

M A C H I N E L E A R N I N G

Team 5

박배서임 최
주수예지 주
미연진이영

CONTENTS

00

Introduction

- 피쳐
- 전처리 방법

01

1차 제출

- 이미지 예시 1
- 1차 제출
- 고난과 역경 1

02

2차 제출

- 이미지 예시 2
- 2차 제출 - 1
- 2차 제출 - 2

03

3차 제출

- 3차 제출 - 1
- 3차 제출 - 2

04

고난과 역경

- 고난과 역경 N

00

Haralick(mean, var, ptp)

Humoments

TAS

Canny Edge

Color Histogram

00

256 * 256

CLAHE

00

코사인 유사도

Cross Validation

01

1차

object 50개, object 당 10장

train object 40개, test object 10개



1.jpg



2.jpg



3.jpg



1.jpg



2.jpg



3.jpg



4.jpg



4.jpg



5.jpg



6.jpg



5.jpg



6.jpg



7.jpg



7.jpg



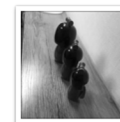
8.jpg



9.jpg



10.jpg



8.jpg



9.jpg



10.jpg

01

사용한 피처

Haralick(mean, var, ptp) /
Humoments

모델

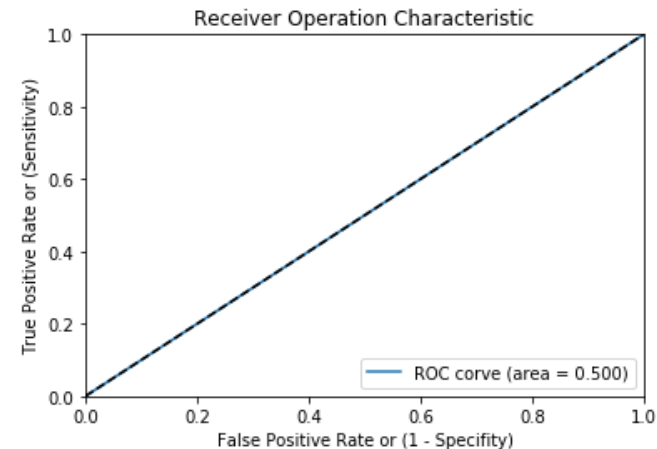
LightGBM

SMOTE, 정규화 했음

자체평가 결과

50.0 %

<matplotlib.legend.Legend at 0x10cadce80>



01

사용한 피처

Haralick(mean, var, ptp) /
Humoments

모델

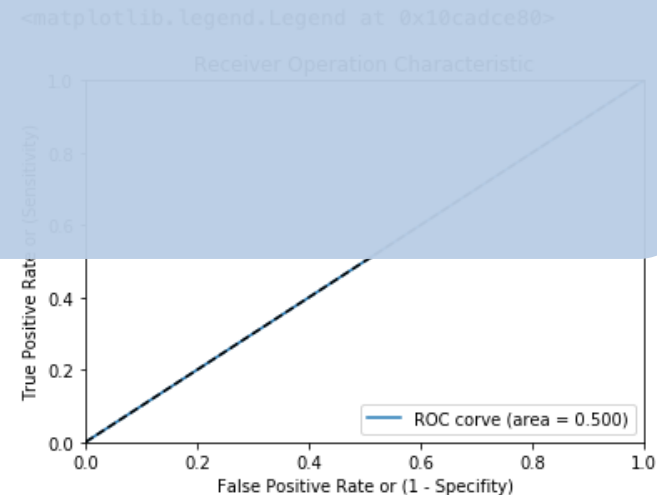
LightGBM

SMOTE 정규화 했음

학습 안됨!!!

