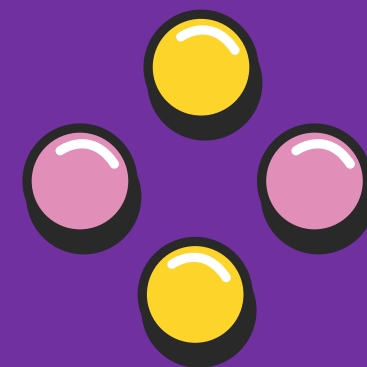
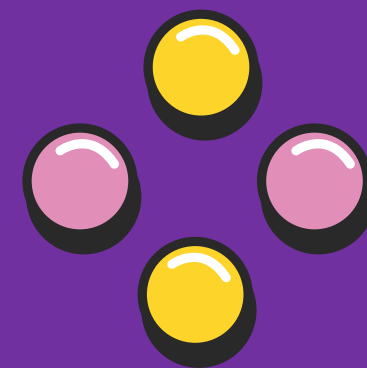
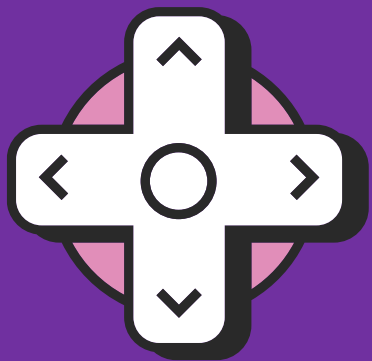


K D T 7





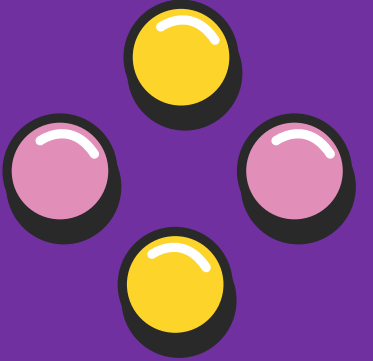
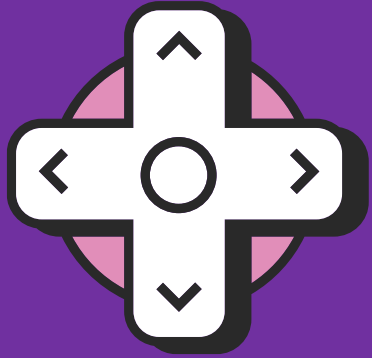
POCKET MONSTER

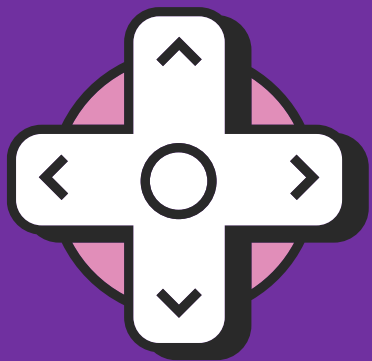
포켓몬 띠부띠부씰 판별모델

사라조

김성익, 신승준, 이정욱, 손병찬

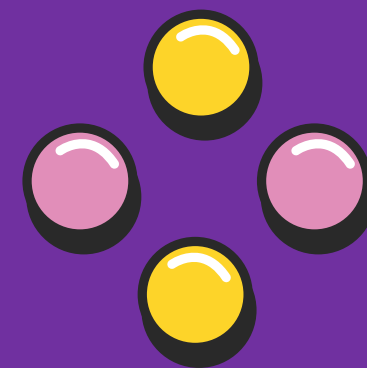


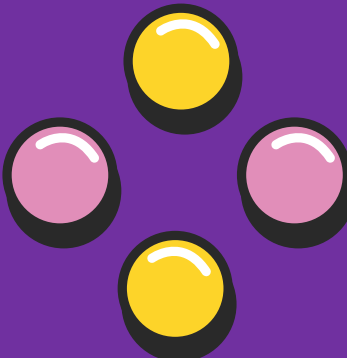
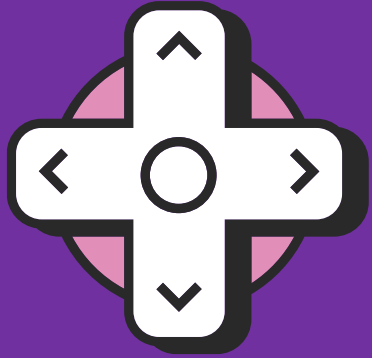




