01. 기본- 결정트리(decision tree)

- Machine Learning with sklearn @ DJ,Lim
- date : 21/07

데이터 셋 다운로드

- UCI : 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 모델이
다.
(나) 스무고개 놀이의 질문과 비슷하다.
```

In [3]:

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

Data Fields

구분	설명
Pregnancies	임신
Glucose	포도당
BloodPressure	혈압
SkinThickness	피부두께
Insulin	인슐린
BMI	BMI
Diabetes Pedigree Function	당뇨병혈통기능
Age	나이
Outcome	결과

In [5]:

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

In [6]:

```
pima.columns
```

Out[6]:

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

In [7]:

```
pima.head()
```

Out[7]:

	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 [8]:

```
pima.columns
```

Out[8]:

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

In [12]:

```
pima.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [13]:

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

Out[13]:

	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 [14]:

```
# 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
```

데이터 나누기

In [15]:

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

In [16]:

```
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 [17]:

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

# 학습
clf = clf.fit(X_train,y_train)

# 예측
y_pred = clf.predict(X_test)
```

모델 평가

In [18]:

```
from sklearn import metrics
```

In [19]:

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

```
Accuracy: 0.670995670995671
```

In [20]:

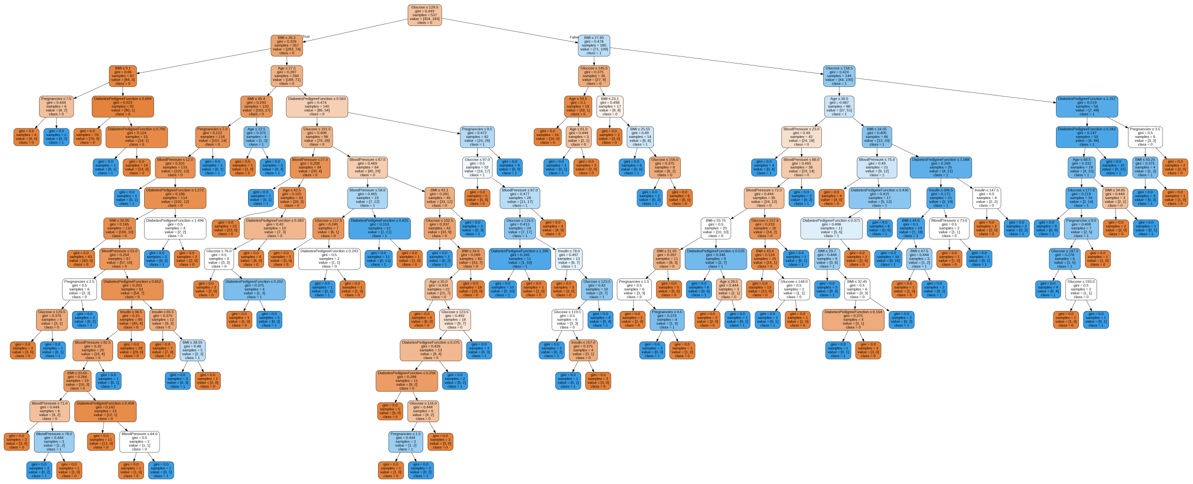
```
from sklearn.tree import export_graphviz
from sklearn.externals.six import StringIO
from IPython.display import Image
import pydotplus
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/externals/six.py:31: FutureWarning: T
he module is deprecated in version 0.21 and will be removed in version 0.23 since w
e've dropped support for Python 2.7. Please rely on the official version of six (htt
ps://pypi.org/project/six/).
  "(https://pypi.org/project/six/).", FutureWarning)
```

In [21]:

```
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[21]:



모델 성능 개선

In [23]:

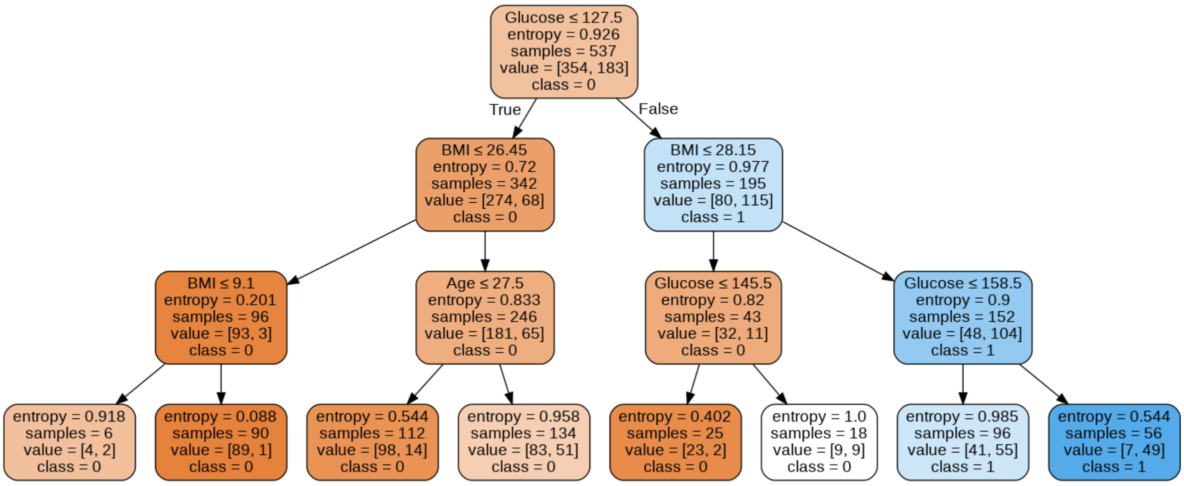
```
# 의사결정트리 모델
clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)

# 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 [24]:

```
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[24]:



In [ ]:

```
```