

PORTFOLIO

문지현, Jihyun Moon

Github: github.com/solidcellaMoon

개인 블로그: star-crab.tistory.com

프로그래밍 프로젝트

001 자연어 처리와 기계학습을 통한 우울 감정 분석과 인식 (3~11p)

002 다양한 기술 스택을 활용한 웹사이트 구현 (12~19p)

003 데이터 분석과 머신러닝을 위한 스터디 (20~22p)

001 자연어 처리와 머신러닝을 통한 우울 감정 분석과 인식

프로젝트 개요

사용자가 한글 자연어 텍스트를 작성하면 글 속에 우울한 감정이 있는지 없는지 여부를 판단합니다. 이 때 Python을 활용하여 텍스트를 분석 및 예측하고, 가장 높은 정확도를 보인 예측모델을 바탕으로 사용자가 텍스트를 입력하면 예측결과를 알 수 있는 일기장 데모사이트를 제작했습니다.

주요 특징

- 한글 텍스트 데이터 사용
- SNS에서 자연어 데이터를 크롤링한 후, 학습에 용이한 형태로 전처리 및 정제하여 사용
- 긍/부정이 아닌, “우울함”이라는 특정 감정을 기준으로 분류하는 예측모델 구현
- 프로젝트 진행 과정과 최종 결과를 정리하여 논문으로 발표

데이터 분석 프로젝트

자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

기술 스택



프로젝트 내의 역할

1. SNS 웹크롤링을 통한 텍스트 데이터셋 수집
2. 데이터셋 전처리 및 분류 작업
3. 최종 데이터셋 개요와 예측모델 성능을 시각화

자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

001

SNS 웹크롤링 - 검색

GetOldTweet3 패키지 사용

```
#1. -----
# 가져올 범위를 정의
days_range = []

start = datetime.datetime.strptime("2019-03-01", "%Y-%m-%d")
end = datetime.datetime.strptime("2020-07-01", "%Y-%m-%d")
date_generated = [start + datetime.timedelta(days=x) for x in range(0, (end-start).days)]

for date in date_generated:
    days_range.append(date.strftime("%Y-%m-%d"))

print("설정된 트윗 수집 기간: {} ~ {}".format(days_range[0], days_range[-1]))
print("총 {}일간의 데이터 수집중!".format(len(days_range)))

# 특정 검색어가 포함된 트윗 검색! (query search)
# str에 검색어 입력
str = "우울"
print("검색 단어: [ %s ]" %str)

#수집 기간 맞추기
start_date = days_range[0]
end_date = (datetime.datetime.strptime(days_range[-1], "%Y-%m-%d")
            + datetime.timedelta(days=1)).strftime("%Y-%m-%d")
# setUntil이 끝을 포함하지 않으므로, day + 1

#트윗 수집 기준 정의
tweetCriteria = got.manager.TweetCriteria().setQuerySearch(str)\
    .setSince(start_date)\
    .setUntil(end_date)\
    .setMaxTweets(-1)

#수집 작업
print("{} 에서 {} 까지 검색 시작".format(days_range[0], days_range[-1]))
start_time = time.time()

tweet = got.manager.TweetManager.getTweets(tweetCriteria)
print("=== 전체 수집 트윗 개수: {} ===".format(len(tweet)))
```

검색할 기간의 범위를 설정

“우울” 감정과 관련된 단어를
검색어로 설정

수집 단계가 다소 시간이 걸리는 작업이기에
특정 기간의 트윗 수가 5000개 이상일 경우
더 짧은 기간으로 검색한 뒤 수집합니다.

자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

001

SNS 웹크롤링 - 수집

```
#2. -----
# 트윗에서 원하는 정보를 골라서 저장한다.
from random import uniform
from tqdm import notebook

# 초기화
tweet_data = []

for index in notebook.tqdm(tweet):
    # 수집 데이터 목록
    # 순서대로 유저이름, 트윗내용, RT수, 마일수, 작성날짜, 작성시간
    username = index.username
    content = index.text
    #retweets = index.retweets
    #favorites = index.favorites
    tweet_date = index.date.strftime("%Y-%m-%d")
    tweet_time = index.date.strftime("%H:%M:%S")

    # 결과 합치기
    data_list = [username, content, tweet_date, tweet_time]
    tweet_data.append(data_list)

    # 과도한 수집 방지를 위한 휴식 (1~2초)
    time.sleep(uniform(1, 2))

#pandas DataFrame으로 변환
import pandas as pd
twitter_df = pd.DataFrame(tweet_data, columns=["user_name", "text", "date", "time"])

#csv 파일 만들기
twitter_df.to_csv("Twt_{0}_to_{1}.csv".format(days_range[0], days_range[-1]), index=False)
print("=== {} 개의 트윗 저장 완료 ===".format(len(tweet_data)))
```

GetOldTweet3 패키지 사용

트윗에서 수집할 정보를 설정

DataFrame으로 변환 후 csv로 저장

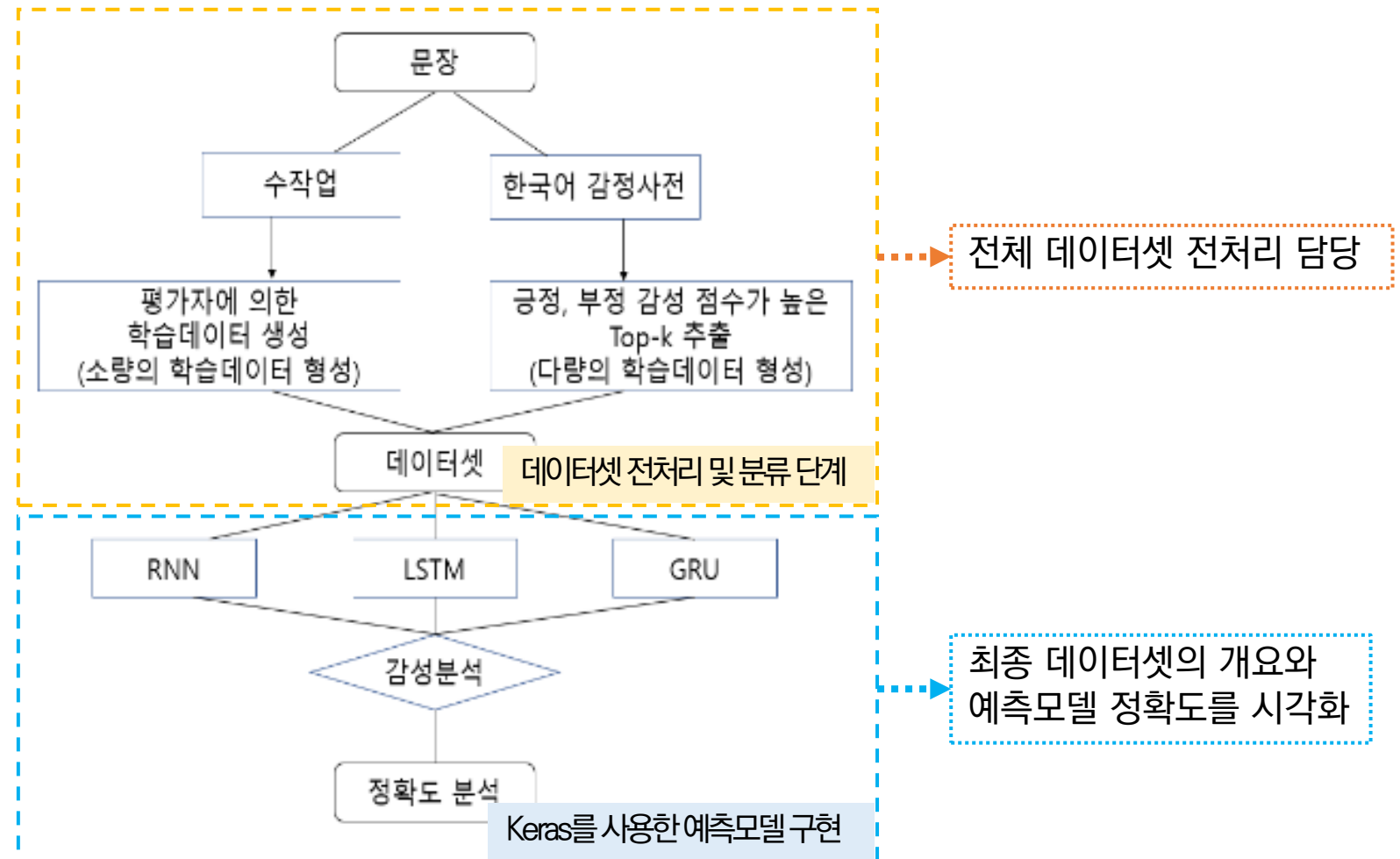
자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

전처리 및 분류 ~ 예측모델 구현 과정



팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

1. KONLPy를 활용하여 형태소, 명사, 어간 등을 추출하고 품사 태깅
2. wordCloud를 만들어 학습 데이터셋의 개요 확인

- 광고, 외부 링크, 외국어, 심한 욕설, 장기간 반복되는 트윗 제외
- KONLPy를 이용하여 형태소, 명사, 어간 등을 추출하고 품사 태깅
- 9만개의 문장에서, 최종 정제 후 총 2232개 문장의 학습 데이터셋 완성



우울함을 나타내는 문장: 1200개



우울함과 반대되는 문장: 1032개

자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

형태소 추출 및 품사 태깅

```
#csv는 ANSI로 저장해야함.
#형태소 추출
def extractMorph(text_list):
    tokens = []
    search = Okt()
    print('형태소 추출 시작')
    for i in text_list:
        word = search.morphs(i, stem=True) #stem=True 하면 어간 추출 (ex: 해서 -> 하다)
        for j in word:
            if j in stopwords:
                #stopwords = 제외할 단어 리스트
                continue
            else:
                tokens.append(j)

    print('형태소 추출 완료!')
    return tokens

#명사 추출
def extractNoun(text_list):...
#어간 추출
def extractPhrase(text_list):...
#용언 추출
def extractPRD(text_list):...

# 품사 태깅 후 해당 품사만 저장 ----- (현재 용언에만 맞춰짐)
def tokenTagging(tokens, word_type):
    pos_arr = []
    print('태깅 시작')
    for i in tokens: # 품사를 붙여서 튜플형태로 저장
        list = [t[0] for t in search.pos(i) if t[1] == word_type]
        if len(list) != 0:
            str = " ".join(list)
            pos_arr.append(str)

    print('태깅 완료!\n',word_type,"만 저장함.")
    return pos_arr
```

품사 추출 시,
필요 없는 특정 단어는 제외

각 품사별로 추출하는 함수 생성

자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

진행기간: 2019.07.21 ~ 2020.06.10

예측 모델 구현 및 시각화 코드

1. 수집한 데이터셋에 0 또는 1의 label 부여

```
labels = []  
# 우울 0 우울아님 1  
for i in range(len(Data)):  
    labels.append(0)  
for i in range(len(Data2)):  
    labels.append(1)
```

2. RNN, LSTM, GRU 총 3가지 모델로 예측, 배치사이즈를 조정하며 정확도 비교

```
from keras.models import Sequential  
from keras.layers import Embedding, Flatten, Dense  
from keras.layers import SimpleRNN  
  
model = Sequential()  
model.add(Embedding(max_words, embedding_dim, input_length=maxlen))  
model.add(SimpleRNN(32, input_shape=(3,1)))  
model.add(Dense(1, activation='sigmoid'))  
model.summary()
```

