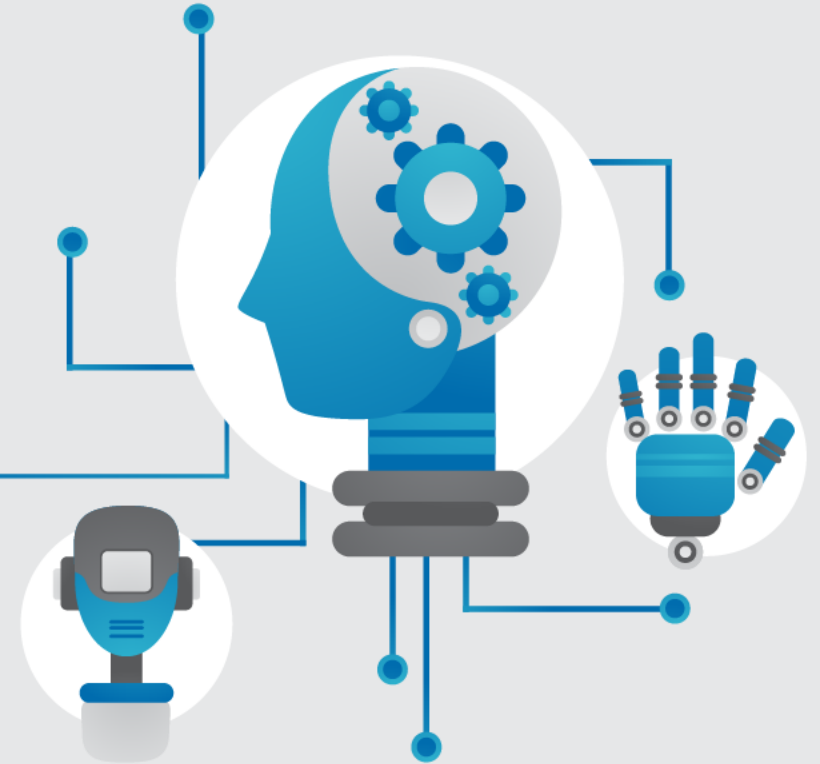


投票法 (Voting)

整體學習





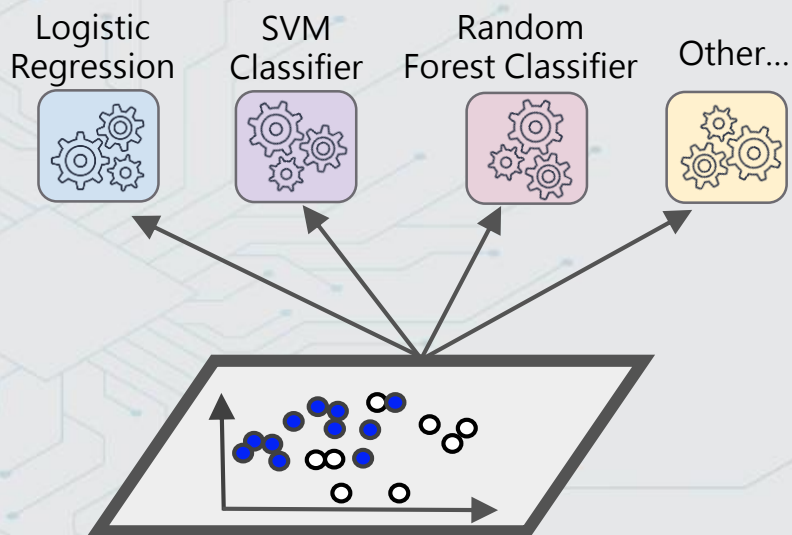
投票法 (Voting)



› 投票分類器 (Voting Classifier)

是一種**後設預測器** (meta-estimator)，它接收一系列 (可能完全不同) 的機器學習分類演算法，再將它們各自預測的結果進行平均，得到最終的預測。

› `sklearn.ensemble.VotingClassifier` 是投票法的實作





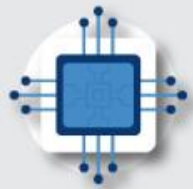
Voting Classifier 參數說明



› `class sklearn.ensemble.VotingClassifier(estimators,
voting='hard', weights=None, n_jobs=None,
flatten_transform=True)`

