

연습문제 – Confusion Matrix (Q10.1)

아래의 matrix는 165명의 환자를 대상으로 질병이 있는지 없는지 예측한 값과, 실제 값을 나타낸 것이다.

- (1) Accuracy 값을 구하여라
- (2) Recall (Sensitivity, True Positive Rate)
- (3) Fall-out (False Alarm Ratio, False Positive Rate)
- (4) Specificity
- (5) Precision
- (6) Precision + Recall = F1 Score

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

연습문제 – Logistic Regression (Q10.2)

유방암 진단 데이터의 Logistic Regression을 수행하여 다음 값을 구하여라.
아래의 matrix는 165명의 환자를 대상으로 질병이 있는지 없는지 예측한 값과, 실제 값을 나타낸 것이다.

- (1) Accuracy
- (2) Precision
- (3) roc_auc

연습문제 – Logistic Regression (Q10.3)

Tennis 데이터 파일([play_tennis.csv](#))를 사용해 의사결정 트리 코드를 구현하세요.
Classification_report를 사용해 precision과 recall, f1-score를 확인하고 마지막으로 Graphviz를 사용해 트리 구조를
그래프 image로 표현해 보세요.
단, 데이터: [play_tennis.csv](#) 데이터를 다음과 같이 바꾸어 계산해보시요.

- (1) 의사결정 트리를 나타내어라
- (2) Accuracy 값을 구하여라

[play_tennis.csv](#)

Day	Outlook	Temperature	Humidity	Play Tennis
D1	Sunny	Hot	High	No
D2	Overcast	Hot	High	Yes
D3	Overcast	Mild	Normal	Yes
D4	Rain	Mild	Normal	Yes
D5	Sunny	Mild	High	Yes
D6	Rain	Hot	High	No
D7	Overcast	Hot	Normal	No

연습문제 – Random Forest (Q10.4)

Iris 데이터를 이용하여 RandomForest Classifier를 테스트 하여라. n_estimators 값에 따라 정확도를 조사하여라.

- (1) n_estimators 값이 얼마일때 정확도가 높은지를 설명하여라. n_estimators 값을 10개정도 시도해본다.
- (2) 의사결정 트리를 그림으로 나타내어라.
- (3) 다음 임의의 데이터가 iris의 어느 종에 속하는지 알아내어라

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
5.6	2.9	3.6	1.3

[참고]

- (2) 의사결정트리 나타내기 – 아래 코드 참조하면 그릴수있음

```
from IPython.display import Image
import pydotplus
model = rfc.estimators_[5]
feature_names = iris.feature_names[0:4] # ['Outlook', 'Temperature',
'Humidity', 'Wind']
target_name = np.array(['iris1', 'iris2', 'iris3'])
dt_dot_data = tree.export_graphviz(model,
    feature_names = iris.feature_names,
    class_names = iris.target_names,
    filled = True, rounded = True,
    special_characters = True)
dt_graph = pydotplus.graph_from_dot_data(dt_dot_data)
Image(dt_graph.create_png())
```

- (3) Iris가 어디에 속하는지 결정트리로 찾아내기

```
myX_test = np.array([[5.6, 2.9, 3.6, 1.3]])
myprediction = rfc.predict(myX_test)
```

