```python
#!/usr/bin/env python

# -*- coding: utf-8 -*-
# 코드 내부에 한글을 사용가능 하게 해주는 부분입니다.

# pandas 라이브러리를 불러옵니다.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 피마 인디언 당뇨병 데이터셋을 불러옵니다. 불러올 때 각 컬럼에 해당하는 이름을 지정합니다.
df = pd.read_csv('../../dataset/pima-indians-diabetes.csv',
                 names = ["pregnant", "plasma", "pressure", "thickness", "insulin", "BMI", "pedigree", "age", "class"])

# 처음 5줄을 봅니다.
print(df.head(5))

# 데이터의 전반적인 정보를 확인해 봅니다.
print(df.info())

# 각 정보별 특징을 좀더 자세히 출력합니다.
print(df.describe())

# 데이터 중 임신 정보와 클래스 만을 출력해 봅니다.
print(df[['plasma', 'class']])
```

데이터 전처리 과정
(결과에 많은 영향이 큰 항목 발견)

* CSV : Comma Separated Value File → 콤마로 구분된 데이터들의 묶음.

* CSV 파일에는 데이터를 묘사해 주는 한 줄이 파일 맨 처음에 나오는데 → header
이 파일엔 헤더가 없으므로 names라는 항목을 통해 속성별 키워드 지정.

각 정보의 형식

정보명, 샘플 수, 평균, 표준편차, 최솟값, 백분위 수, 최댓값

일부 컬럼만 보기

```python
# 데이터 가공학기
print(df[['pregnant','class']].groupby(['pregnant'],as_index=False).mean().sort_values(by='pregnant',ascending=True))
```

가공

이 속성을 기준으로 묶음 → 새로운 인덱스 만들기

pregnant 속성을 기준으로 오름차순 정렬

평균구하라

```python
# 데이터 간의 상관관계를 그래프로 표현해 봅니다.

colormap = plt.cm.gist_heat   #그래프의 색상 구성을 정합니다.
plt.figure(figsize=(12,12))   #그래프의 크기를 정합니다.
# 그래프의 속성을 결정합니다. vmax의 값을 0.5로 지정해 0.5에 가까울 수록 밝은 색으로 표시되게 합니다.
sns.heatmap(df.corr(),linewidths=0.1,vmax=0.5, cmap=colormap, linecolor='white', annot=True)
plt.show()

grid = sns.FacetGrid(df, col='class')
grid.map(plt.hist, 'plasma', bins=10)
plt.show()
```

matplotlib로 그래프 표현

항목간의 상관 관계 표현

색상 밝기 조절

미지정하면 matplotlib 색상의 default 값 불러옴

두 항목 관계를 그래프로 표현

* heatmap()
- 두 항목을 짝 지은 뒤 각각 어떤 패턴으로 변화하는지 관찰하는 함수
→ 두 항목이 전혀 다른 패턴으로 변화할수록 0에 가깝
→ 서로 비슷한 패턴으로 변화할수록 1에 가까움

* 숫자가 1에 가까울수록 밝은 색상.

```python
# 딥러닝을 구동하는 데 필요한 케라스 함수를 불러옵니다.
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# 필요한 라이브러리를 불러옵니다.
import numpy
import tensorflow as tf

# 실행할 때마다 같은 결과를 출력하기 위해 설정하는 부분입니다.
numpy.random.seed(3)
tf.random.set_seed(3)

# 데이터를 불러 옵니다.
dataset = numpy.loadtxt("../dataset/pima-indians-diabetes.csv", delimiter=",")
X = dataset[:,0:8]
Y = dataset[:,8]

# 모델을 설정합니다.
model = Sequential()
model.add(Dense(12, input_dim=8, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

# 모델을 컴파일합니다.
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

# 모델을 실행합니다.
model.fit(X, Y, epochs=200, batch_size=10)

# 결과를 출력합니다.
print("₩n Accuracy: %.4f" % (model.evaluate(X, Y)[1]))
```

케라스를 이용해 당뇨병 예측 실행

넘파이 라이브러리를 사용하면서 텐서플로 기반으로 딥러닝 구현 할때 일정한 값을 얻기 위해 넘파이 seed 값과 텐서플로 seed 값을 모두 설정해야 함!!

층 추가

들 중 하나를 결정 → 이항 분류 사용

샘플 200번 반복 한번에 입력되는 입력값 10개

ramdom.seed( )

ramdom( ) 함수를 써도 내부적으로 랜덤테이블 중 하나를 불러내 표를 순서대로 보여주는 것임.

seed가 같으면 같은 랜덤테이블을 사용.
└ 몇번째 랜덤 테이블을 사용할지.