3. 모델의 정확도 확인을 위한 그래프 표시

그래프

```
import matplotlib.pyplot as plt  
  
acc = history.history['acc']  
val_acc = history.history['val_acc']  
loss = history.history['loss']  
val_loss = history.history['val_loss']
```

```
epochs = range(1, len(acc) + 1)
```

정확도 그래프

```
plt.plot(epochs, acc, 'bo', label='Training acc')  
plt.plot(epochs, val_acc, 'b', label='Validation acc')  
plt.title('RNN : Training and validation accuracy')  
plt.legend()
```

```
plt.figure()
```

손실 그래프

```
plt.plot(epochs, loss, 'bo', label='Training loss')  
plt.plot(epochs, val_loss, 'b', label='Validation loss')  
plt.title('RNN : Training and validation loss')  
plt.legend()
```

```
plt.show()
```

데이터 분석 프로젝트

자연어처리와 기계학습을 통한 우울 감정 분석과 인식

이화여자대학교 캡스톤디자인 수업에서
진행한 졸업 프로젝트 주제
2020년 IPACT JCCT 5월호 논문 게재

팀원 구성: 2인

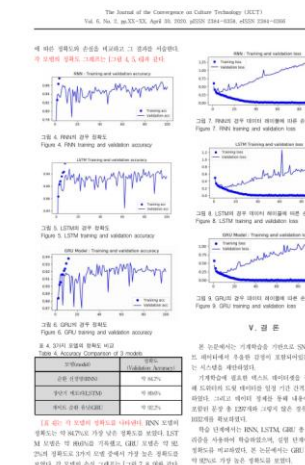
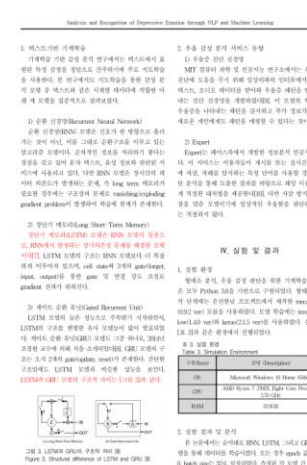
진행기간: 2019.07.21 ~ 2020.06.10

논문 발표

프로젝트 진행 과정과 모델링 결과를 정리하여 논문으로 작성했습니다.
지도교수님의 검토 후 2020년 IPACT JCCT 5월호에 게재되었습니다.

논문 링크: <https://doi.org/10.17703/JCCT.2020.6.2.449>

자연어처리와 기계학습을 통한 우울 감정 분석과 인식
Analysis and Recognition of Depressive Emotion
through NLP and Machine Learning



002 다양한 기술 스택을 활용한 웹사이트 구현

프로젝트 개요

고학번 대상 전공 수업인 “빅데이터 응용” 과목에서 진행한 2인 소규모 프로젝트입니다.
사용자가 영화 데이터를 검색, 분석하고 새로운 데이터를 생성할 수 있는 php 기반 웹사이트를 제작합니다.
필요한 영화 데이터셋을 캐글에서 가져온 뒤, 검색 및 분석이 용이한 형태로 전처리를 진행합니다.
최종 전처리가 끝난 데이터는 phpmyadmin을 통해 사이트DB에 저장합니다.

주요 특징

- Pandas와 Jupyter Notebook을 사용한 빠른 데이터 전처리 및 재구성
- 캐글에서 받은 초기 데이터는 csv파일안에 json 형식이 포함됨

데이터 분석 프로젝트

다양한 기술 스택을 활용한 웹사이트 구현

Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원 구성: 2인

진행기간: 2020.09.28~ 2020.11.23

기술 스택



Pandas



프로젝트 내의 역할

1. 초기 csv 내부의 json 형식을 DataFrame으로 변환
2. 결측치 제거 등 데이터 전처리
3. DB 설계도에 맞춰 최종 csv파일 생성 후 DB에 삽입
4. 페이지 로그인/회원가입/검색기능 구현

다양한 기술 스택을 활용한 웹사이트 구현

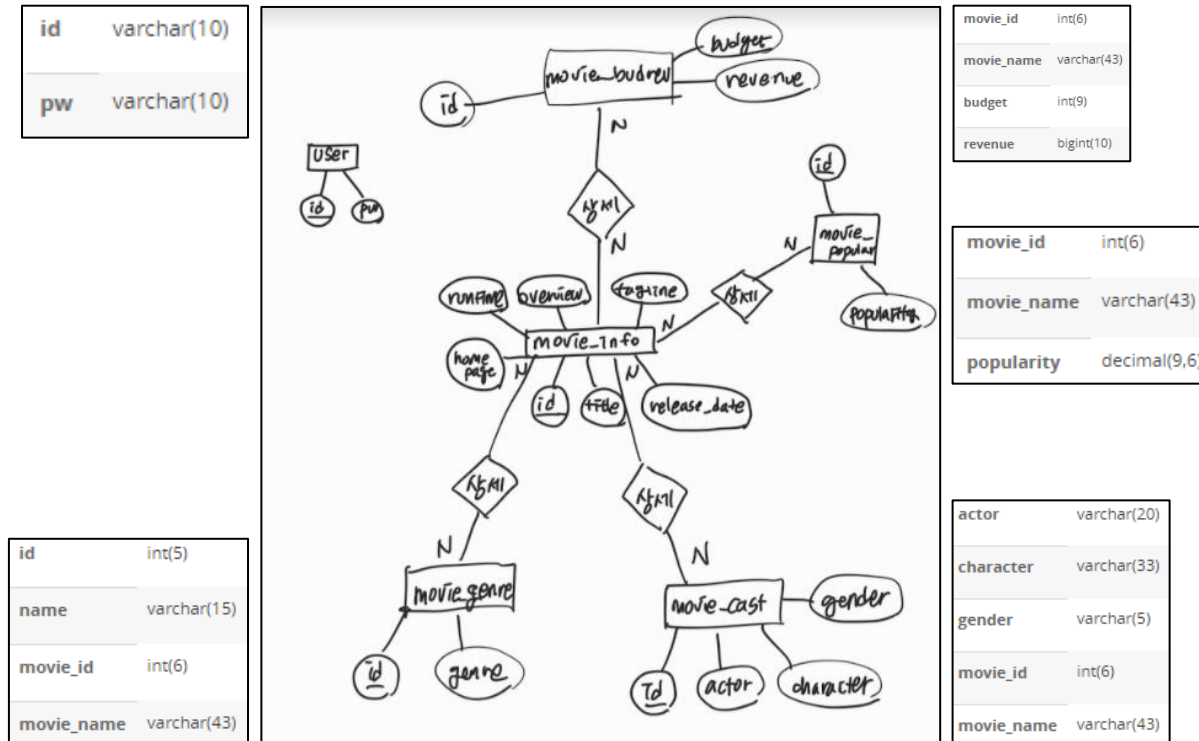
Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원 구성: 2인

