

3. 머신러닝을 위한 확률과 통계

2019. 07. 03, 수

강사소개

- **이력**

포스텍 앱개발 동아리 PoApper

포스텍 수학과 학사 및 이학박사(대수기하)

(주) 비닷두에서 책임연구원으로 근무(스마트 CCTV 관제)

- **취미**

웹툰, 퍼즐

수학

코딩



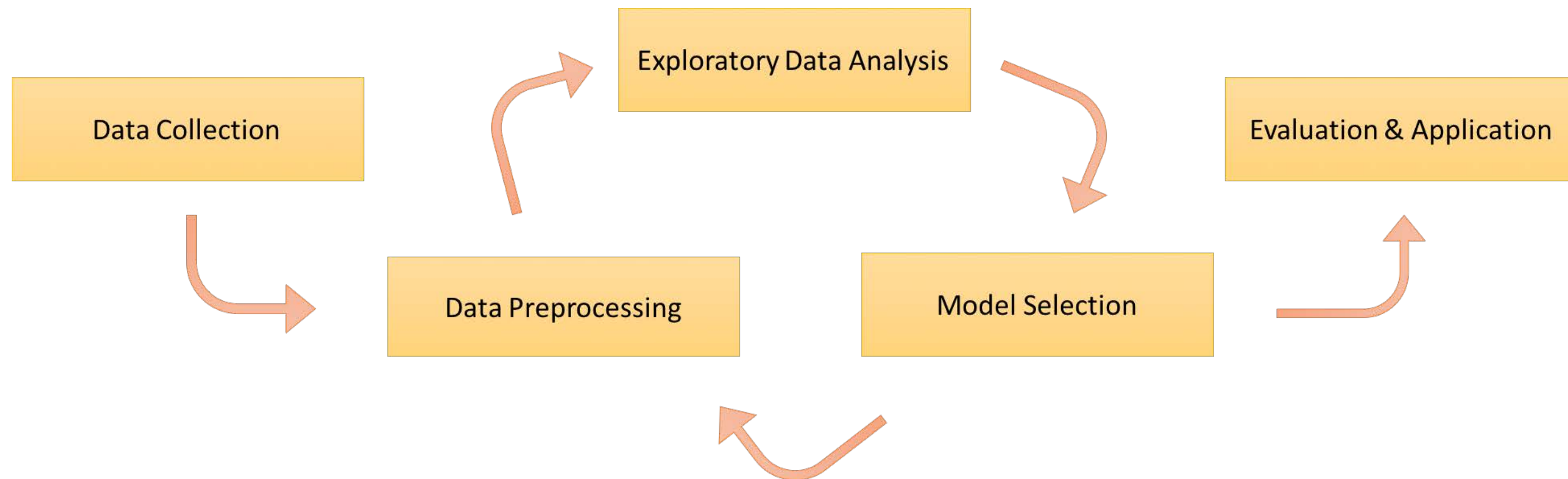
커리큘럼

- 1 ○ 머신러닝 프로젝트 진행과정(Workflow)에 대해 알아보시다.
 1. 데이터의 수집
 2. 전처리(Preprocessing), 3. 탐색적 데이터분석(EDA), 4. 모델 선택
 5. 평가 및 적용
- 2 ○ 탐색적 데이터분석(EDA)에 대해 알아보시다.
 - 통계치 분석 및 시각화를 통한 상관관계 분석
 - 피쳐 엔지니어링
- 3 ○ 확률과 통계의 기본에 대해 알아보시다.
 - 편향(Bias)-분산(Variance) Tradeoff
 - 확률변수 - 이산(이항분포, 포아송분포) 및 연속(정규분포)
 - 빈도주의자 확률 vs. 베이지안 확률 - 베이즈 정리(Bayes' Theorem)
 - 결합, 주변, 조건부 확률분포
 - 최대우도 추정(MLE) vs. 최대사후확률 추정(MAP)

1. 머신러닝 프로젝트 진행과정(Workflow)

1. 머신 러닝 프로젝트 진행과정 (Workflow)

프로젝트에 따라 진행 과정은 조금씩 다르지만, 일반적인 예측 모델링 프로젝트(predictive modeling)는 다음과 같은 프로세스를 따릅니다.



1. 머신 러닝 프로젝트 진행과정 (Workflow)

MNIST, Fashion-MNIST, Titanic과 같은 아주 간단한 데이터셋을 비롯해 CIFAR-10, MS COCO, ImageNet, KITTI, VisualQA, Youtube-8M, CelebFaces 등의 공개(public) 데이터셋은 이미 수집과 전처리까지 다 되어 있지만 커스텀(custom) 데이터셋은 여러분들이 모든 작업을 하셔야 합니다.

1. 데이터 수집 (Data Collection)

- 웹 사이트에 있는 자료 : 웹 크롤링
- 서비스에서 유저들의 행동 데이터 수집 : 서비스 로그

1. 머신 러닝 프로젝트 진행과정 (Workflow)

2. 데이터 전처리 (Data Preprocessing)

- 날 것 그대로의 초기 데이터(Raw data)는 더럽습니다.
(빠진 부분, 중복, 이상한 값 등)
- 더러운 초기 데이터를 정제해서 머신러닝 모델의 입력에 적합한 형태로 바꿔주는 단계를 데이터 정제(Data Cleaning)이라고 합니다.
- 피쳐 엔지니어링(Feature Engineering): 예컨대 집의 가격을 예측할 때는 가로, 세로 길이보다 면적이 더 중요할 것이므로 새로운 피쳐를 만들어줍니다.

1. 머신 러닝 프로젝트 진행과정 (Workflow)



1. 머신 러닝 프로젝트 진행과정 (Workflow)

3. 탐색적 데이터분석 (EDA: Exploratory Data Analysis)

- 데이터 프로젝트의 성공 여부는 얼마나 데이터를 이해하고 있느냐에 좌우된다. 데이터를 이해하기 위해 데이터의 특징을 찾고, 숨겨진 패턴을 발견하는 과정을 EDA라고 부릅니다.
- 구체적으로 데이터의 히스토그램을 그려보고, 두 변수 사이의 산포도를 그려보고, 변수들의 상관관계를 보는 일 등을 포함합니다.

1. 머신 러닝 프로젝트 진행과정 (Workflow)

4. 모델 선택 (Model Selection)

- 첫째, 말그대로 예측값을 계산하는 알고리즘을 선택하는 것
(Logistic Regression, K-NN, SVM, Deep Learning)
- 둘째, 모델이 사용할 속성들(features)을 선택하는 것
- 셋째, 같은 모델이더라도 초모수(Hyperparameter)에 따라서 성능이 천차만별이므로, 초모수를 잘 선택하는 것

1. 머신 러닝 프로젝트 진행과정 (Workflow)

5. 평가 및 적용 (Evaluation & Application)

- 머신러닝 프로젝트에서 가장 중요하지만 가장 실수가 자주 일어나는 과정입니다. 모델 평가에서 반드시 지켜야 하는 점은 테스트 데이터셋은 모델 선택과 모델 학습 과정에서 쓰이지 않아야 한다는 점입니다.
- 이렇게 하는 이유는 평가 과정의 목적이 모델이 새로운 데이터에 대해 얼마나 일반화 (generalization) 가능한지 측정하는 것이기 때문입니다.

1. 머신 러닝 프로젝트 진행과정 (Workflow)

6. 결론

- EDA와 전처리, 모델 선택 과정은 순차적이 아니라 반복적인 관계
- EDA를 통해 어떤 전처리가 필요한지 알 수 있고, 전처리를 한 후에 데이터를 더욱 잘 이해할 수도 있습니다. 예컨대, 데이터를 탐색하던 중에 몰랐던 이상치(outlier)을 발견할 수 있고, 이상치를 제거하는 전처리를 한 후 데이터 탐색이 더 원활해 질 수 있습니다.
- 모델 선택 과정 역시 데이터 탐색과 밀접한 관계가 있습니다. 데이터를 이해하고 나서 데이터에 적합한 모델을 선택할 수 있고, 원하는 만큼 모델의 정확도가 나오지 않을 경우 그 이유를 찾기 위해 데이터 탐색 과정으로 돌아올 수도 있습니다.

2. 탐색적 데이터분석(EDA)

2. 탐색적 데이터 분석(EDA)

캐글의 Titanic 데이터셋(<https://www.kaggle.com/c/titanic>)

변수명	변수 설명
Survived	생존여부(0: 사망, 1: 생존)
Pclass	선실 클래스(1, 2, 3)
Name	이름
Sex	성별
Age	나이
Sibsp	탑승한 형제자매, 배우자의 수
Parch	탑승한 부모, 자식의 수
Ticket	티켓 번호
Fare	운임
Cabin	객실번호
Embarked	탑승장소(C: Cherbourg, Q: Queenstown, S: Southampton)

2. 탐색적 데이터 분석(EDA)

먼저 데이터셋의 앞부분을 살펴보면 다음 그림과 같습니다.

```
df_train = pd.read_csv('./input/train.csv')
df_test = pd.read_csv('./input/test.csv')
```

```
df_train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

2. 탐색적 데이터 분석(EDA)

```
df_train.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
df_test.describe()
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

2. 탐색적 데이터 분석(EDA)

```
for col in df_train.columns:
    msg = '항목: {:>10}##t 비어있는 자료의 비율: {:.2f}%'.format(col, 100 * (df_train[col].isnull().sum() / df_train[col].shape[0]))
    print(msg)
```

항목: PassengerId	비어있는 자료의 비율: 0.00%
항목: Survived	비어있는 자료의 비율: 0.00%
항목: Pclass	비어있는 자료의 비율: 0.00%
항목: Name	비어있는 자료의 비율: 0.00%
항목: Sex	비어있는 자료의 비율: 0.00%
항목: Age	비어있는 자료의 비율: 19.87%
항목: SibSp	비어있는 자료의 비율: 0.00%
항목: Parch	비어있는 자료의 비율: 0.00%
항목: Ticket	비어있는 자료의 비율: 0.00%
항목: Fare	비어있는 자료의 비율: 0.00%
항목: Cabin	비어있는 자료의 비율: 77.10%
항목: Embarked	비어있는 자료의 비율: 0.22%

