

010. 타이타닉 생존자 예측

DataScience

Exported on 11/09/2020

Table of Contents

1	타이타닉 생존자 예측	4
2	디카프리오 형님은 정말 살 수 없었나?	5
2.1	타이타닉?	5
2.2	1910년대 당시 최대 여객선 - 타이타닉	5
2.3	영국에서 미국 뉴욕으로 가던 국제선	6
2.4	당시 최첨단 시설	6
2.5	데이터	7
2.6	컬럼의 의미	7
3	데이터 탐색적 분석 - EDA	8
3.1	그전에 하나더 인스톨	8
3.2	데이터 읽기	8
3.3	생존 상황	8
3.4	38.2%의 생존률	9
3.5	성별에 따른 생존 상황은?	9
3.6	남성의 생존 가능성이 높다	10
3.7	경제력 대비 생존률	10
3.8	선실 등급별 성별 상황	10
3.9	3등실에는 남성이 많았다 - 특히 20대 남성	11
3.10	그러면 나이별 승객 현황도 확인해보자	11
3.11	아이들과 20~30대가 많았다	12
3.12	그러면 등실별 생존률을 연령별로 관찰해 보자	12
3.13	확실히 선실 등급이 높으면 생존률이 높은듯 하다	13
3.14	나이를 5단계로 정리하기	14
3.15	나이, 성별, 등급별 생존자 수를 한번에 파악할 수 있을까?	14
3.16	어리고, 여성이고, 1등실일 수록 생존하기 유리했을까?	15
3.17	남/여 나이별 생존 상황을 보다 더 들여다 보자	15
3.18	이 결과는 어떤 것을 의미할까	16
3.19	탑승객의 이름에서 신분을 알 수 있다	16

3.20 정규식을 이용해서 문장 사이의 신분에 대한 정보를 얻을 수 있다	17
3.21 살짝 응용하면	17
3.22 사회적 신분만 얻을 수 있다	18
3.23 성별별로 본 귀족	18
3.24 사회적 신분을 조금 더 정리하자	19
3.25 이 결과는 또 어떻게 해석할 수 있을까	19
3.26 타이타닉 배의 선장	20
3.27 타이타닉 빙산과 충돌	20
3.28 타이타닉의 탈출 원칙	21
3.29 타이타닉의 최후	21
4 머신러닝을 이용한 생존자 예측	22
4.1 간단히 구조확인	22
4.2 머신러닝을 위해 해당컬럼을 숫자로 변경해야겠다	22
4.3 Label Encode를 사용하면 편하다	23
4.4 결측치는 어쩔 수 없이 포기하자	23
4.5 상관관계	24
4.6 먼저 특성을 선택하고, 데이터를 나누자	24
4.7 일단 DecisionTree	24
5 그런데 디카프리오 형님의 생존율은?	25
5.1 디카프리오를 특정짓자	25
5.2 디카프리오의 생존률	25
5.3 그럼 원슬릿은?	25

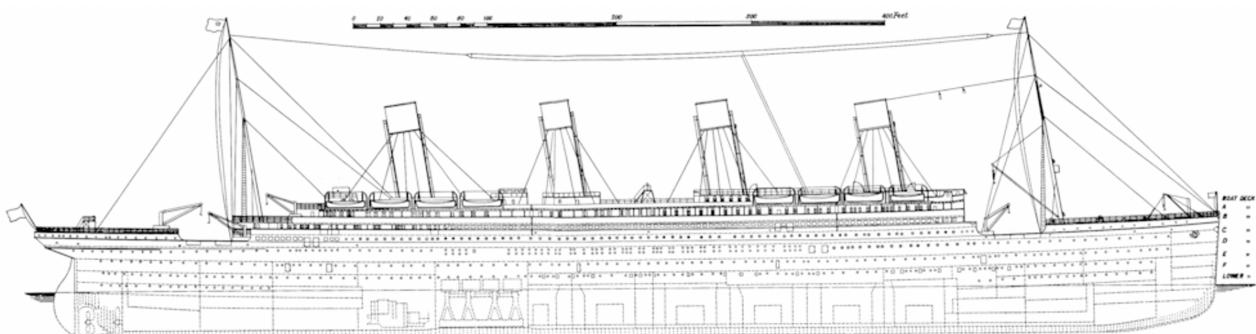
1 타이타닉 생존자 예측

2 디카프리오 형님은 정말 살 수 없었나?

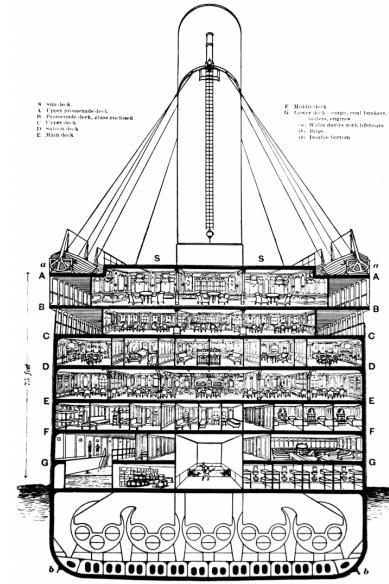
2.1 타이타닉?



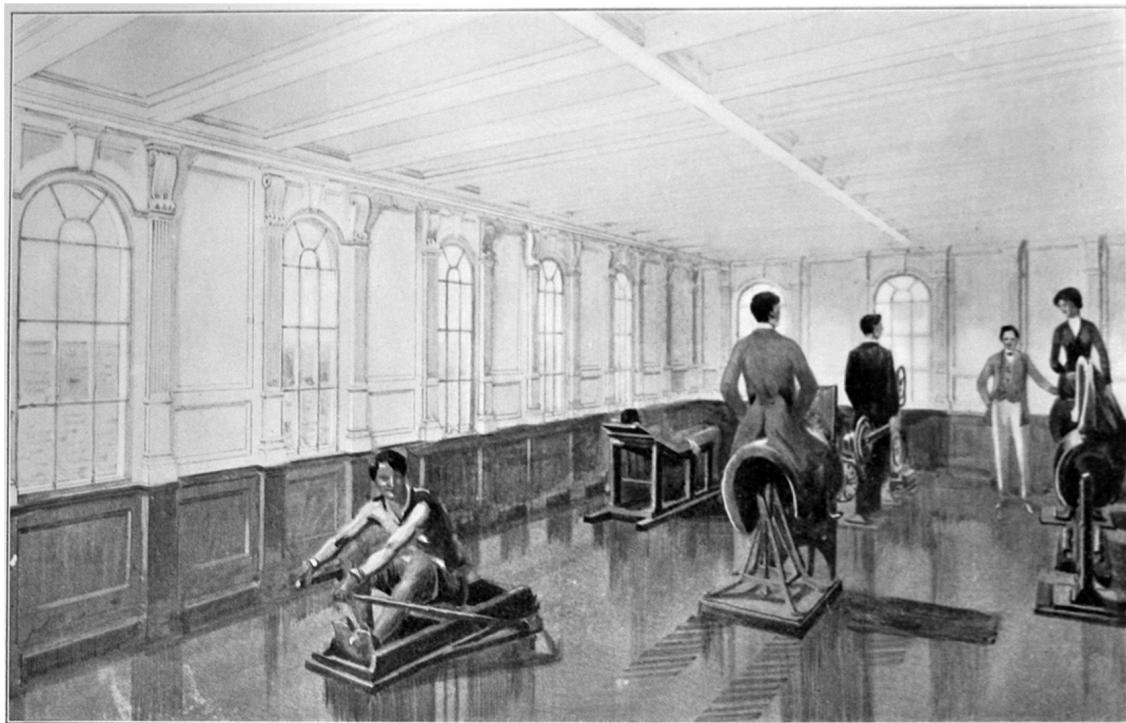
2.2 1910년대 당시 최대 여객선 - 타이타닉



2.3 영국에서 미국 뉴욕으로 가던 국제선



2.4 당시 최첨단 시설



2.5 데이터

PinkWink / ML_tutorial

Branch: master

ML_tutorial / dataset /

titanic.xls	titanic data	Latest commit 753628c 3 hours ago
..		
wine.csv	.	yesterday
winequality-red.csv	wine_data	yesterday
winequality-white.csv	wine_data	yesterday

2.6 컬럼의 의미

pclass	객실 등급
survived	생존 유무
sex	성별
age	나이
sibsp	형제 혹은 부부의 수
parch	부모 혹은 자녀의 수
fare	지불한 요금
boat	탈출을 했다면 탑승한 보트의 번호

3 데이터 탐색적 분석 - EDA

3.1 그전에 하나더 인스톨

- pip install plotly_express

3.2 데이터 읽기

```
import pandas as pd

titanic_url = 'https://raw.githubusercontent.com/PinkWink/ML_tutorial' + \
              '/master/dataset/titanic.xls'
titanic = pd.read_excel(titanic_url)
titanic.head()
```

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville ON
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville ON
			Allison.											

3.3 생존 상황

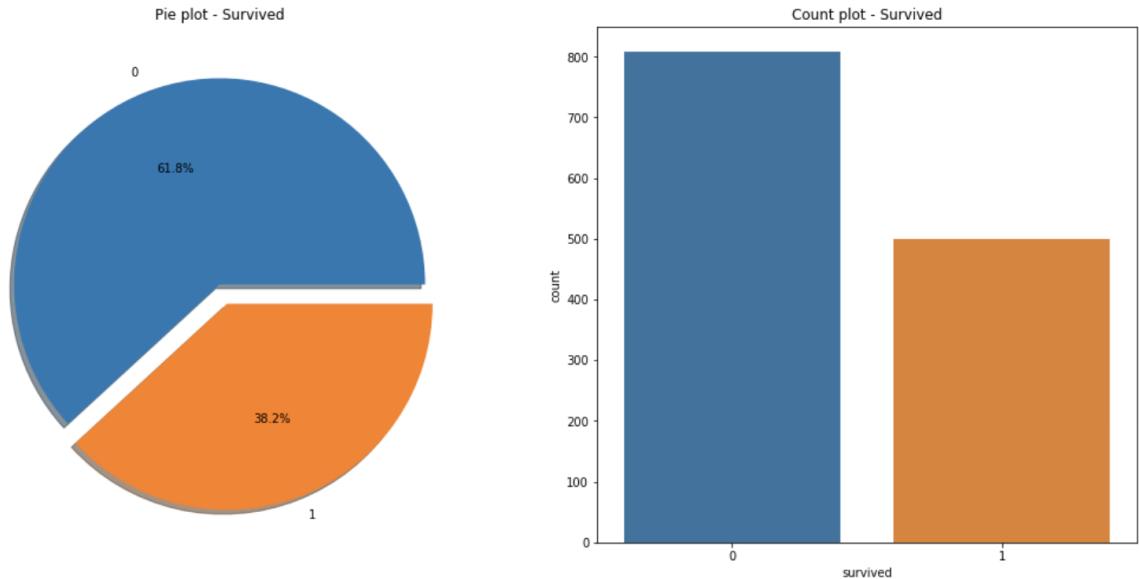
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

