Introduction to model validation

MODEL VALIDATION IN PYTHON

Kasey Jones
Data Scientist

datacamp

What is model validation?

Model validation consists of:

- Ensuring your model performs as expected on new data
- Testing model performance on holdout datasets
- Selecting the best model, parameters, and accuracy metrics
- Achieving the best accuracy for the data given

model의 정확도가 훈련 데이터 및 홀드아웃 데이터에 대해 유사한 경우 모형이 검증되었다고 할 수 있음.

모델 검증은 다음을 선택하는 것으로 구성

scikit-learn modeling review

Basic modeling steps:

```
model = RandomForestRegressor(n_estimators=500, random_state=1111)
model.fit(X=X_train, y=y_train)
```

```
RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
           max_features='auto', max_leaf_nodes=None,
           min_impurity_decrease=0.0, min_impurity_split=None,
           min_samples_leaf=1, min_samples_split=2,
           min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=1,
           oob_score=False, random_state=1111, verbose=0, warm_start=False)
```

Modeling review continued

```
predictions = model.predict(X_test)
print("{0:.2f}".format(mae(y_true=y_test, y_pred=predictions)))
```

```
10.84
```

Mean Absolute Error Formula (평균 절대 오차를 사용하는 경우 y_true, y_pred를 사용하여 반환)

$$\frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n}$$

Review prerequisites

- Intermediate Python
- Supervised Learning with scikit-learn

How often did a fun-sized candy of a given type win its matchups against the rest of the field?

Search...

RK	CANDY	WIN PERCENTAGE
1	Reese's Peanut Butter Cup	84.2%
2	Reese's Miniatures	81.9
3	Twix	81.6
4	Kit Kat	76.8
5	Snickers	76.7

Seen vs. unseen data

Training data = seen data

일반적으로 모델은 보이지 않는 데이터가 모델에 노출되지 않은 특징이나 특성을 가질 수 있기 때문에 이 전에 본 데이터에서 훨씬 더 나은 성능을 발휘.

```
model = RandomForestRegressor(n_estimators=500, random_state=1111)
model.fit(X_train, y_train)
train_predictions = model.predict(X_train)
```

Testing data = unseen data

```
model = RandomForestRegressor(n_estimators=500, random_state=1111)
model.fit(X_train, y_train)
test_predictions = model.predict(X_test)
```

Let's begin!

