

Python Study

Environ. Analysts

<환경을 분석할 줄 아는 사람이 되자!>



Type2 머신 러닝

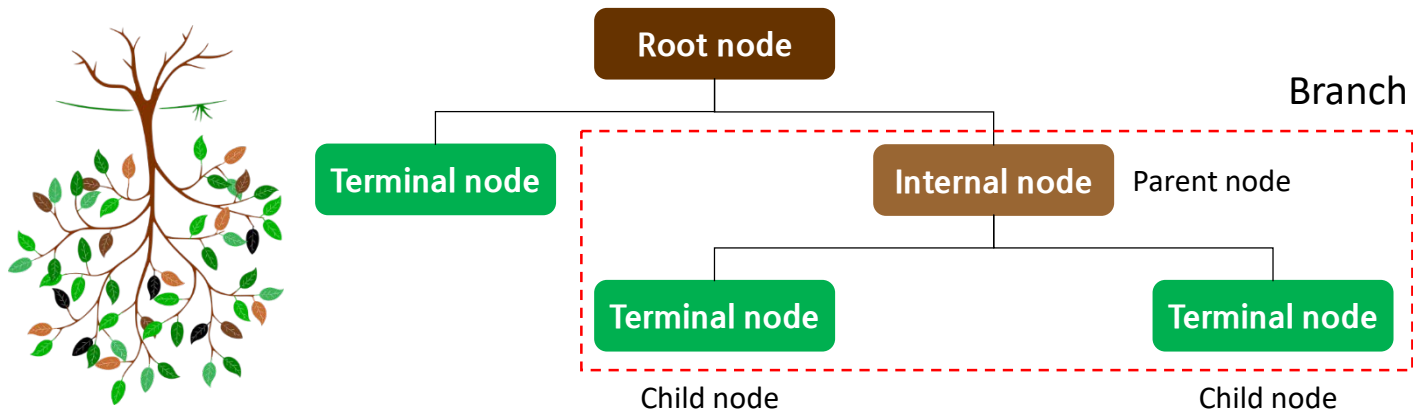


- 결정나무(Decision trees) 이론
- 결정나무(Decision trees) 실습

결정나무(Decision trees)

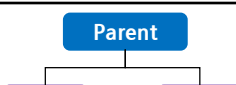
○ 결정나무 모델이란?

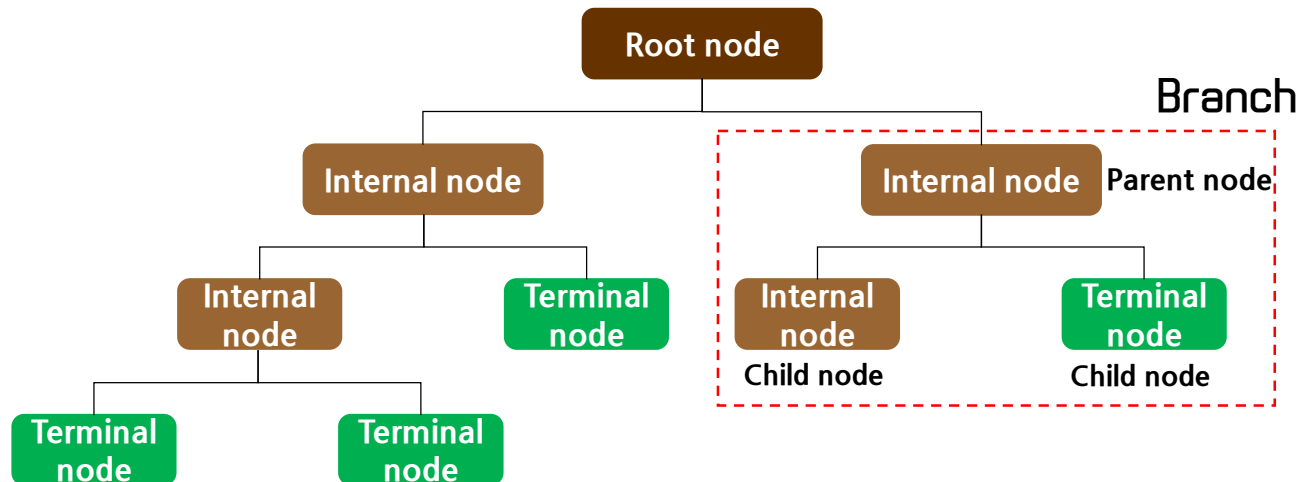
- 의사결정규칙(decision rule)과 그 결과들을 나무(tree) 구조로 도식화 하는 분석 방법
- 데이터를 가장 적합한 기준으로 이분하는 기법으로 탐색과 모형화 두 가지 특징이 있음
- 분류 또는 예측이 나무구조에 의한 추론규칙(induction rule)에 의해서 표현되기 때문에 결과를 해석하고 이해하기 쉬우며, 적용이 간단한 장점이 있음



결정나무(Decision trees)

○ 결정나무 모델의 구조1

이름	형태	설명
뿌리 노드(Root node)	Root node	분류(또는 예측) 대상이 되는 모든 자료집단을 포함
부모 노드(Parent node)		가지(Branch)의 상위 노드
자식 노드(Child node)		가지(Branch)의 하위 노드
내부 노드(Internal node)	Internal node	최종 노드가 아닌
최종 노드(Terminal node)	Terminal node	각 나무줄기의 끝에 위치, 의사결정나무에서 분류 규칙은 끝마디의 개수만큼 생성