f, ax = plt.subplots(1, 2, figsize=(18, 8))

titanic['survived'].value_counts().plot.pie(explode=[0, 0.1],
                                             autopct='%.1f%%', ax=ax[0], shadow=True)
ax[0].set_title('Pie plot - Survived')
ax[0].set_ylabel('')
sns.countplot('survived', data=titanic, ax=ax[1])
ax[1].set_title('Count plot - Survived')

plt.show()
```

3.4 38.2%의 생존률



3.5 성별에 따른 생존 상황은?

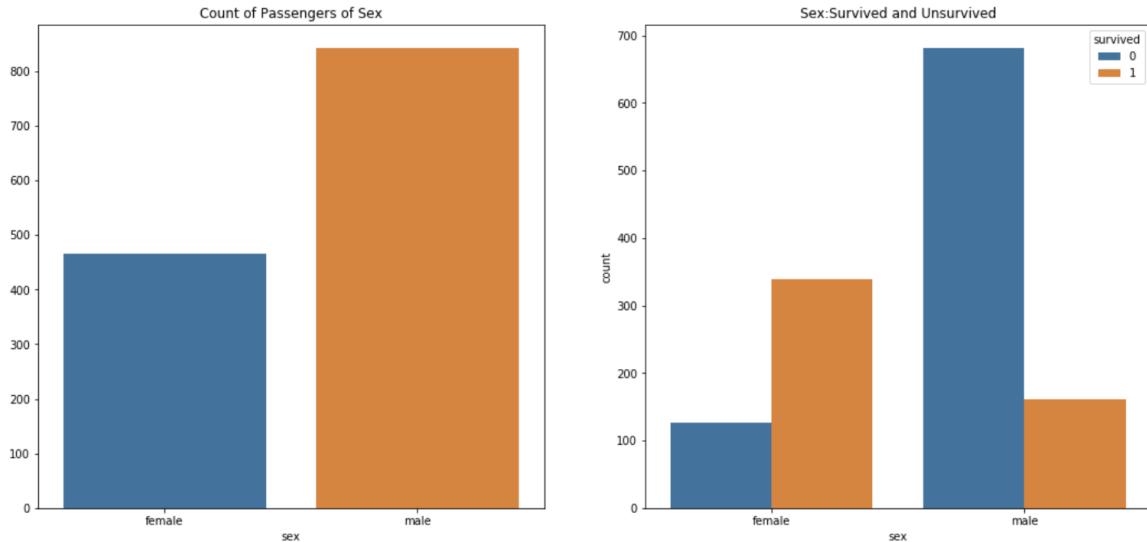
```
f, ax = plt.subplots(1, 2, figsize=(18, 8))

sns.countplot('sex', data=titanic, ax=ax[0])
ax[0].set_title('Count of Passengers of Sex')
ax[0].set_ylabel('')

sns.countplot('sex', hue='survived', data=titanic, ax=ax[1])
ax[1].set_title('Sex:Survived and Unsurvived')

plt.show()
```

3.6 남성의 생존 가능성이를 낮다



3.7 경제력 대비 생존률

```
pd.crosstab(titanic['pclass'], titanic['survived'], margins=True)
```

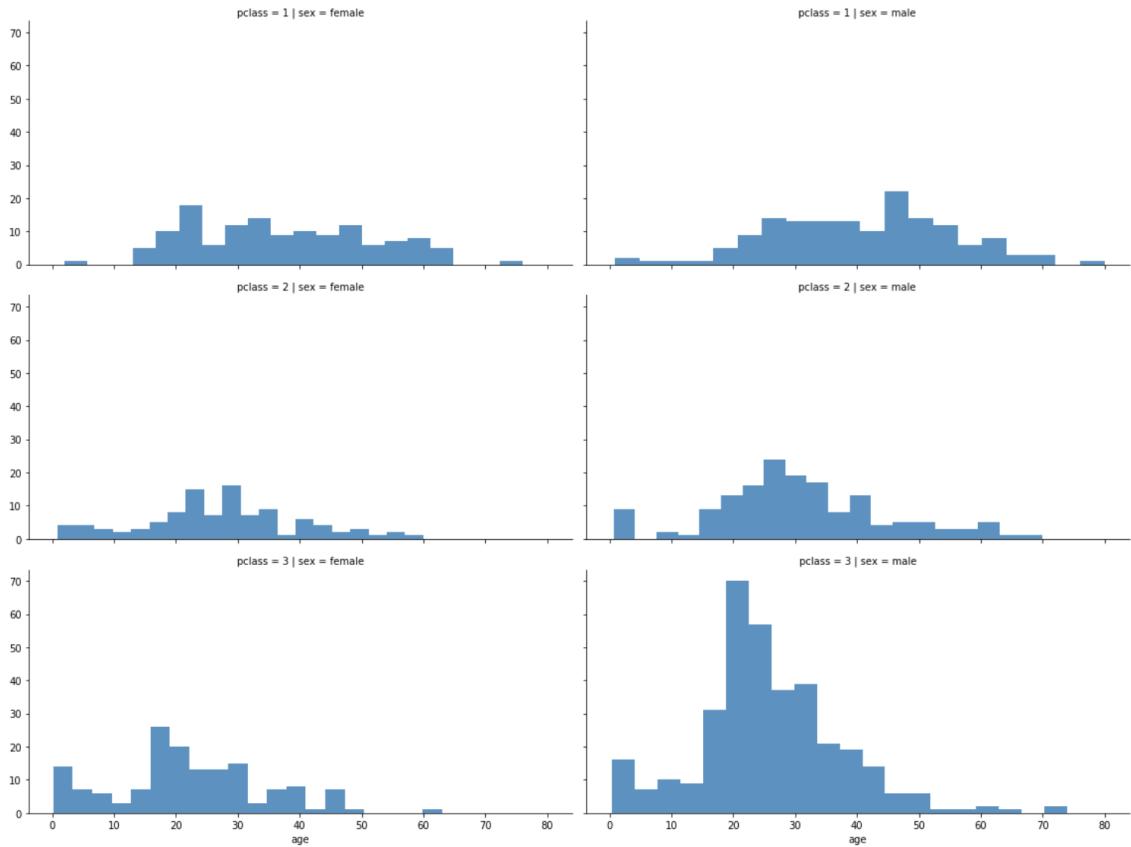
survived	0	1	All
pclass			
1	123	200	323
2	158	119	277
3	528	181	709
All	809	500	1309

- 1등실의 생존 가능성이 아주 높다
- 그런데 여성의 생존률도 높다
- 그럼, 1등실에는 여성이 많이 타고 있었나?

3.8 선실 등급별 성별 상황