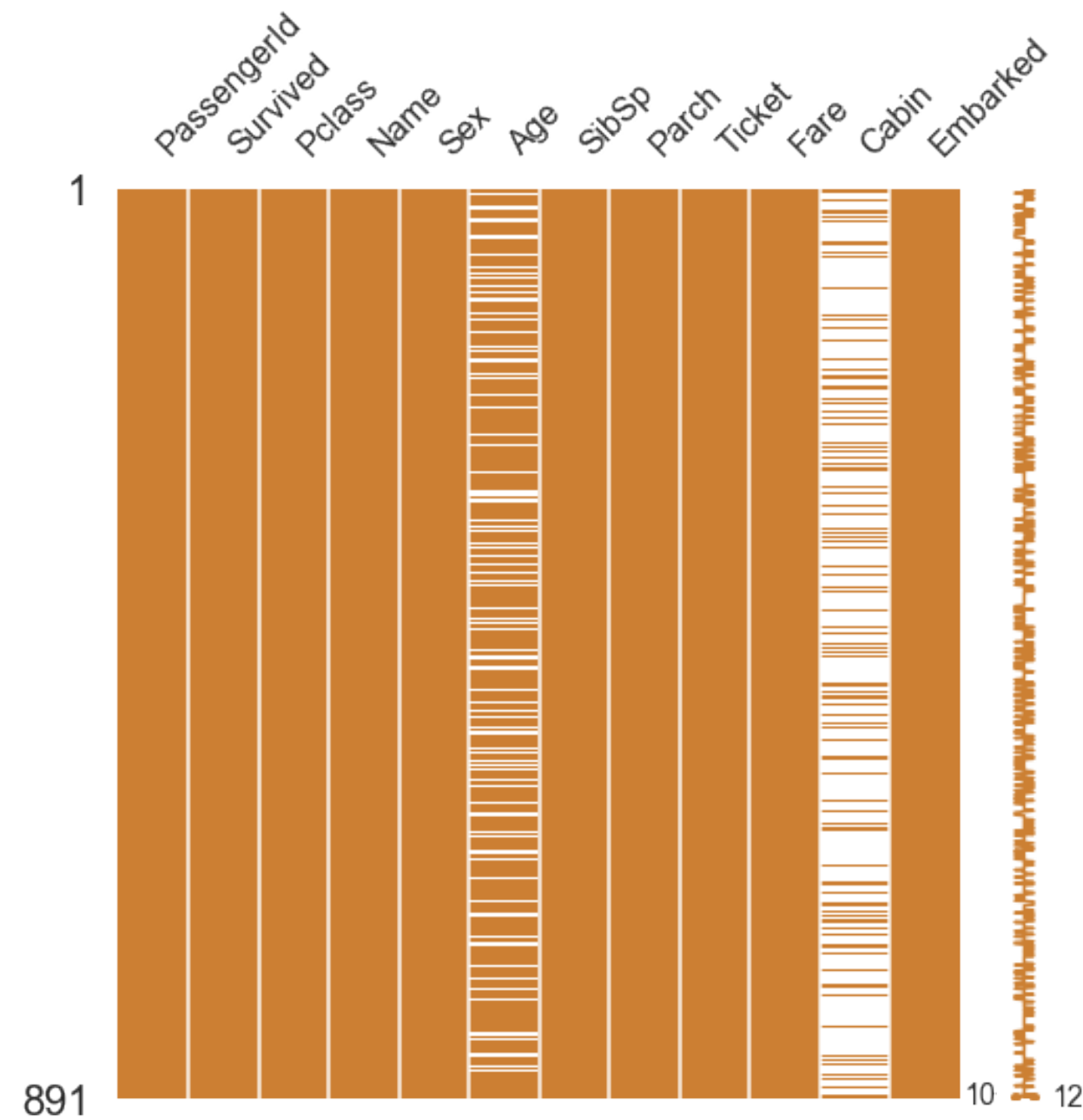
```
for col in df_test.columns:
    msg = '항목: {:>10}##t 비어있는 자료의 비율: {:.2f}%'.format(col, 100 * (df_test[col].isnull().sum() / df_test[col].shape[0]))
    print(msg)
```

항목: PassengerId	비어있는 자료의 비율: 0.00%
항목: Pclass	비어있는 자료의 비율: 0.00%
항목: Name	비어있는 자료의 비율: 0.00%
항목: Sex	비어있는 자료의 비율: 0.00%
항목: Age	비어있는 자료의 비율: 20.57%
항목: SibSp	비어있는 자료의 비율: 0.00%
항목: Parch	비어있는 자료의 비율: 0.00%
항목: Ticket	비어있는 자료의 비율: 0.00%
항목: Fare	비어있는 자료의 비율: 0.24%
항목: Cabin	비어있는 자료의 비율: 78.23%
항목: Embarked	비어있는 자료의 비율: 0.00%

2. 탐색적 데이터 분석(EDA)

앞페이지의 내용을 시각화하면 다음과 같습니다.

```
msno.matrix(df=df_train.iloc[:, :], figsize=(8, 8), color=(0.8, 0.5, 0.2))
```



2. 탐색적 데이터 분석(EDA)

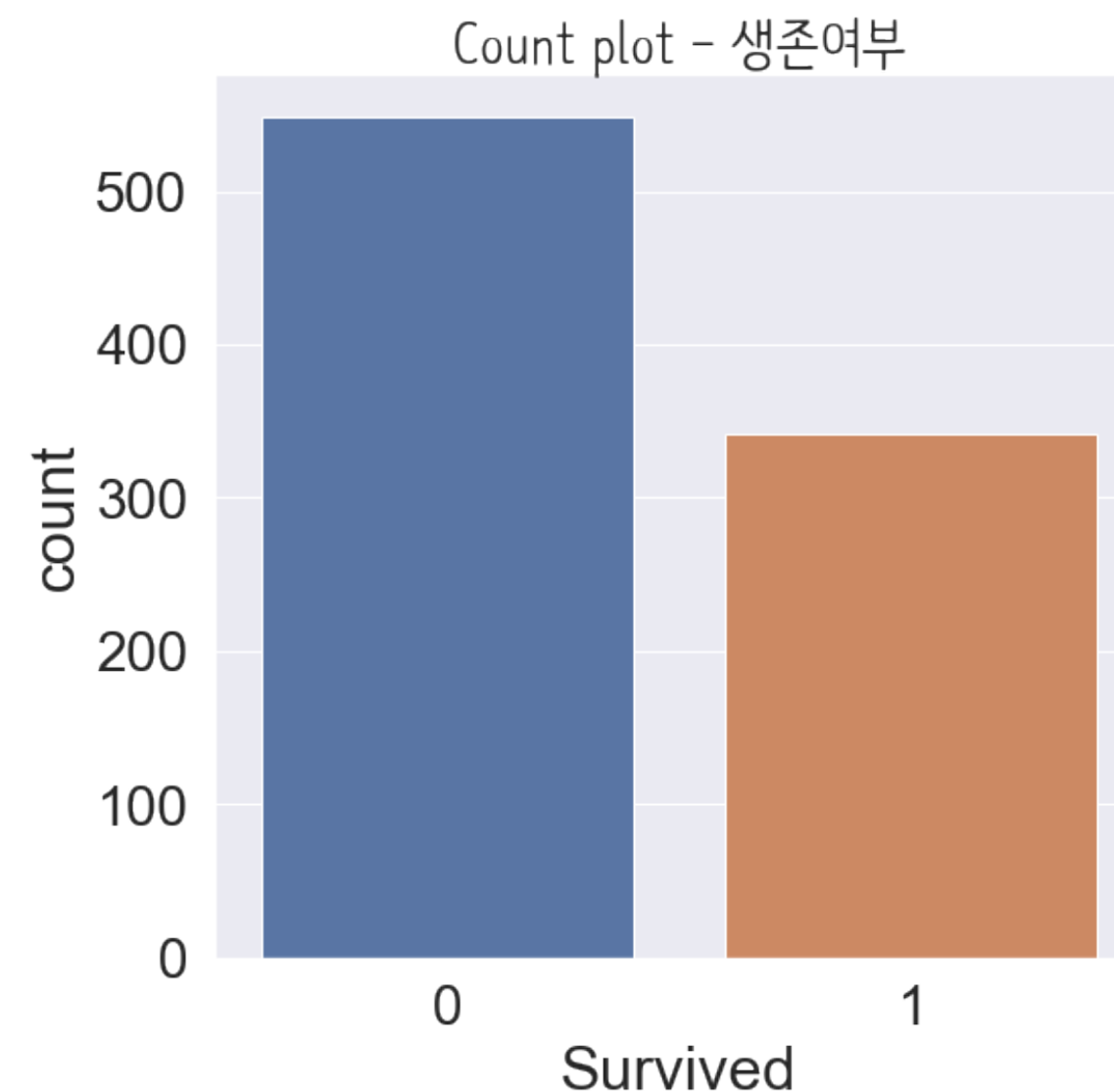
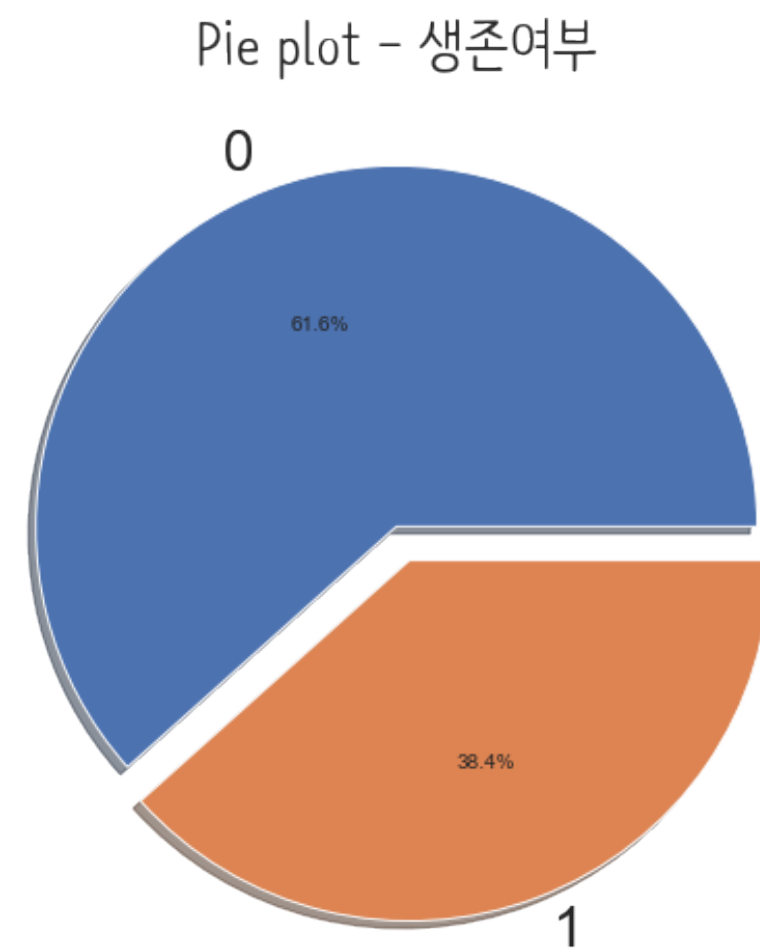
```
path = 'C:\\WINDOWS\\Fonts\\NanumBarunpenR.ttf'
fontprop = fm.FontProperties(fname=path, size=28)

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

df_train['Survived'].value_counts().plot.pie(explode=[0, 0.1], autopct='%1.1f%%', ax=ax[0], shadow=True)
ax[0].set_title('Pie plot - 생존여부', fontproperties=fontprop)
ax[0].set_ylabel('')
sns.countplot('Survived', data=df_train, ax=ax[1])
ax[1].set_title('Count plot - 생존여부', fontproperties=fontprop)

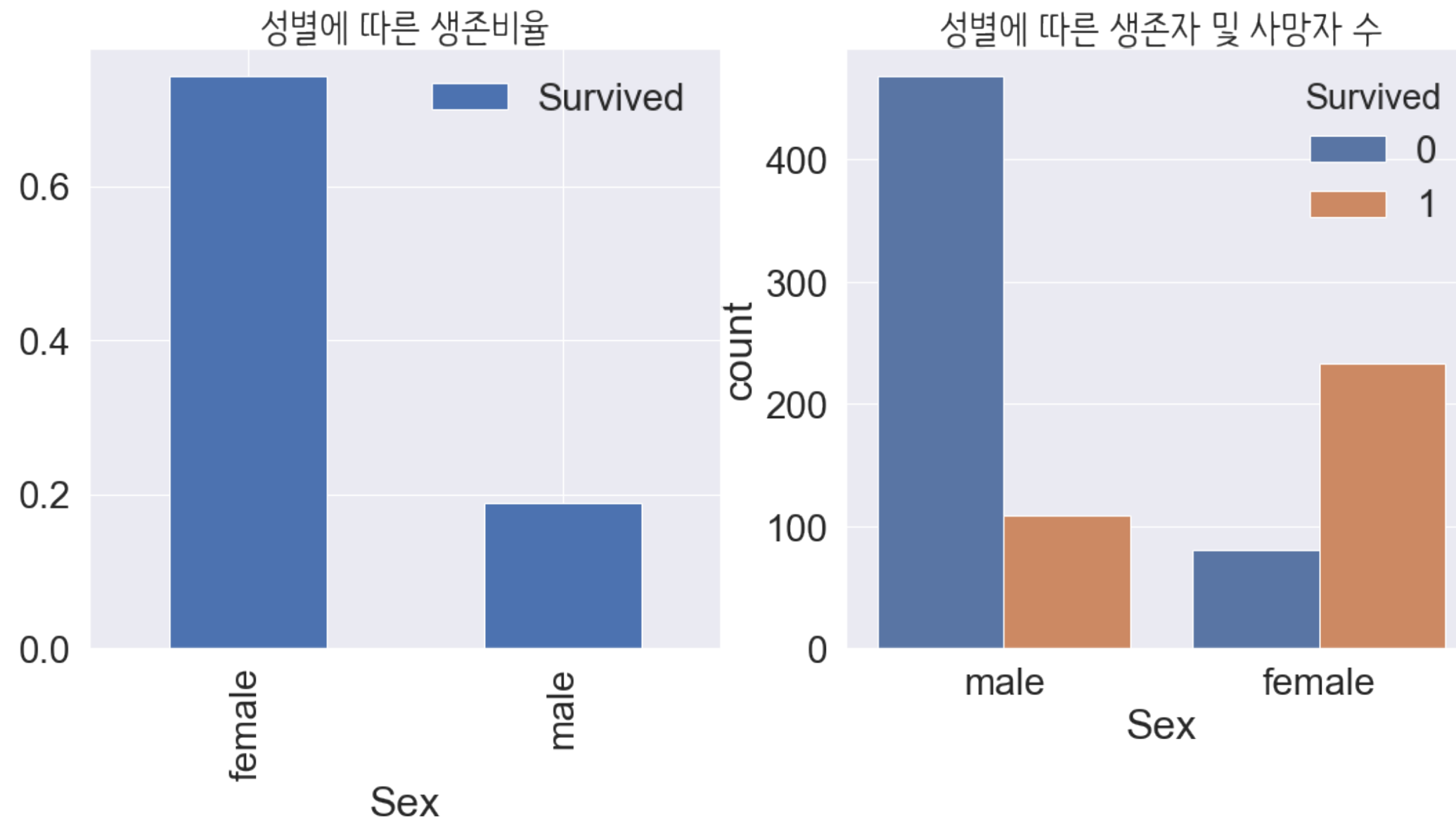
plt.show()
```