CONTENTS

- 🍷 주제 선정
- 🍷 1세대 - 관동지방
- 🍷 4세대 - 신오지방
- 🍷 2세대 - 성도지방
- 🍷 7세대 - 알로라지방







오박사 : 그래, 발표 준비는 다 되었는가?
그럼 바로 시작하도록 하지



오박사 : 포켓몬은 1996년 일본에서 처음 출시된 비디오 게임 시리즈로,
다양한 생물체를 포획하고 훈련하여 배틀하는 게임으로 시작하여
애니메이션, 카드 게임 등으로 확장된 글로벌 프랜차이즈라네



오박사 : 음, 현재까지 9세대까지 도달했으며,
총 1051종의 포켓몬이 발견되었다고 하네!
정말 놀라운 숫자이지! 포켓몬의 세계는 점점 더 확장되고 있구나!



오박사 : ~~황대조원~~ 포켓몬 도감을 만들려고 하는데…….



사라조 : 우리의 이름은 사라조
자, 그럼 포켓몬 도감을 만들어 볼까?

HP :



1. 주제 선정

1. 주제 선정



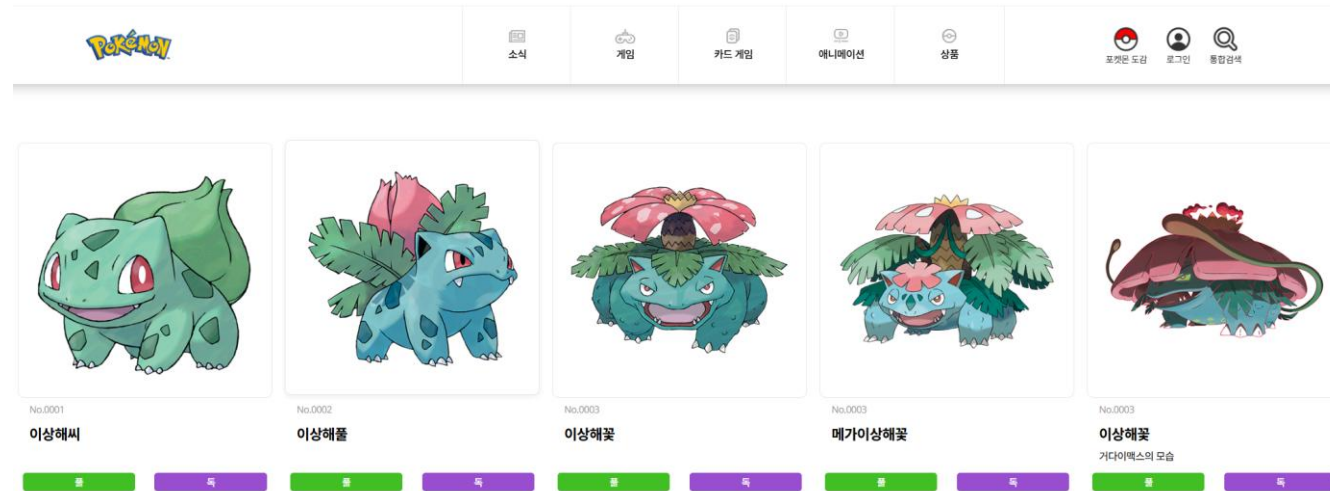
POKÉMON



포켓몬 공식 사이트를 활용하여
이미지를 학습한 후
인터넷 상의 포켓몬을 알아보고자 함



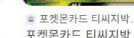
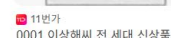
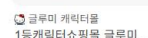
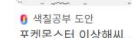
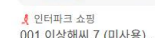
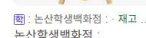
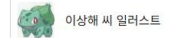
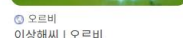
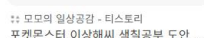
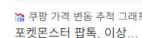
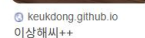
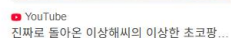
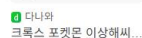
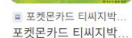
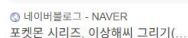
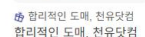
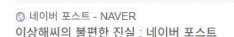
1. 주제 선정



공식 사이트에는 이미 포켓몬 도감이 존재



1. 주제 선정



1. 주제 선정



• Pokemon Wiki - Fandom
이상해씨 (포켓몬) | 포...



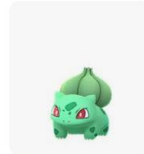
• Fandom
지우의 이상해씨 | 포켓몬 위키 | Fandom



나무위키
이상해씨 - 나무위키



나무위키
이상해씨(한지우) - 나무위키



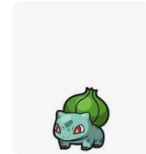
Pokémon GO
이상해씨 (Pokémon) - ...



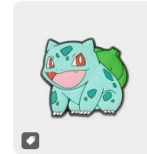
네이버 포스트 - NAVER
이상해씨의 불편한 진실 : 네이버 포스트



합리적인 도매, 천유닷컴
합리적인 도매, 천유닷컴



나무위키
이상해씨 (r105 판) - 나...



SSG
크룩스] 포켓몬 이상해...



나무위키
이상해씨 - 나무위키



네이버블로그 - NAVER
포켓몬 시리즈, 이상해씨 그리기(...



이상해씨
포켓몬카드 티셔지작...
포켓몬카드 티셔지작...



ART street by MediBang
이상해씨. OvvvO - 일...



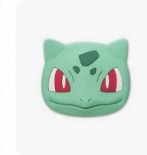
다나와
크룩스 포켓몬 이상해씨...



YouTube
진짜로 돌아온 이상해씨의 이상한 초코팡...



keukdong github.io
이상해씨++



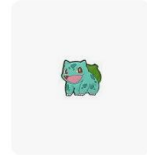
구함 가격 변동 추적 그래프
포켓몬스터 팍팍, 이상...



Naver Blog
이상해씨, 이상해풀, 이...



YouTube
포켓몬스터] 이상해씨 그리기 /How to dr...



이렌드름
포켓몬 이상해씨 지비츠...



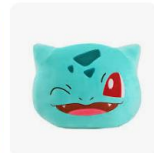
kr.pinterest.com
이상해씨에 관한 인...



모모의 일상공간 - 티스토리
포켓몬스터 이상해씨 색칠공부 도안 ...



오르비
이상해씨 | 오르비



이렌드름
포켓몬스터 말랑말랑 이...



관련 검색어
이상해씨 짤
이상해씨 일러스트
이상해 씨 일러스트



kr.pinterest.com
이상해씨에 관한 인기 아이...



논산학생백화점 : 재고 ...
논산학생백화점 :



인터파크 쇼핑
001 이상해씨 7 (미사용) ...



색칠공부 도안
포켓몬스터 이상해씨...



글루미 캐릭터를
1등캐릭터쇼룸을 글루미 ...



11번가
0001 이상해씨 전 세대 신상품 ...



포켓몬카드 티셔지작...
포켓몬카드 티셔지작...