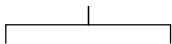
Type2 머신 러닝

<환경을 분석할 줄 아는 사람이 되자!>

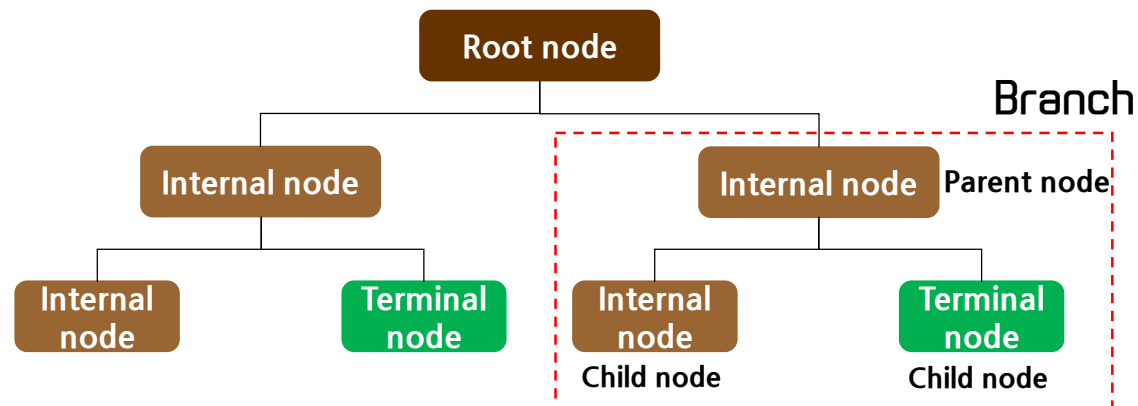


결정나무(Decision trees)

○ 결정나무 모델의 구조2

이름	형태	설명
가지(Branch)		결정나무 마디들의 모임
가지분할(Split)		분리기준에 따라 나무의 가지를 생성하는 과정
가지치기(Pruning)		생성된 가지를 잘라내어 모델을 단순화하는 과정

- 분리기준(Split criterion) : 마디들이 형성될 때, 입력변수(input variable)의 선택과 범주의 병합이 이루어질 기준.
- 가지치기(Pruning) : 분류오류를 크게 할 위험(risk)이 높거나 부적절한 추론규칙(induction rule)을 가지고 있는 가지를 제거.



결정나무(Decision trees)

○ 결정나무 모델의 분리기준(분류나무)

- 분류나무 모델이고 이산형 목표변수의 경우
- 목표변수의 각 범주에 속하는 빈도(frequency)에 기초하여 분리가 일어남.
- 사용되는 분리기준
 - 카이제곱 통계량의 p-값(Chi-square statistic)
 - 지니 지수(Gini index)
 - 엔트로피 지수(entropy index)
- 선택된 기준에 의해 분할이 일어날 때, 카이제곱 통계량의 p-값은 그 값이 작을수록 자식 노드 간의 이질성이 큼을 나타내며
- 자식 노드에서의 지니 지수나 엔트로피 지수는 그 값이 클수록 자식 노드 내의 이질성이 큼을 의미한다. 따라서 이 값들이 가장 작아지는 방향으로 가지분할을 수행

결정나무(Decision trees)

○ 결정나무 모델의 분리기준(이질성 / 순수도)

- 분리기준을 위해 목표변수의 분포를 구별하는 정도를 순수도(purity) 또는 불순도(impurity)에 의해서 측정
- **순수도** : 목표변수의 특정 범주에 개체들이 포함되어 있는 정도.
- **이질성** : 목표변수의 특정 범주에 이질적인 개체가 얼마나 섞였는지 포함하는 정도.

높은 이질성 <=> 낮은 순수도



$$G = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{3}{8}\right)^2 - \left(\frac{1}{8}\right)^2 - \left(\frac{1}{8}\right)^2 = 0.69$$

낮은 이질성 <=> 높은 순수도



$$G = 1 - \left(\frac{7}{8}\right)^2 - \left(\frac{1}{8}\right)^2 = 0.24$$



Type2 머신 러닝

〈환경을 분석할 줄 아는 사람이 되자!〉



결정나무(Decision trees)

○ 결정나무 모델의 분리기준(분류나무)

➤ 지니 지수 :

$$G_i = \sum_{k=1}^n P_{i,k}^2 \quad 0 \leq G \leq 1/2$$

$P_{i,k}$: i 번째 노드에 있는 훈련 샘플 중 클래스 k에 속한 샘플의 비율

➤ 엔트로피 지수 :

$$H_i = - \sum_{k=1}^c P_{i,k} \log_2(P_{i,k}) \quad 0 \leq H_i \leq 1$$

$P_{i,k}$: i 번째 노드에 있는 훈련 샘플 중 클래스 k에 속한 샘플의 비율

결정나무(Decision trees)

○ 결정나무 모델의 분리기준(회귀나무)

- 회귀나무 모델이고 연속형 목표변수의 경우
- 목표변수가 연속형(구간형)인 경우 목표변수의 평균과 표준편차에 기초하여 마디의 분리가 일어난다.
- 사용되는 분리기준
 - 평균(mean) / 표준편차(standard deviation)
 - ANOVA F-통계량

결정나무(Decision trees)

○ 결정나무 모델의 분리기준(정지규칙과 가지치기)

➤ 정지규칙(Stopping rule)

- 더 이상 분리가 일어나지 않고 현재의 마디가 끝마디가 되도록 하는 여러가지 규칙
- 종류 : 카이제곱 검정통계량, 지니지수, 엔트로피 지수, 엔트로피 지수

➤ 가지치기(Pruning)

- 끝마디가 너무 많아 모형이 과대적합된 상태를 방지하기 위한 여러가지 규칙에 따른 방법
- 분류오류를 크게 할 위험(risk)이 높거나 부적절한 추론규칙(induction rule)을 가지고 있는 가지를 제거.