성별



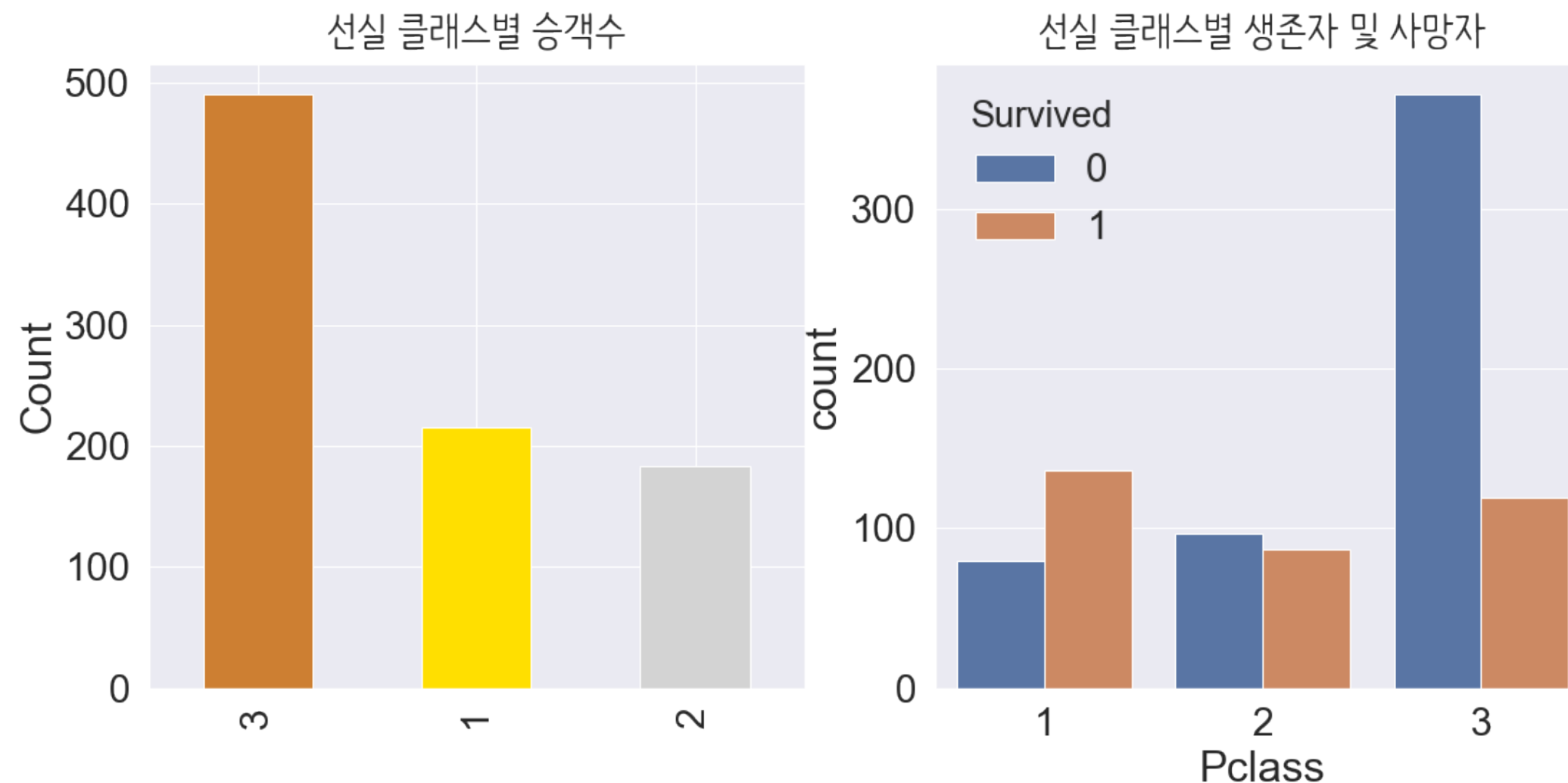
2. 탐색적 데이터 분석(EDA)

```
f, ax = plt.subplots(1, 2, figsize=(18, 8))
df_train[['Sex', 'Survived']].groupby(['Sex'], as_index=True).mean().plot.bar(ax=ax[0])
ax[0].set_title('성별에 따른 생존비율', fontproperties=fontprop)
sns.countplot('Sex', hue='Survived', data=df_train, ax=ax[1])
ax[1].set_title('성별에 따른 생존자 및 사망자 수', fontproperties=fontprop)
plt.show()
```



2. 탐색적 데이터 분석(EDA)

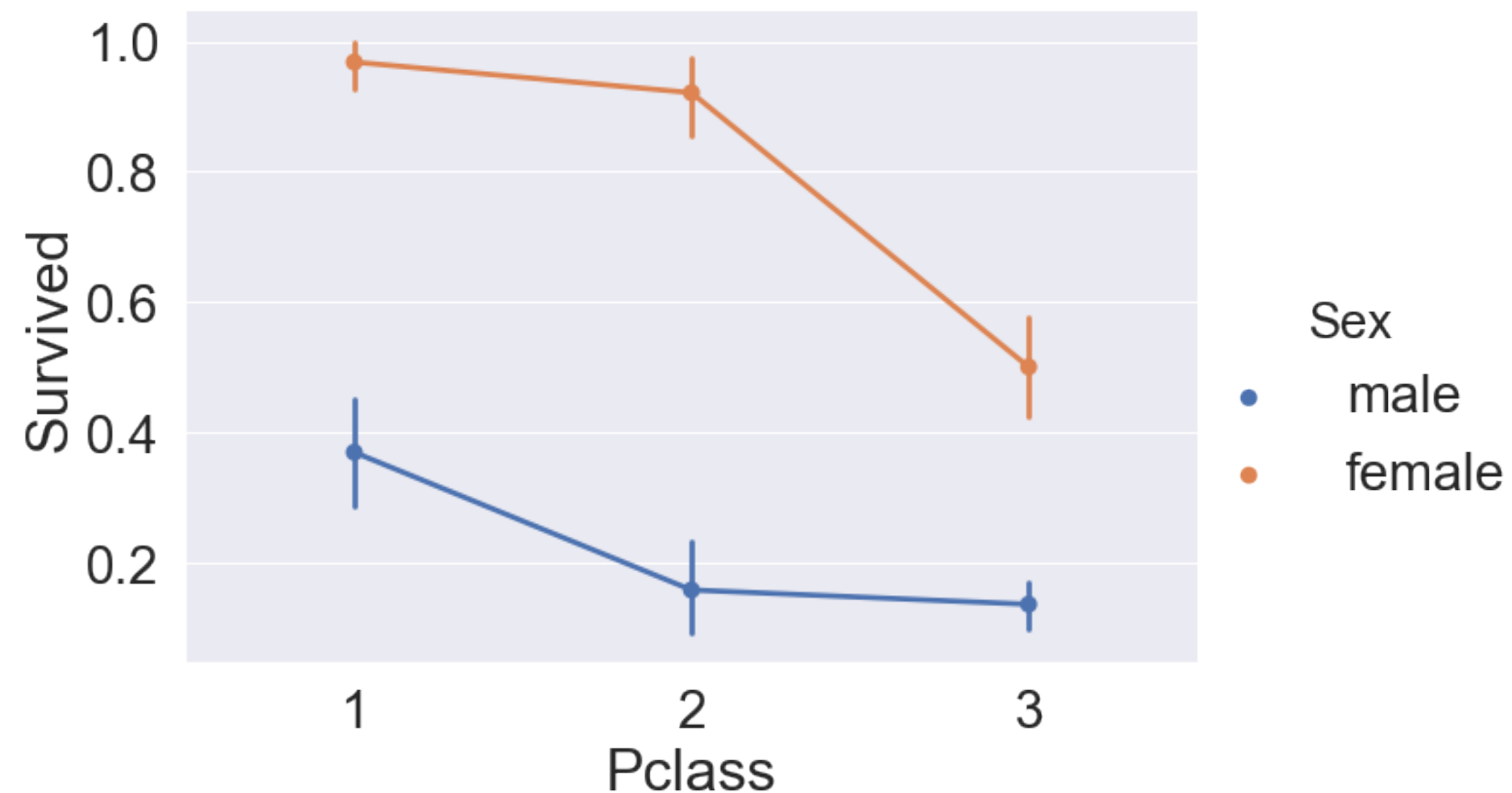
```
y_position = 1.02
f, ax = plt.subplots(1, 2, figsize=(18, 8))
df_train['Pclass'].value_counts().plot.bar(color=['#CD7F32', '#FFDF00', '#D3D3D3'], ax=ax[0])
ax[0].set_title('선실 클래스별 승객수', y=y_position, fontproperties=fontprop)
ax[0].set_ylabel('Count')
sns.countplot('Pclass', hue='Survived', data=df_train, ax=ax[1])
ax[1].set_title('선실 클래스별 생존자 및 사망자', y=y_position, fontproperties=fontprop)
plt.show()
```



2. 탐색적 데이터 분석(EDA)

선실 클래스에 관계없이 여성의 생존률이 높습니다.

```
sns.factorplot('Pclass', 'Survived', hue='Sex', data=df_train, size=6, aspect=1.5)
```



2. 탐색적 데이터 분석(EDA)

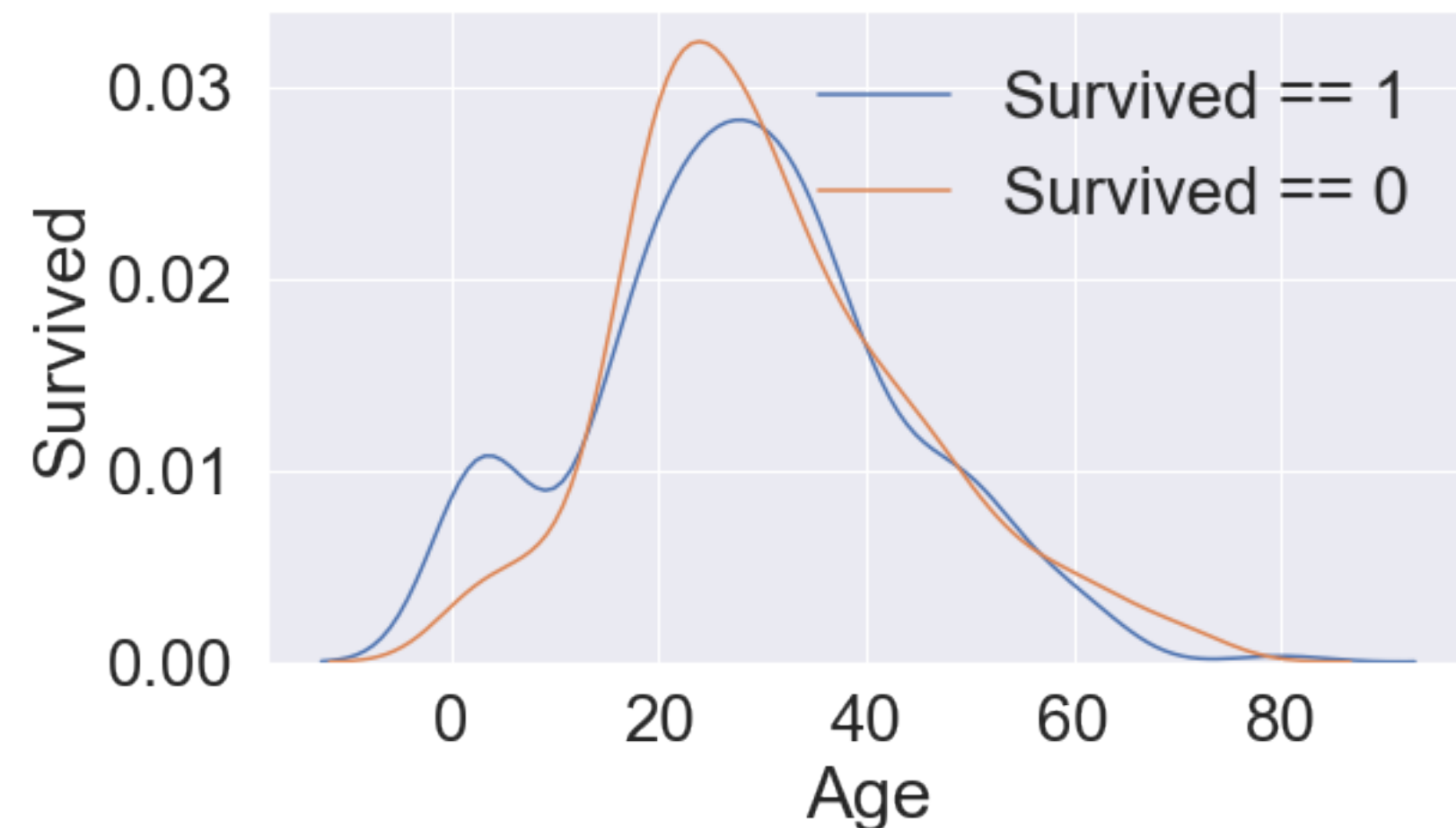
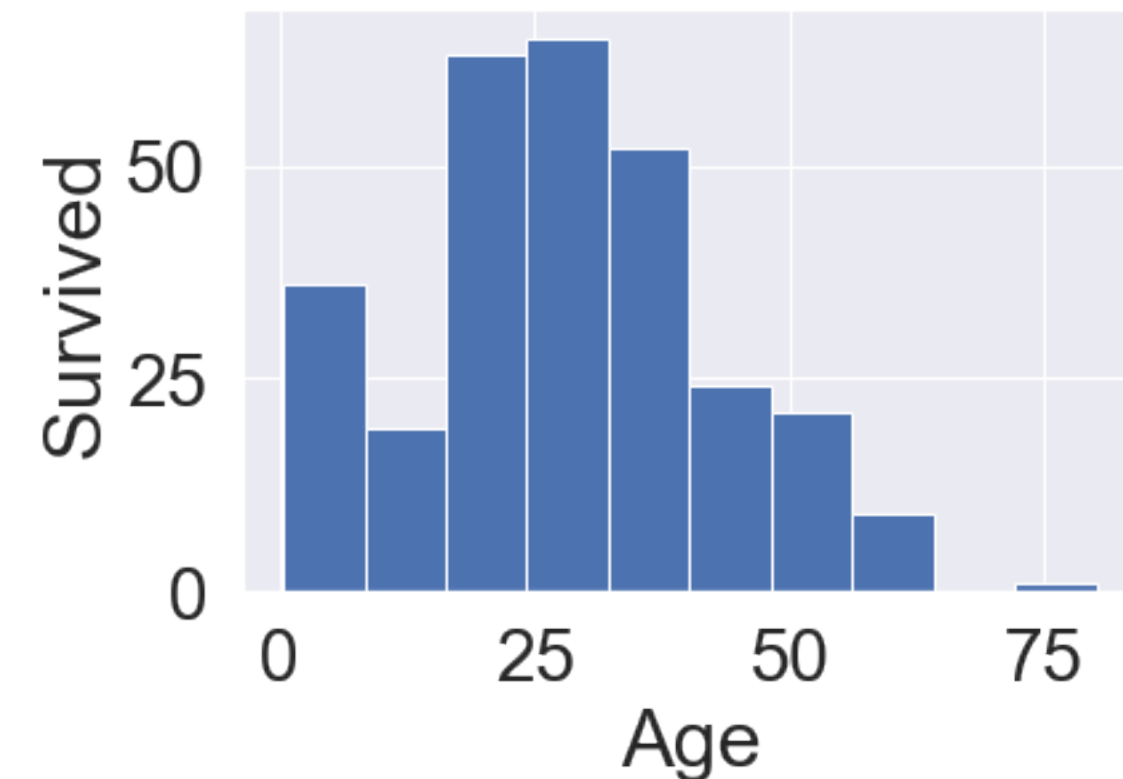
```
df_train[df_train['Survived'] == 1]['Age'].hist()  
plt.xlabel('Age')  
plt.ylabel('Survived')
```