› Voting Classifier 類別常用參數

- `estimators`
- `voting`
- `weights`
- `flatten_transform`

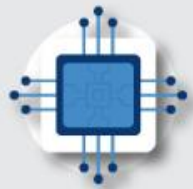


參數 estimators



› estimators : list of (str, estimator) tuples

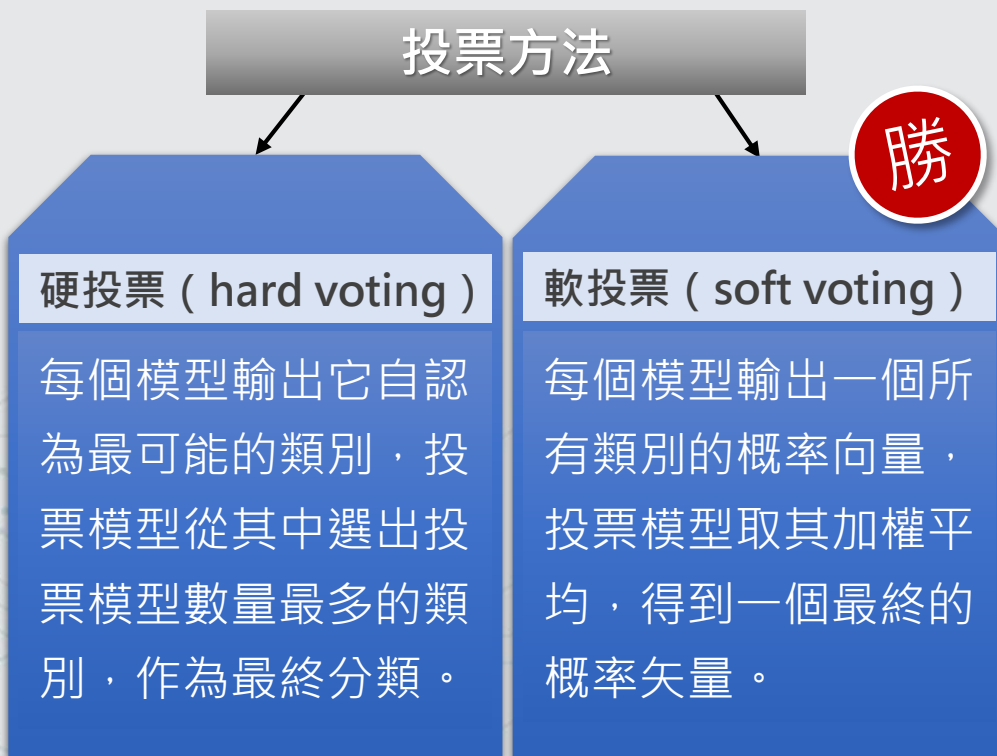
- 一串預測器，以名稱和模型配對格式列出，例如 `[('lr', clf1), ('rf', clf2), ('gnb', clf3)]`。



參數 voting



› voting : str, {'hard', 'soft'} (default='hard')



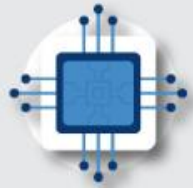
- 一般而言，軟投票的效果好於硬投票
- 預設值為'hard'



參數 weights



- › weights : array-like, shape (n_classifiers,), optional (default=`None`)
 - 權重序列 (float 或 int)
 - hard voting : 對預測類別標籤出現次數的加權
 - soft voting : 對預測類別進行平均前的概率的加權
 - 如果使用None，則使用uniform權重

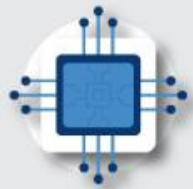


參數 `flatten_transform`



› `flatten_transform` : bool, optional (default=True)

- 只有當 `voting = 'soft'` 時，才會影響變換輸出的shape。
- 如果 `voting = 'soft'` 並且 `flatten_transform = True`，則 `transform` 方法將回傳shape為 (`n_samples`，`n_classifiers * n_classes`) 的矩陣。
- 如果 `flatten_transform = False`，則回傳 (`n_classifiers`，`n_samples`，`n_classes`)