Type2 머신 러닝

〈환경을 분석할 줄 아는 사람이 되자!〉



결정나무(Decision trees)

○ 결정나무 모델 과정

1. 목표변수와 관계가 있는 설명변수들의 선택
2. 분석목적과 자료의 구조에 따라 적절한 **분리기준과 정지규칙**을 정하여
의사결정 나무의 생성
3. 부적절한 나뭇가지는 제거 : **가지치기**
4. 이익(Gain), 위험(Risk), 비용(Cost)등을 고려하여 **모형평가(교차검증)**
5. **분류(Classification) 및 예측(Prediction)** 수행

Type2 머신 러닝



○ 결정나무(Decision trees) 이론

○ 결정나무(Decision trees) 실습



Type2 머신 러닝

<환경을 분석할 줄 아는 사람이 되자!>



결정나무(Decision trees)

0. 모듈 import

0. Data 불러오기

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
%matplotlib inline
```

```
raw_data = pd.read_excel('titanic.xls')
raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 14 columns):
pclass      1309 non-null int64
survived     1309 non-null int64
name        1309 non-null object
sex          1309 non-null object
age         1046 non-null float64
sibsp       1309 non-null int64
parch       1309 non-null int64
ticket      1309 non-null object
fare        1308 non-null float64
cabin       295 non-null object
embarked    1307 non-null object
boat        486 non-null object
body        121 non-null float64
home.dest   745 non-null object
dtypes: float64(3), int64(4), object(7)
memory usage: 143.2+ KB
```

◆ Data Columns의 구분

- pclass : 객실 등급
- survived : 생존 유무
- sex : 성별
- age : 나이
- sibsp : 형제 혹은 부부의 수
- parch : 부모, 혹은 자녀의 수
- fare : 지불한 운임
- boat : 탈출한 보트가 있다면 boat 번호

결정나무(Decision trees)

1. 파악 : Pandas의 describe를 통한 Data의 현황 파악

```
raw_data.describe()
```

	pclass	survived	age	sibsp	parch	fare	body
count	1309.000000	1309.000000	1046.000000	1309.000000	1309.000000	1308.000000	121.000000
mean	2.294882	0.381971	29.881135	0.498854	0.385027	33.295479	160.809917
std	0.837836	0.486055	14.413500	1.041658	0.865560	51.758668	97.696922
min	1.000000	0.000000	0.166700	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	21.000000	0.000000	0.000000	7.895800	72.000000
50%	3.000000	0.000000	28.000000	0.000000	0.000000	14.454200	155.000000
75%	3.000000	1.000000	39.000000	1.000000	0.000000	31.275000	256.000000
max	3.000000	1.000000	80.000000	8.000000	9.000000	512.329200	328.000000

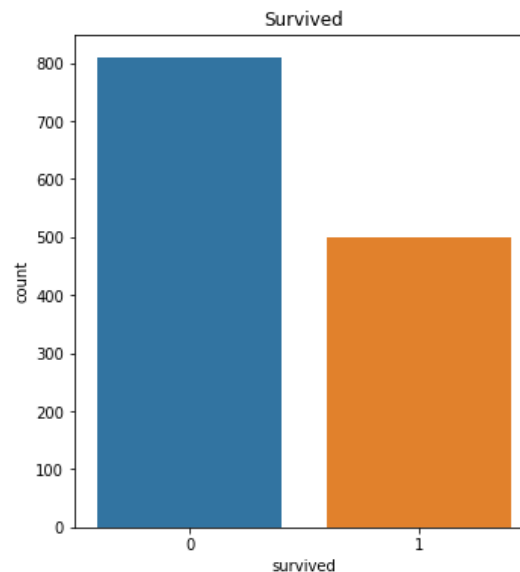
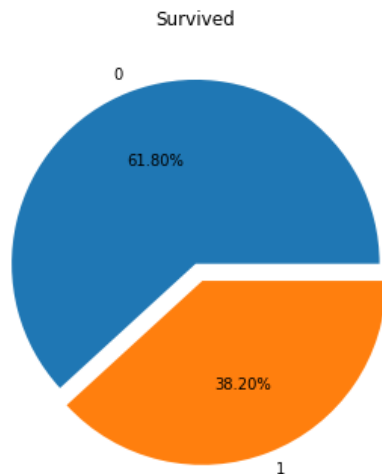
결정나무(Decision trees)

1. 파악 : 시각화를 통한 데이터 분석 - 사망률/생존률