1. 주제 선정

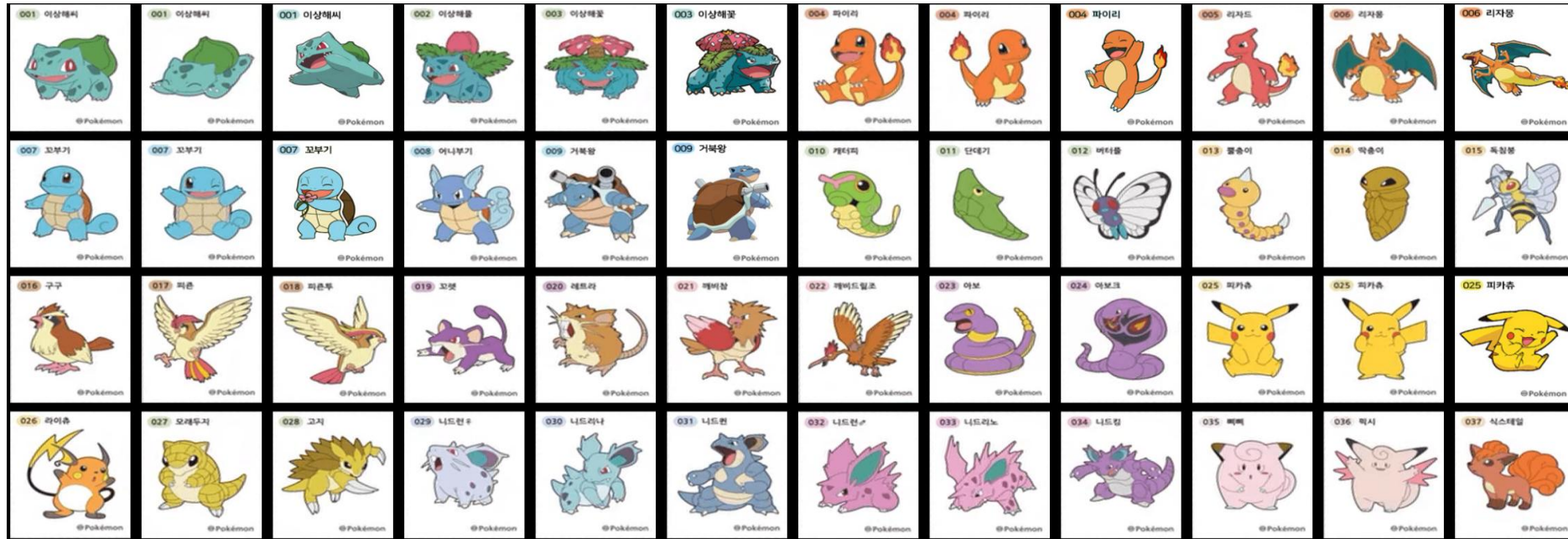


YouTube

진짜로 돌아온 이상해씨의 이상한 초코팡...



1. 주제 선정



출처 : <https://moadocs.github.io/pokedoc/>



띠부띠부실 공식 사이트에서 자료 수집

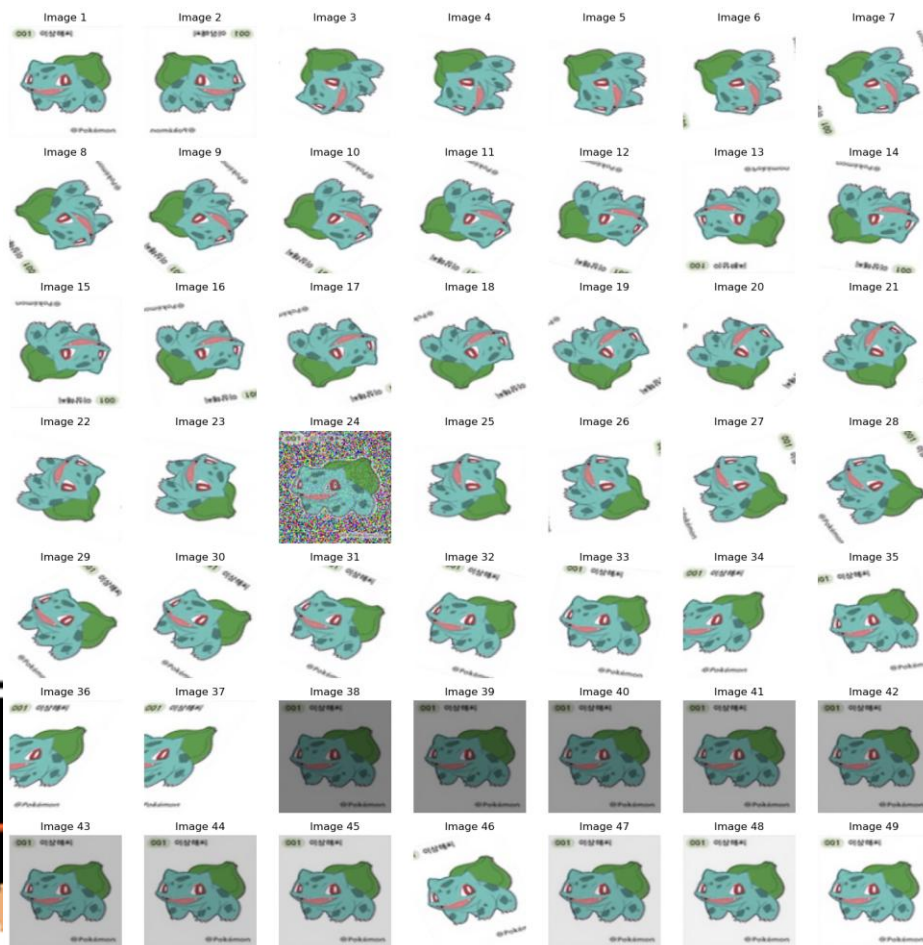


HP :



2. 관동 지방

1. 띄부띄부실 - 모델 전처리



- 좌우 반전
- 상하 반전
- 회전
- 노이즈 처리
- 음영 처리
- 사이즈 : 100X100



268종 캐릭터
16,348장 생성



1. 띄부띄부실 - 데이터 셋 준비



pixel.29998	pixel.29999	target
255	255	101
255	255	101
255	255	101
255	255	101
255	255	101

훈련 셋과 테스트 셋 분리

X_train => 2D (13078, 30000) / X_test => 2D, (3270, 30000)
y_train => 1D (13078,), / y_test => 1D, (3270,)



