2020
데이터 크리에이터 캠프

주제: 머신러닝을 통한 다중분류

명륜이 없는 명륜팀

최재영 양희윤 전서영 한채은

0. 분석 프로세스

1. 문제 설명 → 2. 데이터 전처리 → 3. 모델 학습 → 4. 발전 방안

1. 문제 설명

1. 문제 설명

머신러닝을 통한 다중분류(Multi-Classification) : 이미지의 픽셀 값을 통해 만두, 새우튀김, 순대로 분류

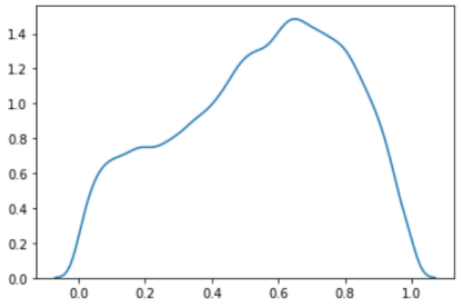
2. 데이터 전처리

2. 데이터 전처리

RGB 특성 탐색

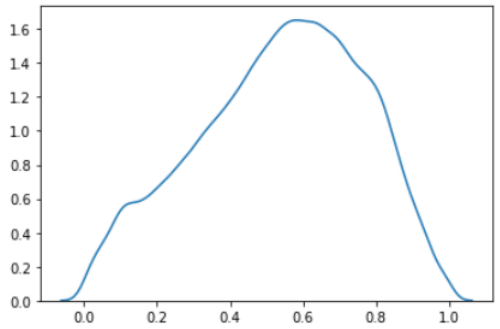
만두 - R

```
In [53]: sns.distplot(mr, hist=False)
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x14501a894a8>
```



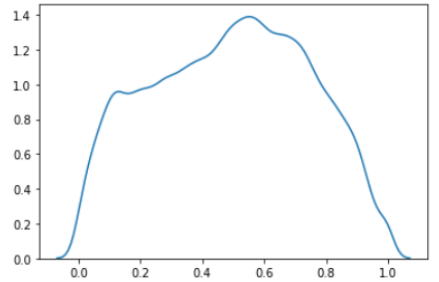
만두 - G

```
In [54]: sns.distplot(mg, hist=False)
Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0x14501af42e8>
```



만두 -B

```
In [55]: sns.distplot(mb, hist=False)
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x1450189c4a8>
```

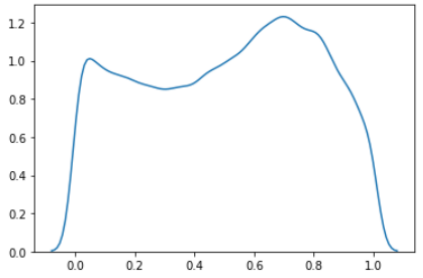


2. 데이터 전처리

RGB 특성 탐색

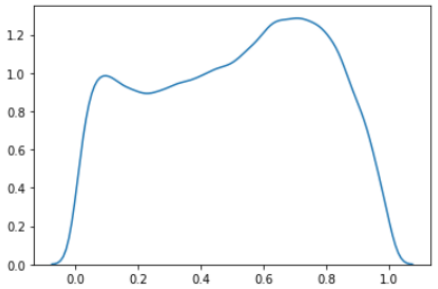
새우튀김 - R

```
In [50]: sns.distplot(shr, hist=False)
Out[50]: <matplotlib.axes._subplots.AxesSubplot at 0x14501954898>
```



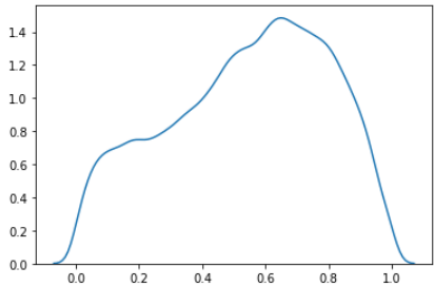
새우튀김 - G

```
In [51]: sns.distplot(shg, hist=False)
Out[51]: <matplotlib.axes._subplots.AxesSubplot at 0x145019bad68>
```