또한 아이들의 생존률이 높습니다.

```
fig, ax = plt.subplots(1, 1, figsize=(9, 5))  
sns.kdeplot(df_train[df_train['Survived'] == 1]['Age'], ax=ax)  
sns.kdeplot(df_train[df_train['Survived'] == 0]['Age'], ax=ax)  
plt.legend(['Survived == 1', 'Survived == 0'])  
plt.xlabel('Age')  
plt.ylabel('Survived')  
plt.show()
```

* KDE(Kernel Density Estimation)

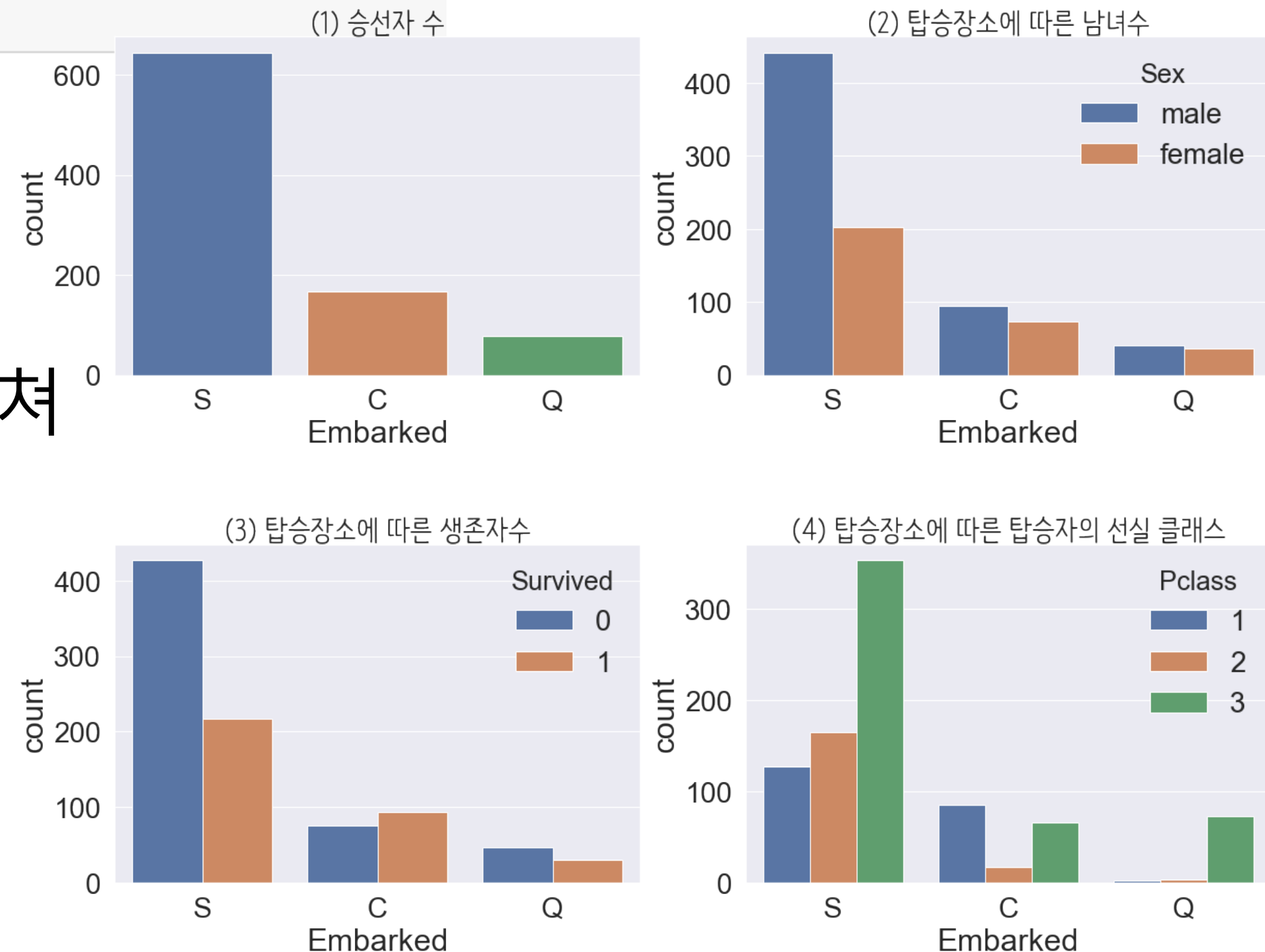
: 커널밀도추정



2. 탐색적 데이터 분석(EDA)

```
f,ax=plt.subplots(2, 2, figsize=(20,15))
sns.countplot('Embarked', data=df_train, ax=ax[0,0])
ax[0,0].set_title('(1) 승선자 수', fontproperties=fontprop)
sns.countplot('Embarked', hue='Sex', data=df_train, ax=ax[0,1])
ax[0,1].set_title('(2) 탑승장소에 따른 남녀수', fontproperties=fontprop)
sns.countplot('Embarked', hue='Survived', data=df_train, ax=ax[1,0])
ax[1,0].set_title('(3) 탑승장소에 따른 생존자수', fontproperties=fontprop)
sns.countplot('Embarked', hue='Pclass', data=df_train, ax=ax[1,1])
ax[1,1].set_title('(4) 탑승장소에 따른 탑승자의 선실 클래스', fontproperties=fontprop)
plt.subplots_adjust(wspace=0.2, hspace=0.5)
plt.show()
```

다소 당연해 보이지 않는 피쳐
탑승장소를 살펴봅시다.



2. 탐색적 데이터 분석(EDA)

모델에 얼마나 큰 영향을 미칠지는 모르겠지만, 그래도
사용하겠습니다. 나중에 모델을 만들고 나면 우리가 사용한 피쳐들이
얼마나 중요한 역할을 했는지 확인해볼 수 있습니다.

앞페이지 그림에서

- 1. 전체적으로 봤을 때, S에서 가장 많은 사람이 탑승했습니다.
- 2. C와 Q는 남녀의 비율이 비슷하고, S는 남자가 더 많습니다.
- 3. S의 경우 생존확률이 많이 낮은 걸 볼 수 있습니다.
- 4. C의 생존확률이 높은건 선실 클래스가 높은 사람이 많이 타서
그리고, S의 생존확률이 낮은건 정 반대의 이유입니다.

2. 탐색적 데이터 분석(EDA)

이제 각 피쳐 간의 상관관계를 한번 보려고 합니다. 두 변수간의 피어슨 상관계수(Pearson correlation coefficient)를 구하면 $(-1, 1)$ 사이의 값을 얻을 수 있습니다. -1 로 갈수록 음의 상관관계, 1 로 갈수록 양의 상관관계를 의미하며, 0 은 상관관계가 없다는 것을 의미합니다. 구하는 수식은 아래와 같습니다.

$$r_{XY} = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sigma_X \sigma_Y}$$

이 값들을 하나의 행렬로 볼수 있게 해주는 것이 히트맵(heatmap)입니다. 공분산(Covariance)이 ± 1 에 가까운 피쳐들은 빼주는 것이 좋기 때문에 히트맵을 살펴보며 피쳐들 사이의 관계를 파악하는 것이 좋습니다.

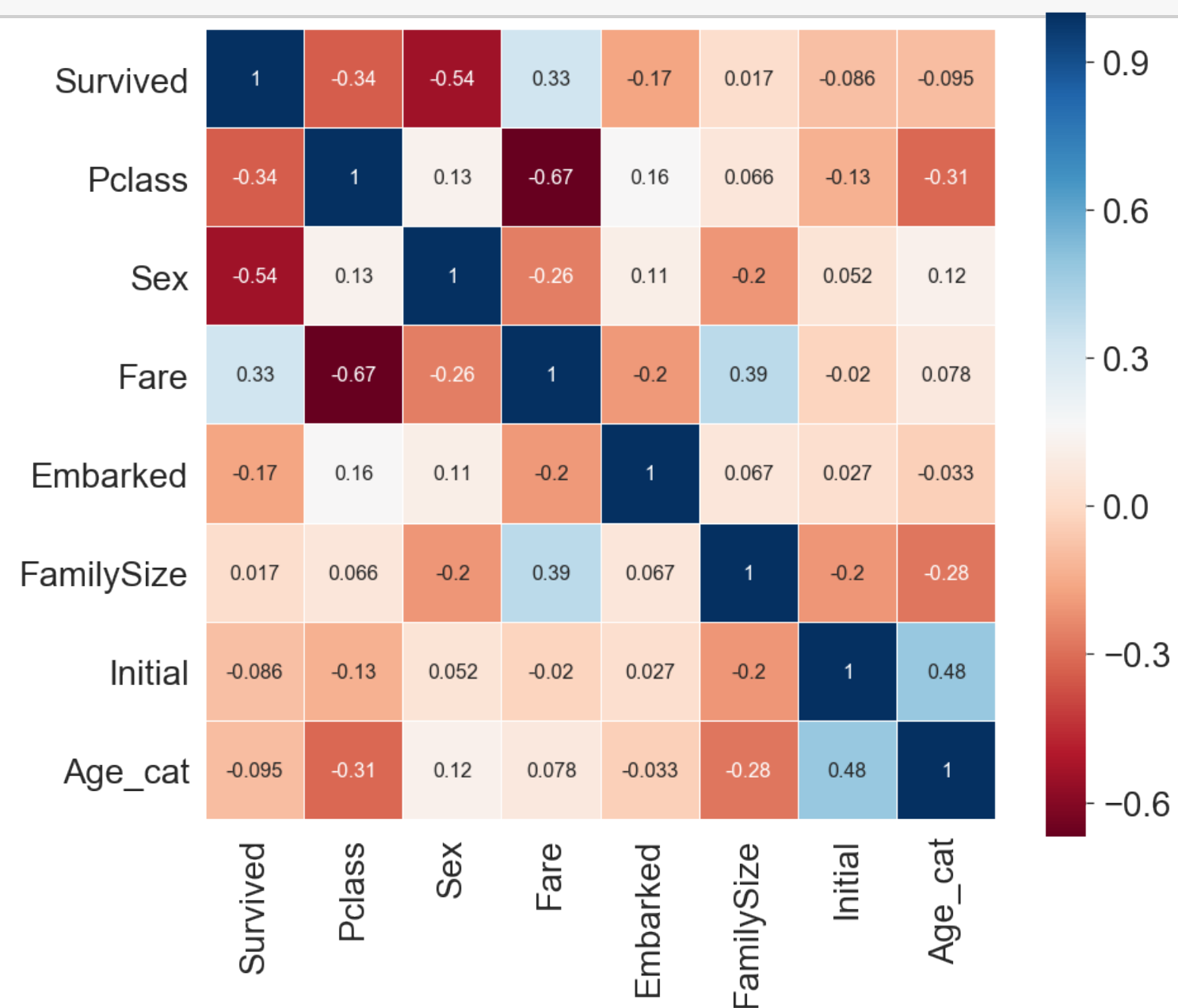
2. 탐색적 데이터 분석(EDA)

```
heatmap_data = df_train[['Survived', 'Pclass', 'Sex', 'Fare', 'Embarked', 'FamilySize', 'Initial', 'Age_cat']]

colormap = plt.cm.RdBu
plt.figure(figsize=(14, 12))
plt.title('Pearson Correlation of Features', y=1.05, size=15)
sns.heatmap(heatmap_data.astype(float).corr(), linewidths=0.1, vmax=1.0,
            square=True, cmap=colormap, linecolor='white', annot=True, annot_kws={"size": 16})

del heatmap_data
```