1. 띠부띠부씰 - RandomForest

rf_model.pkl
- 모델 성능 -

정확도: 0.9498

정밀도: 0.9548

재현율: 0.9498

F1 스코어: 0.9502

오차 (MSE): 3590081.2902

RMS: 1894.7510

띠부띠부씰 분류

파일 선택

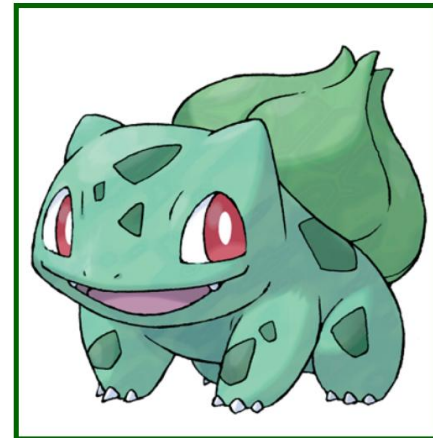
선택된 파일 없음

분석 시작

파일 'noname_000101.png'이(가) 성공적으로 업로드되었습니다.

띠부띠부씰 번호 [001.01.]

예상 도감번호 [[101]]



HP :



3. 포켓몬 분류 모델 - 4세대

2. 개요

포켓몬 4세대 분류 모델 (신오지방)

학습 방법	지도 학습
	분류
사용 알고리즘	랜덤 포레스트



2. 전처리 1



전처리를 흑백으로 했을때

```
[('XGBClassifier', 1.0, 0.688)]
```

	정확도	정밀도	재현율	F1 스코어	오차
Train	1.0	1.0	1.0	1.0	0.343

	정확도	정밀도	재현율	F1 스코어	오차
Test	0.559441	0.59805	0.512216	0.510328	2.903

과대적합 확인



2. 전처리 2



Remove bg 사용 결과



2. 전처리 3



```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 44840 entries, 0 to 44839  
Columns: 14701 entries, 0 to 14700  
dtypes: int64(14701)  
memory usage: 4.9 GB
```



2. 전처리 4



```
# [2-1] 피쳐 타겟 분리
from sklearn.preprocessing import LabelEncoder

featureDF = pokemon_df[pokemon_df.columns[1:]] # 피쳐 : 2차원
targetSR = pokemon_df[pokemon_df.columns[0]]   # 타겟 : 1차원
```

```
# [2-2] 피쳐 전처리 : 데이터 타입 확인, 값의 범위 맞추기

# 데이터 타입 변경 int64 -> uint8
featureDF = featureDF.astype('uint8')
featureDF.values

# 정규화
# 0 ~ 255 -> 0.0 ~ 1.0 정규화 : 효율 증대
featureDF = featureDF/255. # 정규화 과정에서 데이터프레임이나 배열의 모든 값이 동일한 타입으로 처리되도록 보장

featureDF
```



```
# [2-3] 타겟 전처리 : str/object 타입의 경우 수치화 인코딩
lencoder = LabelEncoder()
lencoder.fit(targetSR)
targetSR = lencoder.transform(targetSR)

# 확인
print(lencoder.classes_)
```



2. 학습 결과



	정확도	정밀도	재현율	F1 스코어	오차
Train	1.0	1.0	1.0	1.0	0.272

	정확도	정밀도	재현율	F1 스코어	오차
Test	0.828613	0.835863	0.828613	0.829248	1.713

과대적합

```
# 하이퍼파라미터 dict 생성
param = {'max_depth' : [i for i in range(2,9)], 'criterion' : ['gini', 'entropy', 'log_loss'],
        'min_samples_split': [10, 20], # 분할 최소 샘플 수
        'min_samples_leaf': [5, 10], # 리프 노드 최소 샘플 수
        'class_weight': [None, 'balanced']} # 데이터 불균형 처리
```



2. 예측



```
predict_CSV = './predict.csv'

predict_df = pd.read_csv(predict_CSV, header=None)

featureDF2 = predict_df[predict_df.columns[1:]] # 피쳐 : 2차원
targetSR2 = predict_df[predict_df.columns[0]]   # 타겟 : 1차원

# 데이터 타입 변경 int64 -> uint8
featureDF2 = featureDF2.astype('uint8')
featureDF2.values

# 정규화
# 0 ~ 255 -> 0.0 ~ 1.0 정규화 : 효율 증대
featureDF2 = featureDF2/255. |

new_image = featureDF2

prediction = model.predict(new_image)

print(prediction)
```

0번 : 모부기



```
print(prediction)

✓ 0.2s

[0]
```



HP :



