기초 이론부터 실무 실습까지

머신 러닝 익히기

programmers

Part 05. SVM과 Decision Tree

정 정 민

programmers

Chapter 14. SVM & DT 분류 실습

1. 비행 경험 만족도 데이터
2. SVM을 활용한 풀이
3. Decision Tree를 활용한 풀이

비행 경험 만족도 데이터

강의 실습 코드 : 링크

비행 경험 만족도 데이터 (Airlines Customer satisfaction)

- 이전 선형 분류 실습에서 사용했던 데이터로
- 같은 데이터에 대한 다른 풀이 방법을 비교!

- 이번 실습에서 사용할 데이터로 Kaggle의 공개 데이터 (링크)
  - 혹시 지우셨다면 다시 다운로드 받아주세요!

- 항공사 서비스에 대한 고객 만족도 관련 데이터
- 만족도를 포함한 탑승객의 개인 및 여행 경험 정보 총 23개 특성을 포함

전처리

- 선형 분류 과정에서 진행했던 전처리 과정을 그대로 사용
- 아래의 과정을 포함
  - NA 값 제거
  - 지연 시간 5시간 이상 제거
  - 범주형 데이터 인코딩
  - 상관도를 바탕으로 15개 특성 추출
  - 20% 학습 및 평가 데이터 분할

```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103319 entries, 9997 to 122649
Data columns (total 15 columns):
 #   Column                              Non-Null Count   Dtype
---  ------                              --------------   -----
 0   Inflight entertainment              103319 non-null  int64
 1   Ease of Online booking              103319 non-null  int64
 2   Online support                      103319 non-null  int64
 3   On-board service                    103319 non-null  int64
 4   Online boarding                     103319 non-null  int64
 5   Leg room service                    103319 non-null  int64
 6   Checkin service                     103319 non-null  int64
 7   Baggage handling                    103319 non-null  int64
 8   Cleanliness                         103319 non-null  int64
 9   Seat comfort                        103319 non-null  int64
 10  Inflight wifi service               103319 non-null  int64
 11  Food and drink                      103319 non-null  int64
 12  Customer Type_disloyal Customer     103319 non-null  uint8
 13  Class_Eco                           103319 non-null  uint8
 14  Gender_Male                         103319 non-null  uint8
dtypes: int64(12), uint8(3)
memory usage: 10.5 MB
```

SVM을 활용한 풀이

강의 실습 코드 : 링크

SVM 학습 진행

- 이론 시간에 좋은 성능을 보였던 RBF 커널을 활용해 학습
- SVM 학습의 학습 시간은 선형 모델에 비해 오래 걸림

$$O(n_{sample}^2 \times n_{feat}) \sim O(n_{sample}^3 \times n_{feat})$$

- 비행 만족도 데이터를 기준으로 약 30분 정도 소요
- %%timeit
  - Jupyter Notebook에서 사용 가능
  - 해당 셀을 실행하는데 걸린 시간 측정
  - 셀 실행에 걸린 평균적인 시간과 sd 값을 표시

```
from sklearn.svm import SVC

svm = SVC(kernel='rbf', C=0.1)
%%timeit
svm.fit(X_train, y_train)

# 4min 38s ± 21.2 s per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

SVM 결과 확인

- Logistic Regression을 활용한 정확도 (Accuracy)
  - 학습 데이터 : 82.8 %
  - 평가 데이터 : 82.9 %

- SVM을 활용한 정확도
  - 학습 데이터 : 90.7 %
  - 평가 데이터 : 90.4 %

새로운 분류 평가 척도 : 정밀도(Precision), 재현율 (Recall), F1 점수

- **정밀도 (precision)**
  - 예측한 양성 결과가 실제로 얼마나 진짜 양성인지를 계산
  - 모델이 양성 결과를 잘 찾아내야 하는 상황에서 중요

- **재현율 (recall)**
  - 실제 양성 중 얼마나 양성을 잘 찾아냈는지를 계산
  - 정답을 잘 찾아내는 과정에서 중요

- **F1 점수**
  - 정밀도와 재현율의 조화 평균
  - 조화 평균을 사용해 낮음 점수에 대한 패널티를 늘림
  - 정밀도와 재현율이 전반적으로 좋아야 좋은 F1값을 갖을 수 있음

```
from sklearn.metrics import precision_score, recall_score, f1_score

precision = precision_score(y_test, y_test_pred)
recall = recall_score(y_test, y_test_pred)
f1 = f1_score(y_test, y_test_pred)
```

Decision Tree 을 활용한 풀이

강의 실습 코드 : 링크

Decision Tree 학습 진행

- Entropy 결정 경계를 사용하는 최대 깊이 5의 Tree를 생성

- Decision Tree의 학습 시간은 SVM에 비해 짧음
- Tree를 구성하는 깊이에 따라 변동성이 크지만 일반적으로

$$O\left(n_{sample} \times \log\left(n_{sample}\right) \times n_{feat}\right)$$

정도로 볼 수 있음

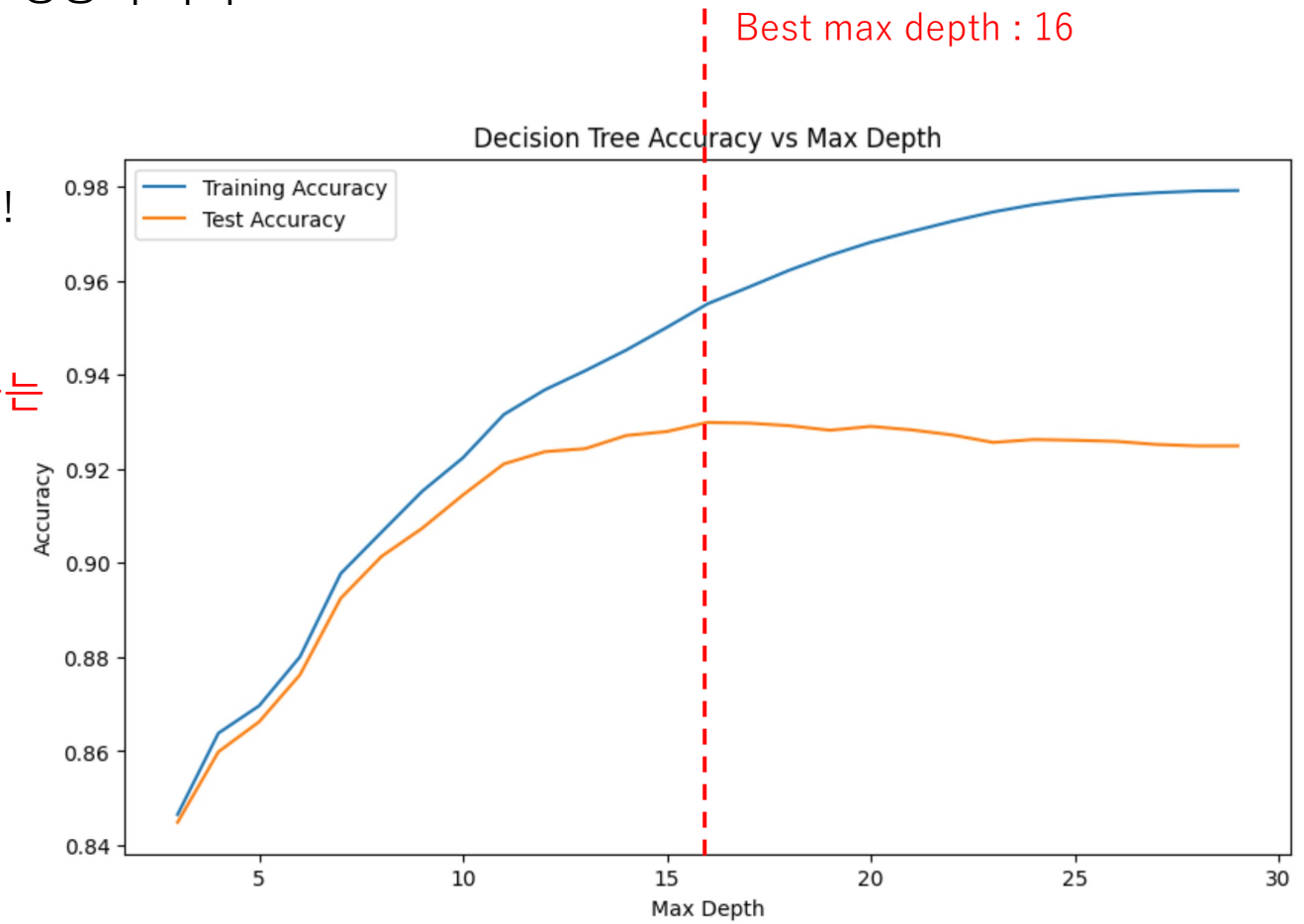
```
from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier(criterion='entropy',
                            max_depth=5,
                            min_samples_split=5)
%%timeit
dt.fit(X_train, y_train)

# 247 ms ± 10.2 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

최고의 모델을 찾아서!

- 머신러닝 모델의 크기가 커지고 복잡도가 증가하면 모델의 성능은 올라감
- 하지만 과적합(Overfitting) 현상이 발생하면 오히려 성능이 하락
  - 학습 데이터에 대한 성능은 지속적으로 상승
  - 평가 데이터에 대한 성능이 하락
  - 학습 데이터를 단순히 암기하는 과정으로 돌입!
- 따라서 평가 데이터에 대한 성능이 낮아지기 시작하는 지점의 세팅을 이용해 최적의 모델을 선택해야 함
  - 옆 그림은 하이퍼파라메터 중 하나인 max depth 값을 이용한 서칭 그래프



Decision Tree 결과 확인

- Logistic Regression을 활용한 정확도 (Accuracy)
  - 학습 데이터 : 82.8 %
  - 평가 데이터 : 82.9 %

- SVM을 활용한 정확도
  - 학습 데이터 : 90.7 %
  - 평가 데이터 : 90.4 %

- DT를 활용한 정확도
  - 학습 데이터 : 95.5 %
  - 평가 데이터 : 93.0 %

[숙제] 더 해보기!

- 수업에서 사용한 학습 및 평가 코드를 활용해 다양한 세팅의 학습을 해보세요

- 데이터
  - 학습 및 평가 데이터 분류 변경
  - 추출한 데이터 특성 변경

- SVM
  - 커널 변경
  - C 상수 값 변경

- Decision Tree
  - 지니 불순도 (Gini Impurity)를 사용
  - min_samples_split 값 변경

- 각 경우를 비교해 최고의 모델은 어떤 상태일 때 가장 좋은 모델의 성능이 나왔나요??

E.O.D