```
grid = sns.FacetGrid(titanic, row='pclass', col='sex', height=4, aspect=2)
grid.map(plt.hist, 'age', alpha=.8, bins=20)
grid.add_legend();
```

3.9 3등실에는 남성이 많았다 - 특히 20대 남성

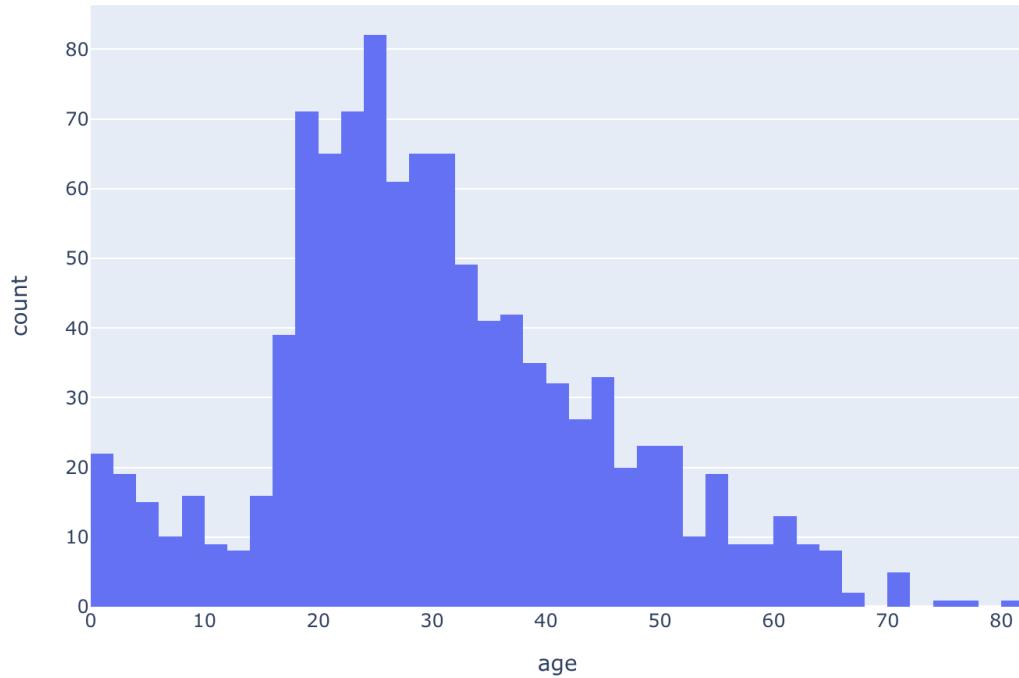


3.10 그러면 나이별 승객 현황도 확인해보자

```
import plotly.express as px

fig = px.histogram(titanic, x="age")
fig.show()
```

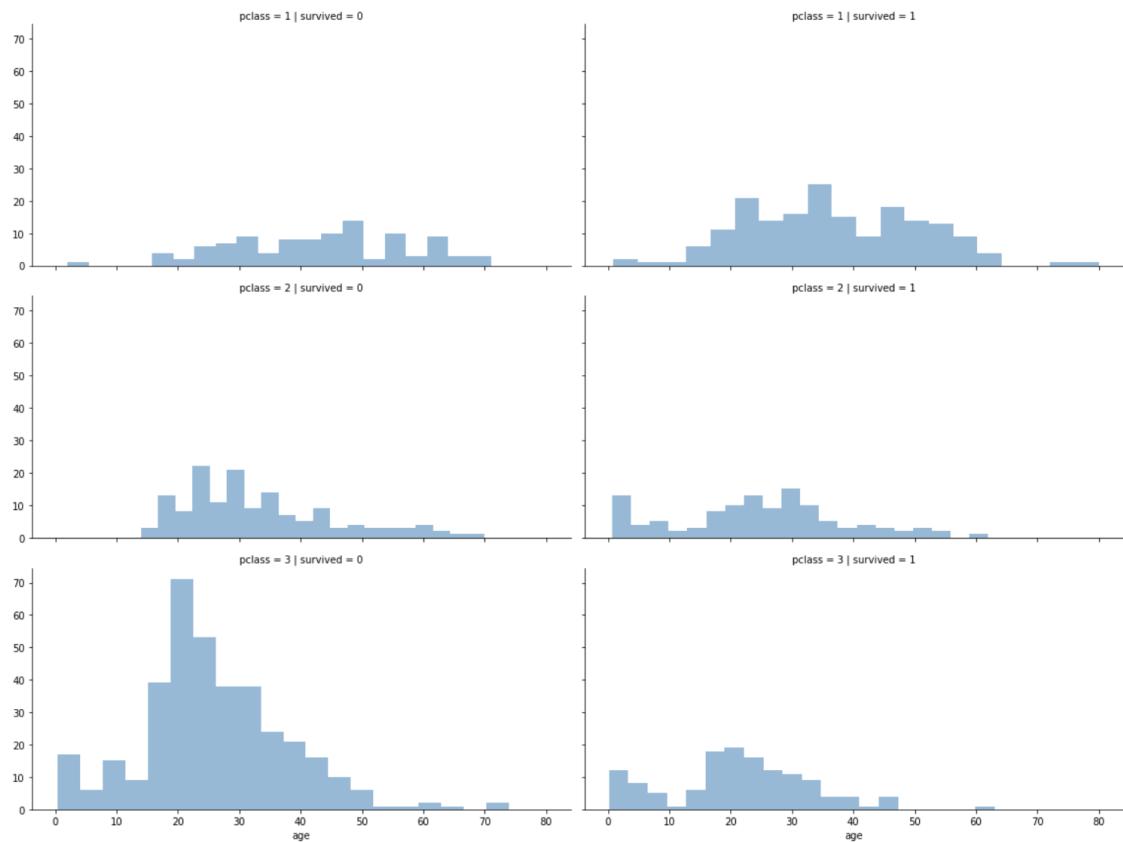
3.11 아이들과 20~30대가 많았다



3.12 그러면 등실별 생존률을 연령별로 관찰해 보자

