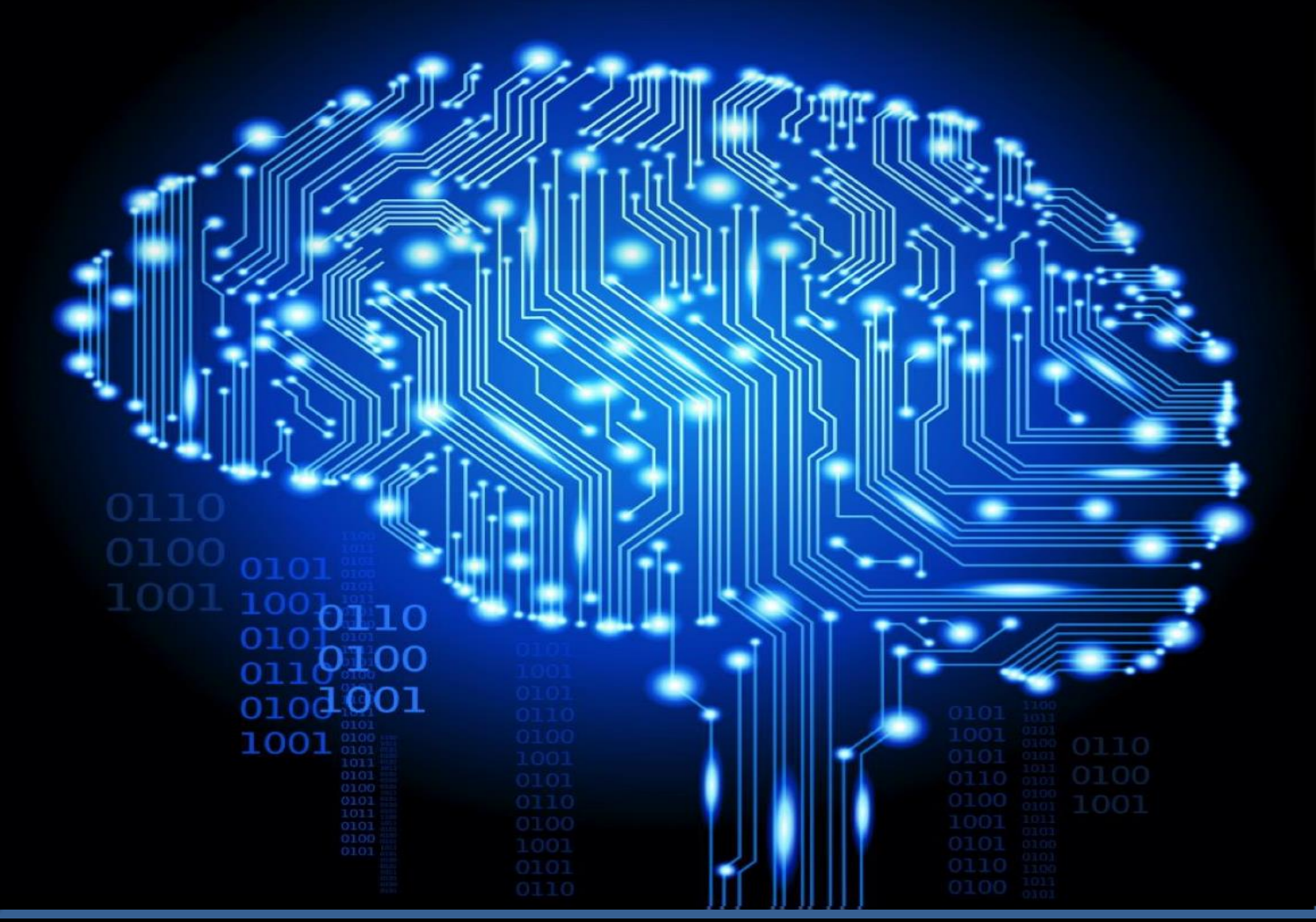


Statistical Learning and Computational Finance Lab.
Department of Industrial Engineering
http://slcf.snu.ac.kr

빅데이터 분석 실습 3: Data exploration with Iris

Iris dataset

- Iris(붓꽃) 데이터셋은 붓꽃의 품종을 나타내는 세 개 클래스(Iris setosa, Iris virginica, Iris versicolor)의 샘플을 50개씩 고르게 가지고 있음 → from sklearn.datasets import load_iris
- Shape: (150, 5), no missing value
- column description

꽃받침

꽃잎

0: setosa
1: virginica
2: versicolor

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

```
df.shape
```

```
(150, 5)
```

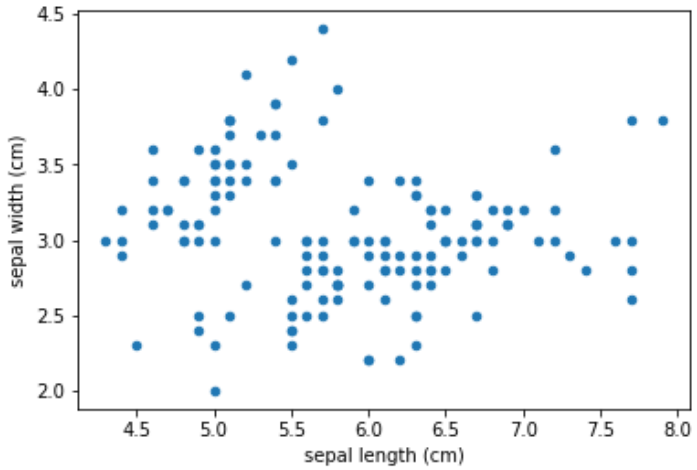
```
df.describe()
```

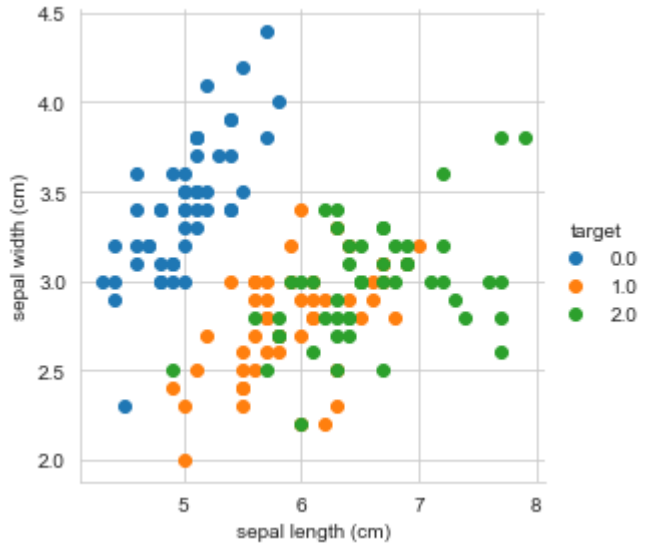
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

Scatter plot

- 산점도: 두 변수 간 관계를 나타내는 그래프 방법
- **import** matplotlib.pyplot **as** plt
- **import** seaborn **as** sns
- 등의 모듈을 사용하여 데이터프레임 정보를 한 눈에 볼 수 있음 → 시각화 수업에서 자세히!

```
df.plot(kind="scatter",x="sepal length (cm)",y="sepal width (cm)")
plt.show()
```





Data exploration

Rename column names

```
df.rename(columns={df.columns[0] : 'SL',
                   df.columns[1] : 'SW',
                   df.columns[2] : 'PL',
                   df.columns[3] : 'PW',
                   df.columns[4] : 'Y'}, inplace = True)
df.head()
```

	SL	SW	PL	PW	Y
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

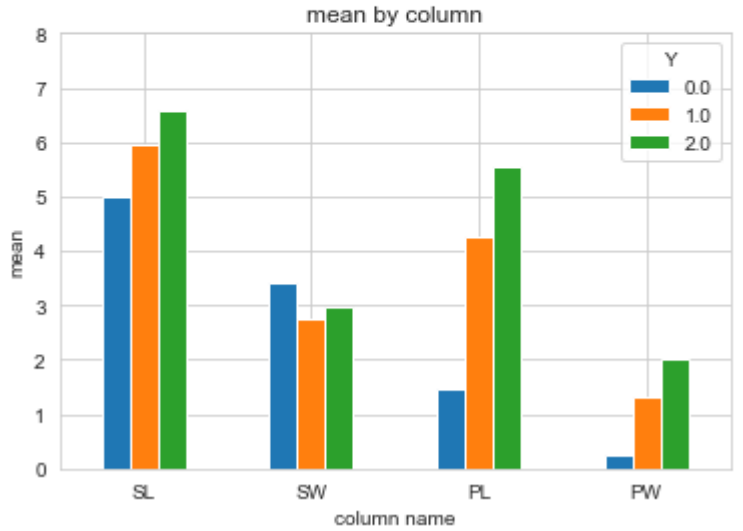
Column 별 평균

```
st = df.groupby(df.Y).mean() # Y를 기준으로 그룹화를 하여 각 그룹의 평균을 구하여 준다.
st.columns.name = "변수" # columns의 이름을 "변수"로 지정한다.
st
```

변수	SL	SW	PL	PW
Y				
0.0	5.006	3.428	1.462	0.246
1.0	5.936	2.770	4.260	1.326
2.0	6.588	2.974	5.552	2.026

barplot

```
st.T.plot.bar(rot=0) # rot : x축 변수명의 기울기
plt.title("mean by column")
plt.xlabel("column name")
plt.ylabel("mean")
plt.ylim(0,8)
plt.show()
```