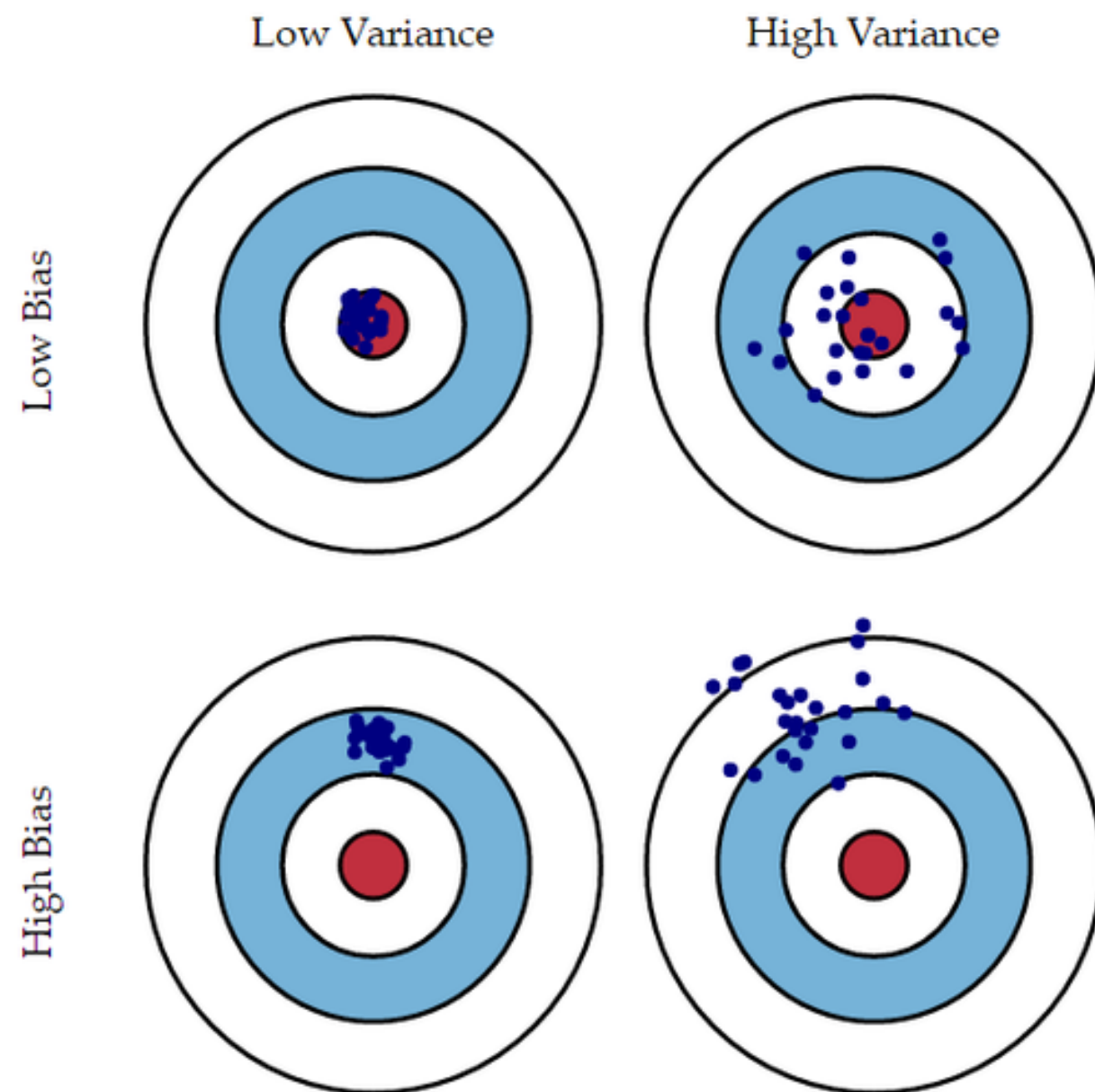
상관관계 히트맵



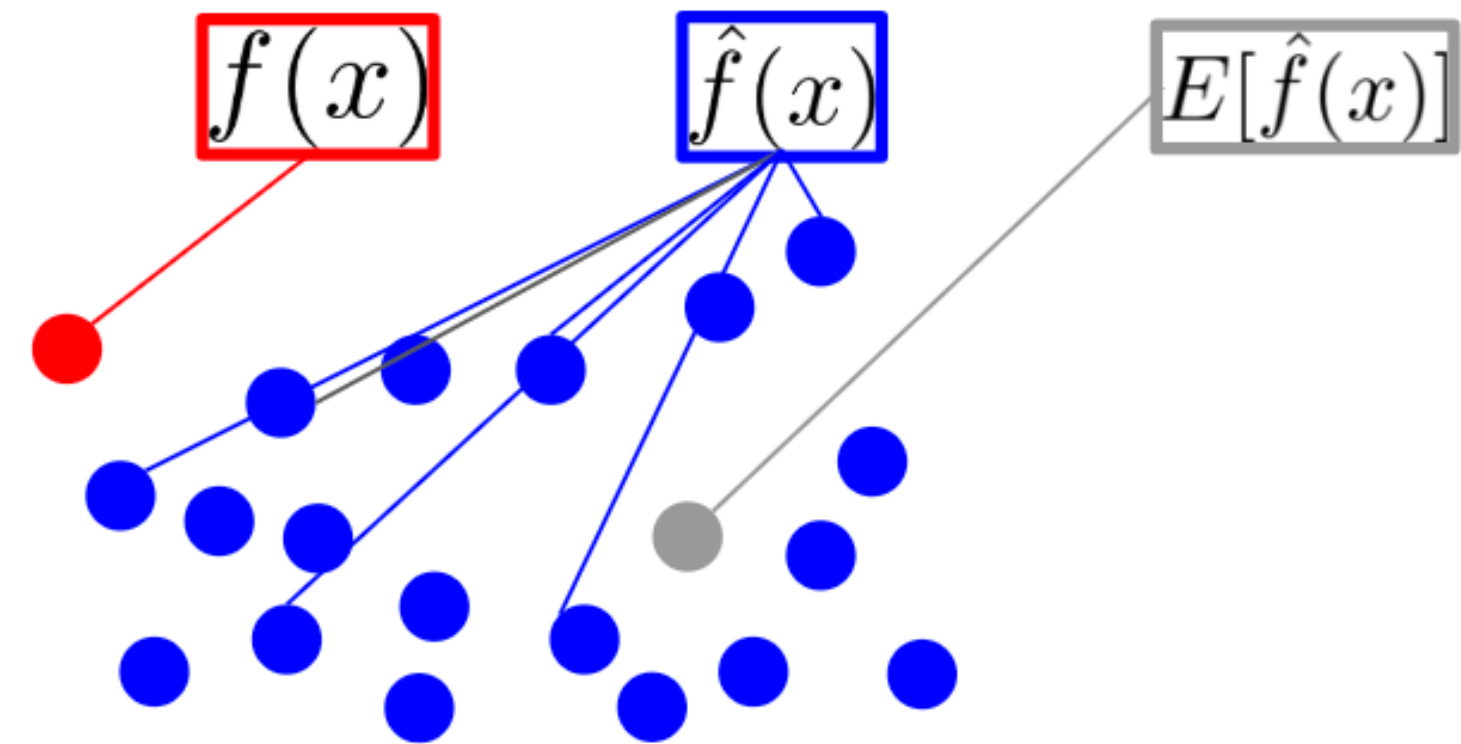
3. 확률과 통계 기본

3-1. 편향(Bias)-분산(Variance) Tradeoff

- 예측값들과 정답이 대체로 멀리 떨어져 있으면 결과의 편향이 크다고 하고, 예측값들이 자기들끼리 대체로 멀리 흩어져 있으면 결과의 분산이 크다고 합니다.



