

## [ML/DL] 앙상블 학습 (Ensemble Learning): 2. Voting(보팅)이란? — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2021-01-08 오후 10:01

URL: <https://continuous-development.tistory.com/179?category=736685>

ML,DL

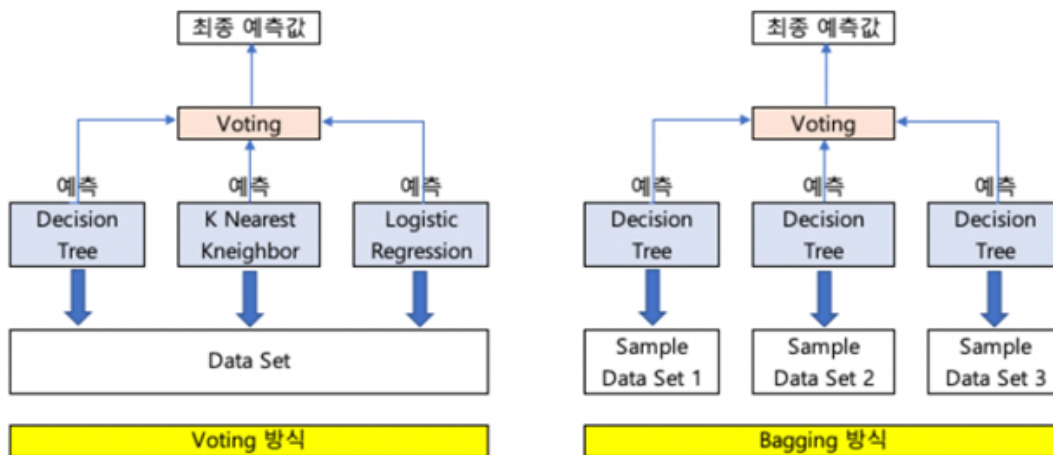
## [ML/DL] 앙상블 학습 (Ensemble Learning): 2. Voting(보팅)이란?

2020. 11. 5. 03:08 수정 삭제 공개

# voting 이란?

다른 알고리즘이 낸 결과물에 대하여 투표를 통해 결정하는 방식이다.

voting과 Bagging은 투표방식이라는 공통점을 가지지만 차이점이 있다.

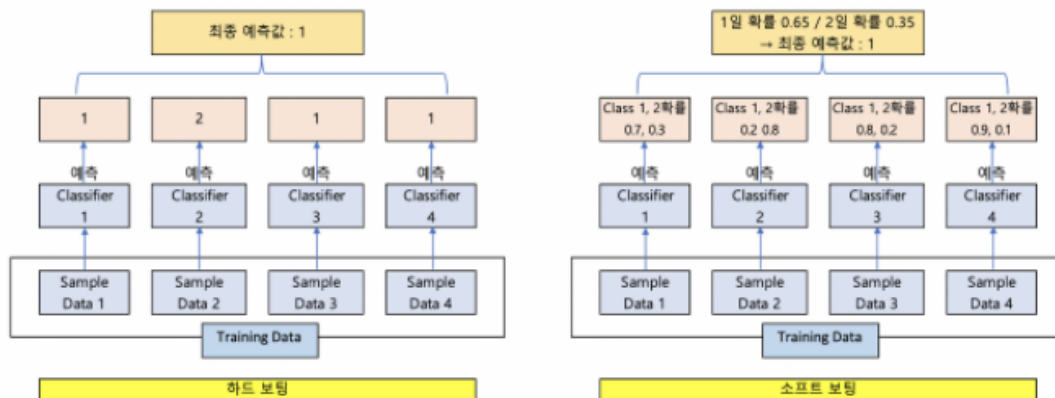


- voting은 다른 알고리즘 model을 조합해서 사용한다.
- bagging은 동일 알고리즘을 다른 sample 조합을 사용한다.

## 보팅의 유형

Hard voting - 각자의 분류를 했을 때 가장 투표를 많이 받은걸 선택한다.

Soft voting - 분류에 따라서 1,2가 있었을때 각각의 확률을 구하고 이것을 더해서 평균을 낸다. 그때 가장 높았던 것을 사용한다.



성능 측면에서는 소프트 보팅이 상대적으로 우수하다.

