3주. Machine Learning Concept			
학번	32170578	이름	김산

Q1. 전통적인 SW 와 머신러닝을 적용한 SW 의 차이점을 설명하시오

전통적인 SW :
- 필요한 동작 과정에 대한 규칙을 인간이 알아내어 알고리즘의 형태로 소프트웨어 안에 구현
머신러닝 SW :
- 규칙을 알아내는 방법을 인간이 제시만 하고 실제 규칙을 알아내는 과정은 머신이 진행한다.

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

설명 변수(Y), 반응변수(X)간의 관계를 찾는 것을 Learning이라고 한다.
- 과거의 주식 변동 데이터 학습 -> 주가 예측
- 건강검진 데이터 학습 -> 간암 발생률 추이 예측
- 과거 대출 및 회수 데이터 학습 -> 대출 신청자의 상환가능성 예측

Q3. 머신러닝이 가능한 이유를 설명하시오.

머신러닝은 과거의 데이터를 통해 숨겨진 규칙을 찾아내는 과정을 머신이 수행한다, 즉 학습을 위해서는 데이터가 많은 환경이 유리하다. 정보화가 됨에 따라 데이터의 양이 크게 증가하게 되었고 이에따라 학습을 위한 데이터의 양이 많아져 머신러닝에 유리한 조건이 되었다.

머신은 인간이 데이터와, 학습 알고리즘을 제공하면, 머신 스스로 데이터를 토대로 예측 모델을 만들어 머신러닝이 가능해진다.

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

regression : 연속된 값을 예측하는것
classification : 데이터의 종류를 예측하는것

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

1) 과거의 연평균 기온 데이터
2) 연평균 기온의 경우 어느 그룹에 속하는 것이 아니라 숫자이므로 회귀 기술을 적용해야 함

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

나누지 않는다면, 모델의 성능을 평가하기 위해서는 미래의 데이터를 가지고 판단해야 하는데 미래의 데이터는 예측을 한다 하더라도 맞는 결과가 무엇인지 알 수 없으므로 모델의 성능을 평가할 수 없다. 과거 데이터 일부를 미래 데이터로 두고, 해당 모델의 성능을 예측할 수 있다.

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

K-means : KMeans 알고리즘은 군집 내 제곱합이라고 알려진 기준을 최소화하면서 동일한 분산의 n개 그룹에서 표본을 분리하여 데이터를 군집화합니다. 이 알고리즘을 사용하려면 클러스터 수를 지정해야 합니다. 많은 수의 샘플로 잘 확장되며 다양한 분야에서 광범위한 응용 분야에 걸쳐 사용되어 왔습니다.

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 :

// source code 의 폰트는 **Courier10 BT Bold**으로 하시오

```
# pandas test
import pandas as ps

# read dataset
cars = pd.read_csv("/home/san/workspace/2021_2/딥러닝클라우드/Data/cars.csv")

# (1) 데이터셋의 위쪽 5행을 보이시오
cars.head()

# (2) 데이터셋의 컬럼들 이름을 보이시오
cars.columns

# (3) 데이터셋의 두 번째 컬럼의 값들만 보이시오.
cars.iloc[:, 1]

# (4) 데이터셋의 11~20행 자료중 speed 컬럼의 값들만 보이시오.
cars['speed'].iloc[11:21]

# (5) speed 가 20 이상인 행들의 자료만 보이시오
result = cars['speed'] >= 20
cars[result]

# (6) speed 가 10 보다 크고 dist 가 50보다 큰 행들의 자료만 보이시오.
cond1 = cars['speed'] > 10
cond2 = cars['dist'] > 50
cars[cond1 & cond2]

# (7) speed 가 15 보다 크고 dist 가 50보다 큰 행들은 몇 개인지 보이시오
cond1 = cars['speed'] > 15
cond2 = cars['dist'] > 50
cars[cond1 & cond2].shape[0]
```

실행화면 캡쳐:

(1)

	speed	dist
0	4	2
1	4	10
2	7	4
3	7	22
4	8	16

(2)

```
Index(['speed', 'dist'], dtype='object')
```

(3)

```
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
```

(4)

```
11    12
12    12
13    12
14    12
15    13
16    13
17    13
18    13
19    14
20    14
Name: speed, dtype: int64
```

(5)

	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

(6)

	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

(7)

```
14
```