자체평가 결과

50.0 %



01

1차 제출에서 사용된 dataset이 적절하지 못해서
학습이 제대로 되지 않았다고 판단
(같은 배경 / 한 object 당 너무 다양한 각도)

->

자연물을 추가한 새로운 object를 구성해서
Dataset을 전부 새롭게 바꿈

02

2차, 3차

object 50개, object 당 10장

train object 40개, test object 10개



1.jpg



2.jpg



3.jpg



1.jpg



2.jpg



3.jpg



4.jpg



4.jpg



5.jpg



6.jpg



5.jpg



6.jpg



7.jpg



7.jpg



8.jpg



9.jpg



10.jpg



8.jpg



9.jpg



10.jpg

02

사용한 피쳐

Humoments(Logged) / Canny Edge

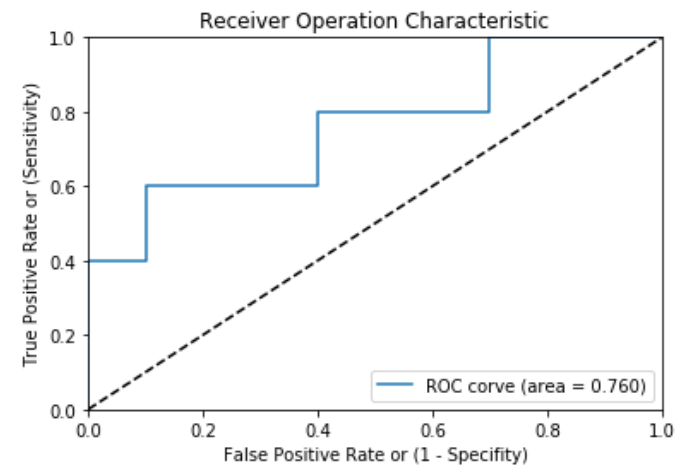
모델

LightGBM

언더샘플링, 정규화 했음

자체평가 결과

76.0 %



02

사용한 피쳐

Humoments(Logged) / TAS / Canny Edge

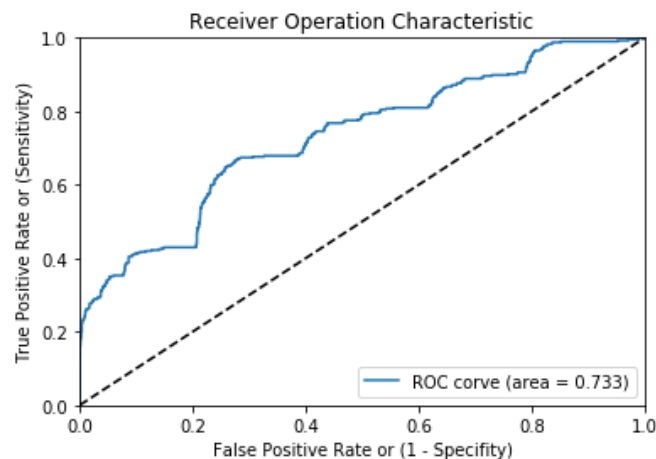
모델

SGDClassifier

오버샘플링, 정규화 했음

자체평가 결과

73.3 %



03

사용한 피처

Humoments / TAS / Canny Edge

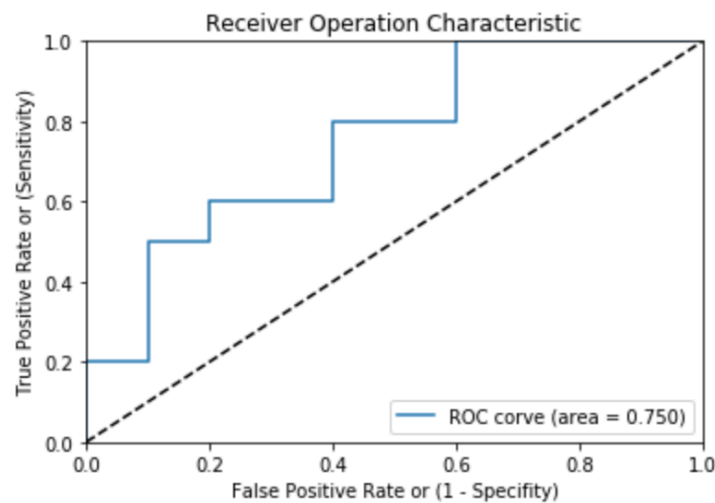
모델

Random Forest

SMOTE, 정규화 했음

자체평가 결과

75.0 %



03

사용한 피쳐

Haralick(mean, var, ptp) / Humoments /
TAS / Color Histogram

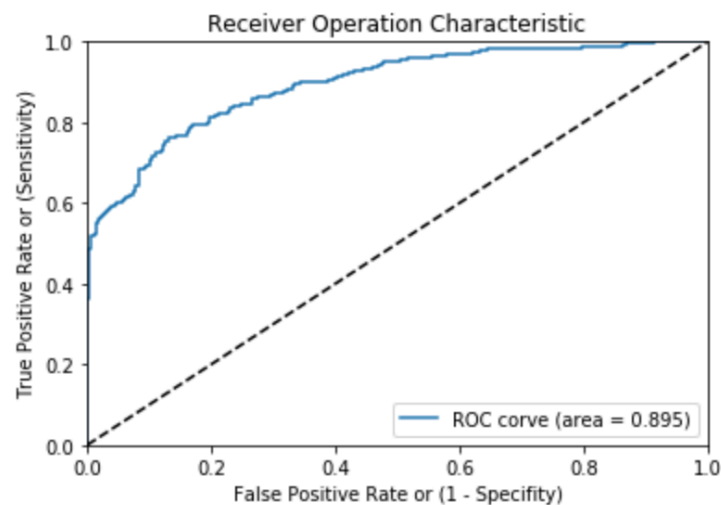
모델

Random Forest

SMOTE, 정규화 했음

자체평가 결과

89.5 %



04

DS.Store 파일이 주기적으로 생성돼서 오류가 발생했고, 매번 없애줘야 하는 것이 번거로움

->

**코드 자체에 .jpg 파일을 제외한 나머지는
걸러낼 수 있도록 추가함**

04

다양한 피처를 사용하려고 열심히 검색했는데,
우리의 코드와 잘 맞지 않았음

(zernike, akaze, SIFT, SURF 등..)
(패키지 설치 안됨, 데이터 형식이 맞지 않음
등의 문제)

04

이미지 전처리 하는 코드가 제대로 적용되지 않음
(함수의 인자에 img파일로 들어가야 하는데
2차까지 이미지의 경로로 넣었음)

->

3차 때는 제대로 전처리해서 적용시킴

```
def img_hist(path) :  
    histogram_arr = []  
    img = cv2.imread(path)  
    hist = cv2.calcHist([img], [0], None, [8], [0, 8])  
  
    for e in hist:  
        e = str(e)  
        e = e.replace('.', '')  
        e = e.replace(']', '')  
        e = e.replace('[', '')  
        e = int(e)  
        histogram_arr.append(e)  
  
    return histogram_arr
```

```
def get_each_feature(path):  
    r_list=[]  
    image = cv2.imread(path,cv2.IMREAD_GRAYSCALE)  
    image1 = cv2.imread(path)  
    humoments = get_hu(path)  
    hu_arr = np.array(humoments)  
    r_list.append(hu_arr)  
    edges = cv2.Canny(image,16,16)  
    canny_mean = edges.mean(axis=0)  
    r_list.append(canny_mean)  
    tas = get_tas(path)  
    tas_arr = np.array(tas)  
    r_list.append(tas_arr)  
    #textures = mt.features.haralick(image1)  
    #ht_mean = textures.mean(axis=0)  
    .....
```

04

**커널이 자꾸 나감....
노트북이 너무 뜨거워짐...
(팬이 상당히 열일 TT)**

**THANK
YOU**