$f(x)$: 정답, $\hat{f}(x)$: 예측값



3-1. 편향(Bias)-분산(Variance) Tradeoff

간단히 정답은 $f(x)$ 대신 t , 예측값은 $\hat{f}(x)$ 대신 f 를 사용하겠습니다. 즉,

$$t = f(\mathbf{x}) + \epsilon$$

이라 하면 다음과 같이 전개 가능합니다.

$$\begin{aligned}\mathbb{E}\{(t - y)^2\} &= \mathbb{E}\{(t - f + f - y)^2\} \\ &= \mathbb{E}\{(t - f)^2\} + \mathbb{E}\{(f - y)^2\} + 2\mathbb{E}\{(f - y)(t - f)\} \\ &= \mathbb{E}\{\epsilon^2\} + \mathbb{E}\{(f - y)^2\} + \underbrace{2[\mathbb{E}\{ft\} - \mathbb{E}\{f^2\} - \mathbb{E}\{yt\} + \mathbb{E}\{yf\}]}_{=0} \\ &= \mathbb{E}\{(f - \mathbb{E}\{y\} + \mathbb{E}\{y\} - y)^2\} + \mathbb{E}\{\epsilon^2\} \\ &= \mathbb{E}\{(f - \mathbb{E}\{y\})^2\} + \mathbb{E}\{(\mathbb{E}\{y\} - y)^2\} + \underbrace{2\mathbb{E}\{(\mathbb{E}\{y\} - y)(f - \mathbb{E}\{y\})\}}_{=0} + \mathbb{E}\{\epsilon^2\} \\ &= \underbrace{\mathbb{E}\{(f - \mathbb{E}\{y\})^2\}}_{\text{bias}^2} + \underbrace{\mathbb{E}\{(\mathbb{E}\{y\} - y)^2\}}_{\text{variance}} + \underbrace{\mathbb{E}\{\epsilon^2\}}_{\text{noise}}\end{aligned}$$

즉, 편향과 분산 사이에는 tradeoff가 존재함을 알 수 있습니다.

3-2. 이산(Discrete) 및 연속(Continuous) 확률분포

1. 이산 확률변수 - pmf(mass)

- 모든 x 에 대해서 $0 \leq p(x) \leq 1$

- $\sum_{i=1}^N p(x_i) = 1$

$$E(X) = \sum_{i=1}^N x_i p(x_i)$$

$$V(X) = \sum_{i=1}^N (x_i - m)^2 p(x_i)$$

$$V(X) = E((X - m)^2) = E(X^2) - m^2$$

2. 연속 확률변수 - pdf(density)

- 모든 x 에 대해서 $0 \leq f(x) \leq 1$

- $\int_{-\infty}^{\infty} f(x) dx = 1$

$$E(X) = \int_{-\infty}^{\infty} x f(x) dx$$

$$V(X) = \int_{-\infty}^{\infty} (x - m)^2 f(x) dx$$

- 이산형에서 확률 $p(x_i)$ 는 연속형에서 $f(x)dx$ 에 대응합니다.

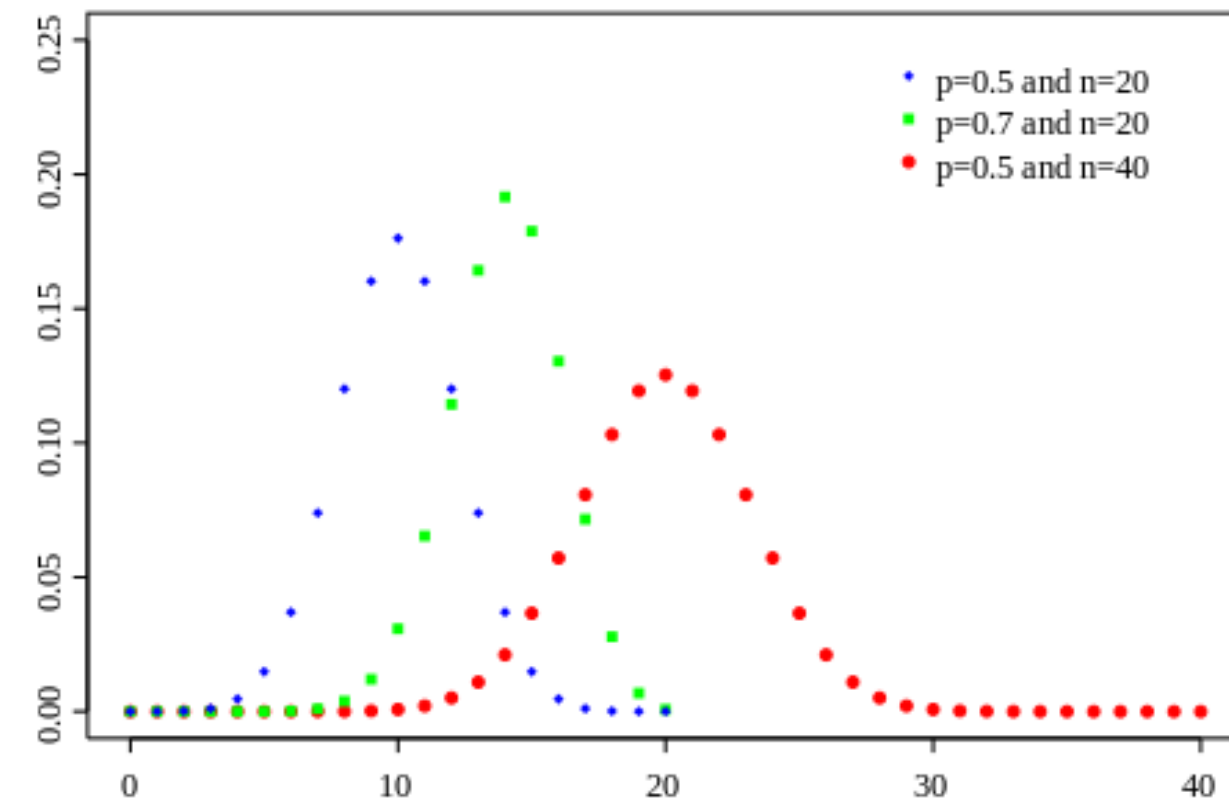
3-3. 이항(Binomial) 분포 및 정규(Normal) 분포

- 이항(Binomial) 분포

$$X \sim B(n, p)$$

$$p(r) = \binom{n}{r} p^r (1-p)^{n-r}$$

$$E(X) = np, \quad V(X) = np(1-p)$$



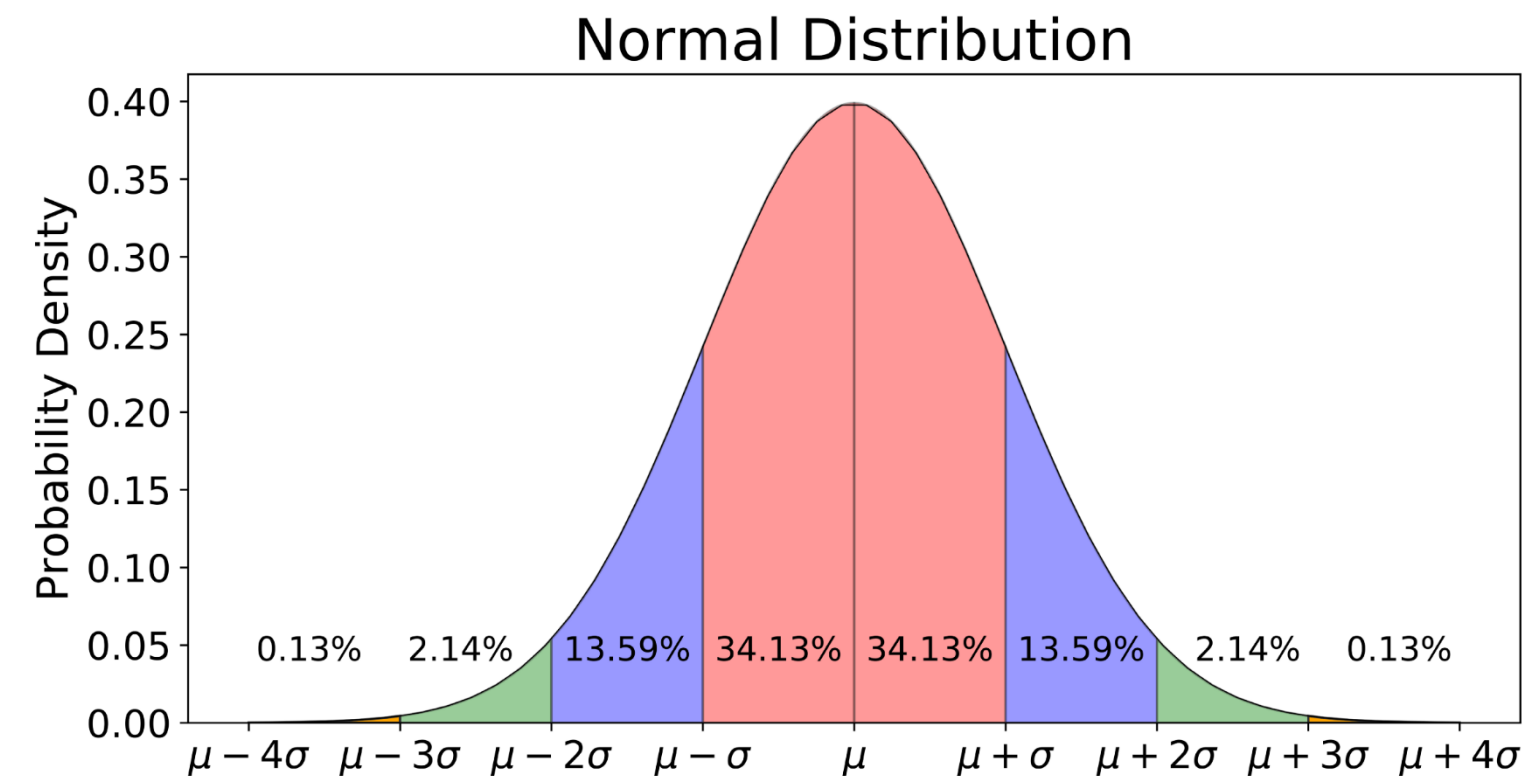
- 정규(Normal, Gaussian) 분포

$$X \sim N(m, \sigma^2)$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

표준화

$$Z := \frac{X - m}{\sigma} \Rightarrow Z \sim N(0, 1)$$



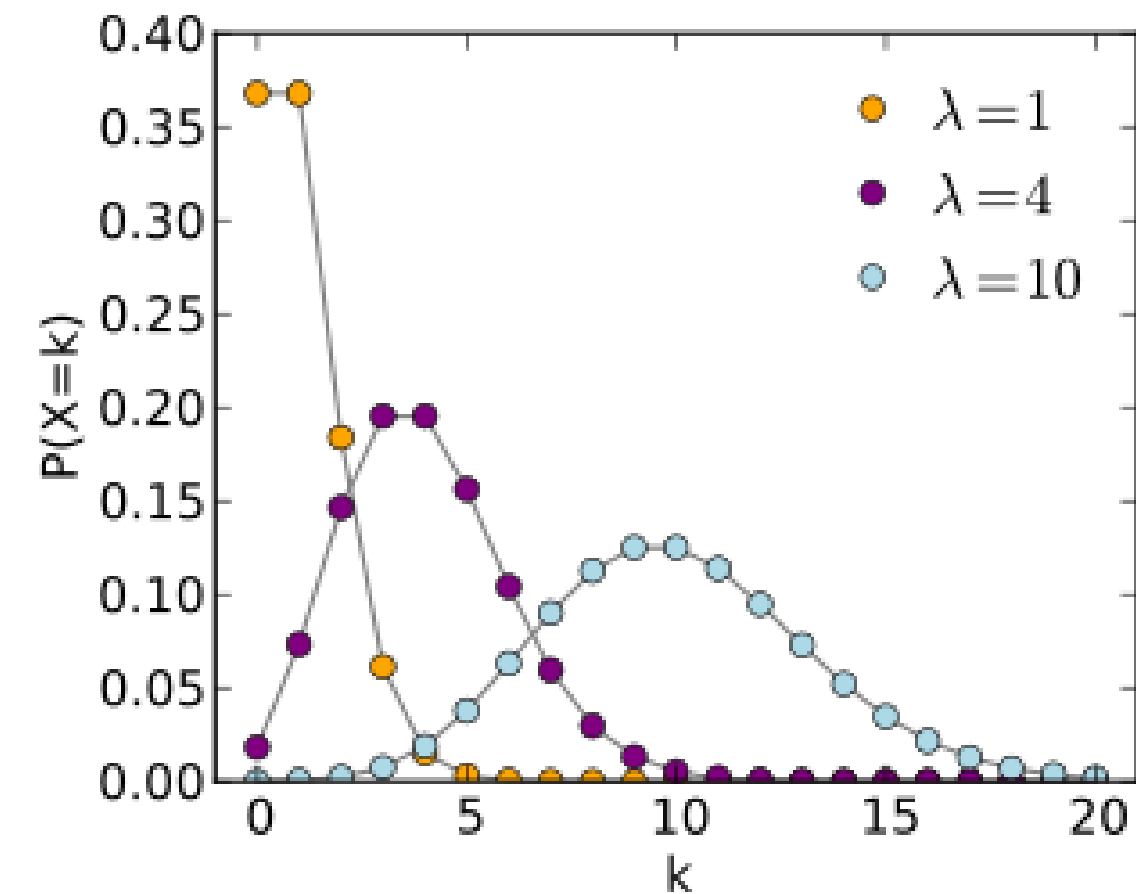
3-4. 포아송(Poisson) 분포

$$X \sim P(\lambda)$$

$$E(X) = V(X) = \lambda$$

단위 시간동안 k 번의 사건이 발생할 확률은

$$p(k) = \frac{\lambda^k e^{-\lambda}}{k!}$$



수학적 모델링에서의 가정

- 두 개 이상의 사건이 동시에 발생할 확률은 0에 가깝다.
- 어떤 단위구간의 사건의 발생은 다른 단위구간의 발생으로부터 독립적이다.
- 특정 구간에서의 사건 발생확률은 그 구간의 크기에 비례한다.

3-5. 빈도주의 및 베이지안(Bayesian) 확률

- **빈도주의 확률**: 기존의 확률 이론은 사건의 빈도수를 분석하기 위해 발전해 왔습니다. 예를 들어 동전 던지거나 카드뽑기와 같은 시행에서, 어떤 사건이 발생할 확률이 p 라고 하는 것은, 그 시행을 무한히 반복했을 때 p 의 비율로 사건이 발생한다는 것을 뜻합니다.
- **베이지안 확률**: 하지만 이런 예는 반복되는 사건이 아니라면 적용하기 어렵습니다. 만일 의사가 환자에게 감기가 걸릴 확률이 40%라고 한다면, 이건 좀 다른 의미를 가집니다. 생물적으로 같은 환자를 무한히 반복하는 것은 말이 안되기 때문입니다. 또한 동전, 카드와 달리 변수들이 굉장히 많죠. 이런 경우에는 **믿음**의 정도를 나타내기 위해 확률을 사용합니다. 이는 실험적으로 베이즈 정리(Bayes' Theorem)를 이용해 구할 수 있습니다.

3-6. 베이즈 정리(Bayes' Theorem)

- 확률 버전

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^n P(B|A_i)P(A_i)}$$

- 확률변수 버전

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)}$$

- 베이즈 정리의 핵심은

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$



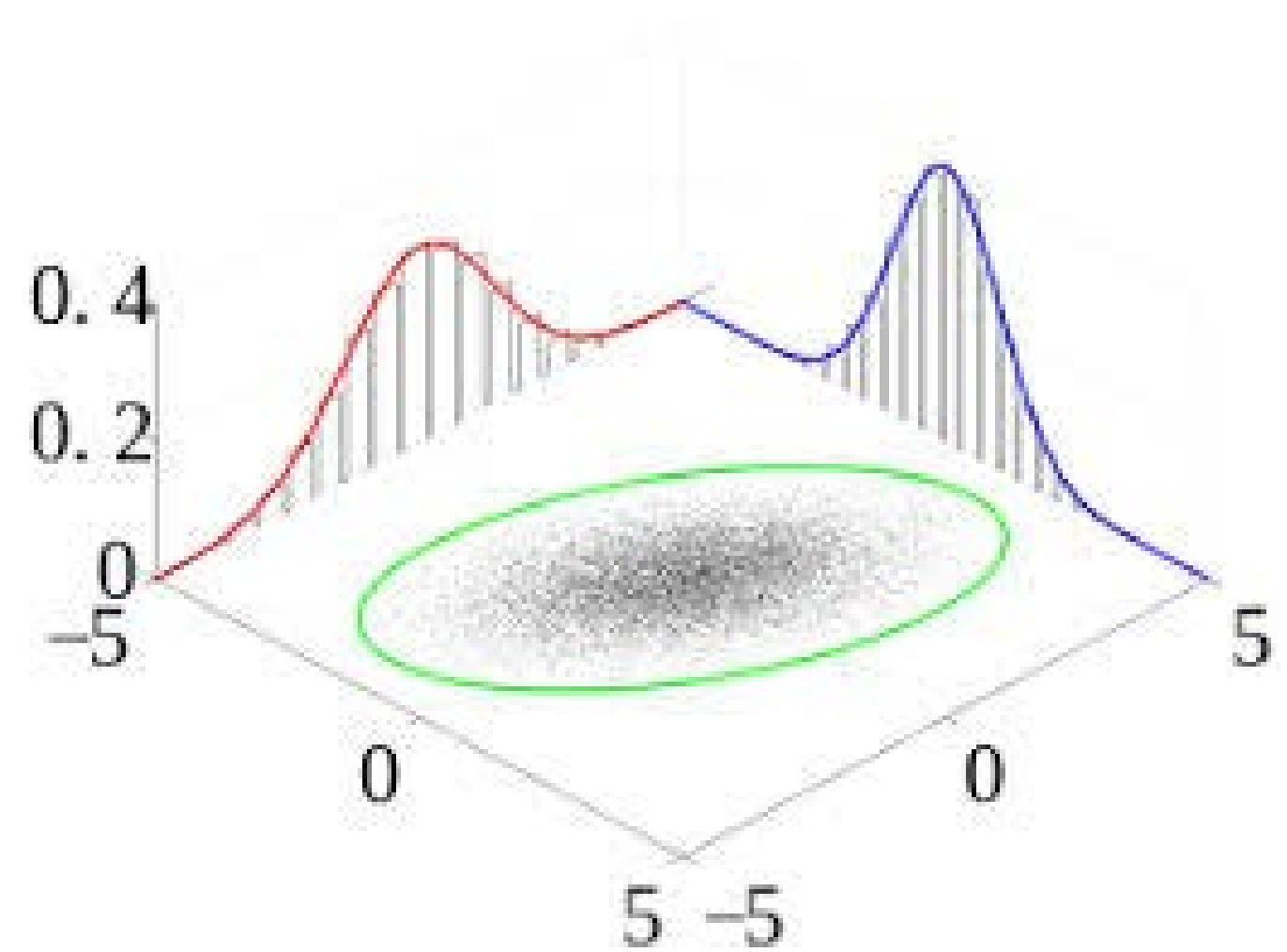
3-7. 결합, 주변, 조건부 확률분포

- 결합(Joint)확률분포 $f(x, y)$ 가 주어져있을 때

$$f_X(x) := \int f(x, y)dy$$

$$f_Y(y) := \int f(x, y)dx$$

를 주변(Marginal) 확률분포라 합니다.



- 조건부(Conditional) 확률분포는 다음과 같이 정의됩니다.

$$f_Y(y|x) := \frac{f(x, y)}{f_X(x)}$$

3-8. 최대우도 추정(MLE)

- MLE(Maximum Likelihood Estimator)는 θ 를 추정하는 방법 중 하나로, likelihood를 최대로 만드는 값으로 선택하는 것입니다.

$$\hat{\theta} = \arg \max_{\theta} f(X|\theta)$$

- 만약 동일한 시행을 독립적으로 반복한다면(i.i.d. : identical and independently distributed)한다면 likelihood는 $f(X|\theta) = \prod_i f(x_i|\theta)$ 가 됩니다.
- MLE는 가장 간단한 추정방법이지만, 관찰에 따라 그 값이 너무 민감하게 변한다는 단점을 가지고 있습니다.

3-9. 최대사후확률 추정(MAP)

- MAP(Maximum A Posterior Estimator)는 θ 를 추정할 때 likelihood가 아니라 사후확률을 최대로 만드는 값으로 선택하는 것입니다.

$$\hat{\theta} = \arg \max_{\theta} f(\theta|X)$$

- prior를 알고 있다면 MLE보다 MAP를 쓰는것이 더 낫습니다.
MLE와 비교해 MAP는 보다 더 자연스러운 결과를 얻습니다. 하지만 오직 지금 주어진 데이터만을 잘 설명하게 될 위험도 있습니다.
- 또한 우리가 옳은 prior를 선택하게 된다면 MLE보다 MAP가 좋겠지만, 잘못된 prior를 선택하게 된다면 오히려 성능이 떨어질 수도 있습니다.

`/* elice */`

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice