타이타닉(Titanic)

데이터 입력

```python
import seaborn as sns
titanic = sns.load_dataset("titanic")

print(titanic.shape)
print(titanic.info())
```

** 데이터 분석(예측)
survived: 생존여부 (alive)
pclass: 객실등급 (class, fare)
sex: 성별 (who)
adult_male: 성인남성 여부
sibsp: 형제자매 및 배우자 수 (alone)
parch: 부모 및 자식 수 (alone)
embarked: 탑승정보 (deck, embark_town)

** 정확한 데이터 분석 필요

```
(891, 15)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   survived     891 non-null    int64
 1   pclass       891 non-null    int64
 2   sex          891 non-null    object
 3   age          714 non-null    float64
 4   sibsp        891 non-null    int64
 5   parch        891 non-null    int64
 6   fare         891 non-null    float64
 7   embarked     889 non-null    object
 8   class        891 non-null    category
 9   who          891 non-null    object
 10  adult_male   891 non-null    bool
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object
 13  alive        891 non-null    object
 14  alone        891 non-null    bool
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.6+ KB
```

https://seaborn.pydata.org/generated/seaborn.load_dataset.html#seaborn.load_dataset

seaborn.countplot

`seaborn.countplot` (*, *x=None, y=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None, palette=None, saturation=0.75, dodge=True, ax=None, **kwargs*)

Show the counts of observations in each categorical bin using bars.

x, y, hue : *names of variables in* `data` *or vector data, optional*

Inputs for plotting long-form data. See examples for interpretation.

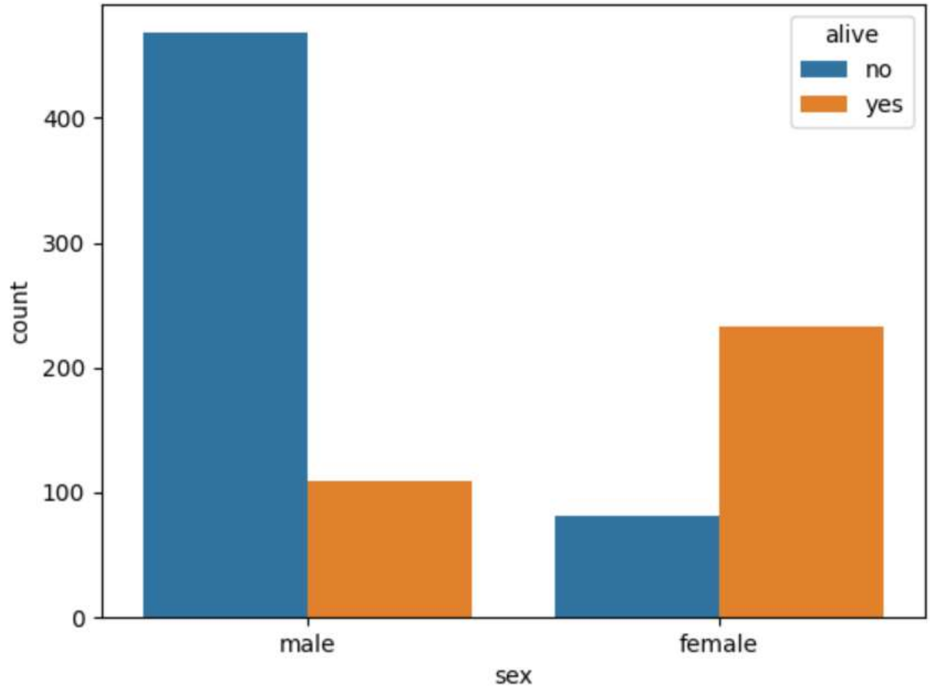
data : *DataFrame, array, or list of arrays, optional*

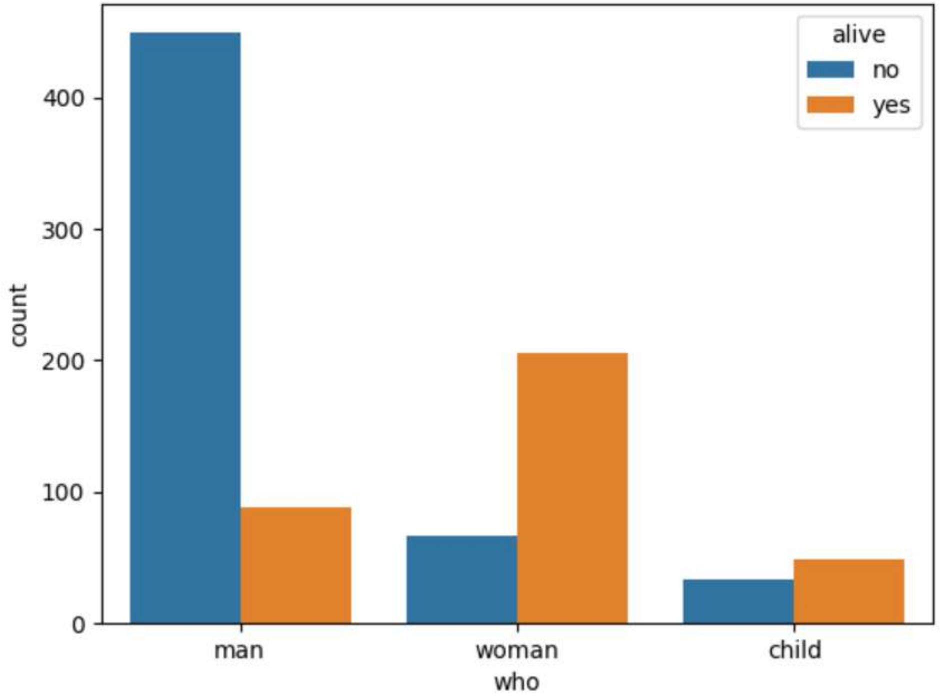
Dataset for plotting. If `x` and `y` are absent, this is interpreted as wide-form. Otherwise it is expected to be long-form.

A special case for the bar plot is when you want to show the number of observations in each category rather than computing a statistic for a second variable. This is similar to a histogram over a categorical, rather than quantitative, variable. In seaborn, it's easy to do so with the `countplot()` function:

https://seaborn.pydata.org/generated/seaborn.countplot.html#seaborn.countplot

데이터 분석(countplot)






```
import matplotlib.pyplot as plt
sns.countplot(data = titanic, x = "sex", hue = "alive")
plt.show()
```

seaborn.catplot

`seaborn.catplot` (**, x=None, y=None, hue=None, data=None, row=None, col=None, col_wrap=None, estimator=<function mean at 0x7ff320f315e0>, ci=95, n_boot=1000, units=None, seed=None, order=None, hue_order=None, row_order=None, col_order=None, kind='strip', height=5, aspect=1, orient=None, color=None, palette=None, legend=True, legend_out=True, sharex=True, sharey=True, margin_titles=False, facet_kws=None, **kwargs*)

Figure-level interface for drawing categorical plots onto a FacetGrid.

x, y, hue : *names of variables in* `data`

Inputs for plotting long-form data. See examples for interpretation.

