train.csv 데이터구성

매우 유명한 titanic 데이터입니다.

배에 승선한 사람들의 정보가 있고 예측하고자 하는 변수는 Survived(0 or 1)입니다.

총 891개의 행, 12개의 열로 구성 되어 있으며, 대략적인 데이터구성은 아래와 같습니다.

	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

데이터분석 과정

1. 데이터 탐색

데이터를 불러오고 데이터에 어떤 변수가 어떤 자료형을 갖고 있는지,

결측치는 얼마나 있는지 알아보는 단계입니다.

별로 알려드릴 건 없지만 csv를 불러오는 것부터 헤매실 수 있으니 tip을 드릴게요!

```
import pandas as pd
import os
os.getcwd()
os.chdir('C:₩₩Users₩₩susie₩₩Desktop')

train=pd.read_csv('train.csv')
```

아래 코드는 결측치를 확인하는 방법이에요! 외워두시고 유용하게 쓰시기 바랍니다.

```
train.isnull().sum()
```

2. 데이터 시각화

하면 좋지만 굳이 안 해도 되는 부분이에요.

다른 방법으로 시각화하셔도 되고, 귀찮다싶으시면 시각화 코드는 제가 드릴게요.

코드 참고해서 그래프가 어떻게 출력되는지 보시면 좋을 것 같습니다.

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

```
sns.set()
```

```
def bar_chart(feature):
    survived=train[train['Survived']==1][feature].value_counts()
                                                    #feature에 따라 생존한 사람 카운트
    dead=train[train['Survived']==0][feature].value_counts()
    df=pd.DataFrame([survived,dead])                #데이터프레임으로 묶고
    df.index=['Survived','Dead']                    #각 열에 인덱스 달아주고
    df.plot(kind='bar', stacked=True, figsize=(10,5)) #차트그리기
```

그래프를 그려주는 함수를 작성했어요. 매개변수는 feature, 즉 변수입니다.
아래처럼 코드를 실행시키면 그래프가 등장해요!!

```
bar_chart('Sex')
```

3. 데이터 전처리

실습 때 text 전처리 되게 오래 걸렸었잖아요. 이것도 마찬가지로 text는 아니지만 전처리가 관건입니다... 하지만 전처리만 하면 nb는 함수 한줄로 해결되니 전처리에 심혈을 기울입시다!!

아래의 표는 각 변수들의 설명과 전처리를 어떻게 해야 하는지 방향입니다.
모든 변수에 일괄적용되는 것은 text를 처리하기 어려워서 text는 모두 숫자화시켰다는 것입니다.
제가 만든 것일 뿐 더 간단한 방법을 쓰셔도 되니 참고 정도 하세요!!

번호	변수명	설명	전처리 방향
1	PassengerId	의미없는 id	변수 삭제
2	Survived	살았느냐? 0=no, 1=yes	전처리 할 필요 x. 모든 전처리가 끝난 후 마지막에 target이라는 변수에 따로 저장할 것 추천
3	Pclass	티켓클래스 1=1등석, 2=2등석, 3=3등석	전처리 할 필요 x
4	Name	이름 Kelly,Mr.James Wilkes,Mrs.James 와 같은 형식	정규표현식을 사용해서 name에서 Mr, Mrs와 같은 정보만 뽑아내서 title이라는 새로운 변수에 저장했어요. title은 text인데 처리하기 힘드니까 각 각 숫자로 바꿔주고 원래 있던 name은 삭제했습니다.
5	Sex	성별. male/female	male=0, female=1로 바꾸었습니다.
6	Age	나이	age에 결측치가 무려 177개가 있다. 177행을 모두 삭제하기 곤란해서 위에서 만든