새우튀김 - B

```
In [53]: sns.distplot(mr, hist=False)
Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x14501a894a8>
```

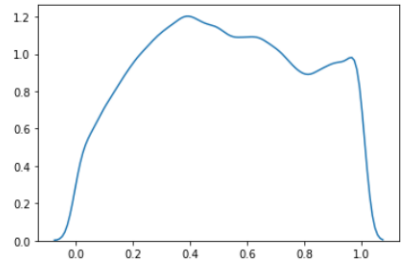


2. 데이터 전처리

RGB 특성 탐색

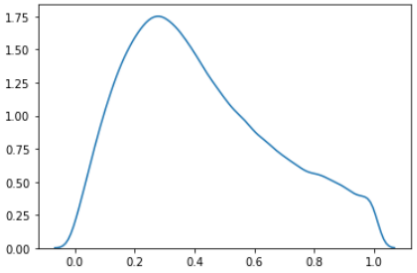
순대 - R

```
In [46]: sns.distplot(sr, hist=False)
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x145000824a8>
```



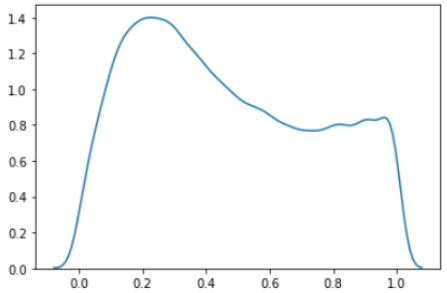
순대 - G

```
In [47]: sns.distplot(sg, hist=False)
Out[47]: <matplotlib.axes._subplots.AxesSubplot at 0x1457c0fd048>
```



순대 - B

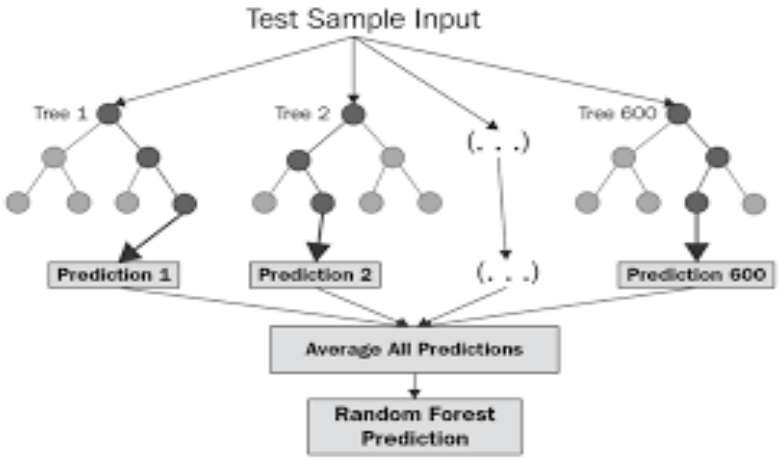
```
In [49]: sns.distplot(sb, hist=False)
Out[49]: <matplotlib.axes._subplots.AxesSubplot at 0x145018bdc50>
```



