

Voting Classifier

2021.11.28 김은희

Ensemble

: 약한 모델의 연결 → 좋은 모델 만들기

Bagging

: 데이터의 복원 추출
→ 각각의 모델 만들기
→ 하나의 모델로 합치기

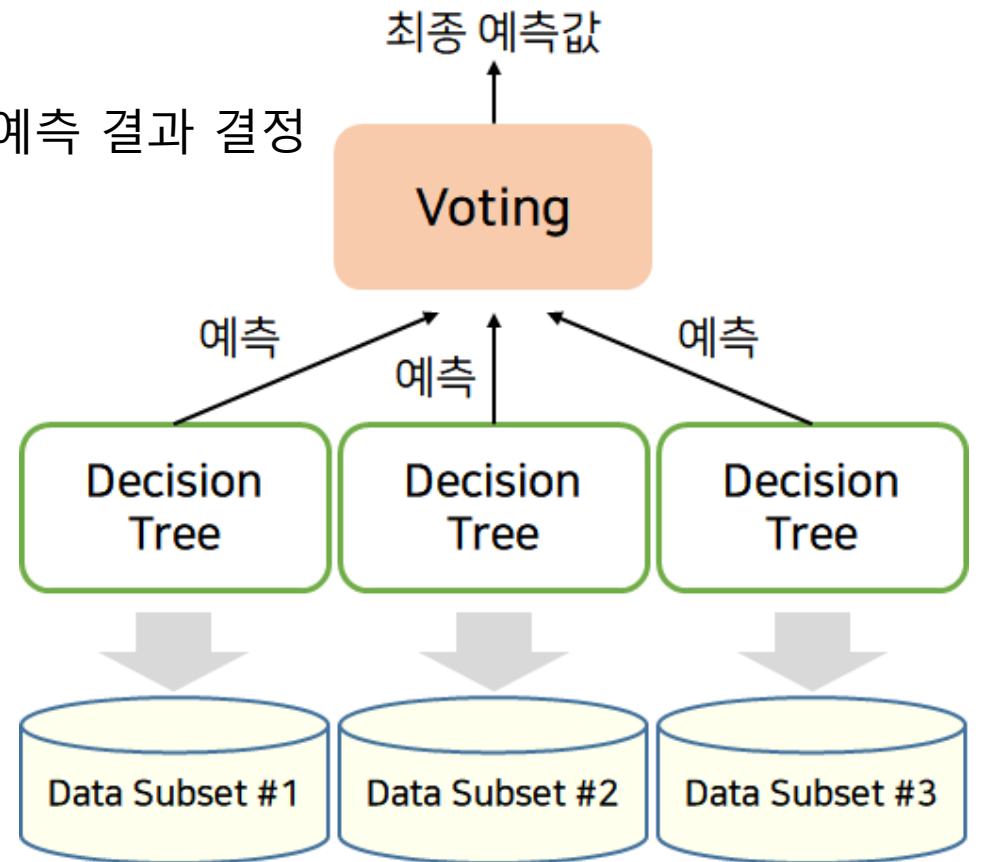
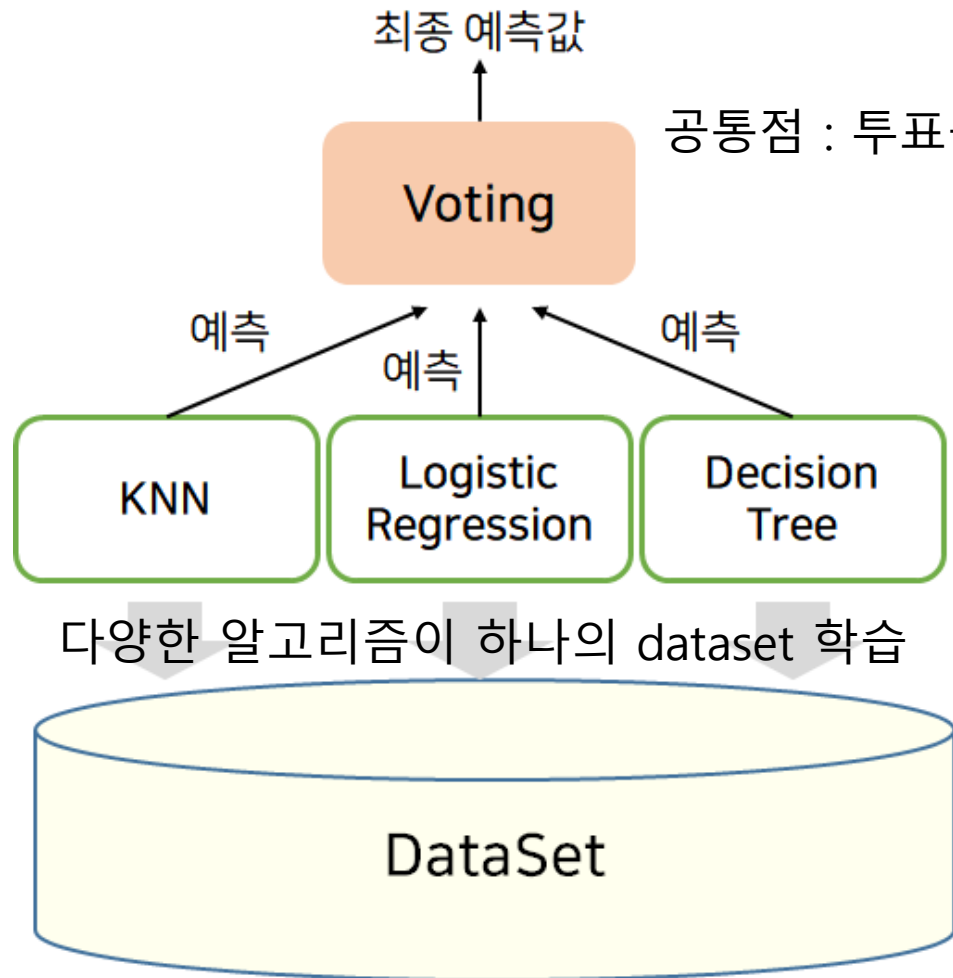
Boosting

: 맞추지 못한 케이스에 높은 가중치

Voting

: 모델 간 투표 → 예측 결과 선정

Voting과 Bagging의 비교



단일 알고리즘이 서로 다른 dataset 학습

Voting Classifier의 두 흐름

- **Hard Voting (majority voting)**

- 각 분류기로부터 예측될 확률이 가장 높은 class를 선정
- Classifier 1, 2, 3이 각각 class A, A, B를 예측 → hard voting → A

- **Soft Voting (weighted voting)**

- 분류기들이 예측한 값의 확률 평균이 가장 높은 class를 선정
- Hard voting보다 합리적이며, 예측 성능이 좋음

	Classifier 1	Classifier 2	Classifier 3	평균
Class A	0.30	0.47	0.53	0.4333
Class B	0.20	0.32	0.40	0.3067

Voting Classifier 사용 방법

[sklearn.ensemble.VotingClassifier](#) ¶

```
class sklearn.ensemble.VotingClassifier(estimators, *, voting='hard',  
weights=None, n_jobs=None, flatten_transform=True, verbose=False) . \[source\]
```

```
logistic_regression = LogisticRegression()
```

```
knn = KNeighborsClassifier(n_neighbors=neighbors)
```

```
voting_model = VotingClassifier(estimators=[ ('LogisticRegression', logistic_regression), ('KNN', knn) ], voting='soft')
```

Voting Classifier 사용 방법

Fit it to the training set

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

Predict

```
y_pred = voting_model.predict(X_test)
```

Evaluate the performance with accuracy score

```
acc = accuracy_score(y_test, y_pred)
```

```
print("Accuracy score : %.3f" %acc )
```