```
f,ax=plt.subplots(1,2,figsize=(12,6))

raw_data['survived'].value_counts().plot.pie(explode=[0,0.1],autopct='%1.2f%%',ax=ax[0])
ax[0].set_title('Survived')
ax[0].set_ylabel('')

sns.countplot('survived',data=raw_data,ax=ax[1])
ax[1].set_title('Survived')
plt.show()
```



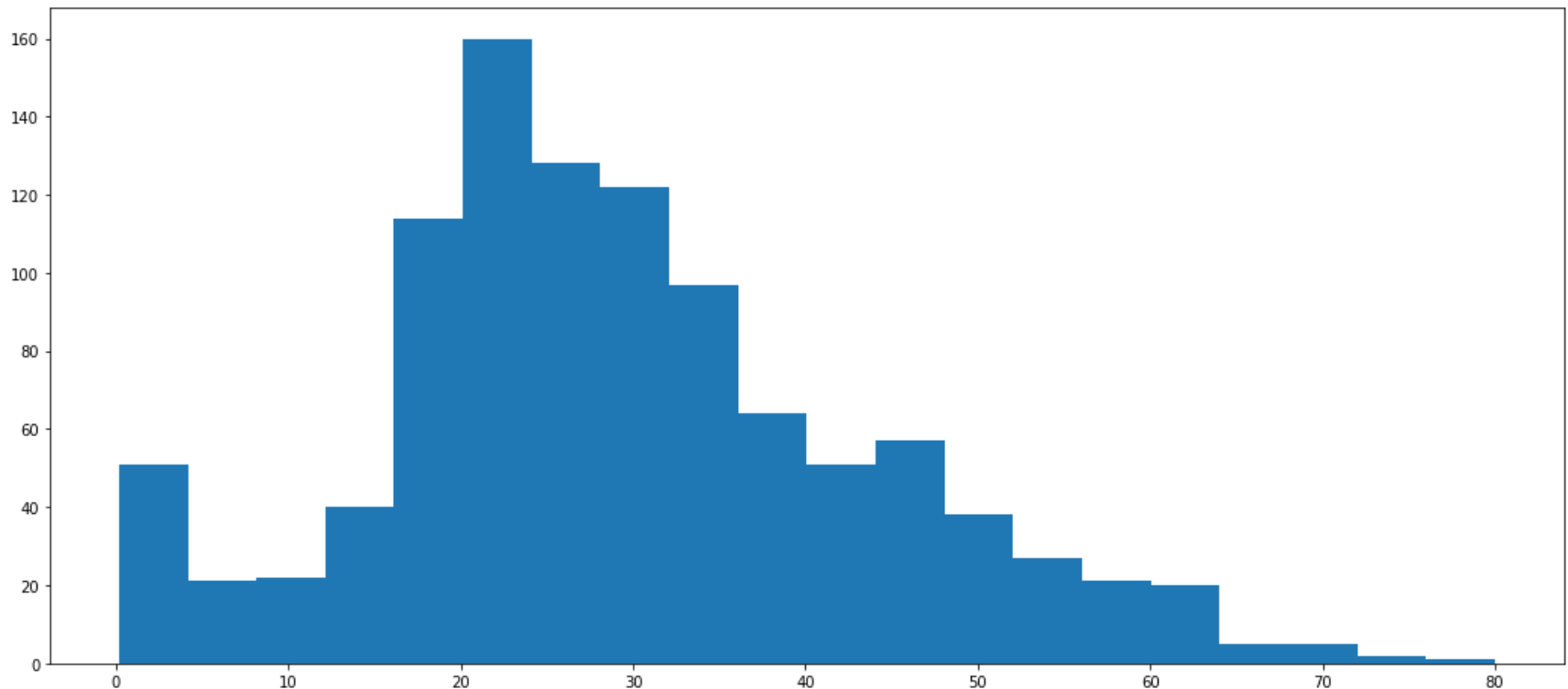
◆ 범주화 : 사망과 생존

- 0: 사망
- 1: 생존

결정나무(Decision trees)

1. 파악 : 시각화를 통한 데이터 분석 - 탑승인원의 연령분포

```
raw_data['age'].hist(bins=20, figsize=(18,8), grid=False);
```



결정나무(Decision trees)

1. 파악 : 시각화를 통한 데이터 분석 – 선실 등급별 분포

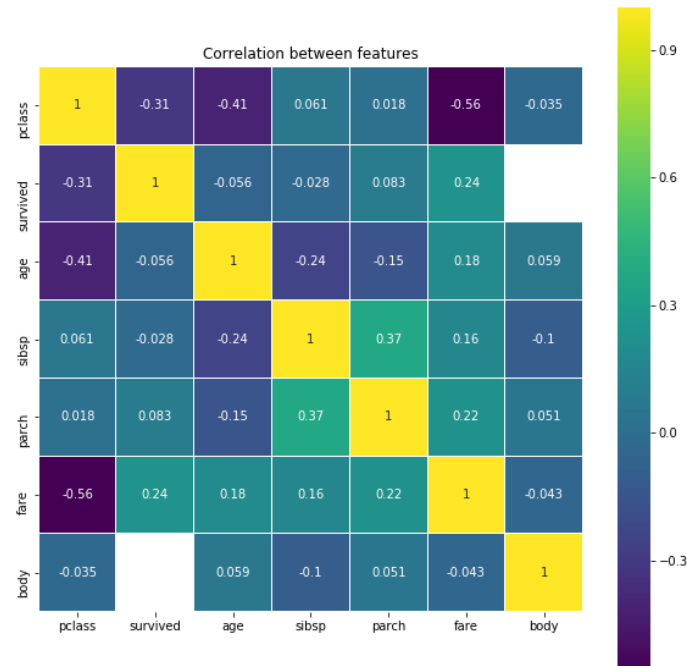
```
raw_data.groupby('pclass').mean()
```

	survived	age	sibsp	parch	fare	body
pclass						
1	0.619195	39.159918	0.436533	0.365325	87.508992	162.828571
2	0.429603	29.506705	0.393502	0.368231	21.179196	167.387097
3	0.255289	24.816367	0.568406	0.400564	13.302889	155.818182

결정나무(Decision trees)

2. 통계분석 : 변수간 상관계수 분석

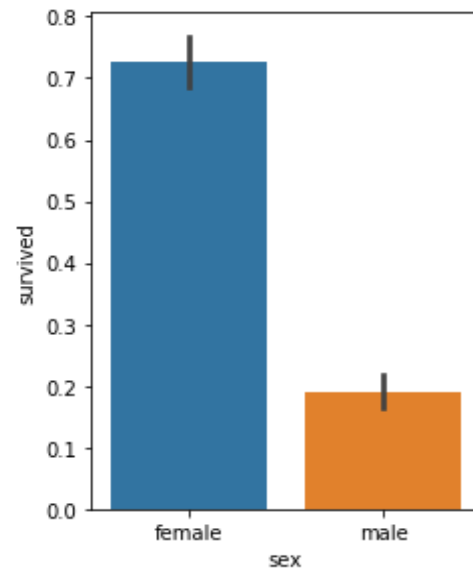
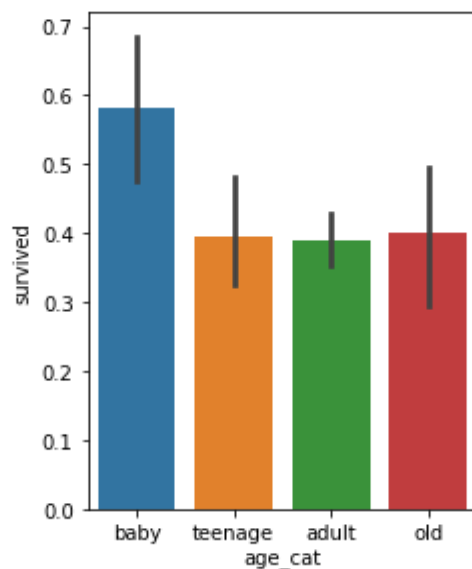
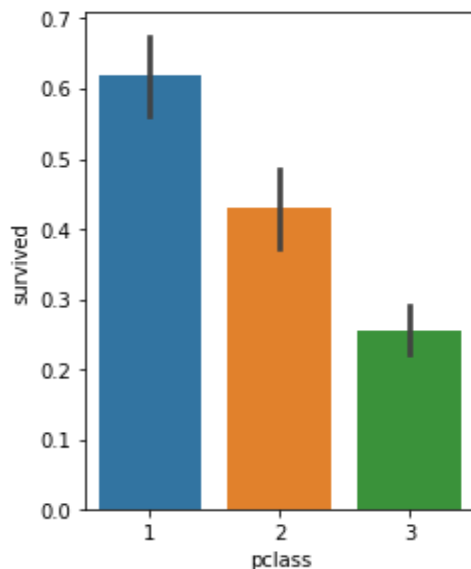
```
plt.figure(figsize=(10, 10))
sns.heatmap(raw_data.corr(), linewidths=0.01, square=True,
            annot=True, cmap=plt.cm.viridis, linecolor="white")
plt.title('Correlation between features')
plt.show()
```



결정나무(Decision trees)

2. 통계분석 : 동일 변수내 변수값간 상관관계 분석

```
raw_data['age_cat'] = pd.cut(raw_data['age'], bins=[0, 10, 20, 50, 100],  
                             include_lowest=True, labels=['baby', 'teenage', 'adult', 'old'])  
plt.figure(figsize=[12,4])  
plt.subplot(131)  
sns.barplot('pclass', 'survived', data=raw_data)  
plt.subplot(132)  
sns.barplot('age_cat', 'survived', data=raw_data)  
plt.subplot(133)  
sns.barplot('sex', 'survived', data=raw_data)  
plt.subplots_adjust(top=1, bottom=0.1, left=0.10, right=1, hspace=0.5, wspace=0.5)  
plt.show()
```

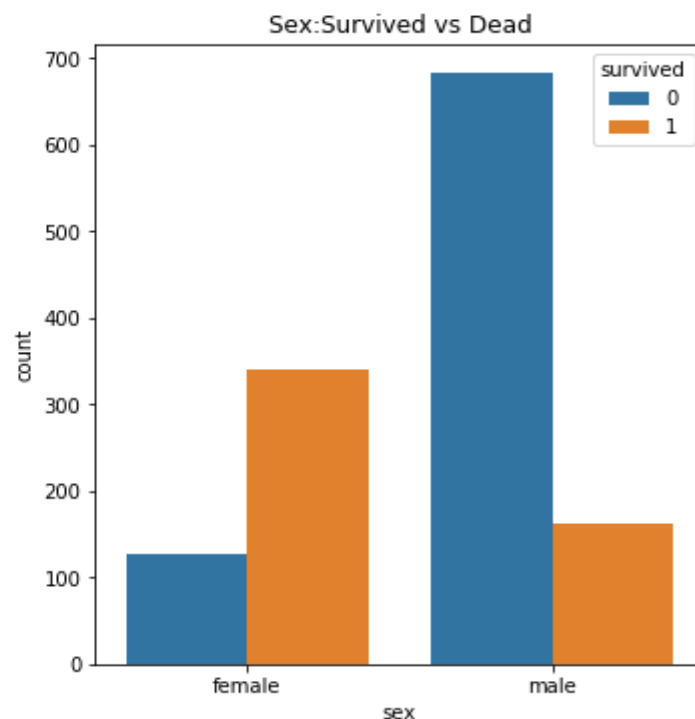
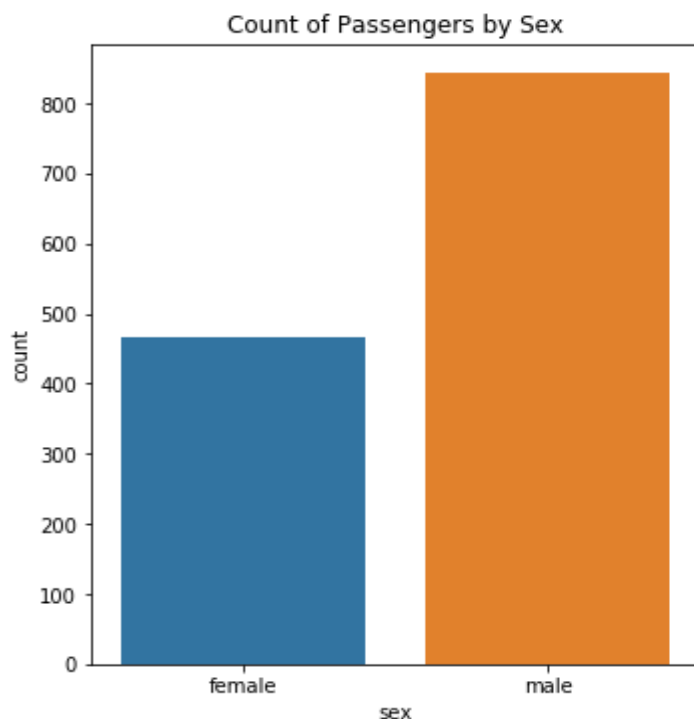


결정나무(Decision trees)

2. 통계분석 : 성별 생존률 분포 파악