3. 모델 학습

3. 모델 학습 (1) 모델 소개

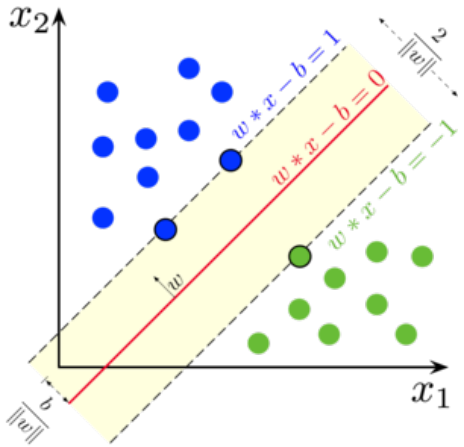
Random Forest



✓ 데이터와 변수를 복원 추출하고 데이터를 하나의 Learner로 정해 학습

✓ K번 반복하여 모델 생성 : 평균치 도출

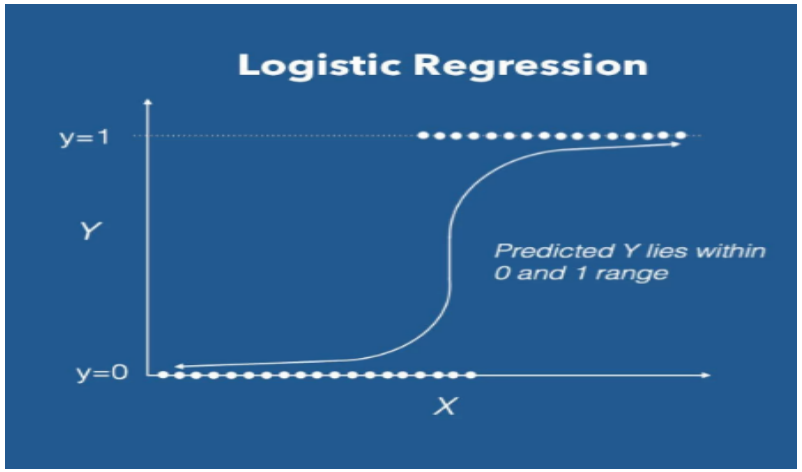
Support Vector Machine



✓ 클래스가 선형 경계에 의해 분리될 수 있는 데이터에 적용 가능

✓ 결정 영역의 초평면을 둘러싸고 있는 margin을 최대화

Logistic Regression

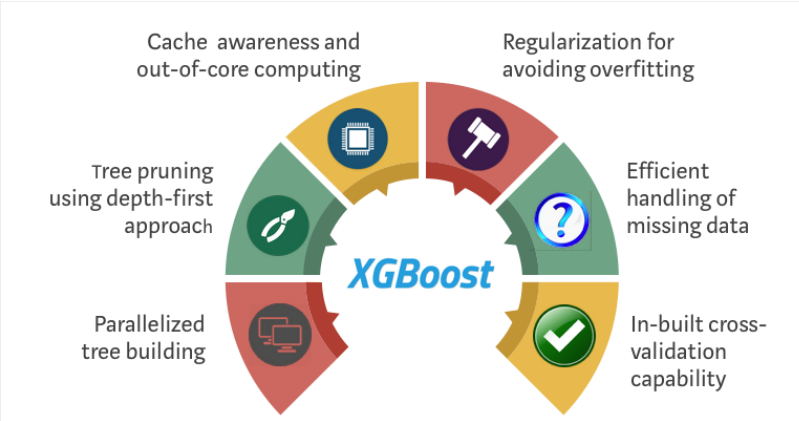


✓ 데이터가 특정 범주에 속할 확률을 예측

✓ 오즈(odds)를 통한 해석 가능

3. 모델 학습 (1) 모델 소개

XGBoost



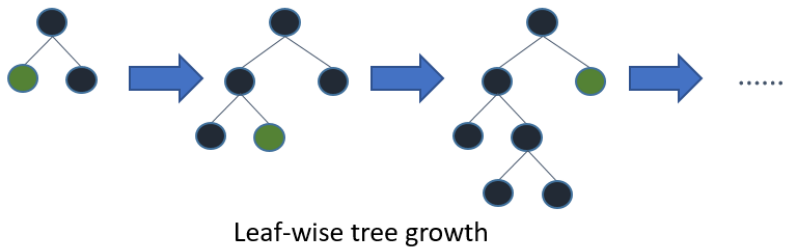
✓ CART(Classification And Regression Tree) 도입
→ 모든 리프들이 모델의 최종 점수에 연관
→ 모델 간의 우위를 비교할 수 있음

✓ 과적합 방지 규제 내장

✓ 조기 종료 제공

✓ 기존의 GB 모델에 비해 빠르고 좋은 성능

LightGBM

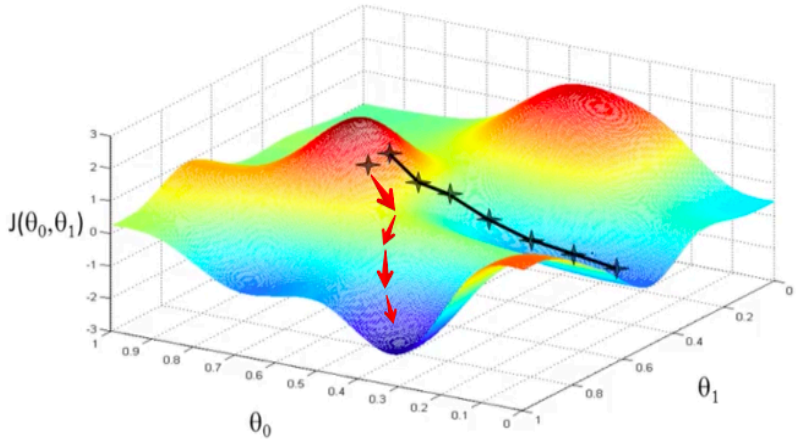


✓ GOSS(Gradient-based One-Side Sampling)기법과 EFB(Exclusive Feature Bundling) 기법 도입

✓ Leaf-wise 방식으로 트리를 성장
→ 분할 기준을 손실 변화로 둠

✓ 기존 GBDT 모델들에 비해 빠르고 효율

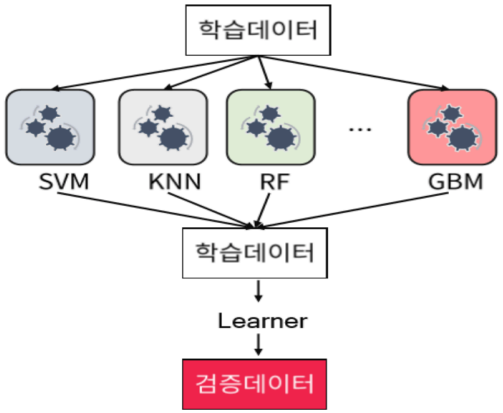
Gradient Boosting



✓ 경사하강법을 이용해 가중치 업데이트

3. 모델 학습 (1) 모델 소개

Stacking



✓ 모델을 결합하여 사용

✓ 여러 모델들을 학습데이터에 학습시킨 후, 각 모델에서의 prediction 값을 독립 변수로 활용하여 단일 모델에 최종 학습

3. 모델 학습 (2) 훈련 프로세스

모델 훈련 방식

Stratified 5-fold로 훈련 진행

Test	Train	Train	Train	Train
Train	Test	Train	Train	Train
Train	Train	Test	Train	Train
Train	Train	Train	Test	Train
Train	Train	Train	Train	Test

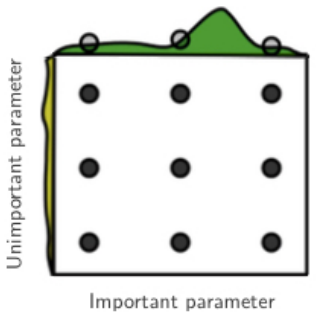
*stratified k fold: y value의 distribution을 지키도록 fold 구성.

각 모델의 훈련 시 5 fold을 진행

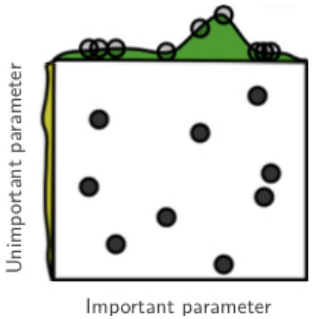
하이퍼파라미터 최적화 방식

세 방식을 혼용하여 최적화 진행

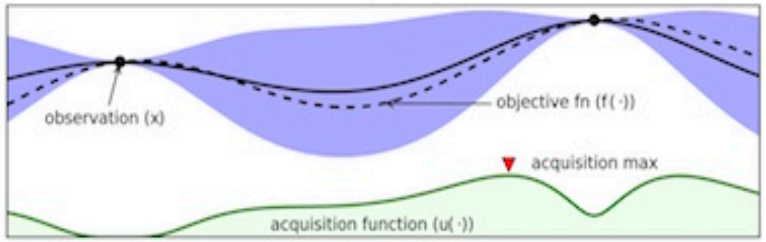
(1) Grid Search



(2) Random Search



(3) Bayesian Optimization



각 모델의 하이퍼파라미터를 최적화

모델 선정 방식

각 모델의 Validation set f1-score로 최종 모델 선정

XGBoost

CatBoost

LightGBM

NEURAL NETWORK

좋은 score를 가진 모델 3개 선정하여 Stacking 진행

3. 모델 학습 (3) 모델 선정

1차 알고리즘 선정

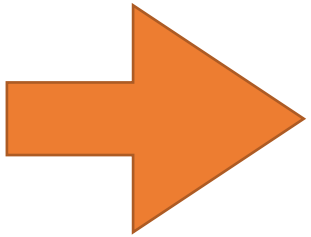
Support Vector	Random Forest	Gradient Boost	Logistic Regression	XgBoost	LightGBM
0.882589	0.872267	0.896747	0.844234	0.887695	0.895205

Support Vector	Random Forest	Gradient Boost	Logistic Regression	XgBoost	LightGBM
0.882589	0.872267	0.896747	0.844234	0.887695	0.895205

3. 모델 학습 (4) 성능 향상

voting : 0.78

```
models = [('xgb',clf1),
          ('lgb',clf2),
          ('svc',clf3),
          ('lr',lr)]
```



	precision	recall	f1-score	support
3	0.72	0.72	0.72	60
4	0.80	0.80	0.80	60
5	0.83	0.83	0.83	60
accuracy			0.78	180
macro avg	0.78	0.78	0.78	180
weighted avg	0.78	0.78	0.78	180

3. 모델 학습 (4) 성능 향상

LGBM : 0.78

	precision	recall	f1-score	support
0	0.74	0.70	0.72	60
1	0.80	0.80	0.80	60
2	0.81	0.85	0.83	60
accuracy			0.78	180
macro avg	0.78	0.78	0.78	180
weighted avg	0.78	0.78	0.78	180

XGBoost : 0.77

	precision	recall	f1-score	support
0	0.71	0.68	0.69	60
1	0.81	0.73	0.77	60
2	0.78	0.88	0.83	60
accuracy			0.77	180
macro avg	0.77	0.77	0.76	180
weighted avg	0.77	0.77	0.76	180

3. 모델 학습 (4) 성능 향상

CV_ STACKING : 진행중..

3. 모델 학습 (4) 성능 향상

LGBM : 0.78

voting : 0.78

XGBoost : 0.77

4. 발전 방안

4. 발전 방안

아쉬운 점 및 느낀 점

1. 이미지 데이터를 다뤄본 적이 없어서 당황 했으나 무사히 마칠 수 있었다.
2. 이미지 데이터와 이를 활용한 알고리즘에 대한 지식이 많았다면 더욱 성능이 좋은 모델을 만들 수 있었을 것 같다.
3. 현장에서 즉석에서 코딩을 하고 결과물을 제출하는 것이 익숙하지 않아 시간이 부족하다는 압박이 컸던 것 같다.

발전 방안

1. 시간이 많았다면 성능이 더욱 좋은 모델을 만들 수 있었을 것 같다.

QUESTIONS?

THANK YOU !

명륜이 없는 명륜팀

최재영 양희윤 전서영 한채은