data : *DataFrame*

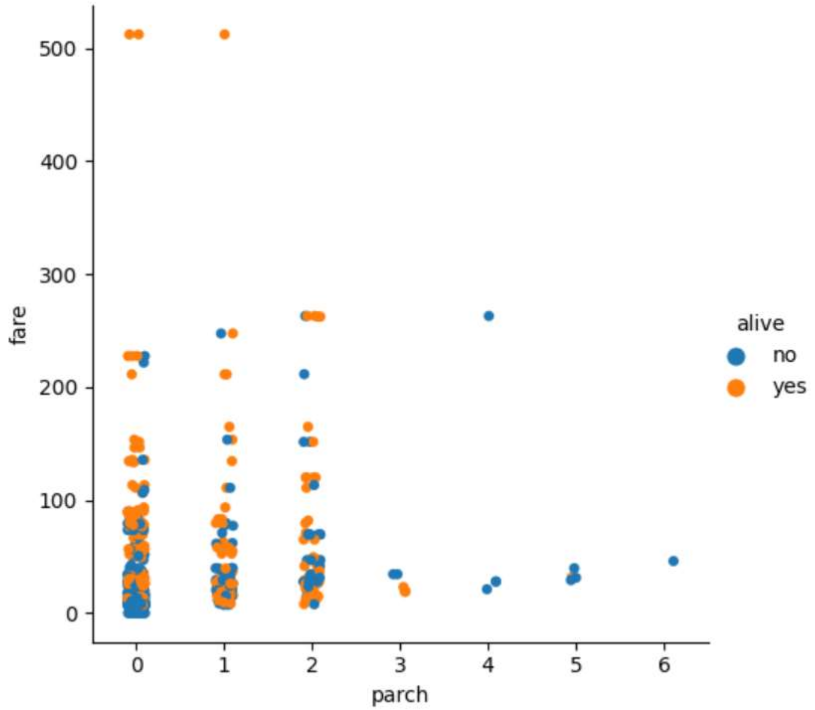
Long-form (tidy) dataset for plotting. Each column should correspond to a variable, and each row should correspond to an observation.

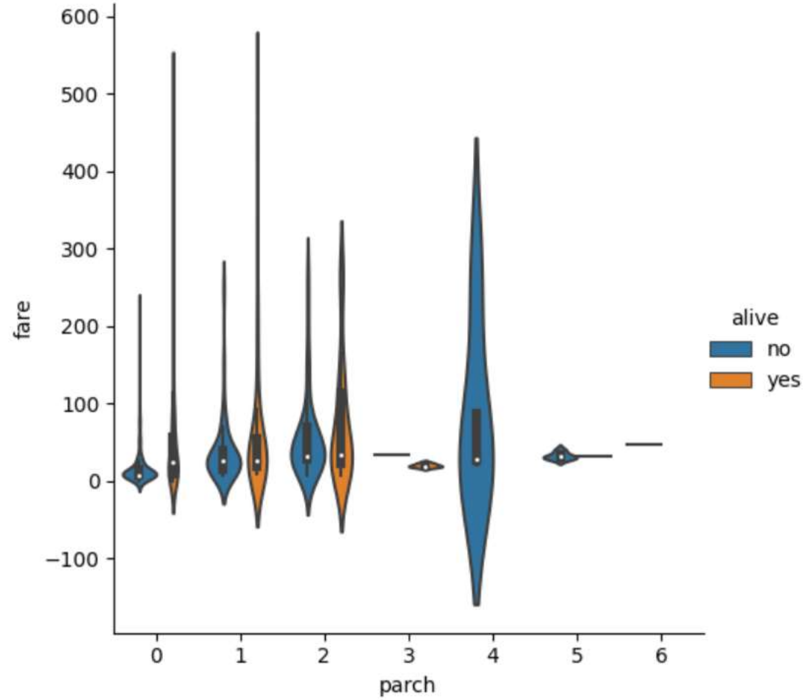
kind : *str, optional*

The kind of plot to draw, corresponds to the name of a categorical axes-level plotting function. Options are: "strip", "swarm", "box", "violin", "boxen", "point", "bar", or "count".

https://seaborn.pydata.org/generated/seaborn.countplot.html#seaborn.countplot

데이터 분석(catplot)






```
import matplotlib.pyplot as plt
sns.catplot(data = titanic, x = "parch", y = "fare", hue = "alive", kind = "violin")
plt.show()
```

데이터 실수화, 정제, 축소

	survived	pclass	sex	age	...	deck	embark_town	alive	alone
0	0	3	male	22.0	...	NaN	Southampton	no	False
1	1	1	female	38.0	...	C	Cherbourg	yes	False
2	1	3	female	26.0	...	NaN	Southampton	yes	True
3	1	1	female	35.0	...	C	Southampton	yes	False
4	0	3	male	35.0	...	NaN	Southampton	no	True
..	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	...	NaN	Southampton	no	True
887	1	1	female	19.0	...	B	Southampton	yes	True
888	0	3	female	NaN	...	NaN	Southampton	no	False
889	1	1	male	26.0	...	C	Cherbourg	yes	True
890	0	3	male	32.0	...	NaN	Queenstown	no	True

```python
data = titanic[["sex", "age", "fare"]].copy()
t = titanic["survived"]
# print(data.tail())
data["age"].fillna(30, inplace = True)
data["sex"].replace("male", 1, inplace = True)
data["sex"].replace("female", 0, inplace = True)
# print(data.tail())
```

	sex	age	fare
886	male	27.0	13.00
887	female	19.0	30.00
888	female	$\boxed{\text{NaN}}$	23.45
889	male	26.0	30.00
890	male	32.0	7.75

	sex	age	fare
886	1	27.0	13.00
887	0	19.0	30.00
888	0	$\boxed{30.0}$	23.45
889	1	26.0	30.00
890	1	32.0	7.75

분류 정확도

구현

```
print("Train-Eval:", model.score(train_data, train_target))
print("Test-Eval :", model.score(test_data, test_target))
```

```
Train-Eval: 0.7881219903691814
Test-Eval : 0.7723880597014925
[[135  30]
 [ 31  72]]
```

로지스틱회귀

```
Train-Eval: 0.9791332263242376
Test-Eval : 0.7686567164179104
[[133  32]
 [ 30  73]]
```

의사결정나무

```
Train-Eval: 0.8073836276083467
Test-Eval : 0.7761194029850746
[[136  29]
 [ 31  72]]
```

의사결정나무(max_depth=3)

데이터(특징)별 성능

```
data = titanic[["sex", "age", "fare"]].copy()
```

```
Train-Eval: 0.8073836276083467
Test-Eval : 0.7761194029850746
```

```
data = titanic[["sex", "age", "sibsp"]].copy()
```

```
Train-Eval: 0.8186195826645265
Test-Eval : 0.7947761194029851
```

```
data = titanic[["sex", "age", "sibsp", "adult_male"]].copy()
```

```
Train-Eval: 0.8234349919743178
Test-Eval : 0.8097014925373134
```

```
data = titanic[["sex", "age", "sibsp", "adult_male", "parch"]].copy()
```

```
Train-Eval: 0.8314606741573034
Test-Eval : 0.8134328358208955
```

데이터 분석
중요

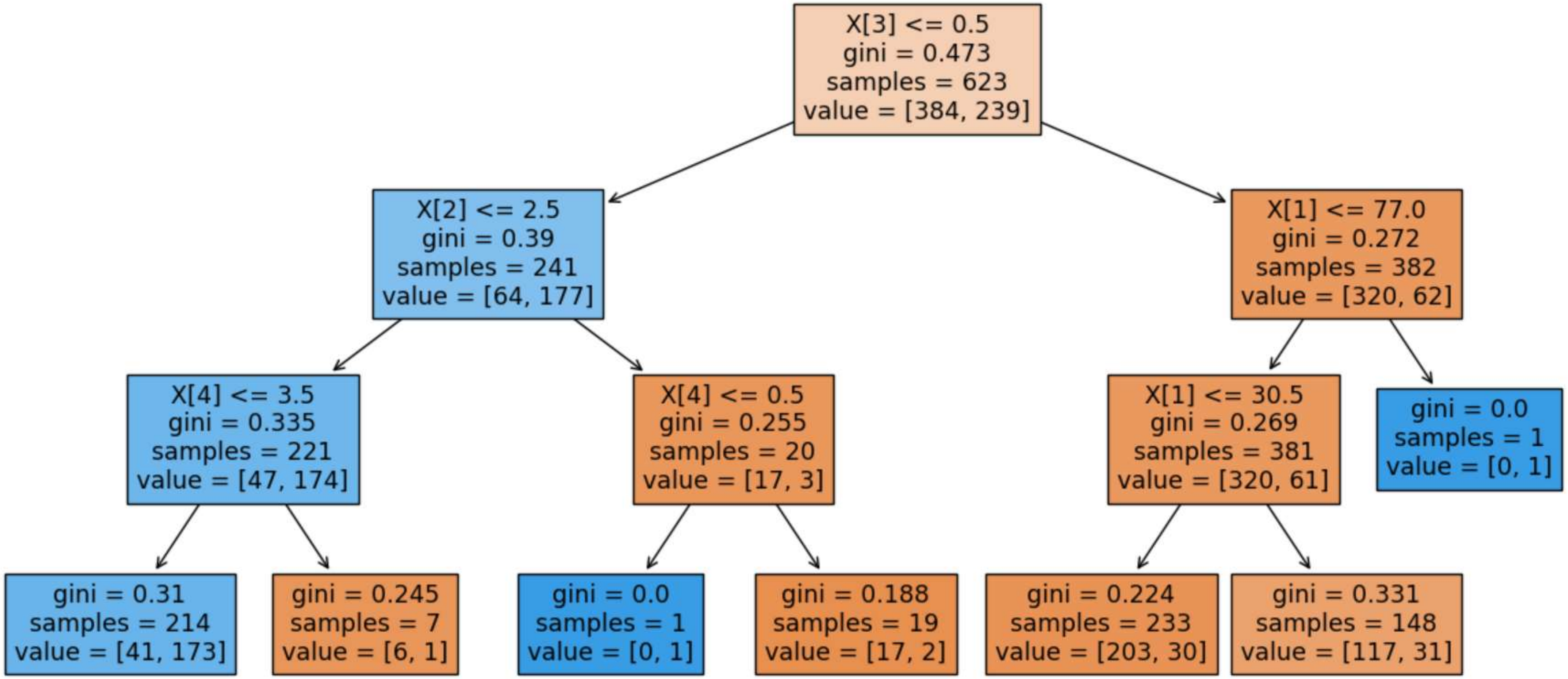
의사결정나무(max_depth=3)

의사결정나무

```
data = titanic[["sex", "age", "sibsp", "adult_male", "parch"]].copy()
```

```
[0.         0.02124066 0.12237088 0.79457493 0.06181353]
```

데이터 중요도

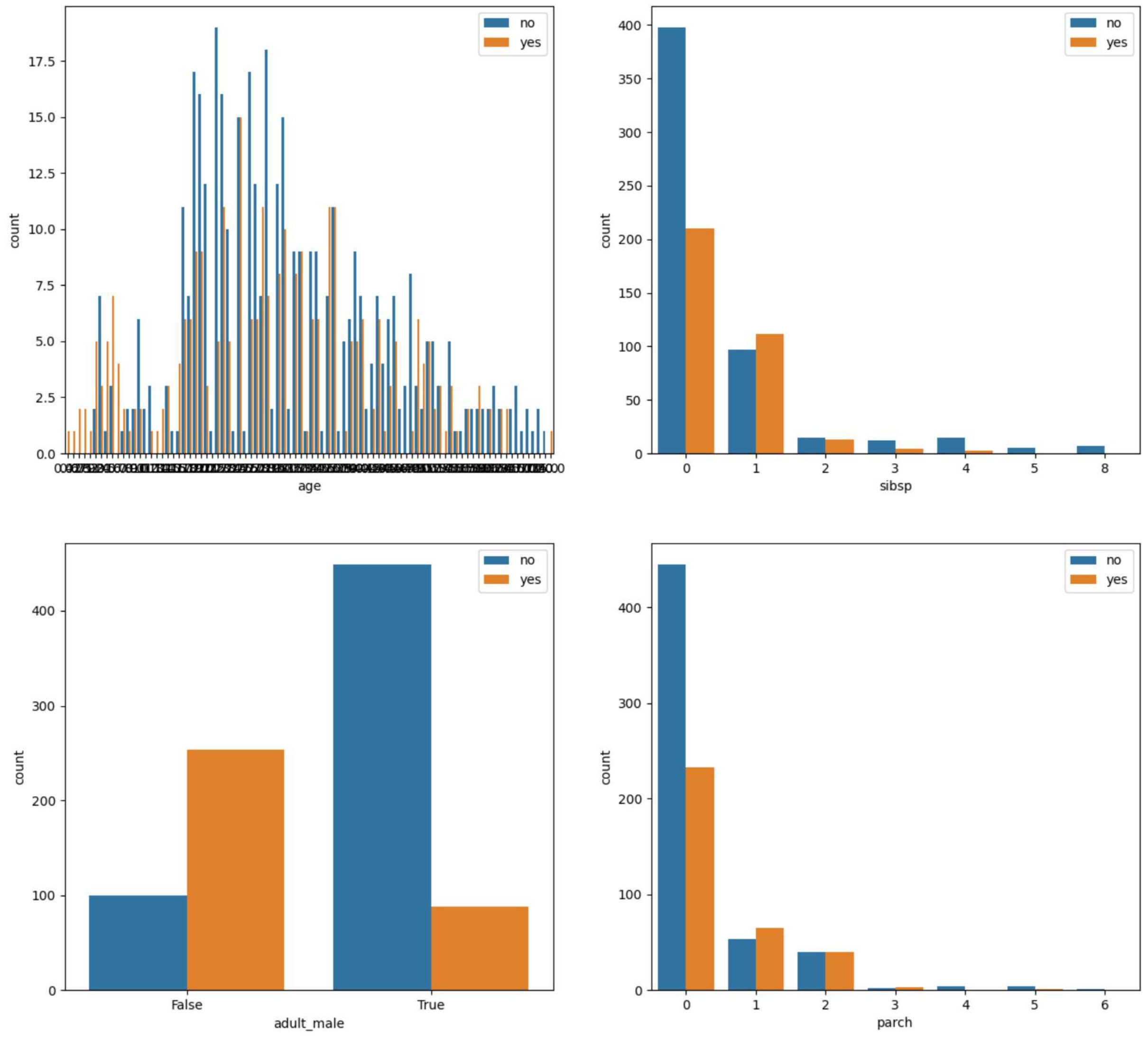


데이터 중요도

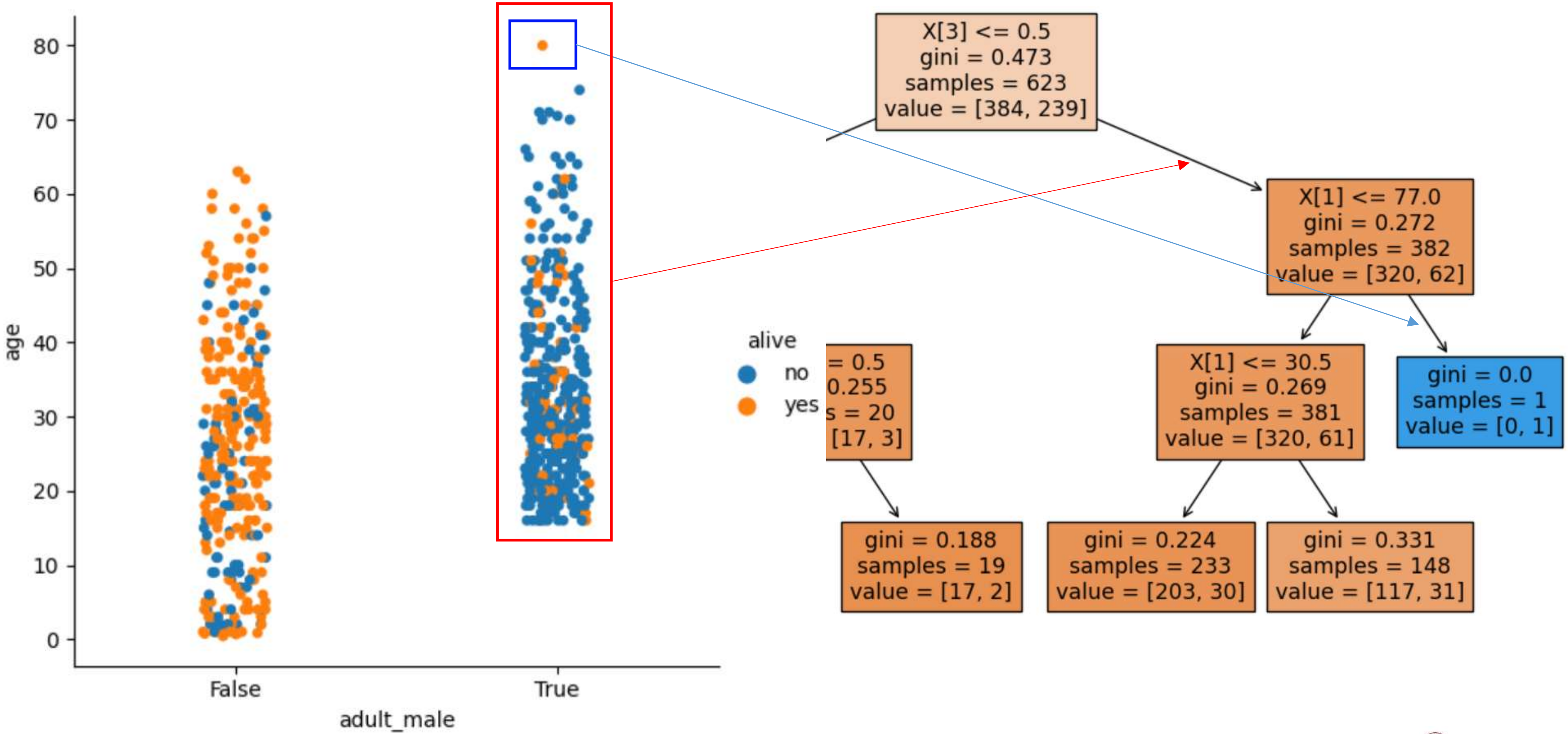
```
import matplotlib.pyplot as plt
```

구현

```
plt.show()
```



의사결정나무 분류과정



age
0
10
20
30
40
50
60
70
80
False
True
adult_male
alive
no
yes
X[3] <= 0.5
gini = 0.473
samples = 623
value = [384, 239]
X[1] <= 77.0
gini = 0.272
samples = 382
value = [320, 62]
= 0.5
0.255
s = 20
[17, 3]
X[1] <= 30.5
gini = 0.269
samples = 381
value = [320, 61]
gini = 0.0
samples = 1
value = [0, 1]
gini = 0.188
samples = 19
value = [17, 2]
gini = 0.224
samples = 233
value = [203, 30]
gini = 0.331
samples = 148
value = [117, 31]

참고자료

- 지능기전공학부 최유경 교수님 자료, https://github.com/sejongresearch/2021.MachineLearning
- 코랩(Colab), https://colab.research.google.com/
- 파이썬(Python), https://www.python.org/doc/
- 사이킷런(sckit-learn), https://scikit-learn.org/stable/index.html
- 판다스(pandas), https://pandas.pydata.org/
- 맷플롯립(matplotlib), https://matplotlib.org/
- 씨본(seaborn), https://seaborn.pydata.org/
- 캐글(Kaggle), https://www.kaggle.com/
- 넘파이(numpy), https://numpy.org/doc/stable/
- 스택오퍼플러우(stackoverflow), https://stackoverflow.com/