Voting Classifier 函式說明

機器學習實務



› Voting Classifier 常用函式

- fit
- predict
- score



Icon made by Eucalyp from www.flaticon.com



訓練



- › 指令：`fit(self, X, y, sample_weight=None)`
- › 參數
 - `X`：訓練向量
 - `y`：相對於`X`的目標向量
- › 回傳：訓練後的voting classifier 模型物件
- › 說明：從訓練中建立一個投票預測器



訓練



› 範例程式

```
from sklearn.ensemble import VotingClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier

clf1 = LogisticRegression(multi_class='multinomial', random_state=1)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)
clf3 = SVC()

votingClassifier = VotingClassifier(estimators=[ ('lr', clf1), ('rf', clf2),
('gnb', clf3)], voting='hard')

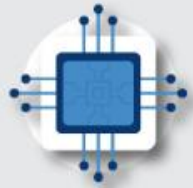
votingClassifier.fit(X_train, y_train)
```



預測



- › 指令：`predict(self, X)`
- › 參數
 - X: 輸入樣本
- › 回傳：每個樣本的預測類別標籤
- › 說明：預測X中樣本的類別標籤



預測



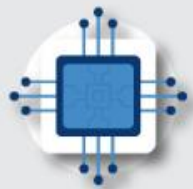
› 範例程式

```
clf1 = LogisticRegression(multi_class='multinomial', random_state=1)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)
clf3 = SVC()

votingClassifier = VotingClassifier(estimators=[ ('lr', clf1), ('rf', clf2),
('gnb', clf3)], voting='hard')

votingClassifier.fit(X_train, y_train)

predictions = votingClassifier.predict(X_test)
```



評分



- › 指令：`score(self, X, y, sample_weight=None)`
- › 參數
 - `X`：測試樣本
 - `y`：測試樣本的正確答案
- › 回傳：測試樣本的平均準確度
- › 說明：返回給定測試數據和標籤上的平均準確度



評分



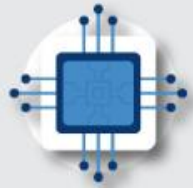
› 範例程式

```
clf1 = LogisticRegression(multi_class='multinomial', random_state=1)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)
clf3 = SVC()
```

```
votingClassifier = VotingClassifier(estimators=[ ('lr', clf1), ('rf', clf2),
('gnb', clf3)], voting='hard')
```

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

```
accuracy = votingClassifier.score(X_test, y_test)
```



投票法 (Voting) 實作



› 以威斯康辛乳癌數據集作為資料，
實作投票法 (Voting)，以下列分類器實作

- Logistic Regression
- Random Forest Classifier
- Support Vector Classifier



資料前處理

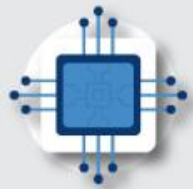


› 從sklearn.datasets載入數據資料

載入資料

```
from sklearn import datasets  
bunch = datasets.load_breast_cancer()
```

```
data = bunch.data  
labels = bunch.target
```

資料前處理



› 將data以及labels分割成train和test資料

切割資料

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,Y_train,Y_test=train_test_split(data, labels,  
test_size=0.3,shuffle=True,stratify=labels)
```



建立模型與參數設定



› 建立Logistic Regression, Random Forest Classifier和Support Vector Classifier模型

建立模型

```
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
```

```
clf1 = LogisticRegression(C=1e5)
clf2 = RandomForestClassifier(n_estimators=50,
                             class_weight='balanced')
clf3 = SVC(C=1e5, kernel='linear', gamma='scale',
           class_weight='balanced')
```

```
from sklearn.ensemble import VotingClassifier
votingClassifier = VotingClassifier( estimators=[ ('lr', clf1),
('rf', clf2), ('gnb', clf3)], voting='hard')
```



模型訓練與評估



› 模型訓練與評估

進行訓練

```
votingClassifier.fit(X_train, Y_train)
```

進行預測

```
acc = votingClassifier.score(X_test, Y_test)
```

```
print('Accuracy:', acc)
```

```
Accuracy: 0.9532163742690059
```