01. 기본- 결정트리(decision tree)

- Machine Learning with sklearn @ DJ,Lim
- date : 19/09

데이터 셋 다운로드

- https://www.kaggle.com/uciml/pima-indians-diabetes-database (https://www.kaggle.com/uciml/pima-indians-diabetes-database)

(가) decision tree는 classification(분류)와 regression(회귀) 문제에 널리 사용하는 모델이다.
(나) 스무고개 놀이의 질문과 비슷하다.

In [1]:

```
# 라이브러리 불러오기
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics
```

In [0]:

```
pima = pd.read_csv("diabetes.csv")
```

In [28]:

```
pima.head()
```

Out[28]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

Feature Selection

In [30]:

```
pima.columns
```

Out[30]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [33]:

```
pima.head(3)
```

Out[33]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67

In [0]:

```
# 데이터 셋 (feature와 target 변수로 나누기)
feature_cols = ['Pregnancies', 'Insulin', 'BMI', 'Age','Glucose',
                'BloodPressure','DiabetesPedigreeFunction']
X = pima[feature_cols] # Features
y = pima.Outcome       # Target variable
```

Splitting Data

In [0]:

```
# 데이터 셋 나누기
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% trair
```

In [36]:

```
print(X_test.columns)
print(X_train.columns)
print(y_train.shape)
```

```
Index(['Pregnancies', 'Insulin', 'BMI', 'Age', 'Glucose', 'BloodPressure',
       'DiabetesPedigreeFunction'],
      dtype='object')
Index(['Pregnancies', 'Insulin', 'BMI', 'Age', 'Glucose', 'BloodPressure',
       'DiabetesPedigreeFunction'],
      dtype='object')
(537,)
```

In [0]:

```
# 의사결정 트리 모델 생성
clf = DecisionTreeClassifier()

# 학습
clf = clf.fit(X_train,y_train)

# 예측
y_pred = clf.predict(X_test)
```

모델 평가

In [39]:

```
# Model Accuracy, 얼마나 정확한가? 정확도
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.6926406926406926

In [40]:

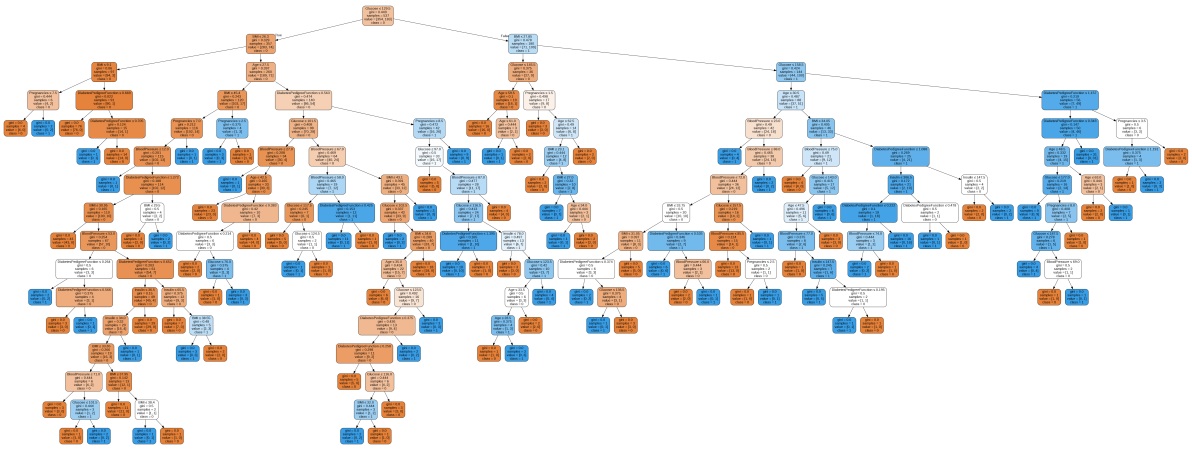
```
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/externals/six.py:31: DeprecationWarni
ng: The module is deprecated in version 0.21 and will be removed in version 0.23 sin
ce we've dropped support for Python 2.7. Please rely on the official version of six
(https://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", DeprecationWarning)
```

In [41]:

```
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True,feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

Out[41]:



성능 최적화

In [42]:

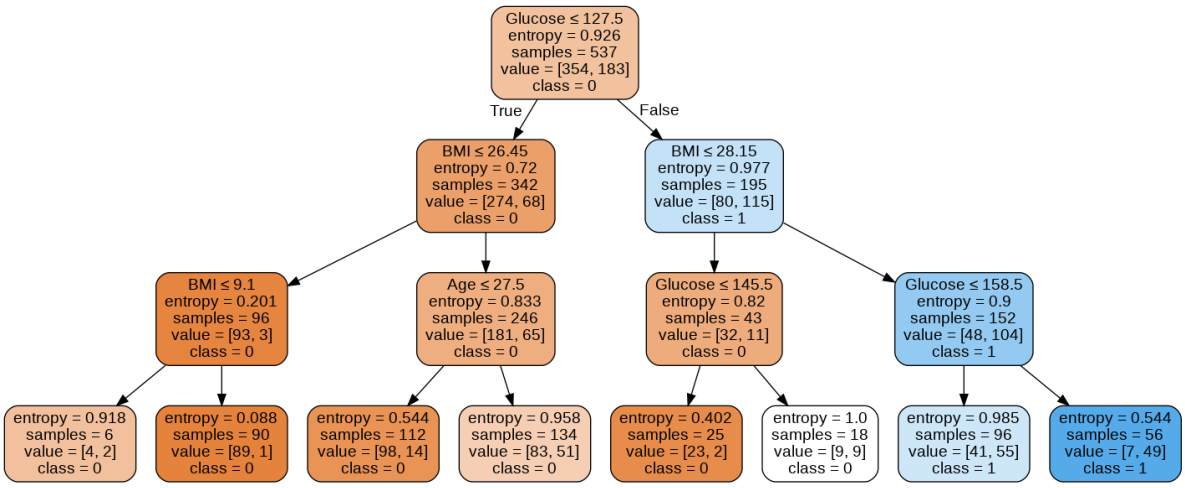
```
# 의사결정트리 모델
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

# 학습
clf = clf.fit(X_train,y_train)

# 데이터 셋 예측
y_pred = clf.predict(X_test)

# 정확도 확인
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

Accuracy: 0.7705627705627706

In [43]:

```
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(clf, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names = feature_cols,class_names=['0','1'])
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('diabetes.png')
Image(graph.create_png())
```

Out[43]:



In [0]: