

유호원 YU HOWON

mail : yoohowon@gmail.com
github: <https://github.com/HowardHowonYu>
blog : <https://howardhowonyu.github.io/>

EDUCATION

- **패스트 캠퍼스**
데이터 사이언스 스쿨 2020년 01월 - 2020년 05월
- **Tsinghua Univ. (청화대학교 - 清華大學)**
Journalism and Communication 학사 2009년 09월 - 2015년 07월

PROJECTS

Machine Learning Project

- **회귀분석** - Craigslist(미국의 중고 매물, 부동산, 구인등을 다루는 커뮤니티)의 중고차 매물 데이터를 활용한 가격 예측 프로젝트. 약 51만 건의 데이터. 허위 및 광고성 매물 데이터에 존재하는 이상치 제거를 위해, 미국 정부기관에서 관리하는 차량 이력 데이터베이스를 활용. 허위 매물 제거 및 차량 주행거리, 연식에 대해 신뢰도 높은 데이터 획득. 중고차 가격이 급격히 떨어지는 지점에 대한 가설 검증. 국내 중고차 시장 주행거리 5만km가 기점, 가설을 검증 결과 미국 중고차 시장에서도 약 3만 마일(4만8천km)를 기점으로 가격이 급격히 떨어지는 모습을 확인.
- **교통 표지판 분류** - keras를 이용한 표지판 분류 프로젝트, CNN기법을 활용. 정부기관에서 pdf 형태로 배포하는 표지판 이미지를 가공, keras를 이용해 Image augmentation 진행. 실제 도로에서 획득할수 있는 형태의 이미지로 전환하여 프로젝트 진행.
- **카카오 아레나 대회** - 멜론 플레이리스트 곡 목록 및 태그 예측.

Works

- **Job Hunter 프로젝트** - Slack에서 여러 구인 사이트의 공고를 한번에 볼수있도록 하는 app 제작 프로젝트. 사람인, 잡코리아, 로켓펀치등의 웹사이트를 크롤링, Scrapy Framework을 활용, AWS EC2에 Crontab을 활용하여 주기적인 크롤링 실행. Flask를 이용하여 Slack app의 server를 제작
- **EDA 프로젝트** - Instacart 신선 식품 구매 데이터 탐색 및 분석 프로젝트.

EXPERIENCE

2018년 03월 - 2019년 02월

- **블릿츠 : 영상 콘텐츠, 광고, 의류 제작** - 브랜드 콘텐츠, 오리지널 콘텐츠 제작 총괄, BPH 화장품 “하이드로 컴플렉스 수딩 아쿠아젤” SNS 바이럴 콘텐츠, 뮤직 드라마, 15초 내외의 마이크로 필름 제작 담당, 사업 초기 자금 투자 유치 업무 진행, 사업기획서 및 콘텐츠 제작 Workflow 설계

2016년 09월 - 2017년 11월

- **더저스트폴미스핏츠 : 소셜미디어 JustForMe개발 (공동창업)** - “JustForMe” 앱 중화권 마케팅 담당 : 소셜미디어 콘텐츠 마케팅 플랜 현지화를 위한 번역 및 감수 진행. 중화권 DAU 분기별 약 200% 성장, 중문(간체, 번체) 버전 번역 및 디자인 담당, 번체 사용 지역(대만, 홍콩 및 동남아)과 간체 사용 지역(중국 대륙)을 이원화하여 각 특성에 맞는 기능 설계, 정부 지원 사업 (사무 공간 지원, 마케팅 자금 지원) 커뮤니케이션 담당, Adobe XD, Illustrartor, Sketch등을 활용한 Artwork 제작

SKILLS

- **Programming Language** : Python, HTML, CSS, JavaScript
- **Framework & Packages** : Tensorflow, Scikit Learn, Pandas, Numpy
- **Tools** : Adobe Premier, After effect, Illustrator, Sketch
- **외국어** : 중국어 (상급)

회귀분석

미국 중고차 가격 예측

Notebook : https://nbviewer.jupyter.org/github/HowardHowonYu/usedcar_regression_project/blob/howard/howard/used_car_regression_final.ipynb

Technical Skills : Python, Scikit-learn, Pandas, Numpy

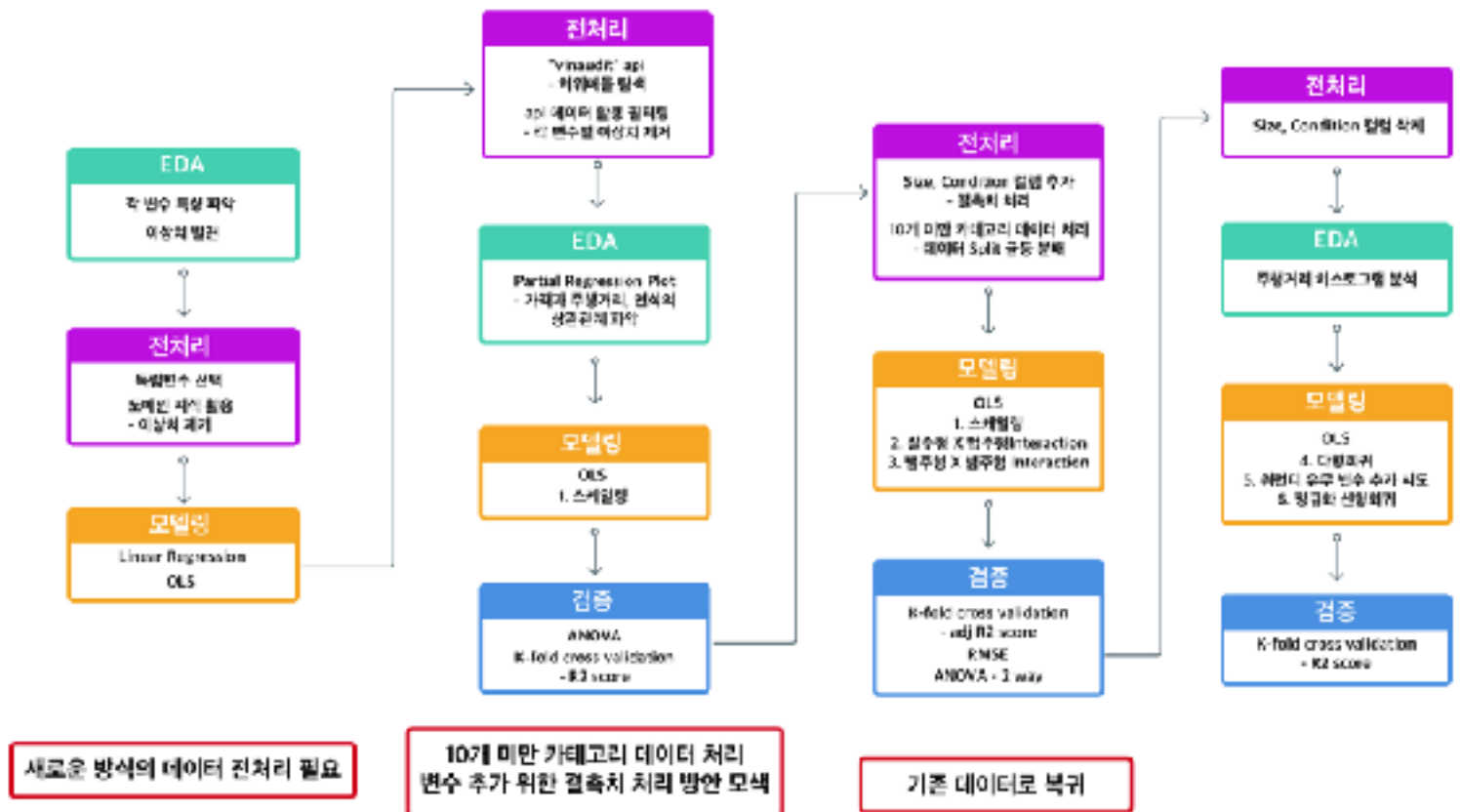
연구 가설

- 주행거리 5만km 이하일때 차량을 파는것이 가격적으로 유리할 것이다.
- 지역별 가격차이가 있을 것이다.

