3주. Machine Learning Concept			
학번	32152339	이름	송준영

Q1. 전통적인 SW 와 머신러닝을 적용한 SW 의 차이점을 설명하시오

▶ **전통적인 SW**

규칙을 인간이 직접 알아내어 알고리즘의 형태로 SW안에 구현.

▶ **머신러닝을 적용한 SW**

규칙을 알아내는 방법은 인간이 제시.

실제 규칙을 알아내는 과정은 머신이 진행.

-머신러닝 방법론 예시

1) 과거 데이터를 수집, 정리
2) 훈련(머신 입장에서는 학습) 방법 결정 : regression, decision tree 등
3) 머신이 학습
4) 예측모델 도출(학습 방법에 따라 다양한 형태)
5) 예측에 활용

→ **즉, 규칙(관계)을 알아내는 주체가 인간이냐 머신이냐가 주요한 차이점.**

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

1. **과거의 데이터를 기반으로**
2. **머신이** (SW, Program... 즉, 주체가 사람이 아니라는 의미)
3. **인간이 제시해준 학습방법을 통해** (regression, decision tree 등)
4. **반응변수**(Y, label, class 등)**와 설명변수**(X, 독립변수) **간의 관계를 찾아**
5. **예측모델**(Predictor)**을 구축하는 것**

Q3. 머신러닝이 가능한 이유를 설명하시오.

▶ 빅데이터를 다룰 수 있을 정도로 머신의 성능적인 발전이 이루어짐.

▶ 머신이 학습할 수 있도록 하는 머신러닝 방법론들의 이론적 기반 마련.

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

▶ 회귀(regression)

반응변수(Y)가 수치형 변수인 것. ex) 주가, 기온 등

▶ 분류(classification)

반응변수(Y)가 범주형 변수인 것. ex) 환자 여부, 성별 등

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) 기후변화에 따른 연평균 기온의 **과거 데이터**
즉, X(기후를 나타내는 다양한 변수들)와 Y(연평균 기온)으로 구성된 데이터가 필요.
2) **회귀**
연평균 기온이라는 수치형 변수를 예측하기 때문에 지도학습 중에서도 회귀를 적용해야 함.

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

▶데이터셋을 나누는 이유 : 구축한 모델이 일반화된 모델인가를 판단하기 위하여 학습에 사용되지 않은 데이터로 평가해야 하기 때문이다.
머신러닝 모델을 개발할 때는 일반적으로 데이터를 training data와 test data로 분할한다. 이는 training data로 모델을 구축하고 해당 모델이 새로운 데이터(미래 데이터)에도 비슷한 성능을 가지는가(즉, 일반화된 모델인가)를 알아보는 시험을 test data로 진행해야하기 때문이다.
물론 모델의 예측성능 평가를 위하여 미래 데이터를 대입하여 비교해보면 좋으나, 미래 데이터는 일반적으로 구할 수 없으므로, 학습에 사용되지 않은 test data가 그 역할을 대신한다.

▶나누지 않을 때의 발생하는 문제점 : 과적합(overfitting)인가에 대하여 평가할 수 없음.
만약 나누지 않고 하나의 데이터셋으로 모델을 개발한다면 해당 모델이 개발에 사용한 데이터 외에도 우수한 성능을 보이는가(즉, training data에 과적합되었는가)에 대한 평가를 할 수 없다.
즉, 새로운 미래데이터가 입력되었을 때 성능이 유지되는지에 대한 평가를 할 수 없다.

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 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작한다. 이 알고리즘은 비지도 학습의 일종으로, 레이블이 달려 있지 않은 입력 데이터에 레이블을 달아주는 역할을 수행한다. 이 알고리즘은 EM 알고리즘을 이용한 클러스터링과 비슷한 구조를 가지고 있다.

▶Affinity propagation
통계와 데이터 마이닝에서 Affinity propagation(AP)는 데이터 포인트 간 메시지 전달(message passing) 개념을 기반으로 하는 클러스터링 알고리즘이다. k-평균이나 k-medoids와 같은 군집화 알고리즘과 달리, Affinity propagation는 알고리즘을 실행하기 전에 군집 수를 결정하거나 추정할 필요가 없다. k-medoids와 유사하게, 친화도 전파는 클러스터를 대표

하는 입력 세트의 멤버인 "exemplar"를 찾는다.

▶Mean-shift
밀도 함수의 최대값(mode-seeking algorithm)을 찾기 위한 비모수 형상 공간 분석 기법이다. 응용분야는 컴퓨터 비전과 이미지 처리에서 클러스터 분석등이 있다.

