3주. Machine Learning Concept			
학번	32153180	이름	이상민

Q1. 전통적인 SW 와 머신러닝을 적용한 SW 의 차이점을 설명하시오

전통적인 SW에서는 인간이 규칙을 알아내어 알고리즘의 형태로 SW 안에 구현한다. 기상 예측을 예로 들면 규칙을 구현해놓고 현재의 날씨를 입력하면 내일의 날씨를 예측하는 형태로 이용한다.
머신러닝을 적용한 SW에서는 머신이 규칙을 찾는 방식이다. 규칙을 알아내는 방법을 인간이 제시해주고, 실제로 규칙을 알아내는 과정 자체는 머신이 학습을 통해 진행한다.

Q2. 머신러닝에서 러닝(learning)의 실제적인 의미를 설명하시오.

설명변수와 반응변수로 구성된 데이터셋을 학습 방법에 집어 넣어 예측 모델을 구축하는 과정이다. 머신 입장에서는 러닝이고 인간 입장에서는 훈련이다. 과거의 주식 변동 데이터를 학습하여 주가를 예측하거나 건강검진 데이터를 학습하여 암 발생률을 예측하는 등이 있다.

Q3. 머신러닝이 가능한 이유를 설명하시오.

머신러닝은 과거의 데이터를 학습하여 미래를 예측하는 기술로 반복된 학습을 통해 규칙을 찾고 그것을 바탕으로 예측한다. 따라서 학습 데이터가 많을수록 머신러닝에 유리하다.

Q4. 회귀(regression) 와 분류(classification)의 차이점을 설명하시오

회귀와 분류 모두 설명변수, 반응변수가 존재하는 지도학습이다. 회귀는 답이 수치형으로 크기가 있고 대소비교가 가능한 반면 분류는 답이 범주형 자료로 주어진다.

Q5. 기후변화에 따른 연평균 기온을 예측하는 머신러닝 모델을 만들려고 한다. (2점)
1) 모델을 만들기 위해 필요한 것은 무엇인가
2) 이 모델은 회귀, 분류, 군집화, 강화학습중 어느 기술을 적용해야 하는가? 그 이유는 무엇인가

모델을 만들기 위해 과거 데이터셋이 필요하다. 반응변수는 연평균 기온이며 설명변수는 바람, 온도 등 기후와 관련된 데이터로 이루어져야 한다.
모델은 수치형인 기온을 예측하는 것이기 때문에 지도학습 중 회귀 방법을 적용해야 한다.

Q6. 머신러닝 모델을 개발할 때 데이터셋을 training data 와 test data 로 나누는 이유는 무엇인가? 나누지 않는다면 어떤 문제가 발생하는가

주로 모델 학습을 위해 training data를 이용하고, 모델을 평가하기 위해 test data를 이용한다. 머신러닝은 미래를 예측하는 것이 목표인데 미래의 데이터란 존재하지 않기 때문에 데이터셋을 7(train) : 3(test) 정도 비율로 나눠 테스트한다.
이렇게 나누지 않는다면 모델이 한정된 데이터에만 치우쳐 과적합(overfitting)이 발생한다.

Q7. scikit-learn 홈페이지(https://scikit-learn.org/stable/)를 방문하여 scikit-learn에서 제공하는 군집화(clustering) 일고리즘에는 어떤 것들이 있는지 찾아서 제시하시오

K-Means : 주어진 데이터를 K개의 클러스터로 묶는 알고리즘
Affinity propagation : 모든 데이터가 특정한 기준에 따라 자신을 대표할 데이터 선택
Mean-shift : 슬라이딩 윈도우에 기반한 알고리즘으로 가장 밀도 높은 지역을 찾음
Spectral clustering : 그래프 기반의 군집화 기법
Ward hierarchical clustering : 두 군집 간 유사성을 오차 제곱합에 기반해서 측정
Agglomerative clustering : 시작할 때 각 포인트를 하나의 클러스터로 지정하고, 그 다음 종료 조건을 만족할 때까지 가장 비슷한 두 클러스터를 합침
DBSCAN : 데이터가 밀집한 정도 즉 밀도를 이용하여 측정
OPTICS : DBSCAN의 일반화
Gaussian mixtures : 가우시안 분포가 여러개 혼합된 클러스터링 알고리즘
Birch : 주어진 메모리 크기 안에서 클러스터링을 효율적으로 할 수 있는 알고리즘

Q8. Pandas 모듈을 이용하여 배포된 데이터셋중 cars 데이터셋을 읽어온 후 다음 문제를 해결하시오 (2점)
(1) 데이터셋의 위쪽 5행을 보이시오
(2) 데이터셋의 컬럼들 이름을 보이시오
(3) 데이터셋의 두 번째 컬럼의 값들만 보이시오.
(4) 데이터셋의 11~20행 자료중 speed 컬럼의 값들만 보이시오.
(5) speed 가 20 이상인 행들의 자료만 보이시오
(6) speed 가 10 보다 크고 dist 가 50보다 큰 행들의 자료만 보이시오.
(7) speed 가 15 보다 크고 dist 가 50보다 큰 행들은 몇 개인지 보이시오

Source code :

```
import pandas as pd
data = pd.read_csv("C:/Users/sangmin/Desktop/학교생활/4-2/딥러닝클라우드
/dataset_0910/cars.csv")

# (1)
data.head(5)
# (2)
data.columns
# (3)
data.iloc[:, 1]
# (4)
data.iloc[11:21, 0]
# (5)
data[data['speed'] >= 20]
# (6)
speed = data['speed'] > 10
dist = data['dist'] > 50
data[speed & dist]
# (7)
speed = data['speed'] > 15
dist = data['dist'] > 50
data[speed & dist].shape[0]
```

실행화면 캡쳐:

```
In [49]: data.head(5)
Out[49]:
   speed  dist
0      4     2
1      4    10
2      7     4
3      7    22
4      8    16
```

```
In [50]: data.columns
Out[50]: Index(['speed', 'dist'], dtype='object')
```

```
In [51]: data.iloc[:, 1]
Out[51]:
0       2
1      10
2       4
3      22
4      16
5      10
6      18
7      26
8      34
9      17
10     28
11     14
12     20
13     24
14     28
15     26
16     34
17     34
18     46
19     26
20     36
21     60
22     80
23     20
24     26
25     54
26     32
27     40
28     32
29     40
30     50
```

```
31     42
32     56
33     76
34     84
35     36
36     46
37     68
38     32
39     48
40     52
41     56
42     64
43     66
44     54
45     70
46     92
47     93
48    120
49     85
Name: dist, dtype: int64
```

```
In [52]: data.iloc[11:21, 0]
Out[52]:
11    12
12    12
13    12
14    12
15    13
16    13
17    13
18    13
19    14
20    14
Name: speed, dtype: int64
```

```
In [53]: data[data['speed'] >= 20]
Out[53]:
    speed  dist
38     20    32
39     20    48
40     20    52
41     20    56
42     20    64
43     22    66
44     23    54
45     24    70
46     24    92
47     24    93
48     24   120
49     25    85
```

```
In [54]: speed = data['speed'] > 10
    ...: dist = data['dist'] > 50
    ...: data[speed & dist]
Out[54]:
    speed  dist
21     14    60
22     14    80
25     15    54
32     18    56
33     18    76
34     18    84
37     19    68
40     20    52
41     20    56
42     20    64
43     22    66
44     23    54
45     24    70
46     24    92
47     24    93
48     24   120
49     25    85
```

```
In [55]: speed = data['speed'] > 15
    ...: dist = data['dist'] > 50
    ...: data[speed & dist].shape[0]
Out[55]: 14
```