2020-10-22 오전 3:09	Microsoft Excel 워...	778KB
 1_Programming.xlsx	2020-10-22 오후 12:38	Microsoft Excel 워...	8KB
 2_ComputerScience.xlsx	2020-10-22 오전 12:38	Microsoft Excel 워...	84KB
 4_ProgrammingLanguages.xlsx	2020-10-22 오전 12:56	Microsoft Excel 워...	97KB
 5_Networking_And_Cloudcomputing.xlsx	2020-10-22 오전 1:07	Microsoft Excel 워...	59KB
 6_Database_And_BigData.xlsx	2020-10-22 오전 1:26	Microsoft Excel 워...	97KB
 7_Hardware_And_DIY.xlsx	2020-10-22 오전 1:39	Microsoft Excel 워...	61KB
 8_Graphics_And_Design.xlsx	2020-10-22 오전 2:02	Microsoft Excel 워...	117KB
 9_Software.xlsx	2020-10-22 오전 2:22	Microsoft Excel 워...	86KB
 10_Certification.xlsx	2020-10-22 오전 2:44	Microsoft Excel 워...	102KB
 11_WebDevelopment_And_Design.xlsx	2020-10-22 오후 12:29	Microsoft Excel 워...	71KB
 12_OperatingSystem.xlsx	2020-10-22 오전 3:09	Microsoft Excel 워...	70KB



## BestSellers 서적 데이터

 Amazon_2020_Best.xlsx	2020-10-21 오후 9:53	Microsoft Excel 워...	15KB
 Packt_2019_Best.xlsx	2020-10-19 오후 8:13	Microsoft Excel 워...	7KB
 Packt_2020_Best.xlsx	2020-10-22 오후 12:41	Microsoft Excel 워...	15KB

## SVD 모델을 적용한 이후의 모델 예측도에 영향을 미치는 특성

사용된 특성 수: 14511

가장 낮은 tfidf를 가진 특성:

```
['money', 'development raspberry', 'database upgrade', 'soul', '2019.1 ?',
'collaboration', '12 :', 'christian', 'hackers', 'programming perl',
'tensorflow lite', 'kibana 7', 'ultimate white', ': adventures', ', 10th',
'network kali', 'xlmner', 'lightning-fast data', 'parallelize distribute',
'docbook']
```

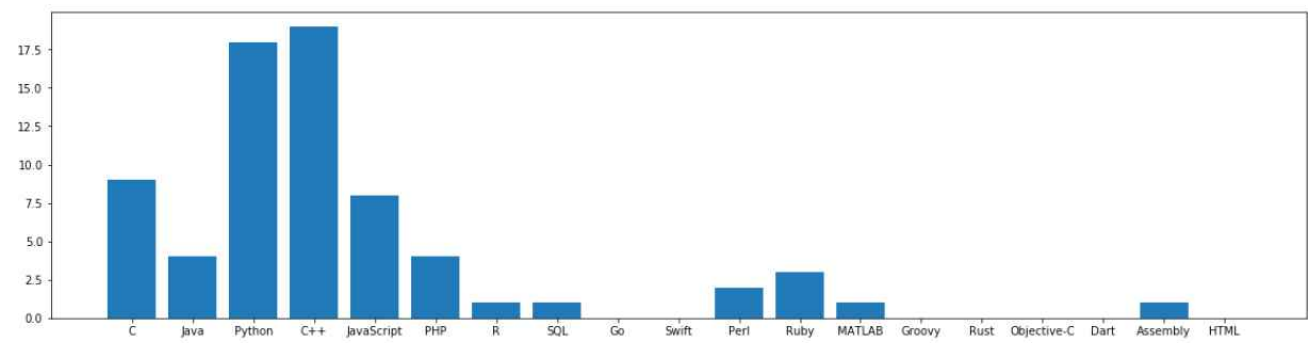
가장 높은 tfidf를 가진 특성:

```
['algorithms', 'compiling', 'microcontroller', 'method', 'common',
'associations', 'public', 'beginning', 'python', 'instant', 'java', 'fine',
'sql', '3', 'using', 'freeswitch', 'mastering', 'virtualization', 'monitoring',
'metadata']
```