▶Spectral clustering
다변량 통계 및 데이터 클러스터링에서 스펙트럼 클러스터링 기술은 데이터의 유사성 매트릭스 스펙트럼을 사용하여 더 적은 차원에서 클러스터링하기 전에 차원 축소를 수행합니다. 유사성 행렬은 입력으로 제공되며 데이터 집합의 각 쌍의 점에 대한 상대 유사성의 정량적 평가를 한다.

▶Ward hierarchical clustering
Ward의 최소 분산 방법을 활용한 계층적 클러스터 분석이다.

▶Agglomerative clustering
데이터 마이닝 및 머신러닝에서 계층적 클러스터링하는 방법이다.

▶DBSCAN : 밀도기반 클러스터링
점이 세밀하게 몰려 있어서 밀도가 높은 부분을 클러스터링 하는 방식이다.
즉, 어느점을 기준으로 반경 x내에 점이 n개 이상 있으면 하나의 군집으로 인식하는 방식이다.

▶OPTICS
실제로 클러스터를 생성하지는 않지만 밀도 기반 클러스터링 구조를 표현하는 데이타베이스의 인자를 순서화하여 하나의 전역 파라미터의 제한을 받지 않고 밀도 기반 클러스터링과 동일한 정보를 나타낸다.

▶Gaussian mixtures
Gaussian 분포가 여러 개 혼합된 clustering 알고리즘이다. 현실에 존재하는 복잡한 형태의 확률 분포를 K개의 Gaussian distribution을 혼합하여 표현하자는 것이 GMM의 기본 아이디어이다. 이때 K는 데이터를 분석하고자 하는 사람이 직접 설정해야 한다.

▶Birch
한 번만 Dataset을 검사하여 cluster를 만들며, 모든 데이터나 클러스터링을 스캔하지 않고도 클러스터링을 결정을 내린다는 점에서 local한 cluster방법이라고 할 수 있다. 즉, 보통 새로운 데이터가 들어오면 모든 데이터와의 거리를 구하거나 모든 cluster와 거리를 구하고 그 중 가까운 곳에 할당을 받는 그런 군집화 기법과는 조금 다르다는 의미이다.

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 Bold으로 하시오
df=ps.read_csv("C:/Users/ATIV/Desktop/딥러닝_클라우드
/dataset_0914/cars.csv")

#(1) 데이터셋의 위쪽 5행을 보이시오
df[:5]
#(2) 데이터셋의 컬럼들 이름을 보이시오
df.columns
#(3) 데이터셋의 두 번째 컬럼의 값들만 보이시오.
df.iloc[:,1]
#(4) 데이터셋의 11~20행 자료중 speed 컬럼의 값들만 보이시오.
df.iloc[10:20].speed
#(5) speed 가 20 이상인 행들의 자료만 보이시오
df[df.speed>=20]
#(6) speed 가 10 보다 크고 dist 가 50보다 큰 행들의 자료만 보이시오.
df[(df["speed"]>10) & (df["dist"]>50)]
#(7) speed 가 15 보다 크고 dist 가 50보다 큰 행들은 몇 개인지 보이시오
len(df[(df["speed"]>15) & (df["dist"]>50)])
```

실행화면 캡쳐:

#(1) 데이터셋의 위쪽 5행을 보이시오

```
In [70]: df[:5]
Out[70]:
   speed  dist
0      4     2
1      4    10
2      7     4
3      7    22
4      8    16
```

#(2) 데이터셋의 컬럼들 이름을 보이시오

```
In [71]: df.columns
Out[71]: Index(['speed', 'dist'], dtype='object')
```

#(3) 데이터셋의 두 번째 컬럼의 값들만 보이시오.

```
In [72]: df.iloc[:,1]
Out[72]:
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
```

#(4) 데이터셋의 **11~20**행 자료중 **speed** 컬럼의 값들만 보이시오**.**

```
In [73]: df.iloc[10:20].speed
Out[73]:
10    11
11    12
12    12
13    12
14    12
15    13
16    13
17    13
18    13
19    14
Name: speed, dtype: int64
```

#(5) speed 가 **20** 이상인 행들의 자료만 보이시오

```
In [74]: df[df.speed>=20]
Out[74]:
    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 가 **10** 보다 크고 **dist** 가 **50**보다 큰 행들의 자료만 보이시오**.**

```
In [75]: df[(df["speed"]>10) & (df["dist"]>50)]
Out[75]:
    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) speed 가 **15** 보다 크고 **dist** 가 **50**보다 큰 행들은 몇 개인지 보이시오

```
In [76]: len(df[(df["speed"]>15) & (df["dist"]>50)])
Out[76]: 14
```