진행기간: 2020.09.28~2020.11.23

ER Diagram for DB

기획 단계에 작성한 데이터베이스의 ER 다이어그램



최종 데이터 파일이 위 구조도와 동일하도록 초기 데이터셋 전처리 진행

다양한 기술 스택을 활용한 웹사이트 구현

Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원구성: 2인

진행기간: 2020.09.28~2020.11.23

전처리 코드 일부 - json 형식 변환

내용을 정리한 블로그 글: <https://star-crab.tistory.com/18>

장르+영화 테이블

```
movieGen = pd.DataFrame({"id": [], "name": [], "movie_id": []}) #최종결과를 저장할 DF

for i in gen.index:
    #임시저장용 DF에 개별 영화의 정보를 저장한다.
    df2 = pd.DataFrame({"id": [], "name": []}) #임시로 저장할 DF
    df2 = df2.append(pd.read_json(gen[i])) #json형식이 포함된 컬럼은 read_json으로 변환
    df2['movie_id'] = df['id'][i]
    df2['movie_Name'] = df['original_title'][i]
    #개별 영화의 정보들을 최종결과 DF에 저장한다.
    movieGen = movieGen.append(df2)

movieGen = movieGen.astype({'id': 'int', 'movie_id': 'int'})
movieGen.reset_index(inplace=True) #인덱스 재설정
movieGen.drop('index', axis=1, inplace=True)
movieGen.to_csv('movie_genres.csv', mode='w') #최종결과를 csv로 저장
```

개별 영화마다 다수의 레코드를 갖기에
임시저장용 DF에 저장

임시저장용 DF의 인덱스는 정렬되지 않았기에
최종 DF의 인덱스를 재설정해준 뒤 csv로 저장

	id	name	movie_id	movie_Name
0	28	Action	19995	Avatar
1	12	Adventure	19995	Avatar
2	14	Fantasy	19995	Avatar
3	878	Science Fiction	19995	Avatar
4	12	Adventure	285	Pirates of the Caribbean: At World's End
...
57	28	Action	1930	The Amazing Spider-Man
58	12	Adventure	1930	The Amazing Spider-Man
59	14	Fantasy	1930	The Amazing Spider-Man
60	28	Action	20662	Robin Hood
61	12	Adventure	20662	Robin Hood

62 rows x 4 columns

코드 실행 후 DataFrame 형태

다양한 기술 스택을 활용한 웹사이트 구현

Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원 구성: 2인

진행기간: 2020.09.28~ 2020.11.23

로그인, 회원가입 페이지 구현

```
$id=$_POST['id'];
$pw=$_POST['pw'];

// 입력받은 데이터를 DB에 저장
$query = "insert into user (id, pw) values ('$id','$pw')";

$result = $connect->query($query);

// 저장이 됐다면 (result = true) 가입 완료
if($result) {
    ?>
    <script>
        alert('가입 되었습니다.');
```

City

Dhaka

Dilli

Newyork

Islamabad

Primary movie

signin

```
        location.replace("login.php");
    </script>
}
else{
    <script>
        alert("fail");
    </script>
}
```

사용자에게 입력 받은 insert를 통해 pw값과 id값을 저장합니다.
가입 여부를 결정할 때 PHP-MySQL 트랜잭션을 사용했습니다.

다양한 기술 스택을 활용한 웹사이트 구현

Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원 구성: 2인

진행기간: 2020.09.28~2020.11.23

회원 정보 수정 페이지 구현

가입 후, 회원 정보 수정 페이지를 통해
회원의 개인 정보를 수정할 수 있습니다.

정보를 update/delete 하는 코드는
아래와 같습니다.

```
session_start();  
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");  
$id = $_SESSION['userid'];  
$pw = $_POST['pw'];  
  
$query = "update user set id='$id',pw='$pw'";  
$result = $connect->query($query);
```

update_action.php - UPDATE 기능

```
session_start();  
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");  
$id = $_SESSION['userid'];  
  
$query = "delete from user where id=$id";  
$result = $connect->query($id);
```

signin_action.php - DELETE 기능

다양한 기술 스택을 활용한 웹사이트 구현

Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원 구성: 2인

진행기간: 2020.09.28~2020.11.23

다양한 검색 기능 구현 (일부)

MySQL 쿼리를 PHP 코드에 삽입하여, 사이트 내에 다양한 검색을 구현합니다.
아래는 SQL문을 활용하여 연도별 데이터 개수를 구현한 내용입니다.

```
$connect = mysqli_connect('localhost', 'team21', 'team21', 'team21') or die ("connect fail");  
$query = "SELECT year(release_date) as yeardate, COUNT(release_date) AS cnt FROM movie_info GROUP BY Year(release_date) desc;";  
$result = $connect->query($query);  
$total = mysqli_num_rows($result);
```

index	year	count
11	2016	1
10	2015	2
9	2014	1
8	2013	페이지 내 결과

다양한 기술 스택을 활용한 웹사이트 구현

Python, PHP, MySQL, JavaScript를
활용한 데이터 분석 웹사이트 구현

팀원 구성: 2인

진행기간: 2020.09.28~2020.11.23

다양한 검색 기능 구현 (일부)

MySQL 쿼리를 PHP 코드에 삽입하여, 사이트 내에 다양한 검색을 구현합니다.
아래는 SQL문을 활용하여 인기도별 랭킹을 구현한 내용입니다.

```
<?php
    while($rows = mysqli_fetch_assoc($result)){ //DB에 저장된 데이터 수 (열 기준)
    }
    <tr class="even">
        <td width = "50" align = "center"><?php echo ($all-$total+1)?></td>
        <td width = "500" align = "center">
            <?php echo $rows['movie_name']?></td>
        <td width = "200" align = "center"><?php echo $rows['popularity']?></td>
        </tr>
    <?php
        $total--;
    }
?>
```

rank	title	popularity
1	Batman v Superman: Dawn of Justice	155.790452
2	Avatar	150.437577
3	Pirates of the Caribbean: Dead Man's Chest	145.847379
4	The Avengers	144.448633
5	Pirates of the Caribbean: At World's End	페이지 내 결과

003 데이터 분석과 머신러닝을 위한 스터디

프로젝트 개요 및 주요 활동 요약

평소 데이터 분석과 머신러닝 관련으로 여러 자료를 찾아보며 개인 스터디를 진행했습니다.

- Pandas, Numpy의 기초를 챕터 단위로 정리
- 영문 자료와 공식 사이트를 참고하며 Pycaret 라이브러리 사용법 공부
- 그 외에도 관련 분야의 개인 혹은 팀 스터디를 진행하고 기록

스터디 Github 링크: <https://github.com/solidcellaMoon/studynote>

개인 블로그 링크: <https://star-crab.tistory.com/>

데이터 분석과 머신러닝을 위한 스터디

평소 다양한 자료를 찾아보면서 진행한 데이터 관련 개인 스터디

- Pandas, Numpy 기초 정리
- Pycaret 사용법 공부

Pandas, Numpy 기초 정리

Pandas와 Numpy 기초를 익힐 때 작성한 코드입니다.
언제든지 빠르게 복습, 참고할 수 있도록 정리하고 Github에 올렸습니다.

3. groupby() (★)

기능은 SQL의 group by와 유사하지만 다른 면이 있음. 마찬가지로 분석에 자주 쓰인다.

DataFrame에 groupby()를 호출하면 DataFrameGroupBy라는 또다른 형태의 DataFrame을 반환한다.

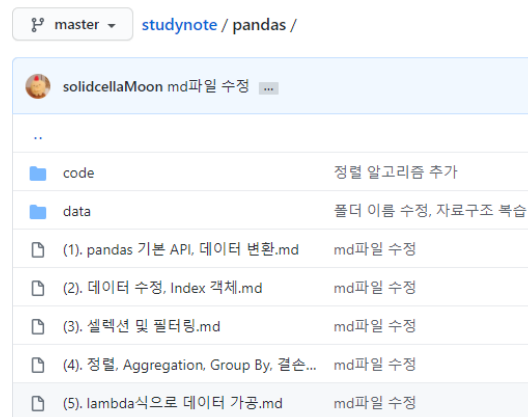
```
groupby_df = titanic_df.groupby(by='Pclass')  
print(type(groupby_df))
```

```
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>
```

3-1. SQL과의 차이점

1. DataFrame.groupby() 결과에 aggregation 함수를 호출하면 groupby() 대상 칼럼을 제외한 모든 칼럼에 해당 aggregation을 적용한다.

```
groupby_df = titanic_df.groupby(by='Pclass').count()  
groupby_df
```



master	studynote / pandas /
solidcellaMoon md파일 수정	
..	
code	정렬 알고리즘 추가
data	폴더 이름 수정, 자료구조 복습
(1). pandas 기본 API, 데이터 변환.md	md파일 수정
(2). 데이터 수정, Index 객체.md	md파일 수정
(3). 선택 및 필터링.md	md파일 수정
(4). 정렬, Aggregation, Group By, 결손...	md파일 수정
(5). lambda식으로 데이터 가공.md	md파일 수정

Pandas 정리 폴더:

<https://github.com/solidcellaMoon/studynote/tree/master/memo/pandas>

Numpy 정리 폴더:

<https://github.com/solidcellaMoon/studynote/tree/master/memo/numpy>

데이터 분석과 머신러닝을 위한 스터디

평소 다양한 자료를 찾아보면서 진행한 데이터 관련 개인 스터디

- Pandas, Numpy 기초 정리
- Pycaret 사용법 공부

Pycaret 사용법 공부

스터디용 자료를 찾아보던 도중 새로운 라이브러리를 알게 되어 영문 블로그와 공식 사이트 자료를 참고하며 사용법을 익혔습니다. 다양한 예측모델의 성능을 한번에 확인할 수 있는게 인상적이었습니다.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.8363	0.8588	0.6795	0.8506	0.7531	0.6333	0.6440	0.6870
gbc	Gradient Boosting Classifier	0.8331	0.8571	0.6882	0.8363	0.7525	0.6287	0.6374	0.0140
lr	Logistic Regression	0.8282	0.8511	0.7274	0.7922	0.7575	0.6251	0.6271	0.4770
ridge	Ridge Classifier	0.8218	0.0000	0.7275	0.7763	0.7505	0.6123	0.6136	0.0050
lda	Linear Discriminant Analysis	0.8186	0.8526	0.7275	0.7696	0.7472	0.6061	0.6074	0.0060
lightgbm	Light Gradient Boosting Machine	0.8123	0.8534	0.7060	0.7698	0.7347	0.5902	0.5932	0.0120
ada	Ada Boost Classifier	0.8105	0.8395	0.7406	0.7471	0.7427	0.5929	0.5940	0.0150
xgboost	Extreme Gradient Boosting	0.8059	0.8551	0.7190	0.7531	0.7329	0.5810	0.5840	0.2280
rf	Random Forest Classifier	0.8042	0.8450	0.7100	0.7568	0.7282	0.5761	0.5806	0.0380
et	Extra Trees Classifier	0.7834	0.8164	0.6926	0.7167	0.7015	0.5321	0.5348	0.0340
dt	Decision Tree Classifier	0.7705	0.7496	0.6837	0.7040	0.6875	0.5072	0.5127	0.0050
knn	K Neighbors Classifier	0.7094	0.7080	0.5286	0.6249	0.5681	0.3539	0.3589	0.2900
svm	SVM - Linear Kernel	0.6872	0.0000	0.4835	0.5824	0.4688	0.2910	0.3257	0.0060
nb	Naive Bayes	0.4014	0.8043	0.9870	0.3814	0.5501	0.0326	0.1043	0.0050
qda	Quadratic Discriminant Analysis	0.3978	0.0000	0.9000	0.3343	0.4875	0.0000	0.0000	0.0110

<catboost, core, CatBoostCl Pycaret으로 타이타닉 생존자 예측 정확도 확인

관련 블로그 글 1: <https://star-crab.tistory.com/13>

관련 블로그 글 2: <https://star-crab.tistory.com/14>

Github 업로드:

https://github.com/solidcellaMoon/studynote/blob/master/%E4%B3%BC%EC%A0%9C/titanic/titanic_analysis.ipynb

감사합니다

문지현, Jihyun Moon