불균형한 클래스 다루기

- 타깃 벡터가 매우 불균형한 클래스로 이루어진 경우
- 잘 동작하지 않는다면 모델에 내장된 클래스 가중치 매개변수를 사용하거나 다운샘플링이나 업샘플링을 고려해 볼 수 있음
- 불균형 데이터셋을 만들기 위해 Iris setosa 샘플 50개 중 40개를 삭제 → shape: (110, 5)
- Iris setosa 샘플(클래스 0) 10개와 Iris setosa가 아닌 샘플(클래스 1) 100개로 이루어진 불균형 데이터 생성

꽃받침 / 꽃잎 / 0: setosa, 1: not setosa

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0.0
1	4.9	3.0	1.4	0.2	0.0
2	4.7	3.2	1.3	0.2	0.0
3	4.6	3.1	1.5	0.2	0.0
4	5.0	3.6	1.4	0.2	0.0

Balanced dataset

```
iris_df['target'].value_counts()

1.0    100
0.0     10
Name: target, dtype: int64
```

Imbalanced dataset

RandomForestClassifier

- 사이킷런에 있는 많은 알고리즘은 훈련할 때 불균형한 영향을 줄일 수 있도록 클래스에 가중치를 부여할 수 있는 매개변수를 제공
- RandomForestClassifier는 class_weight 매개변수를 가진 인기 높은 분류 알고리즘
- 매개변수값에 원하는 클래스 가중치를 직접 지정할 수 있음
- 또는 balanced로 지정하여 클래스 빈도에 반비례하게 자동으로 가중치를 만들 수 있음

```
# 가중치를 만듭니다.
weights = {0: .9, 1: 0.1}

# 가중치를 부여한 랜덤 포레스트 분류기를 만듭니다.
RandomForestClassifier(class_weight=weights)
```

```
RandomForestClassifier(class_weight={0: 0.9, 1: 0.1})
```

```
# 균형잡힌 클래스 가중치로 랜덤 포레스트 모델을 훈련합니다.
RandomForestClassifier(class_weight="balanced")
```

```
RandomForestClassifier(class_weight='balanced')
```

Upsampling, downsampling

Source: https://www.researchgate.net/figure/Downsample-flow-left-and-Upsample-flow-right_fig1_337303049

- Downsampling: 다수 클래스의 샘플을 줄임 / Upsampling: 소수 클래스의 샘플을 늘림
- 다운샘플링에서는 다수 클래스(즉, 더 많은 샘플을 가진 클래스)에서 중복을 허용하지 않고 랜덤하게 샘플을 선택하여 소수 클래스와 같은 크기의 샘플 부분집합을 만들어 줌
- 업샘플링에서는 다수 클래스의 샘플만큼 소수 클래스에서 중복을 허용하여 랜덤하게 샘플을 허용하여 랜덤하게 샘플 선택

불균형한 데이터셋 다루기

- 실전에는 불균형한 클래스가 아주 많음
- 가장 좋은 방법은 소수 클래스의 샘플을 더 많이 모으는 것. 하지만 불가능한 경우가 많기 때문에 다른 선택 사항을 고려해야 함
- 불균형한 데이터셋을 다루는 경우, 그에 잘 맞는 모델 평가 지표를 사용해야 함.
  - 정확도(accuracy)는 모델 성능을 평가하는데 자주 사용되는 평가 지표
  - 클래스가 불균형할 때 정확도 설명력 떨어짐
  - 예를 들어 희귀한 암을 가진 샘플이 0.5%라면 아무도 암에 걸리지 않았다고 예측하는 단순한 모델도 99.5%의 정확도를 얻을 것임
  - 더 나은 지표로 오차 행렬, 정밀도, 재현율, F1점수, ROC곡선 등이 있음

불균형한 데이터셋 다루기

- 일부 모델에서 제공하는 클래스 가중치 매개변수 사용
  - 알고리즘이 불균형한 클래스를 조정할 수 있음
  - 사이킷런에 있는 많은 분류기들은 이에 적합한 class_weight 매개변수를 가지고 있음
- 샘플링
  - 다운샘플링에서 소수 클래스 크기와 동일하게 다수 클래스의 랜덤한 부분집합 추출
  - 업샘플링에서는 다수 클래스 크기와 동일하게 소수 클래스로부터 중복을 허용하여 반복적으로 샘플 추출
  - 다운샘플링과 업샘플링 중 어떤 것을 사용할지 여부는 문제에 따라 달라짐
  - 일반적으로 두 전략을 모두 시도해보고 더 나은 결과를 내는 것을 선택