```
f,ax=plt.subplots(1,2,figsize=(12,6))
sns.countplot('sex',data=raw_data, ax=ax[0])
ax[0].set_title('Count of Passengers by Sex')

sns.countplot('sex',hue='survived',data=raw_data, ax=ax[1])
ax[1].set_title('Sex:Survived vs Dead')
plt.show()
```



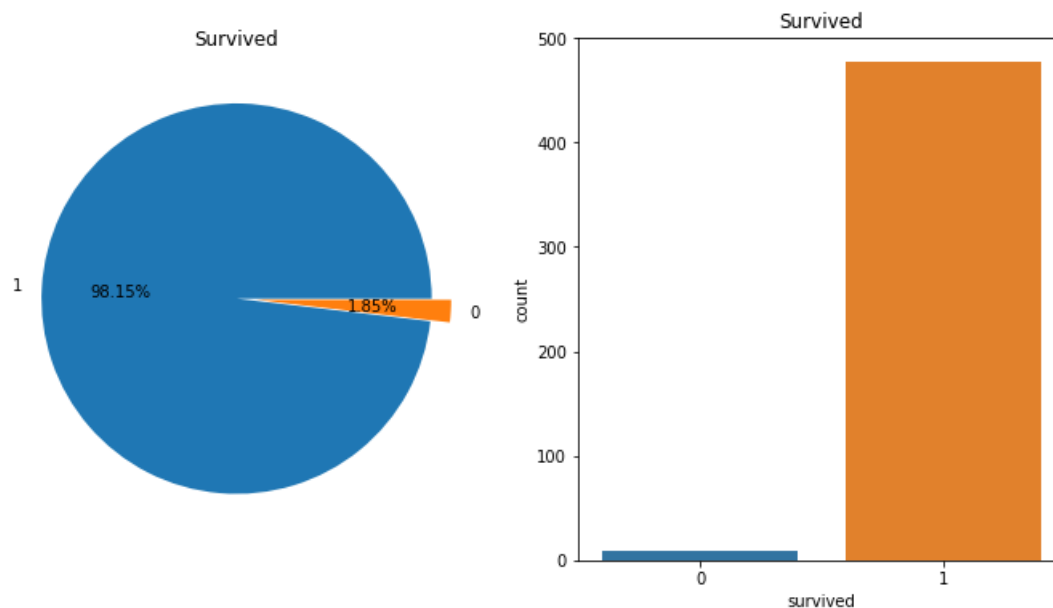
결정나무(Decision trees)

2. 통계분석 : 보트 탑승인원의 생존률 분포

```
boat_survivors = raw_data[raw_data['boat'].notnull()]
f, ax = plt.subplots(1, 2, figsize=(12, 6))

boat_survivors['survived'].value_counts().plot.pie(explode=[0, 0.1], autopct='%1.2f%%', ax=ax[0])
ax[0].set_title('Survived')
ax[0].set_ylabel('')

sns.countplot('survived', data=boat_survivors, ax=ax[1])
ax[1].set_title('Survived')
plt.show()
```



결정나무(Decision trees)

3. 결정나무 모델 구축 : 결측치 제거

```
tmp = []
for each in raw_data['sex']:
    if each == 'female':
        tmp.append(1)
    elif each == 'male':
        tmp.append(0)
    else:
        tmp.append(np.nan)

raw_data['sex'] = tmp

raw_data['survived'] = raw_data['survived'].astype('int')
raw_data['pclass'] = raw_data['pclass'].astype('float')
raw_data['sex'] = raw_data['sex'].astype('float')
raw_data['sibsp'] = raw_data['sibsp'].astype('float')
raw_data['parch'] = raw_data['parch'].astype('float')
raw_data['fare'] = raw_data['fare'].astype('float')

raw_data = raw_data[raw_data['age'].notnull()]
raw_data = raw_data[raw_data['sibsp'].notnull()]
raw_data = raw_data[raw_data['parch'].notnull()]
raw_data = raw_data[raw_data['fare'].notnull()]

raw_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1045 entries, 0 to 1308
Data columns (total 15 columns):
pclass      1045 non-null float64
survived     1045 non-null int64
name        1045 non-null object
sex         1045 non-null float64
age         1045 non-null float64
sibsp       1045 non-null float64
parch       1045 non-null float64
ticket      1045 non-null object
fare        1045 non-null float64
cabin       272 non-null object
embarked     1043 non-null object
boat        417 non-null object
body        119 non-null float64
home.dest    685 non-null object
age_cat      1045 non-null category
dtypes: category(1), float64(7), int64(1), object(6)
memory usage: 123.7+ KB
```

결정나무(Decision trees)

3. 결정나무 모델 구축 : 목표변수와 관계가 있는 설명변수들의 선택

```
train_pre = raw_data[['pclass', 'sex', 'age', 'sibsp', 'parch', 'fare']]
train_pre.head()
```

	pclass	sex	age	sibsp	parch	fare
0	1.0	1.0	29.0000	0.0	0.0	211.3375
1	1.0	0.0	0.9167	1.0	2.0	151.5500
2	1.0	1.0	2.0000	1.0	2.0	151.5500
3	1.0	0.0	30.0000	1.0	2.0	151.5500
4	1.0	1.0	25.0000	1.0	2.0	151.5500

◆ 선택한 설명변수

- pclass : 객실 등급
- sex : 성별
- age : 나이
- sibsp : 형제 혹은 부부의 수
- parch : 부모, 혹은 자녀의 수
- fare : 지불한 운임

결정나무(Decision trees)

3. 결정나무 모델 구축 : Train / Test 데이터 나누기

- Random Sampling
- Train : 80%
- Test : 20%

```
#Test Train split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train_pre, raw_data[['survived']], test_size=0.2, random_state=5)
```

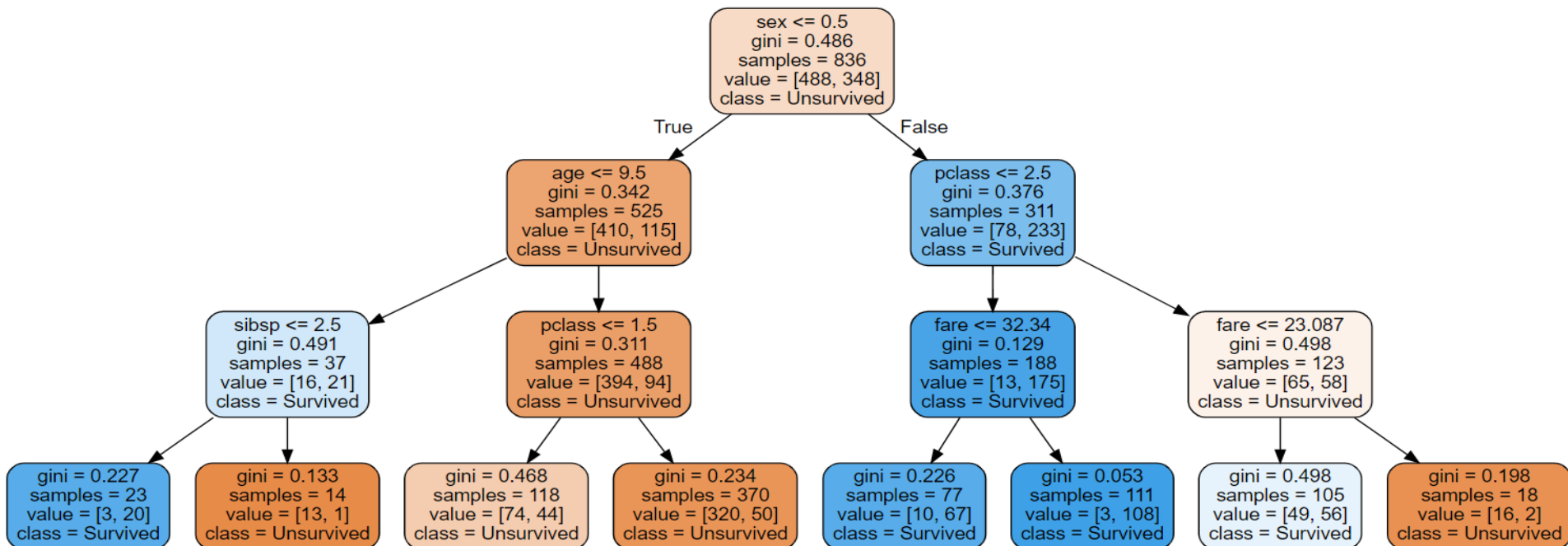

결정나무(Decision trees)

4. 결정나무 모델 생성 : 모델1 생성

- Depth : 3 Layer
- Criterion : Gini
- Splitter : Best