4. 뮤 모델

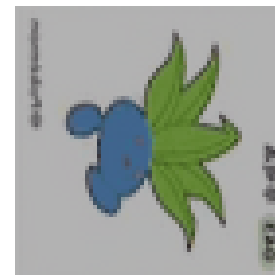
3. 듀 모델

전처리과정

띠부띠부씰 데이터 268장

리사이즈, 돌리고, 반전

약 2500장



3. 듀 모델

전처리과정

띠부띠부씰 데이터 268장

리사이즈, 돌리고, 반전

약 2500장



```
...
accuracy          0.99          650
macro avg         0.99          650
weighted avg      0.99          650
```

Output is truncated. View as a [scrollable element](#) or open in a [text editor](#)

	train_score	test_score	diff	train_loss	test_loss
rf	1.0	0.986154	0.013846	0.0	0.013846



3. 듀 모델

전처리과정

띠부띠부씰 데이터 268장

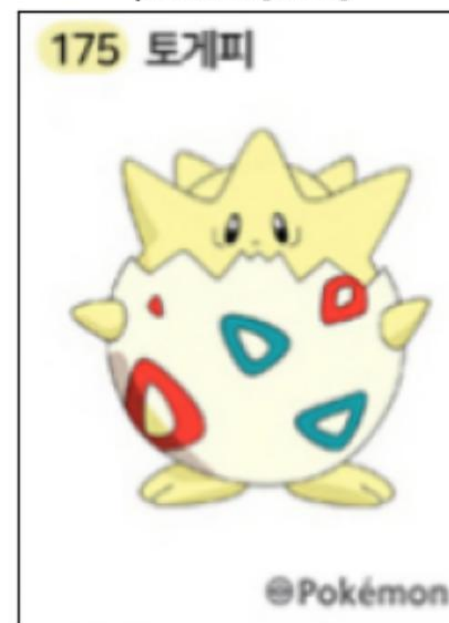
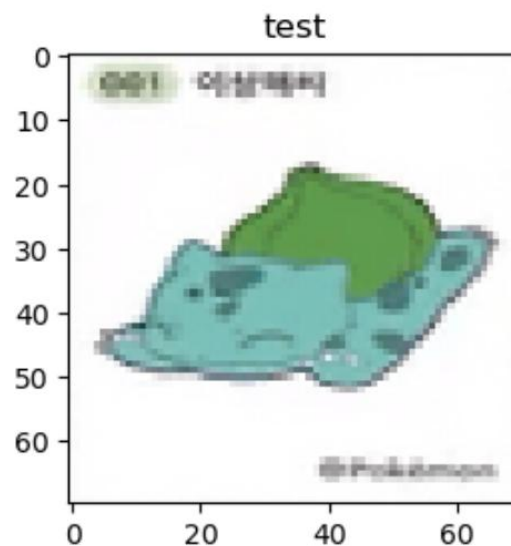
리사이즈, 돌리고, 반전

약 2500장

000102.png



predict [175]