MODEL VALIDATION IN PYTHON

Regression models

MODEL VALIDATION IN PYTHON

Kasey Jones
Data Scientist

datacamp

Random forests in scikit-learn

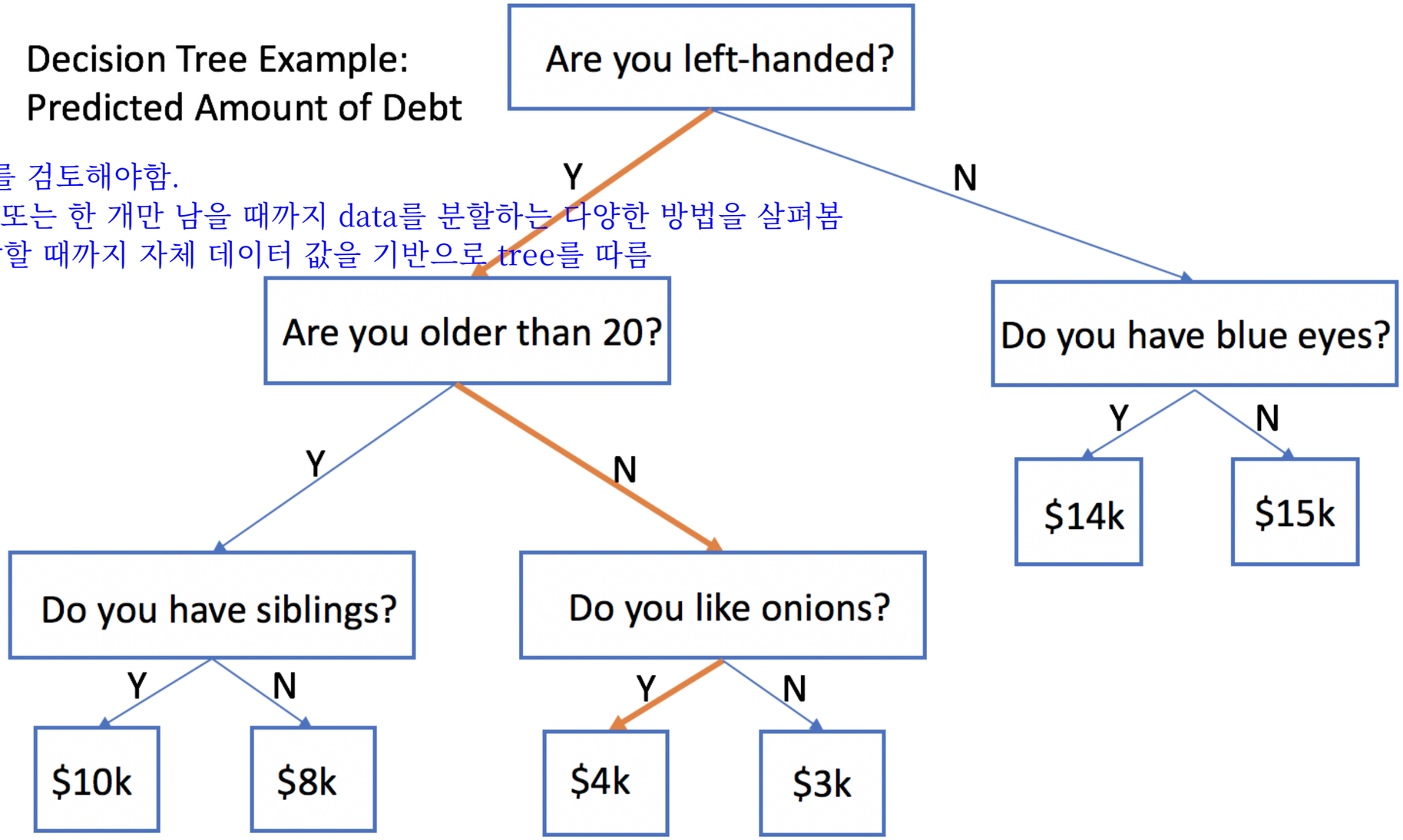
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import RandomForestClassifier
```

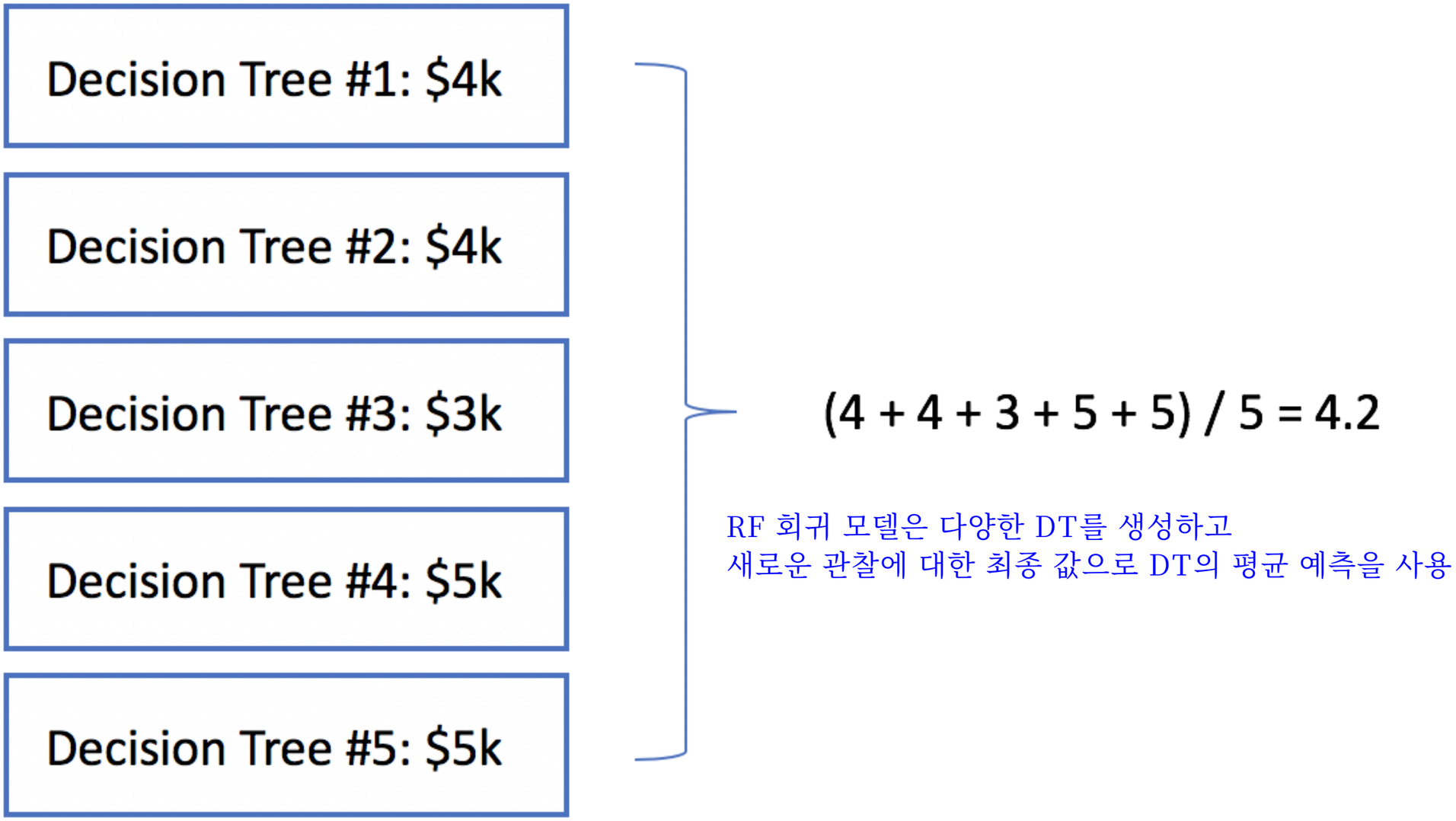
동일한 방식으로 호출