```
from sklearn.tree import DecisionTreeClassifier
tree_clf = DecisionTreeClassifier(max_depth=3, random_state=5)
tree_clf.fit(X_train, y_train)
print('Score: {}'.format(tree_clf.score(X_train, y_train)))
```

Score: 0.80622009569378



결정나무(Decision trees)

4. 결정나무 모델 생성 : 모델2 생성

- Depth : 3 Layer
- Criterion : entropy
- Splitter : random

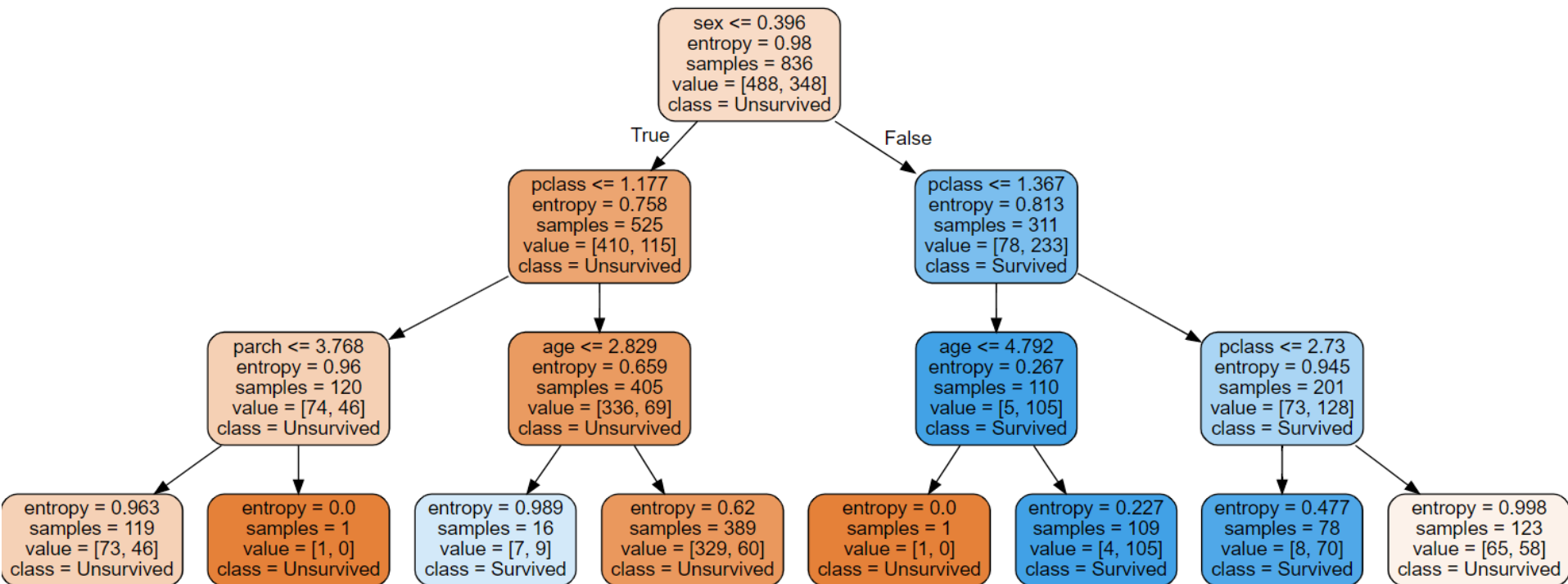
```
from sklearn.tree import DecisionTreeClassifier
tree_clf = DecisionTreeClassifier(criterion='entropy', max_depth=3, min_samples_split=3, splitter='random', random_state=5)
tree_clf.fit(X_train, y_train)
print('Score: {}'.format(tree_clf.score(X_train, y_train)))
```

Score: 0.7811004784688995

결정나무(Decision trees)

4. 결정나무 모델 생성 : 모델2 생성

- Depth : 3 Layer
- Criterion : entropy
- Splitter : random



결정나무(Decision trees)

5. 결정나무 모델 분류 및 예측 : 생성된 모델의 Test 데이터에 대한 정확도 확인

➤ 모델1 결과

```
from sklearn.metrics import accuracy_score  
  
y_pred = tree_clf.predict(X_test)  
print("Test Accuracy is ", accuracy_score(y_test, y_pred)*100)
```

Test Accuracy is 85.16746411483254

➤ 모델2 결과

```
from sklearn.metrics import accuracy_score  
  
y_pred = tree_clf.predict(X_test)  
print("Test Accuracy is ", accuracy_score(y_test, y_pred)*100)
```

Test Accuracy is 82.77511961722487

결정나무(Decision trees)

5. 결정나무 모델 분류 및 예측 : 생성된 모델로 특정 인원의 사망 생존률 예측

```
# pclass, sex, age, sibsp, parch, fare
dicaprio = [3., 0., 19., 0., 0., 5.]
winslet = [1., 1., 17., 1., 2., 100.]

def isSurvived(name, person):
    isSurvive = 'not survived' if tree_clf.predict([person])[0] == 0 else 'survived'
    print(name, ' is ', isSurvive,
          ' --> ', max(tree_clf.predict_proba([person])[0]))

isSurvived('Dicaprio', dicaprio)
isSurvived('Winslet', winslet)
```

- **모델1 결과** Dicaprio is not survived --> 0.8648648648648649
Winslet is survived --> 0.972972972972973
- **모델2 결과** Dicaprio is not survived --> 0.8457583547557841
Winslet is survived --> 0.963302752293578

결정나무(Decision trees)

5. 결정나무 모델 분류 및 예측 : 생성된 모델로 특정 인원의 사망 생존률 예측

```
# pclass, sex, age, sibsp, parch, fare
dicaprio = [3., 0., 19., 0., 0., 5.]
winslet = [1., 1., 17., 1., 2., 100.]

def isSurvived(name, person):
    isSurvive = 'not survived' if tree_clf.predict([person])[0] == 0 else 'survived'
    print(name, ' is ', isSurvive,
          ' --> ', max(tree_clf.predict_proba([person])[0]))

isSurvived('Dicaprio', dicaprio)
isSurvived('Winslet', winslet)
```

- **모델1 결과** Dicaprio is not survived --> 0.8648648648648649
Winslet is survived --> 0.972972972972973
- **모델2 결과** Dicaprio is not survived --> 0.8457583547557841
Winslet is survived --> 0.963302752293578