```
grid = sns.FacetGrid(titanic, col='survived', row='pclass', height=4, aspect=2)
grid.map(plt.hist, 'age', alpha=.5, bins=20)
grid.add_legend();
```

3.13 확실히 선실 등급이 높으면 생존률이 높은듯 하다



3.14 나이를 5단계로 정리하기

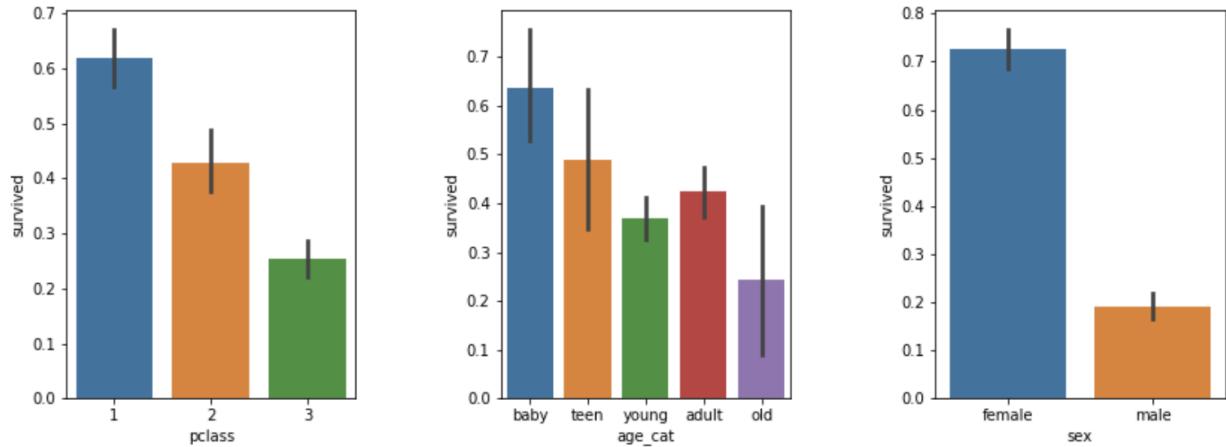
```
titanic['age_cat'] = pd.cut(titanic['age'], bins=[0,7,15,30,60,100],
                             include_lowest=True,
                             labels=['baby','teen','young','adult','old'])
titanic.head()
```

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest	a
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO	
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON	
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON	
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON	
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON	

3.15 나이, 성별, 등급별 생존자 수를 한번에 파악할 수 있을까?

```
plt.figure(figsize=(12,4))
plt.subplot(131)
sns.barplot('pclass', 'survived', data=titanic)
plt.subplot(132)
sns.barplot('age_cat', 'survived', data=titanic)
plt.subplot(133)
sns.barplot('sex', 'survived', data=titanic)
plt.subplots_adjust(top=1, bottom=0.1, left=0.1, right=1, hspace=0.5, wspace=0.5)
```

3.16 어리고, 여성이고, 1등실일 수록 생존하기 유리했을까?



3.17 남/여 나이별 생존 상황을 보다 더 들여다 보자

```

fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(14, 6))