## 다중 라벨링 적용 이후 해당 언어와 가장 연관이 깊은 기술의 관계 표

	language	label
0	C	WebDevelopment
1	Java	WebDevelopment
2	Python	WebDevelopment
3	C++	WebDevelopment
4	JavaScript	WebDevelopment
5	PHP	Data
6	R	Data
7	SQL	Data
8	Go	WebDevelopment
9	Swift	WebDevelopment
10	Perl	WebDevelopment
11	Ruby	WebDevelopment
12	MATLAB	Business&Other   Data
13	Groovy	WebDevelopment
14	Rust	WebDevelopment
15	Objective-C	WebDevelopment
16	Dart	WebDevelopment
17	Assembly	WebDevelopment
18	HTML	WebDevelopment

최근 도서에서 가장 많이 언급되는 언어 순위 시각화



## 제5장. 결론

### 제1절. 문제점 분석 및 처리결과

#### 1. 과제에 대한 문제 분석

##### 가. 문제점

단일 라벨링 및 자료 부족으로 인한 낮은 예측 결괏값.

##### 나. 개선방안 및 처리결과

추가 데이터 공급.

다중 라벨링 작업.

### 제3절. 총평

고원준 : 이번 지능형 시스템 프로젝트를 진행하면서 처음 접해보는 내용이 많아 다른 과목보다 어려움이 많았습니다. 하지만 포기하지 않고 노력한 결과 이렇게 끝을 맞이하였고 개인적으로 생각하기에는 좋은 결과를 얻을 수 있던 것 같습니다. 처음에는 단순히 캐글 데이터만을 가지고 해당 데이터에서 자주 언급되는 언어와 기술을 알아보려고 하였으나 그렇게 진행될 경우 저희가 진행한 수업 내용과는 맞지 않는다고 생각하여 현재의 주제로 변경하게 되었습니다. 아직 성능이 그렇게 좋지는 않아 개선할 부분이 있지만 만족할만한 결과를 얻었습니다. 얻은 결과를 가지고 앞으로의 학습 방향을 설정하는 데에 조금이나마 도움이 될 수 있을 것 같습니다.

송인용: 처음 자연어 처리를 하다 보니 기술적 미숙함 등을 이유로 초기 프로젝트 가정처럼 정확한 분류가 나오지는 않았습니다.

하지만 이번 프로젝트를 통해서 자연어 처리 모델을 만드는 법을 배울 수 있었고, 아직 개선 가능성이 충분하다고 생각해, 모델을 개선할 방법을 계속 찾아보고 어느 정도의 완성도가 나오면 웹에서 구동할 수 있게끔 만들어 보려고 합니다.

## 부록(시제품 사진)

0_AllBookName.xlsx	2020-10-20 오후 11:15	Microsoft Excel 워...	158KB
1_Business_And_Other.xlsx	2020-10-20 오후 10:59	Microsoft Excel 워...	18KB
2_Cloud_And_Networking.xlsx	2020-10-20 오후 11:02	Microsoft Excel 워...	32KB
3_Data.xlsx	2020-10-20 오후 11:04	Microsoft Excel 워...	31KB
4_GameDevelopment.xlsx	2020-10-20 오후 11:05	Microsoft Excel 워...	16KB
5_IoT_And_Hardware.xlsx	2020-10-20 오후 11:06	Microsoft Excel 워...	11KB
6_Mobile.xlsx	2020-10-20 오후 11:07	Microsoft Excel 워...	16KB
7_Programming.xlsx	2020-10-20 오후 11:10	Microsoft Excel 워...	31KB
8_Tech.xlsx	2020-10-20 오후 11:12	Microsoft Excel 워...	31KB
9_WebDevelopment.xlsx	2020-10-20 오후 11:15	Microsoft Excel 워...	31KB

사진내용 : Packt 사이트 크롤링 결과

0_total.xlsx	2020-10-22 오전 3:09	Microsoft Excel 워...	778KB
1_Programming.xlsx	2020-10-22 오후 12:38	Microsoft Excel 워...	8KB
2_ComputerScience.xlsx	2020-10-22 오전 12:38	Microsoft Excel 워...	84KB
4_ProgrammingLanguages.xlsx	2020-10-22 오전 12:56	Microsoft Excel 워...	97KB
5_Networking_And_Cloudcomputing.xlsx	2020-10-22 오전 1:07	Microsoft Excel 워...	59KB
6_Database_And_BigData.xlsx	2020-10-22 오전 1:26	Microsoft Excel 워...	97KB
7_Hardware_And_DIY.xlsx	2020-10-22 오전 1:39	Microsoft Excel 워...	61KB
8_Graphics_And_Design.xlsx	2020-10-22 오전 2:02	Microsoft Excel 워...	117KB
9_Software.xlsx	2020-10-22 오전 2:22	Microsoft Excel 워...	86KB
10_Certification.xlsx	2020-10-22 오전 2:44	Microsoft Excel 워...	102KB
11_WebDevelopment_And_Design.xlsx	2020-10-22 오후 12:29	Microsoft Excel 워...	71KB
12_OperatingSystem.xlsx	2020-10-22 오전 3:09	Microsoft Excel 워...	70KB

사진내용 : Amazon 사이트 크롤링 결과

Amazon_2020_Best.xlsx	2020-10-21 오후 9:53	Microsoft Excel 워...	15KB
Packt_2019_Best.xlsx	2020-10-19 오후 8:13	Microsoft Excel 워...	7KB
Packt_2020_Best.xlsx	2020-10-22 오후 12:41	Microsoft Excel 워...	15KB

사진내용 : BestSellers 데이터 크롤링 결과.

<pre>from IPython.display import display import numpy as np import matplotlib.pyplot as plt import pandas as pd import mglearn</pre>		
<pre>from nltk.tokenize import word_tokenize</pre>		
<pre>Adata = pd.read_csv('데이터_최종.csv') # 데이터의 NaN값 제거 Adata2 = Adata.dropna(axis=0)</pre>		
<pre>#일부 크롤링한 데이터중 #nBook 같은 잘못된 데이터가 들어 있어 제거 Adata2['title'] = Adata2['title'].str.rstrip('#nBook')</pre>		
<pre>Adata2.head()</pre>		
	title	theme
0	Mastering QuickBooks 2020	Business&Other
1	Hands-On Microsoft Teams	Business&Other
2	Microsoft 365 and SharePoint Online Cookb	Business&Other
3	Microsoft Power Platform Enterprise Architecture	Business&Other
4	Odoo 12 Development Cookbook - Third Edition	Business&Other

사진내용 : 데이터 전처리.

```

from sklearn.naive_bayes import MultinomialNB

from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.pipeline import Pipeline

from sklearn.feature_extraction.text import TfidfVectorizer

# 트위터나 페이스북 같은 SNS 텍스트를 분석할때 사용하는 토큰화수
from nltk.tokenize.casual import casual_tokenize

#SVD 알고리즘 적용 모델
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

```

```

# casual_tokenize 사용한 모델
#모델에 Tf-IDF 추가
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(tokenizer=casual_tokenize,min_df=5, ngram_range=(1,2),stop_words="english")),
    ('classifier', MultinomialNB()), # train on Tf-IDF vectors w/ Naive Bayes classifier
])

```

사진내용 : 라이브러리 선언 및  
모델에 Tf-IDF 적용.

```

pipeline.fit(X,y)

Pipeline(memory=None,
       steps=[('tfidf',
                TfidfVectorizer(analyzer='word', binary=False,
                                decode_error='strict',
                                dtype=<class 'numpy.float64'>,
                                encoding='utf-8', input='content',
                                lowercase=True, max_df=1.0, max_features=None,
                                min_df=5, ngram_range=(1, 2), norm='l2',
                                preprocessor=None, smooth_idf=True,
                                stop_words='english', strip_accents=None,
                                sublinear_tf=False,
                                token_pattern='(?u)#####\w+\b',
                                tokenizer=<function casual_tokenize at 0x0000016C3680840>,
                                use_idf=True, vocabulary=None)),
               ('classifier',
                MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True))],
       verbose=False)

```

사진내용 : Pipeline Fit.

```

#SVD 알고리즘 적용 모델

tfidf_model = TfidfVectorizer(tokenizer=casual_tokenize,min_df=5, ngram_range=(1,2),stop_words="english")
tfidf_docs = tfidf_model.fit_transform(X).toarray()

LX_train, LX_test, Ly_train, Ly_test = train_test_split(tfidf_docs, y, random_state=50)

lda = LinearDiscriminantAnalysis()

model = lda.fit(LX_train, Ly_train)

```

사진내용 : SVD 알고리즘 적용 및  
Model Fit.

```

#다중 리플링으로 모델을 평가하기에 test 세트들 정확도를 측정하지는 않음
#최종 모델에는 train, test 구분없이 사용
#어떤 특성이 모델 예측도에 영향을 끼쳤나 확인

vectorizer = pipeline("tfidf")
train = vectorizer.transform(X_train)
max_value = train.max(axis=0).toarray().ravel()
sorted_by_tfidf = max_value.argsort()

feature_names = np.array(vectorizer.get_feature_names())
print("사용된 특성 수:", len(feature_names))
print("가장 낮은 tfidf를 가진 특성:\n", feature_names[sorted_by_tfidf[:20]])
print("가장 높은 tfidf를 가진 특성: \n", feature_names[sorted_by_tfidf[-20:]])

사용된 특성 수: 7934
가장 낮은 tfidf를 가진 특성:
['bulldog |' 'pages |' 'story pages' 'write |' 'primary' 'space |'
 'draw write' 'journal kids' '100 story' '|' dotted' 'exercise book'
 '|' grades' 'dotted midline' '2 school' 'kids draw' 'grades k'
 'picture space' 'outer space' 'midline picture' '(' outer']
가장 높은 tfidf를 가진 특성:
['advanced' 'design' 'java' 'scanning' 'ii' 'microservices' 'software'
 'libreoffice' 'soa' 'android' 'freewitch' 'cryptology' 'workshop' ':'
 'unity' 'mobile' 'beginning' 'working' 'technology' 'essential']

```

사진내용 : SVD 모델을 적용시키지 않은  
상태의 모델 예측도에 영향을 미치는  
특성 확인.

```
#SVD 모델 측정

max_value = LX_train.max(axis=0).ravel()

sorted_by_tfidf = max_value.argsort()

feature_names = np.array(tfidf_model.get_feature_names())
print("사용된 특성 수:", len(feature_names))
print("가장 낮은 tfidf를 가진 특성: ", feature_names[sorted_by_tfidf[:20]])
print("가장 높은 tfidf를 가진 특성: ", feature_names[sorted_by_tfidf[-20:]])
```

사용된 특성 수: 14511  
가장 낮은 tfidf를 가진 특성:  
['money', 'development raspberry', 'database upgrade', 'soul', '2019.1 ?',  
'collaboration', '12', 'christian', 'hackers', 'programming perl',  
'tensorflow lite', 'kibana 7', 'ultimate white', 'adventures', '10th',  
'network kali', 'xlmr', 'lightning-fast data', 'parallelize distribute',  
'docbook']  
가장 높은 tfidf를 가진 특성:  
['algorithms', 'compiling', 'microcontroller', 'method', 'common',  
'associations', 'public', 'beginning', 'python', 'instant', 'java', 'fine',  
'sql', '3', 'using', 'freeswitch', 'mastering', 'virtualization', 'monitoring',  
'metadata']

사진내용 : SVD 모델을 적용시킨 상태의 모델 예측도에 영향을 미치는 특성 확인.

#데이터가 테스트 데이터가 되었으므로 train이 아닌 데이터 전부를 사용해 모델 구축

```
#model.fit(tfidf_docs,y)
predictions = pipeline.predict(prog_book['Book_title'])
```

probe = pipeline.predict\_proba(predictions) #일반 모델

```
#label_result에 사용할 컬럼
column = Adata2['theme'].unique()
```

df\_prediction = pd.DataFrame(probe, columns=column)

df\_prediction.head()

	Business&Other	Cloud&Networking	Data	Game Development	IoT&Hardware	Mobile	Programming	Security	WebDevelopment
0	0.389221	0.107236	0.242714	0.005677	0.080422	0.011945	0.092814	0.056026	0.013046
1	0.030565	0.048029	0.052244	0.002776	0.749548	0.004651	0.073855	0.031797	0.004434
2	0.057616	0.045134	0.135910	0.014499	0.108406	0.018751	0.589394	0.010105	0.020106
3	0.049407	0.334987	0.072929	0.003841	0.137898	0.015159	0.029247	0.340527	0.019825
4	0.110802	0.565544	0.105528	0.001157	0.140165	0.002576	0.032144	0.030044	0.011910

사진내용 : 모델 구축.

```
#다중 라벨링 작업
count = len(column)
for i in range(0, len(df_prediction)):

    label = []
    pmax = max(df_prediction.loc[i])

    for j in range(0, count):
        if pmax * 0.8 < df_prediction.loc[i][j]:

            label.append(column[j])
            print(label)
    result['label'][i] = " | ".join(label)
```

```
['Business&Other']
['IoT&Hardware']
['Programming']
['Cloud&Networking']
['Cloud&Networking', 'Security']
['Cloud&Networking']
['Data']
['Cloud&Networking']
['IoT&Hardware']
['Data']
['Programming']
['Data']
['IoT&Hardware']
['Business&Other']
['Data']
['Programming']
['Cloud&Networking']
['Cloud&Networking', 'Security']
['IoT&Hardware']
['IoT&Hardware']
```

사진내용 : 다중 라벨링 작업 1.

```
#다중 라벨링 작업
count = len(column)
for i in range(0, len(df_prediction)):
```

```
label = []
pmax = max(df_prediction.loc[i])
```

```
for j in range(0, count):
    if pmax * 0.5 < df_prediction.loc[i][j]:
```

```
label.append(column[j])
```

```
result['label'][i] = " | ".join(label)
```

사진내용 : 다중 라벨링 작업 2.

#각언어가 어떤 테마랑 가장 관련 있는지 보여줄

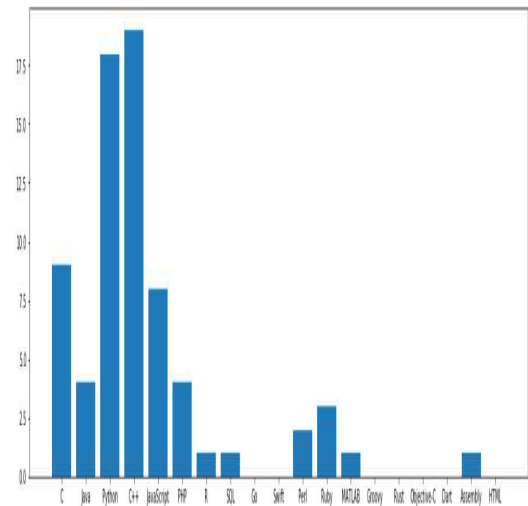
result\_lang

	language	label
0	C	WebDevelopment
1	Java	WebDevelopment
2	Python	WebDevelopment
3	C++	WebDevelopment
4	JavaScript	WebDevelopment
5	PHP	Data
6	R	Data
7	SQL	Data
8	Go	WebDevelopment
9	Swift	WebDevelopment
10	Perl	WebDevelopment
11	Ruby	WebDevelopment
12	MATLAB	Business&Other   Data
13	Groovy	WebDevelopment
14	Rust	WebDevelopment
15	Objective-C	WebDevelopment
16	Dart	WebDevelopment
17	Assembly	WebDevelopment
18	HTML	WebDevelopment

사진내용 : 언어와 테마의 상관 관계.

```
lang['count'] = lang_count
plt.figure(figsize=(20, 6))
plt.bar(lang['language'], lang['count'])

plt.show()
```



사진내용 : 최근 도서에서 가장 많이 언급되는 언어 순위

사진내용 :

사진내용 :