데이터 출처

- Craigslist(<https://craigslist.org>): 중고 매물, 구인 구직, 주택, 자유 주제 토론등을 다루는 커뮤니티 웹사이트
- 미국내 중고차 매물 약 51만 건(업데이트 : 2020년 1월)

Workflow



Issue

- 데이터 특성상 허위 및 광고성 매물로 인한 주행거리, 연식등의 이상치들이 다수 존재.
- 예측 모델 성능 향상을 위한 이상치 탐색 및 제거가 관건

Issue solving

- vin(차대번호)를 활용한 이상치 탐색 작업 진행
- <https://www.vinaudit.com/> 에서 제공하는 api를 이용하여 허위 매물 탐색 및 이력 조회
- 위 api는 미국 정부기관에서 관리하는 데이터베이스를 기반으로 제작, 데이터 신뢰도가 높음

Machine Learning Project

모델링

Model 1 : 종속변수(가격)을 Log 연산, 실수형 독립변수 연식과 주행거리를 정규화 시킨 기본 선형 모델

Model 2 : 실수형 변수와 범주형 변수의 Interaction(상호 작용) formula 추가

연식(year)

- 연도별 인기있는 자동차 제조사가 다르기 때문에, 연식이 가격예측에 미치는 영향이 제조사에 따라 달라진다.
- 금융위기를 기점으로 소형 SUV등의 점유율이 상승 하는 등, 연식이 가격예측에 미치는 영향이 차량 종류에 따라 달라진다.

주행거리(odometer)

- 장거리 혹은 단거리 운행에 적합한 연료 종류가 다르기 때문에, 주행거리가 가격 예측에 미치는 영향이 연료 종류에 따라 달라진다.
- 장거리 운행에 적합한 특정 실린더 종류가 있기 때문에, 주행거리가 가격 예측에 미치는 영향이 실린더 종류에 따라 달라진다.
- 장거리 운행을 하는 특정 차종이 있기 때문에, 주행거리가 가격 예측에 미치는 영향이 차량 종류에 따라 달라진다.

Model 3 : 범주형변수와 범주형 변수의 interaction 추가

- 제조사별로 가격에 미치는 영향이 실린더 갯수에 따라 달라질 수 있기 때문에, 제조사와 실린더의 interaction은 가격 예측에 영향을 준다.
- 차종이 가격에 미치는 영향이 구동방식에 따라 달라질 수 있기 때문에, 제조사와 실린더의 interaction 은 가격 예측에 영향을 준다.

Model 4 : 연식과 주행거리에 다항식추가

- Partial Regression Plot에서 곡선의 그래프를 발견



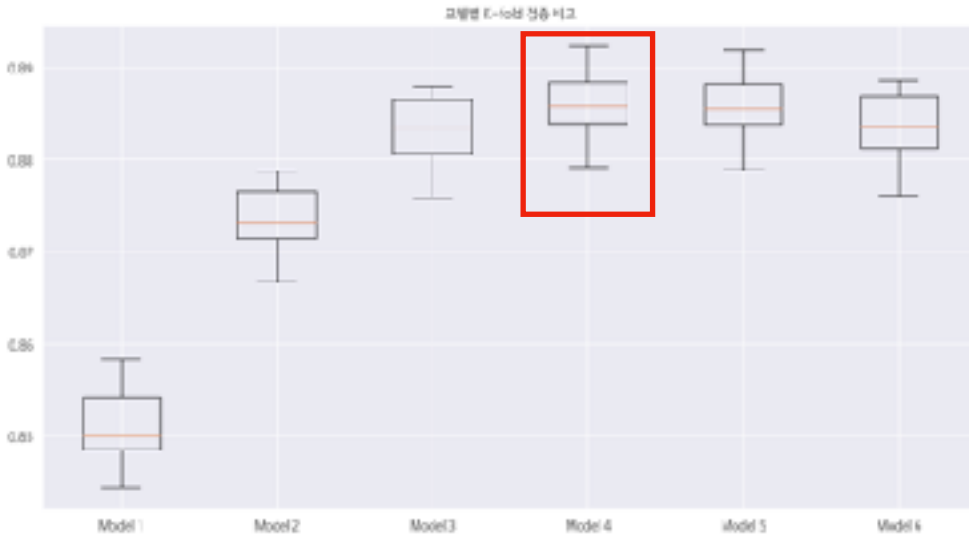
Model 5 : 중고차 보증수리 유무에 관한 데이터를 추가

- 포드, 토요타, 현대 등의 중고차 워런티 서비스 제공 기준 약 6만마일
- 보증수리 유무 카테고리 데이터를 추가 하여 모델링

Model 6 : Ridge, Lasso, Elastic Net을 이용

- 정규화 선형회귀 방법을 이용 선형회귀 계수에 대한 제약 조건을 추가

결과 : Model 4번이 안정적으로 최상의 성능을 보임



연구 가설 검증

가설 1 : 주행거리 5만km 이하일때 차량을 파는것이 가격적으로 유리할 것이다.



- 데이터에서 가장 높은 빈도를 보이는 2012 Ford F-150 FX4모델의 가격이 약 3만마일(4만8천km)부터 급격히 떨어짐

가설 2 : 지역별 가격 차이가 있을 것이다.

	sum_sq	df	F	PR(>F)
C(year)	227.0251634	12.0000000	283.6542885	0.0000000
C(manufacturer)	492.5864538	35.0000000	211.0138517	0.0000000
C(transmission)	24.7197562	2.0000000	185.8148418	0.0000000
C(title_status)	27.6030497	5.0000000	82.7720300	0.0000000
C(state)	108.8557151	50.0000000	32.6420761	0.0000000
C(point_color)	12.6745019	11.0000000	17.6846719	0.0000000
Residual	2874.6001218	41602.0000000	nan	nan

- Anova 독립 검정 결과 State 카테고리 데이터의 F 검정통계량을 확인 할수 있고, 유의확률도 신뢰가능한 수준인것을 확인



- 실제 데이터 상의 워싱턴 주와 코네티컷 주에서 각각 판매되는 2012 Ford F-150 FX4모델 가격의 차이는 약 1만불

한계 및 개선점

- 자동차 보증수리 여부에 대한 명확한 데이터의 부재로, Model 5의 아이디어를 좀더 발전 시키지 못함

Job Hunter 프로젝트

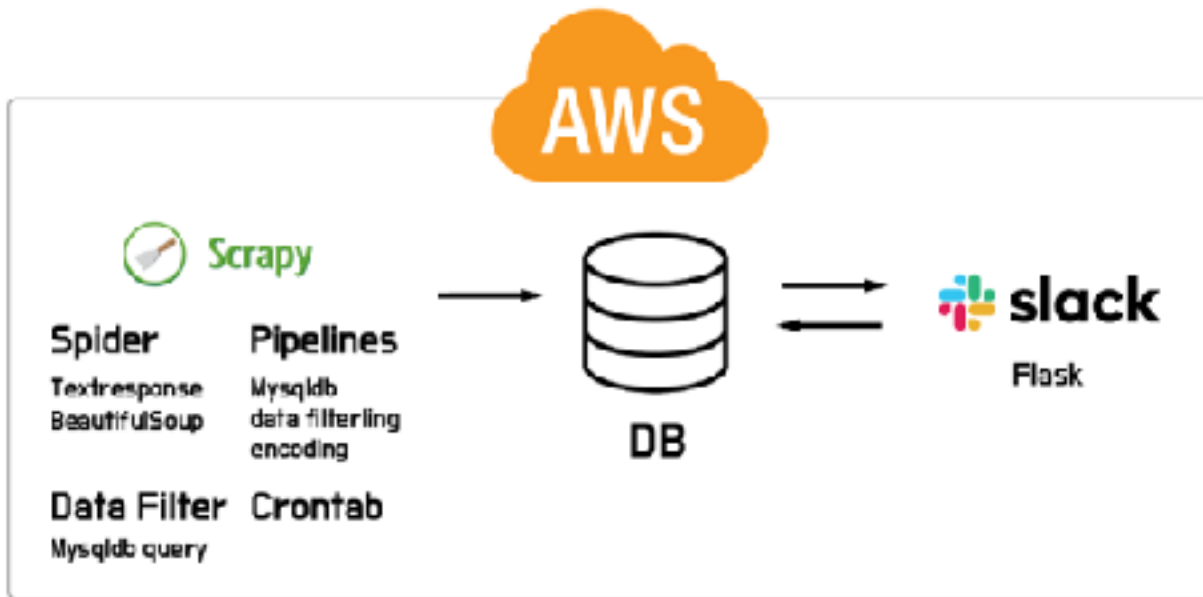
구직 공고 탐색 Slack app 제작

Technical Skills : Python, Scikit-learn, Pandas, Numpy, Scrapy, AWS

Goal

- 매일 업데이트 되는 데이터 관련 구직 정보를 획득할수 있는 Slack app 제작

Structure



1. BeautifulSoup로 HTML로 파싱하여 Css-selector를 활용한 데이터 추출
2. TextResponse로 xpath를 활용
3. 구직 공고 사이트별 두가지 방식의 크롤링 코드를 Scrapy 프레임워크에 적용
4. AWS EC2 에서 crontab을 이용해 주기적으로 크롤링 실행, DB에 데이터 저장
5. Flask를 이용해 Slack app 제작에 필요한 server 구축

Detail

수집 대상

- 사람인, 잡코리아, 로켓펀치등 구직 사이트

수집 주기

- 매일 새벽 2시 크롤링 진행

수집 데이터

- 회사명, 사업분야, 직무, 공고 링크, 연봉 및 조건, 기한, 직무관련 키워드, 회사(사무실 위치)

DB

- SQL Injection을 방지하기 위한 방법 적용 (SQL ALchemy 등 api방식으로 구현)

Slack app

- 현재 AWS EC2에서 tmux로 Session을 나누어 Flask 실행 중

Works

Issue

- 로켓펀치는 HTML 전체 코드를 json안의 String 형태로 반환함
- 잡코리아는 요청 횟수에 따라 크롤링을 차단

Issue solving

```

- 로켓펀치
- 잡코리아

class Company:
    def __init__(self, name, address, phone, email, website, description):
        self.name = name
        self.address = address
        self.phone = phone
        self.email = email
        self.website = website
        self.description = description

    def __str__(self):
        return f'Company: {self.name}, {self.address}, {self.phone}, {self.email}, {self.website}, {self.description}'

class Job:
    def __init__(self, title, company, location, salary, experience, education, description):
        self.title = title
        self.company = company
        self.location = location
        self.salary = salary
        self.experience = experience
        self.education = education
        self.description = description

    def __str__(self):
        return f'Job: {self.title}, {self.company}, {self.location}, {self.salary}, {self.experience}, {self.education}, {self.description}'

class Spider:
    def __init__(self, url, headers):
        self.url = url
        self.headers = headers

    def get_html(self):
        response = requests.get(self.url, headers=self.headers)
        return response.text

    def parse_html(self):
        # HTML 파싱 로직
        pass

    def parse_json(self):
        # JSON 파싱 로직
        pass

    def save_data(self):
        # 데이터 저장 로직
        pass
```

로켓펀치

- Scrapy Spider 생성자 함수 설정시 json의 String을 추출하고, String데이터를 HTML로 파싱하는 과정 설정

잡코리아

- 에러를 만날때마다 AWS EC2 서버를 재시작 하는 방식으로 매번 새로운 ip로 접근하는 방식을 고려했
- 최적화된 Term을 찾는 방식도 진행

결과

- 현재 패스트캠퍼스 데이터사이언스 스쿨 12기 Slack Workspace에서 Job Hunter app 서비스 중



한계 및 개선점

- 잡코리아의 server단에서의 blcok을 회피하는 더 좋은 방안에 대한 구상 필요
- SQL injection을 피하기 위한 SQL Alchemy등 api방식을 활용할 필요

PROJECTS

Machine Learning Project

CNN 이미지 분류

Keras를 활용한 표지판 분류 프로젝트

Technical Skills : Python, Keras, Pandas, Numpy

Notebook : https://nbviewer.jupyter.org/github/HowardHowonYu/traffic-sign-recognition/blob/master/traffic_sign.ipynb

Workflow

Step 1 : “도로교통공단 교통안전표지 일람표”에서 adobe illustrator를 이용 이미지 추출



출처 : https://www.koroad.or.kr/kp_web/safeDataView.do?board_code=DTBBS_030&board_num=100162

Step 2 : Image Augementation 진행

- 각 표지판 이미지를 1000장씩 augmentation 진행
- 영상에서 표지판이 인식되는 이미지의 모양을 고려하여, 이미지 왜곡, 회전, 명도등을 설정



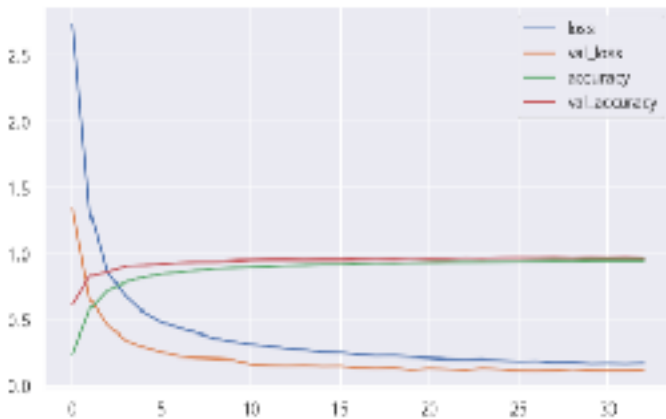
Step 3 : CNN 모델링

- Keras로 CNN 구현
- EarlyStopping을 사용해 val_loss를 모니터링함. 총 26번의 epoch

PROJECTS

Machine Learning Project

Accuracy, Loss



```
model.evaluate(X_test, y_test)
executed in 18.9s, finished 16:14:58 2020-06-03
17600/17600 [=====] - 19s 1ms/step
[0.12164360738888547, 0.9614117612938745]
```

- loss값과 validation loss 값이 과적합 없이 수렴하는 것을 확인
- accuracy 역시 비슷한 양상을 보임
- test 데이터에 대한 accuracy 약 0.962

Step 4 : 결과 분석

Label : 야생동물보호, Predict : 횡단보도



Label : 내리막경사, Predict : 우회



Label : 횡단, Predict : 낙석도로



Label : 자위차통행금지, Predict : 앞지르기금지



- 사람의 눈으로 판단하기 힘든 데이터들은 예측해 내지 못함
- 실제 표지판 사진으로 예측



```
result = model.predict(X)
label_to_str[np.argmax(result, axis=1)[0]]
executed in 7ms, finished 16:08:25 2020-06-07
'작무로이중급온도로'
```

한계 및 개선점

- Image Augmentation의 옵션을 더 조정해 모델 성능 개선이 필요
- 이미지의 수량을 단계적으로 확장시켜 모델 학습
- 모델링(레이어 구성)을 위한 하이퍼 파라미터 조정 필요

Machine Learning Project

카카오 아레나 대회

멜론 플레이리스트 곡 목록 및 태그 예측 - 진행중

Technical Skills : Python, Implict, Pandas, Numpy

Goal

- 플레이리스트의 일부 정보를 가지고, 노래 100곡과 태그 10개를 생성

데이터셋 개요

노래 장르, 곡 별 메타 데이터, Train, Validation, Test 데이터, Mel-spectrogram 데이터(약 280G)

Matrix Factorization + Cosine Similarity

노래 100곡은 협업 필터링 방식

Matrix factorization with Implicit feedback - binary sparse matrix

playlist

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \approx \underbrace{\begin{pmatrix} \\ \\ \\ \\ \\ \end{pmatrix}}_f \left(\begin{pmatrix} \\ \\ \\ \\ \\ \end{pmatrix} \right) \} f$$

songs + tags

implicit의 AlternatingLeastSquares를 활용해 추천시스템 구축

태그 10개는 Cosine Similarity 를 활용

$$magnitude = \sqrt{(x^2 + y^2 + z^2 + \dots)}$$


가중치 계산

$$vector = \left(\frac{x}{magnitude}, \frac{y}{magnitude}, \frac{z}{magnitude}, \dots \right)$$

playlist 벡터 계산

Playlist x Tags 행렬의 값은 0과1이 아닌, 0~1사이의 실수값을 가지게 됨
이후 cosine similarity matrix를 이용해 Tag 예측

결과

용산꽃주먹 (My team)		0.048762	0.007069 (93)	0.284849 (16)	11
Song 예측 순위 93위(93/101) Tag 예측 순위 16위 (16/101) 현재 대회 진행중					