## Voting 구현

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScale
```

```

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix, precision_recall_curve, roc_curve
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import VotingClassifier
from sklearn.datasets import load_breast_cancer

```

```

cancer = load_breast_cancer()

cancer.keys()

```

```
dict_keys(['data', 'target', 'target_names', 'DESCR', 'feature_names'])
```

```

cancer_df = pd.DataFrame(cancer.data, columns=cancer.feature_names)

cancer_df.head()

```

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst radius	worst texture	worst perimeter	worst area	worst smoothness
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07671	...	25.38	17.33	154.80	2019.0	0.185
1	20.57	17.77	132.90	1326.0	0.08474	0.07664	0.0869	0.07617	0.1812	0.05667	...	24.99	23.41	150.80	1956.0	0.122
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	...	23.57	25.53	152.50	1709.0	0.144
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	...	14.91	26.50	88.87	567.7	0.201
4	20.29	14.34	135.10	1297.0	0.10030	0.13200	0.1980	0.10430	0.1809	0.05083	...	22.54	16.67	152.20	1575.0	0.137

## # 소프트 보팅

*# 앙상블을 구현하는 보팅 분류기 - votingClassifier*

```
dtc = DecisionTreeClassifier()  
lr = LogisticRegression()  
rfc = RandomForestClassifier()  
knn = KNeighborsClassifier()
```

*# 앙상블 모델을 구현하기 위해서는 구현을 위한 분류기(votingClassifier)*

```
voting_model = VotingClassifier([('dtc',dtc),('lr',lr),('knn',knn)],voting='soft')
```

*# 데이터 세트를 분리*

```
X_train , X_test, y_train, y_test = train_test_split(cancer.data,cancer.target, test_size=
```



*# 보팅 분류기를 이용해서 학습/예측/평가*

```
voting_model.fit(X_train, y_train)  
y_pred = voting_model.predict(X_test)  
print('VOTING ACCURACY:', accuracy_score(y_test,y_pred))
```

```
# 개별 학습기에 대한 학습/예측/평가

classifier = [dtc,lr,knn]

for c in classifier:
    c.fit(X_train,y_train)
    pred=c.predict(X_test)
    print(c.__class__.__name__, accuracy_score(y_test,pred))
```

```
VOTING ACCURACY: 0.9385964912280702
DecisionTreeClassifier 0.9385964912280702
LogisticRegression 0.9473684210526315
KNeighborsClassifier 0.9035087719298246
```

## # 하드 보팅

```
# 앙상블을 구현하는 보팅 분류기 - votingClassifier
```

```
dtc = DecisionTreeClassifier()
lr = LogisticRegression()
rfc = RandomForestClassifier()
knn = KNeighborsClassifier()
```

```
# 앙상블 모델을 구현하기 위해서는 구현을 위한 분류기(votingClassifier)
```

```
voting_model = VotingClassifier([('dtc',dtc),('lr',lr),('knn',knn)],voting='hard')
```

*# 데이터 세트를 분리*

```
X_train , X_test, y_train, y_test = train_test_split(cancer.data,cancer.target, test_size=
```



*# 보팅 분류기를 이용해서 학습/예측/평가*

```
voting_model.fit(X_train, y_train)
y_pred = voting_model.predict(X_test)
print('VOTING ACCURACY:', accuracy_score(y_test,y_pred))
```

*# 개별 학습기에 대한 학습/예측/평가*

```
classifier = [dtc,lr,knn]

for c in classifier:
    c.fit(X_train,y_train)
    pred=c.predict(X_test)
    print(c.__class__.__name__, accuracy_score(y_test,pred))
```

```
VOTING ACCURACY: 0.9385964912280702
DecisionTreeClassifier 0.9473684210526315
LogisticRegression 0.9473684210526315
KNeighborsClassifier 0.9035087719298246
```

---

-참조 : [injo.tistory.com/22](http://injo.tistory.com/22)

## [Chapter 4. 분류] 앙상블 학습

[injo.tistory.com](http://injo.tistory.com)

4-2. 앙상블 학습 In [1]: from IPython.core.display import display, HTML display(HTML(" ")) 1. 앙상블 학습이란? 앙상블(Ensemble) 학습 앙상블이란 여...

[ariz1623.tistory.com/207](http://ariz1623.tistory.com/207)

## 앙상블 학습 개요

[ariz1623.tistory.com](http://ariz1623.tistory.com)

앙상블 학습이란 앙상블 학습을 통한 분류는 여러 개의 분류기를 생성하고 그 예측을 결합함으로써 보다 정확한 최종 예측 도출하는 기법을 말한다. ...

### 'ML,DL' 카테고리의 다른 글

[ML/DL] XGboost의 정의와 구현 및 hyper parameter 설정

[ML/DL] 앙상블 학습 (Ensemble Learning): 3.Boosting(부스팅)이란?

**[ML/DL] 앙상블 학습 (Ensemble Learning): 2. Voting(보팅)이란?**

[ML/DL] 앙상블 학습 (Ensemble Learning): 1. bagging(배깅)이란?

[ML/DL] 앙상블 학습 (Ensemble Learning): bagging,voting,boosting

[ML/DL] DecisionTree 구현 및 hyper parameter 설정

VOTING

보팅

앙상블 voting

앙상블 보팅



나아무늘보

혼자 끄적끄적하는 블로그 입니다.