

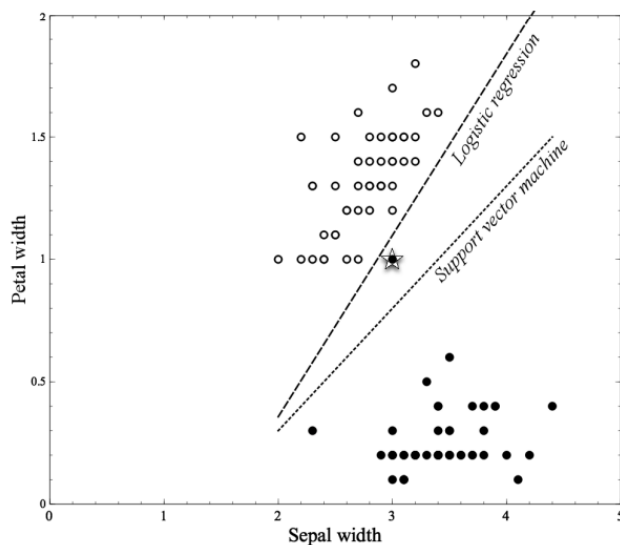
1. key strategic assets in data science 빈칸채우기

( ) and the capability to extract useful ( ) from data

2. Decision tree 보고 IG 구해서 어떤 변수로 분류할 것인지 구하기

3. Classification, Regression, Similarity matching, Clustering, Association rule discovery, Profiling, Link Prediction, Data reduction, Casual modeling에 대한 설명만 주어진 후 어떤 data mining task인지 작성

4. Logistic regression, svm의 결정경계를 보여준 뒤, 이를 비교 및 평가



5. logistic regression의 cost function과  $H(x)$ 값의 관계 서술

[파이썬 코드]

6. 코드 결과 작성하기 -> 강의안 파이썬 리스트, 튜플, 딕셔너리, 넘파이 등

```
# 항목 추가
capitals = {}
capitals["Korea"] = "Seoul"
capitals["USA"] = "Washington"
capitals["UK"] = "London"
capitals["France"] = "Paris"
print(capitals)
```

```
{'Korea': 'Seoul', 'USA': 'Washington', 'UK': 'London', 'France': 'Paris'}
```

```
y = ages > 20
y
```

```
array([False, False,  True,  True,  True])
```

```
# 파이썬 리스트 슬라이싱과 넘파이 스타일 슬라이싱 비교
np_array = np.array([[ 1,  2,  3,  4],
                     [ 5,  6,  7,  8],
                     [ 9, 10, 11, 12],
                     [13, 14, 15, 16]])
print(np_array[:,2])
print(np_array[:,2][:,2]) # Nested 인덱싱, 슬라이싱, 첫 슬라이싱: 0행, 2행 선택, 두 번째 슬라이싱: 그 중 0행 선택
print(np_array[:,2, ::2]) # 행 슬라이싱: 0행, 2행 선택, 열 슬라이싱: 0열 2열 선택
print(np_array[1::2])
print(np_array[1::2, 1::2])
```

```
[[ 1  2  3  4]
 [ 9 10 11 12]]
[[1 2 3 4]]
[[ 1  3]
 [ 9 11]]
[[ 5  6  7  8]
 [13 14 15 16]]
[[ 6  8]
 [14 16]]
```

7. 넘파이 코드 빈칸 채우기 -> 슬라이싱 코드 채우기

[웨카]

8. 특정 변수를 클릭한 창을 보여주고 해당 변수의 타입 및 평균 작성

9. 산점도 보고 어떤 변수를 선택하는 것이 class 구분에 용이한지 작성

10. confusion matrix를 보고 정확도를 구하고 b클래스로 예측된 데이터의 개수를 작성