```
rfr = RandomForestRegressor(random_state=1111)
rfc = RandomForestClassifier(random_state=1111)
```

Decision Tree Example:
Predicted Amount of Debt

rf를 이해하려면 dt를 검토해야함.
DT는 관찰이 몇 개 또는 한 개만 남을 때까지 data를 분할하는 다양한 방법을 살펴봄
마지막 node에 도달할 때까지 자체 데이터 값을 기반으로 tree를 따름





RF 회귀 모델은 다양한 DT를 생성하고
새로운 관찰에 대한 최종 값으로 DT의 평균 예측을 사용

Random forest parameters

`n_estimators` : the number of trees in the forest

n_estimators, max_depth만으로도 model accuracy에 큰 영향을 미칠 수 있음

`max_depth` : the maximum depth of the trees

`random_state` : random seed 재현 가능

```
from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor(n_estimators=50, max_depth=10)
```

```
rfr = RandomForestRegressor(random_state=1111)
rfr.n_estimators = 50
rfr.max_depth = 10
```

Feature importance

Print how important each column is to the model

```
for i, item in enumerate(rfr.feature_importances_):
    print("{0:s}: {1:.2f}".format(X.columns[i], item))
```

```
weight: 0.50
height: 0.39
left_handed: 0.72
union_preference: 0.05
eye_color: 0.03
```

dataframe, X인 경우 열 이름에 액세스하고 중요도 점수를 쉽게 print가능
(숫자가 클 수록 중요)

Let's begin

MODEL VALIDATION IN PYTHON

Classification models

MODEL VALIDATION IN PYTHON

Kasey Jones
Data Scientist

datacamp

Classification models

이 과정에서는 분류 모델 또는 반응 변수가 범주형일 때 사용하기 위해 만들어진 무델을 검토하는 데 초점을 맞춤

- Categorical Responses:
  - Newborn's hair color
  - Winner of a basketball game
  - Genre of the next song on the radio

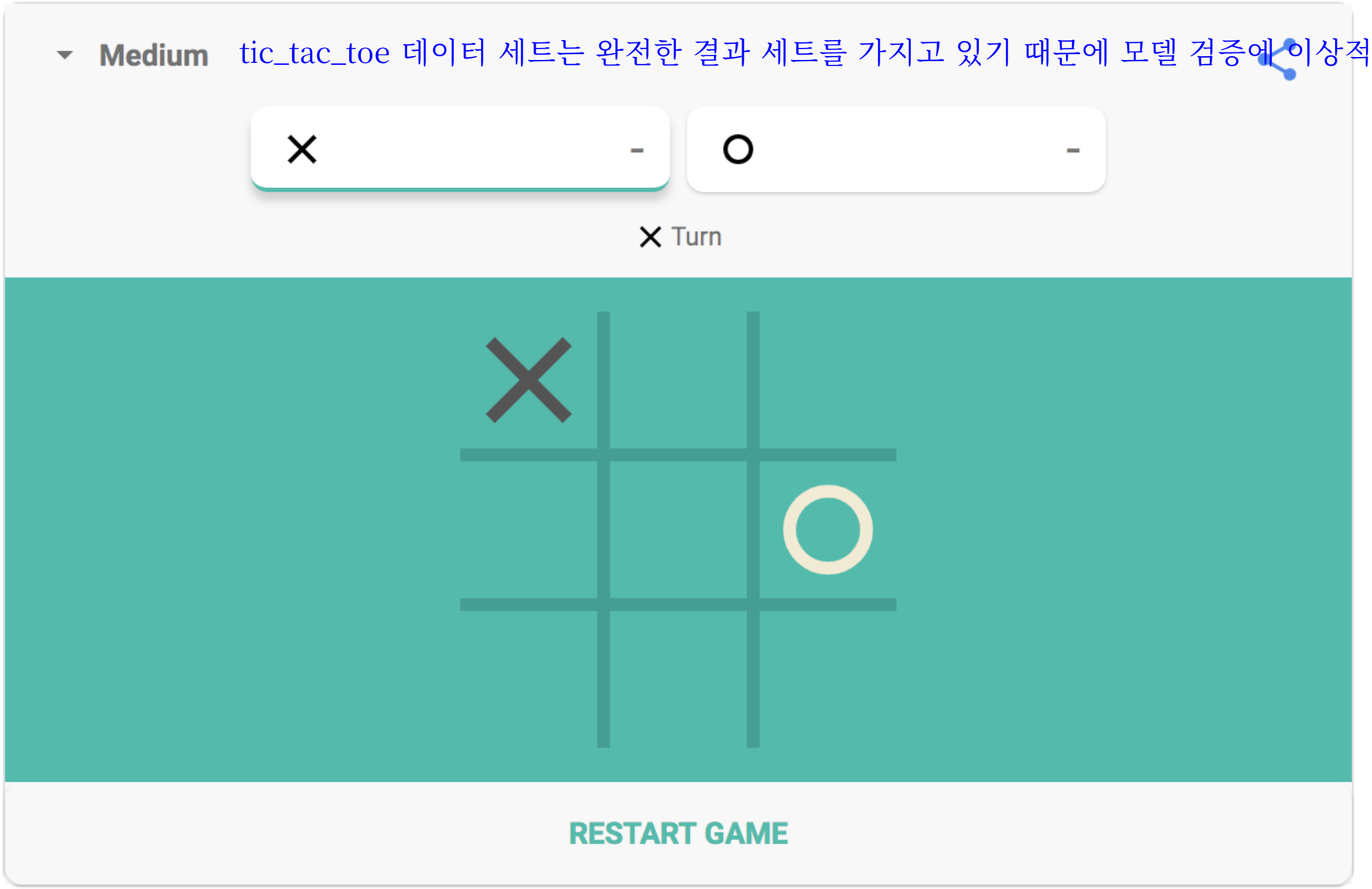
The Tic-Tac-Toe dataset

분류 모델의 예

...	Bottom-Left	Bottom-Middle	Bottom-Right	Class
...	X	O	O	positive
...	O	X	O	positive
...	O	O	X	positive
...	X	X	O	negative
...	...	...	...	...

'b' : 사각형이 비어 있음
'X' : player 1
'O' : player 2
마지막 열 : first player1이 이겼는지 나타냄

Medium tic_tac_toe 데이터 세트는 완전한 결과 세트를 가지고 있기 때문에 모델 검증에 이상적임.



Using .predict() for classification

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state=1111)
rfc.fit(X_train, y_train)
rfc.predict(X_test)
```

각 class print하기

```
array([1, 1, 1, 1, 0, 1, ...])
```

```
pd.Series(rfc.predict(X_test)).value_counts()
```

class 개수 count

```
1    627
0    331
```

Predicting probabilities

```
rfc.predict_proba(X_test)
```

예측 확률 배열을 반환
배열의 각 항목에는 합계가 1인 확률이 포함

```
array([[0. , 1. ],
       [0.1, 0.9],
       [0.1, 0.9],
       ...])
```

이 데이터 포인트의 경우, 현재 게임 보드를 고려할 때 플레이어 1이 질 확률이 10%이고, 이길 확률이 90%라는 것을 나타낸다.

```
rfc = RandomForestClassifier(random_state=1111)
rfc.get_params()
```

두 가지 추가 방법

1. .get_params() 사용하여 매개변수를 검토 후
매개변수 사전과 그 값을 출력하여 정확히 어떤 매개변수가 사용되었는지 확인
매개변수를 아는 것은 model 퀄리티를 평가하고, model은 다시 실행하고 매개변수를 조정할 때 필수적

```
{'bootstrap': True,
 'class_weight': None,
 'criterion': 'gini',
 ...}
```

```
rfc.fit(X_train, y_train)
rfc.score(X_test, y_test)
```

분류 모델의 전체 정확도를 빠르게 확인할 수 있는 방법

```
0.8989
```

Let's classify Tic-Tac-Toe end-game scenarios

MODEL VALIDATION IN PYTHON