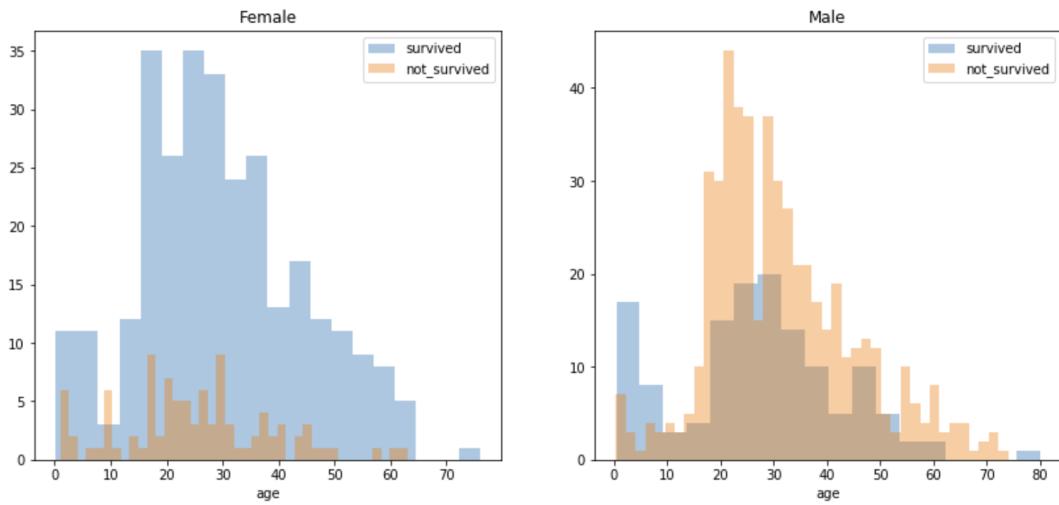
women = titanic[titanic['sex']=='female']
men = titanic[titanic['sex']=='male']

ax = sns.distplot(women[women['survived']==1]['age'], bins=20,
                  label = 'survived', ax = axes[0], kde=False)
ax = sns.distplot(women[women['survived']==0]['age'], bins=40,
                  label = 'not_survived', ax = axes[0], kde=False)
ax.legend(); ax.set_title('Female')

ax = sns.distplot(men[men['survived']==1]['age'], bins=18,
                  label = 'survived', ax = axes[1], kde=False)
ax = sns.distplot(men[men['survived']==0]['age'], bins=40,
                  label = 'not_survived', ax = axes[1], kde=False)
ax.legend(); ax = ax.set_title('Male')

```

3.18 이 결과는 어떤 것을 의미할까



3.19 탑승객의 이름에서 신분을 알 수 있다

```
| for idx, dataset in titanic.iterrows():
    print(dataset['name'])
```

Allen, Miss. Elisabeth Walton
 Allison, Master. Hudson Trevor
 Allison, Miss. Helen Loraine
 Allison, Mr. Hudson Joshua Creighton
 Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
 Anderson, Mr. Harry
 Andrews, Miss. Kornelia Theodosia
 Andrews, Mr. Thomas Jr
 Appleton, Mrs. Edward Dale (Charlotte Lamson)
 Artagaveytia, Mr. Ramon
 Astor, Col. John Jacob

3.20 정규식을 이용해서 문장 사이의 신분에 대한 정보를 얻을 수 있다

```
| import re

for idx, dataset in titanic.iterrows():
    tmp = dataset['name']
    print(idx)
    print(re.search('\,\s\w+(\s\w+)?\.', tmp).group())

, Mr.
1302
, Mr.
1303
, Mr.
1304
, Miss.
1305
...
```

3.21 살짝 응용하면

```
| import re

title = []
for idx, dataset in titanic.iterrows():
    title.append(re.search('\,\s\w+(\s\w+)?\.', dataset['name']).group()[2:-1])

titanic['title'] = title
titanic.head()
```

3.22 사회적 신분만 얻을 수 있다

survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest	age_cat	title
	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO	young	Miss
	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON	baby	Master
	Allison, Miss. Helen Lorraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON	baby	Miss
	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON	young	Mr
	Allison													

3.23 성별별로 본 귀족

```
pd.crosstab(titanic['title'], titanic['sex'])
```

title	sex	female	male
Capt	0	1	
Col	0	4	
Don	0	1	
Dona	1	0	
Dr	1	7	
Jonkheer	0	1	
Lady	1	0	
Major	0	2	
Master	0	61	
Miss	260	0	
Mlle	2	0	
Mme	1	0	
Mr	0	757	
Mrs	197	0	
Ms	2	0	
Rev	0	8	
Sir	0	1	
the Countess	1	0	

3.24 사회적 신분을 조금 더 정리하자

```
titanic['title'] = titanic['title'].replace('Mlle', 'Miss')
titanic['title'] = titanic['title'].replace('Ms', 'Miss')
titanic['title'] = titanic['title'].replace('Mme', 'Mrs')

Rare_f = ['Dona', 'Dr', 'Lady', 'the Countess']

Rare_m = ['Capt', 'Col', 'Don', 'Major', 'Rev', 'Sir', 'Jonkheer', 'Master']

for each in Rare_f:
    titanic['title'] = titanic['title'].replace(each, 'Rare_f')

for each in Rare_m:
    titanic['title'] = titanic['title'].replace(each, 'Rare_m')

titanic['title'].unique()

array(['Miss', 'Rare_m', 'Mr', 'Mrs', 'Rare_f'], dtype=object)
```

3.25 이 결과는 또 어떻게 해석할 수 있을까

```
titanic[['title','survived']].groupby(['title'], as_index=False).mean()
```

	title	survived
0	Miss	0.678030
1	Mr	0.162483
2	Mrs	0.787879
3	Rare_f	0.636364
4	Rare_m	0.443038

3.26 타이타닉 배의 선장



선장 E. J. 스미스



선원과 승객들에게 인기가 많아서 어떤 승객들은 스미스 선장이 책임지는 배만 탑승하기도 함



은퇴가 예정되어 있었는데 선사의 부탁을 거절하지 못하고 마지막 운항을 하기로 했다고 함



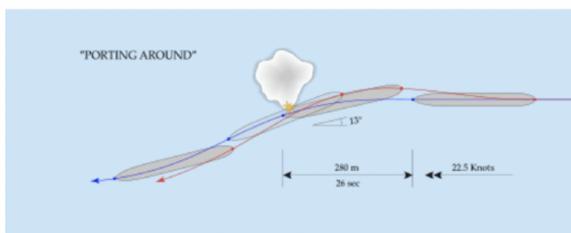
배의 침몰 직전까지도 승객의 대피와 구조 요청을 지휘하다가 배와 함께 운명을 했다고 함

3.27 타이타닉 빙산과 충돌

▼ 2.1. 빙산과 충돌하다

[편집]

그리고 운명의 1912년 4월 14일 밤 11시 40분, 레지놀드 리와 함께 견시를 보던 갑판 선원 프레드릭 플리트가 전방 450m에 높이 20m 미만의 빙산을 육안으로 발견했다.^[2] 출항할 때 쌍안경을 사용할 수 없었고 빙산의 10분의 9는 숨어있었기 때문에 빙산을 발견했을 때에는 이미 늦은 뒤였다.^[3]



3.28 타이타닉의 탈출 원칙

◀ 2.1. 빙산과 충돌하다 [편집]

그리고 운명의 1912년 4월 14일 밤 11시 40분, 레지놀드 리와 함께 견시를 보던 갑판 선원 프레드릭 플리트가 전방 450m에 높이 20m 미만의 빙산을 육안으로 발견했다. [2] 출항할 때 쌍안경을 사용할 수 없었고 빙산의 10분의 9는 숨어있었기 때문에 빙산을 발견했을 때에는 이미 늦은 뒤였다. [3]

4월 15일 0시 15분 조난 신호 CQD(CQ= sécurité) 또 유키로 보내 왔자 유키는 모든 기지로 후송 D= Distress 조나 등을 학처 구조요청이 되다. [13] 를 발시 인근 선박에 구조를 요청했다. 불과 20km 정도의 거리에 정박중인 화물선 캐리포니안호가 있었지만 1명밖에 없는 통신사가 오랜 1인 근무로 인한 피로누적으로 취침 중 (...)이라 연락을 받지 못했다. 이 일로 인해 타이타닉 사고 이후 무전을 상시 커놓고 교대근무를 통해 통신사가 항상 대기하도록 하는 국제규약이 생겨났다. 대략 90km 떨어져 있던 여객선 RMS 카파시아호가 응답해 전속으로 구조로 향했지만 배의 최대 속도가 17노트(약 시속 31.48km)에 그쳐 카파시아호가 현장에 도착한 시간은 오전 4시, 생존자들의 증언에 따른 타이타닉의 완전침몰시각은 2시 20분 경으로 침몰 1시간 40분 뒤에 도착했다. 그나마 이것도 아래에 나오듯이 카파시아호가 선박의 동력을 한계까지 필사적으로 쥐어짜서 최선을 다해 달려온 것이었으며, 침수된 접이식 보트 A, B호 생존자 등은 카파시아호가 빠르게 도착한 덕에 그 추위에도 많이 살아남았다.

2등 항해사 라이틀러는 선장에게 여자와 어린이를 먼저 태울 것을 건의하고 선장은 승인했다. [14] 이건 당시 사회 통념상 어떻게 보면 당연한 일이었다. 1등 항해사 머독은 더 이상 여성과 아이들이 보이지 않으면 남자를 태우는 것을 협용하는 등 비교적 남성에게도 관대한 대응을 했지만 라이틀러는 "여자와 어린이 먼저"를 "여자와 어린이 맨"으로 받아들여 혼란을 막기 위해 여성과 어린이 우선의 관습을 철저히 실시했다. [14] 이게 현장에선 도리어 억효과를 내기도 했다. 정원이 덜 찼음에도 성인 남자라는 이유로 탑승이 거부되거나 우현에 비해 보트를 내리는 시간이 오래 걸리는 등 부작용을 일으킨 것이다. 다만, 좌현에서 100% 남성 승객의 보트 승선이 거부된 건 아니고, 보트에 선원이 한 명 뿐이라는 여성 승객의 의의로 인해 선원 대신 타게 된 오트 선수와 몇 명의 노인 등 약간의 예외도 있었다.

3.29 타이타닉의 최후

◀ 2.1. 빙산과 충돌하다 [편집]

그리고 운명의 1912년 4월 14일 밤 11시 40분, 레지놀드 리와 함께 견시를 보던 갑판 선원 프레드릭 플리트가 전방 450m에 높이 20m 미만의 빙산을 육안으로 발견했다. [2] 출항할 때 쌍안경을 사용할 수 없었고 빙산의 10분의 9는 숨어있었기 때문에 빙산을 발견했을 때에는 이미 늦은 뒤였다. [3]



조나 등을 학처 구조요청이 되다. [13] 를 발시 인근 통신사가 오랜 1인 근무로 인한 피로누적으로 취침 중 항상 대기하도록 하는 국제규약이 생겨났다. 대략 90km 속 31.48km)에 그쳐 카파시아호가 현장에 도착한 시간은 오전 4시, 생존자들의 증언에 따른 타이타닉의 완전침몰시각은 2시 20분 경으로 침몰 1시간 40분 뒤에 도착했다. 그나마 이것도 아래에 나오듯이 카파시아호가 선박의 동력을 한계까지 필사적으로 쥐어짜서 최선을 다해 달려온 것이었으며, 침수된 접이식 보트 A, B호 생존자 등은 카파시아호가 빠르게 도착한 덕에 그 추위에도 많이 살아남았다.

통념상 어떻게 보면 당연한 일이었다. 1등 항해사 머독은 있지만 라이틀러는 "여자와 어린이 먼저"를 "여자와 어린이 억효과를 내기도 했다. 정원이 덜 찼음에도 성인 남자라면, 좌현에서 100% 남성 승객의 보트 승선이 거부된 건 아니고, 보트에 선원이 한 명 뿐이라는 여성 승객의 의의로 인해 선원 대신 타게 된 오트 선수와 몇 명의 노인 등 약간의 예외도 있었다.

2시 10분에는 바닷물이 최상층인 보트 갑판까지 다다랐다. 그때쯤 체육관 바깥쪽 휴식터에서 월리스 하틀리가 지휘를 맡은 악단이 구슬픈 찬송가를 마지막으로 연주하기 시작했다. (어떤 곡이 연주되었는지에 대해서는 의견이 분분하다. 유력한 곡은 'Nearer, My God, to Thee' [21] 와 'Autourme'이다.)

선장은 확성기를 전 채로 선교에 들어가 그대로 그곳에 남았다고 하며 배의 설계자 토머스 앤드루스는 흡연실에서 구명조끼를 벗은 채 그림을 응시하는 모습이 마지막으로 목격되었다고 한다.

2시 15분에서 17분 사이에는 배가 보트 갑판을 본격적으로 삼키기 시작했다. 배가 전과는 비교할 수도 없을 정도로 빠르게 가라앉기 시작했다. 통신사 잭 필립스와 조수 해럴드 브리드는 더 이상은 무리라는 것을 깨닫고 마지막 구조 신호를 보낸 후 침수되는 통신실에서 빠져나왔다.

4 머신러닝을 이용한 생존자 예측

4.1 간단히 구조확인