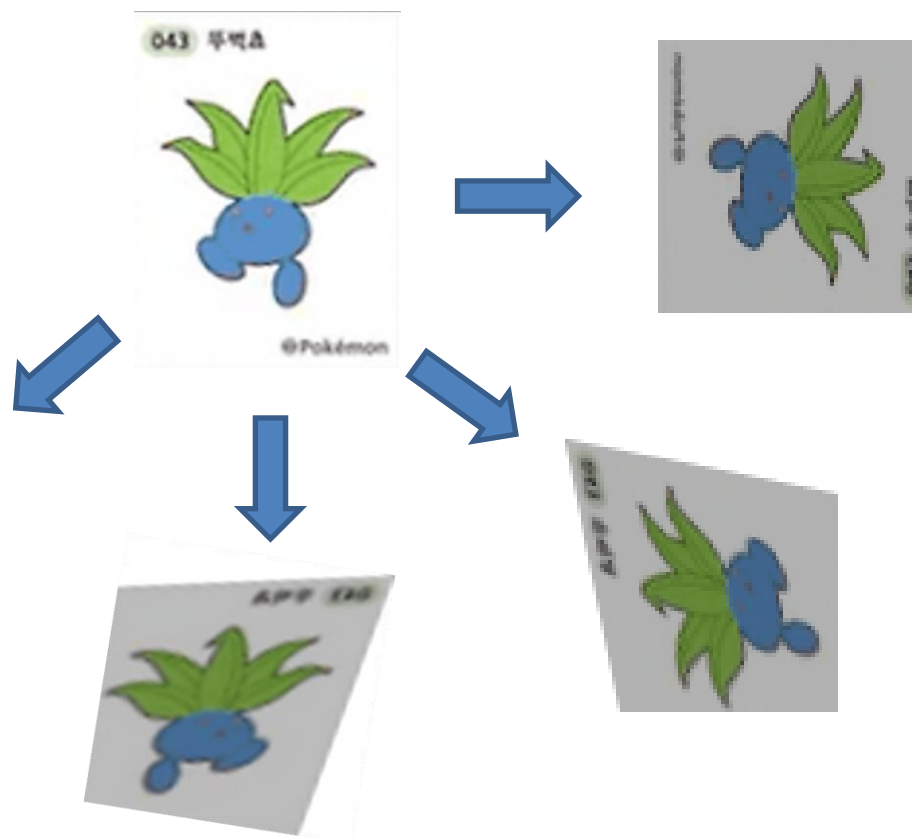
3. 듀 모델

전처리과정

띠부띠부씰 데이터 268장

리사이즈, 돌리고, 반전

약 2500장



3. 듀 모델



전처리과정

띠부띠부씰 데이터 268장

리사이즈, 돌리고, 비틀고, 음영처리, 반전 약 18000장

	train_score	test_score	diff	train_loss	test_loss
rf	0.999767	0.609375	0.390392	0.000233	0.390625



3. 듀 모델



전처리과정

띠부띠부씰 데이터 268장

리사이즈, 돌리고, 비틀고, 음영처리, 반전 약 18000장

	train_score	test_score	diff	train_loss	test_loss
rf	0.999767	0.609375	0.390392	0.000233	0.390625



3. 듀 모델

전처리과정

띠부띠부씰 데이터 268장

리사이즈, 돌리고, 비틀고,

음영처리, 반전

약 18000장

012101.png



test



predict [151]



HP:



5. 알로라 지방 (7세대)

4. 알로라 지방

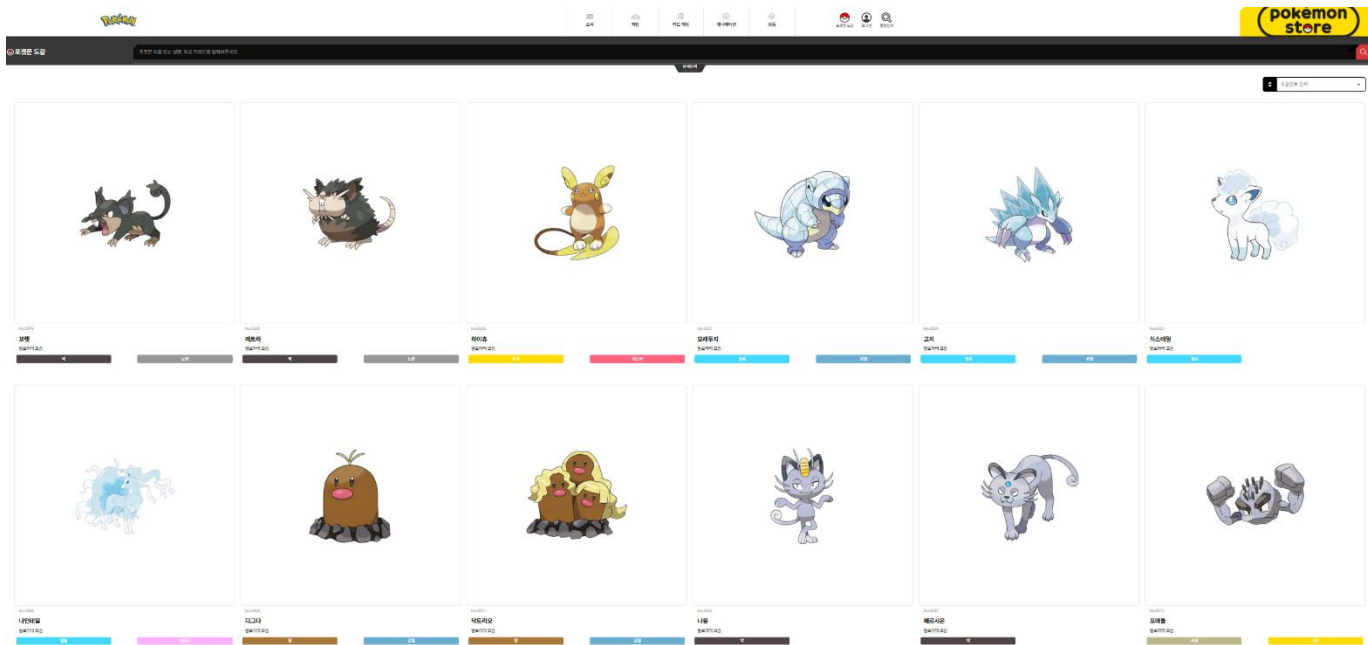


미국의 하와이를 모티브로 제작된
포켓몬 7세대의 메인 무대

포켓몬 최초로
리전폼(Region form)
시스템 도입



4. 알로라 지방



포켓몬 도감 사이트에서
Img downloader로
데이터 셋 수집



4. 모델링 – 최근접 이웃

```
# 최근접 이웃 분석  
k_model = KNeighborsClassifier()  
k_model.fit(X_train,y_train)
```

✓ 0.8s

▼ KNeighborsClassifier ⓘ ?

KNeighborsClassifier()



acc_score: 0.5904030710172745
pre_score: 0.8111441439726497
rc_score: 0.5901205217035691
F1_score: 0.6118012027883518
loss: 0.898039758866473



acc_score: 0.3254029163468918
pre_score: 0.5387312072601929
rc_score: 0.33160743023339445
F1_score: 0.32360059605622116
loss: 0.898039758866473

좋지 못한 성적을 보여줌



4. 모델링 – 서포트 벡터 머신

```
# 서포트 벡터 머신  
s_model = SVC()  
s_model.fit(X_train,y_train)
```

▼ SVC ⓘ ?
SVC ()

```
acc_score: 0.5760076775431862  
pre_score: 0.8348289755279066  
rc_score: 0.5715477460082814  
F1_score: 0.6011597735027565  
loss: 0.898039758866473  
acc_score: 0.3561013046815042  
pre_score: 0.5313146055916742  
rc_score: 0.342448450480095  
F1_score: 0.3419262846610268  
loss: 0.898039758866473
```



좋지 못한 성적을 보여줌



4. 모델링 – 랜덤 포레스트 모델

```
# 랜덤 포레스트
```

```
r_model = RandomForestClassifier()
```

```
r_model.fit(X_train, y_train)
```

✓ 56.7s

▼ RandomForestClassifier ⓘ ?

RandomForestClassifier()

acc_score: 1.0

pre_score: 1.0

rc_score: 1.0

F1_score: 1.0

loss: 0.898039758866473

acc_score: 0.9293937068303914

pre_score: 0.9478981116088326

rc_score: 0.9267384336677815

F1_score: 0.9307043564466877

loss: 0.898039758866473



과대적합 존재하나 가장 고성능



4. 예측 - 랜덤 포레스트 모델



No.0019

꼬렛

실제 학습에 사용된 이미지



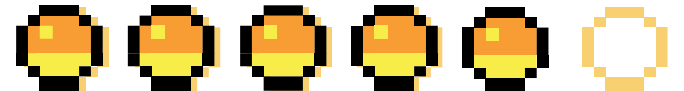
모델에 넣은 이미지



```
new_image = featureDF2  
  
prediction = r_model.predict(new_image)  
  
print(prediction)
```

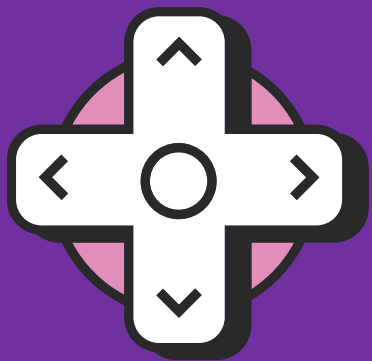
✓ 0.2s

[19]



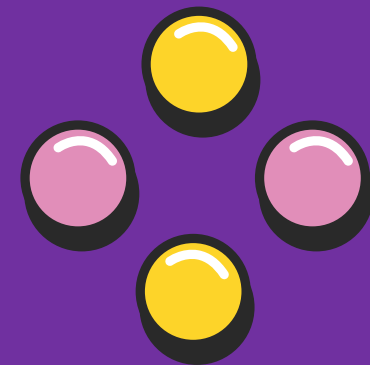


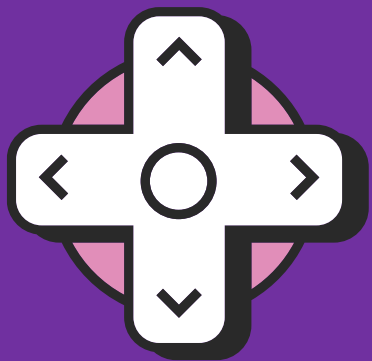
홍길동 : 야호, 모든 발표를 다 끝냈다!
자, 그럼 이제 실전이다



THE END

발표를 경청해주셔서 감사합니다!





Q&A



IF YOU HAVE ANY QUESTION

질문이 있으시면 말씀해주세요!