			title 변수를 이용했어요. title이 같은 것들끼리 그룹을 만들어서, 해당 그룹의 평균값을 age의 Nan 값으로 대체했습니다.
7	SibSp	형제자매 수	둘 다 가족의 수입니다. 두 변수를 합쳐서 familySize라는 새로운 변수를 만들었어요 그리고 원래 있던 두 변수는 삭제했습니다.
8	Parch	부모님 수	
9	Ticket	의미 없는 티켓 고유번호	변수 삭제
10	Fare	요금	그대로 쓰셔도 됩니다. 저는 fare의 범위가 넓어서 ~17이면 0, 18~30이면 1, 31~100이면 2, 100초과이면 3으로 묶어주었어요.
11	Cabin	직원들의 사번 B96, B98, C23, C25처럼 '알파벳+숫자'로 구성	text에다가 결측치도 많아서 가장 힘듭니다 저는 앞에 알파벳만 따오고 그것도 숫자로 바꿔주었어요. 그리고 687개의 결측치는 어떻게 처리해야 할까요?
12	Embarked	어느 선착장에서 승선했는지! C,S,Q가 있는데 각 각 장소를 의미합니다.	text자체로 처리하기 힘드니까 S=0, C=1, Q=2 로 바꿨어요. 결측치는 2개 있는데 S로 채워넣었어요!

아래는 제가 전처리한 후 train.head(5)를 실행한 결과입니다. 달라도 전혀 상관없어요~!

	Survived	Pclass	Sex	Age	Fare	Cabin	Embarked	Title	FamilySize
0	0	3	0	22.0	0.0	2.0	0	0	2
1	1	1	1	38.0	2.0	0.8	1	2	2
2	1	3	1	26.0	0.0	2.0	0	1	1
3	1	1	1	35.0	2.0	0.8	0	2	2
4	0	3	0	35.0	0.0	2.0	0	0	1

4. NB모델 훈련, 예측, 정확도 구하기

제가 강의자료에 올려두었던 Hint가 쓰이는 부분입니다. Hint 여기에도 써둘게요

```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
```

추가적으로 5줄 정도만 더 코딩하면 정확도를 구할 수 있으니 힘내세요!!
참고로 저는 10번의 정확도를 구해서 평균 낸 결과 78.78%가 나왔습니다.

추가적으로 드릴 말씀은

① 전체적인 구성과 분석 방향을 제시해 드렸습니다. 혼자서 차근차근 해보시고, 그래도 못 하겠으면 저에게 갠톡주세요. .ipynb 파일로 과제폼을 드릴게요. 코드 중간 중간 빈칸으로 뚫어 놓고 채워가는 방식입니다.

② 지난 강의 때 제가 제대로 대답하지 못 한 '언더플로우'에 대해서 익명의 10기분이 아주 잘 설명해 주셔서 그것으로 저의 대답을 갈음합니다. 다음부터 수업 때 알고 계시면 같이 도와주세요 ㅜㅜ

> 컴퓨터의 경우 0,1로 수를 구분하며 연산을 하는데 16비트 연산, 32비트 연산, 64비트 연산 등이 있으며 2018년 보급형 컴퓨터는 대부분 64비트 연산을 합니다. 이는 0,1이 들어갈 수 있는 공간이 64개라는 뜻이며, 제일 앞의 공간은 표현하려는 수의 부호를 의미하고 나머지는 그 값을 표현합니다. 이러한 표현법도 여러 가지가 있습니다.
>
> 이렇게 표현을 할 경우 10진수로 변환을 거치면 일정 소수점 이상은 잘리게 되어있습니다. double형은 소수점 14자리, decimal은 소수점 28가지 표현이 되며 0.0000으로 0이 14개 이상일 경우 소수점 자리를 더 이상 확인 할 수 없게 됩니다. 이런 현상을 underflow가 발생했다고 합니다. 0이 14개가 넘어가서 무조건 0으로 표시되는 문제가 발생하게 되는 것으로, 크기 구별이 불가능해집니다.
>
> 반대 개념으로는 overflow가 있으며 수가 2^64이나 2^32을 넘어가면 수를 알 수 없게 되는 현상이 있습니다. 때문에 log함수를 사용하여 수치를 크게 확대시키며 크기 구별을 가능하게 하는 것으로 보입니다.

③ 제 강의에 대한 피드백을 봤습니다. 모든 답변 토씨 하나 허투루 보지 않았습니다. 전부 다 제가 듣기에 과분한 칭찬들이더군요ㅠㅠ 좋은 말씀 해주신 분들 너무 감사해요ㅠ 진짜 감동받아서 힘들 때 마다 그거 보려구요ㅎㅎ 앞으로 더 열심히 하겠습니다 ☺