```
titanic.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
---  --  
 0   pclass      1309 non-null   int64  
 1   survived    1309 non-null   int64  
 2   name        1309 non-null   object  
 3   sex         1309 non-null   object  
 4   age         1046 non-null   float64 
 5   sibsp       1309 non-null   int64  
 6   parch       1309 non-null   int64  
 7   ticket      1309 non-null   object  
 8   fare        1308 non-null   float64 
 9   cabin       295 non-null   object  
 10  embarked    1307 non-null   object  
 11  boat        486 non-null   object  
 12  body        121 non-null   float64 
 13  home.dest   745 non-null   object  
 14  age_cat     1046 non-null   category
dtypes: category(1), float64(3), int64(4), object(7)
memory usage: 144.8+ KB
```

4.2 머신러닝을 위해 해당컬럼을 숫자로 변경해야겠다

```
titanic['sex'].unique()

array(['female', 'male'], dtype=object)
```

4.3 Label Encode를 사용하면 편하다

```
| from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
le.fit(titanic['sex'])
titanic['gender'] = le.transform(titanic['sex'])
titanic.head()
```

ved	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest	age_cat	gender
	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO	young	0
	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON	baby	1
	Allison, Miss. Helen Lorraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON	baby	0

4.4 결측치는 어쩔 수 없이 포기하자

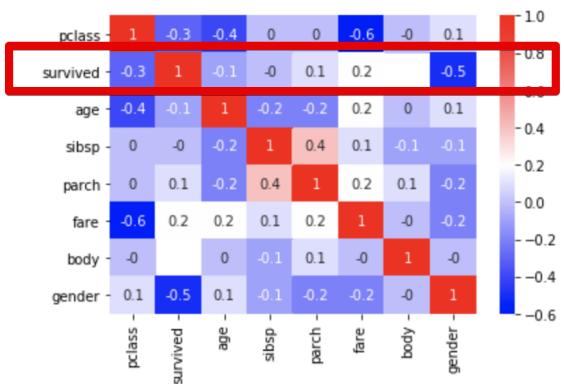
```
| titanic = titanic[titanic['age'].notnull()]
titanic = titanic[titanic['fare'].notnull()]
titanic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1045 entries, 0 to 1308
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   pclass      1045 non-null   int64  
 1   survived    1045 non-null   int64  
 2   name        1045 non-null   object  
 3   sex         1045 non-null   object  
 4   age         1045 non-null   float64 
 5   sibsp       1045 non-null   int64  
 6   parch       1045 non-null   int64  
 7   ticket      1045 non-null   object  
     ...
```

4.5 상관관계

```
correlation_matrix = titanic.corr().round(1)
sns.heatmap(data=correlation_matrix, annot=True, cmap='bwr')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f8151a8f9d0>



4.6 먼저 특성을 선택하고, 데이터를 나누자

```
from sklearn.model_selection import train_test_split

X = titanic[['pclass', 'age', 'sibsp', 'parch', 'fare', 'gender']]
y = titanic['survived']

X_train, X_test, y_train, y_test = \
    train_test_split(X, y, test_size=0.2, random_state=13)
```

- 특성은 pclass, age, sibsp, parch, fare, gender로 선정

4.7 일단 DecisionTree

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

dt = DecisionTreeClassifier(max_depth=4, random_state=13)
dt.fit(X_train, y_train)

pred = dt.predict(X_test)
print(accuracy_score(y_test, pred))
```

0.7655502392344498

5 그런데 디카프리오 형님의 생존율은?

5.1 디카프리오를 특정짓자



5.2 디카프리오의 생존률

```
| import numpy as np  
  
dicaprio = np.array([[3, 18, 0, 0, 5, 1]])  
print('Decaprio : ', dt.predict_proba(dicaprio)[0,1])
```

Decaprio : 0.2857142857142857

5.3 그럼 윈슬릿은?

```
| winslet = np.array([[1, 16, 1, 1, 100, 0]])  
print('Winslet : ', dt.predict_proba(winslet)[0,1])
```

Winslet : 1.0