Week2-4

1. 학습정리
    1. **pandas**
        1. groupby
            1. 인덱스가 같은 것 끼리 뭉쳐줌

Groupby

Groupby

```
df.groupby("Team")["Points"].sum()
```

적용받는 연산

묶음의 기준이 되는 컬럼

학교 남녀

적용받는 컬럼

	Points	Rank	Team	Year
0	876	1	Riders	2014
1	789	2	Riders	2015
2	863	2	Devils	2014
3	673	3	Devils	2015
4	741	3	Kings	2014

```
Team
Devils     1536
Kings      2285
Riders     3049
Royals     1505
kings       812
Name: Points, dtype: int64
```

결과
TEAM을 기준으로
Points을 Sum

2. 여러 개로 묶을 수 있음

```
df.groupby("Team")["Points"].std()
```

3. hierarchical index
    1. 그룹바이로 계층이 있는 인덱스를 만들 수 있음
    2. reset_index() 함수로 원래대로 풀어줄 수 있음
    3. swaplevel() 함수로 인덱스 레벨을 바꿀 수 있음
    4. sort_index(level=0)
        1. 인덱스 레벨 0 을 기준으로 소팅
    5. 멀티 인덱스일때는 sum(level=~), std(level=~) 등 연산함수를 레벨을 지정해 주면 사용 가능
    6. grouped 상태는 제네레이터 상태임

Groupby – gropued

Groupby

- Groupby에 의해 Split된 상태를 추출 가능함

```
grouped = df.groupby("Team")

for name,group in grouped:
    print (name)
    print (group)
```

```
Devils
   Points  Rank    Team  Year
2     863     2  Devils  2014
3     673     3  Devils  2015
Kings
   Points  Rank   Team  Year
4     741     3  Kings  2014
6     756     1  Kings  2016
7     788     1  Kings  2017
```

Tuple 형태로 그룹의 key 값
Value값이 추출됨

Groupby – gropued

Groupby

- 추출된 group 정보에는 세 가지 유형의 apply가 가능함
- Aggregation: 요약된 통계정보를 추출해 줌
- Transformation: 해당 정보를 변환해줌
- Filtration: 특정 정보를 제거 하여 보여주는 필터링 기능

2. **pivot table**
   1. 컬럼에 추가로 라벨링 값을 추가하여
   2. 밸류에 뉴메릭 타입값을 어그리게이션 하는 형태

Pivot Table

Pivot table & Cros

```
df_phone.pivot_table(["duration"],
                     index=[df_phone.month,df_phone.item],
                     columns=df_phone.network, aggfunc="sum", fill_value=0)
```

		duration								
	network	Meteor	Tesco	Three	Vodafone	data	landline	special	voicemail	world
month	item									
2014-11	call	1521	4045	12458	4316	0.000	2906	0	301	0
	data	0	0	0	0	998.441	0	0	0	0
	sms	10	3	25	55	0.000	0	1	0	0
2014-12	call	2010	1819	6316	1302	0.000	1424	0	690	0
	data	0	0	0	0	1032.870	0	0	0	0
	sms	12	1	13	18	0.000	0	0	0	4

3. **Crosstab**

1. 피벗 테이블의 특수한 형태
2. 두 칼럼의 교차 빈도, 비율 등을 구할 때 사용
3. User-Item Rating Matrix 를 만들때 사용

```
pd.crosstab(index=df_movie.critic,columns=df_movie.title,values=df_movie.rating,
            aggfunc="first").fillna(0)
```

title	Just My Luck	Lady in the Water	Snakes on a Plane	Superman Returns	The Night Listener	You Me and Dupree
critic						
Claudia Puig	3.0	0.0	3.5	4.0	4.5	2.5
Gene Seymour	1.5	3.0	3.5	5.0	3.0	3.5
Jack Matthews	0.0	3.0	4.0	5.0	3.0	3.5
Lisa Rose	3.0	2.5	3.5	3.5	3.0	2.5
Mick LaSalle	2.0	3.0	4.0	3.0	3.0	2.0
Toby	0.0	0.0	4.5	4.0	0.0	1.0

4. **merge**
   1. 두 데이터 프레임을 합침

subject_id 기준으로 merge

```
pd.merge(df_a, df_b, on='subject_id')
```

	subject_id	test_score
0	1	51
1	2	15
2	3	15
3	4	61
4	5	16
5	7	14

	subject_id	test_id	first_name	last_nam
0	4	61	Billy	Bond
1	5	16	Brian	Blac
2	7	14	Bryce	Brice
3	8	15	Betty	Btisan

	subject_id	first_name	last_name
0	4	Billy	Bonder
1	5	Brian	Black
2	6	Bran	Balwner
3	7	Bryce	Brice
4	8	Betty	Btisan

   2. 컬럼명이 다를 때

merge

merge & concat

left right

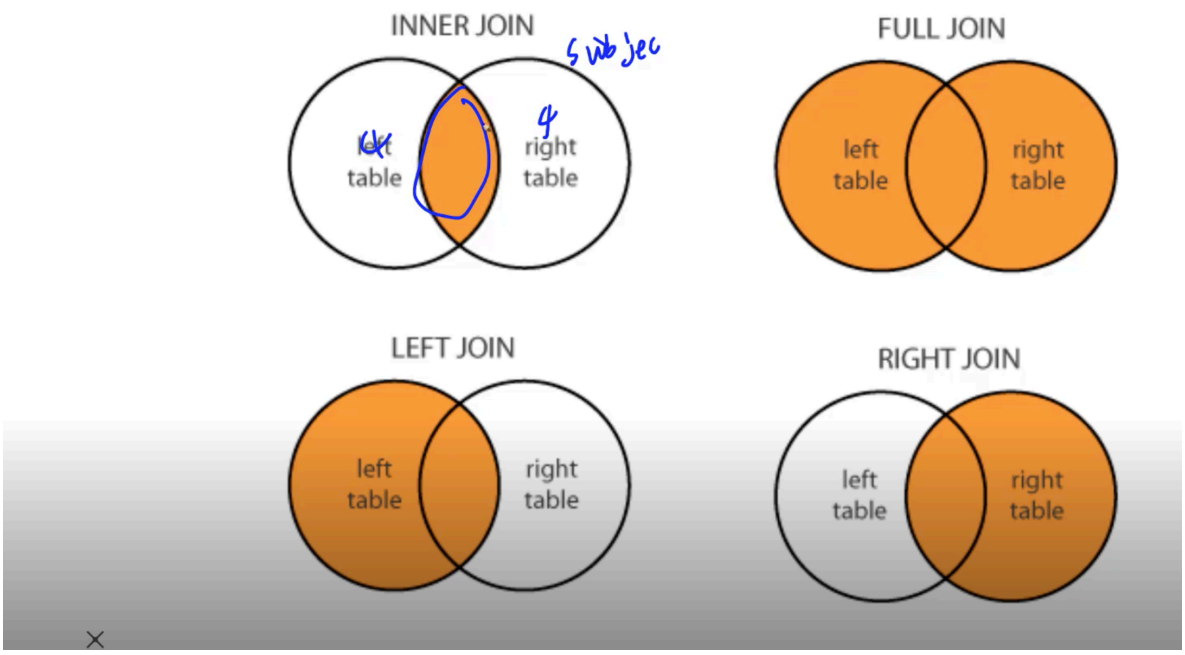
두 dataframe이 column이름이 다를 때

```
pd.merge(df_a, df_b, left_on='subject_id', right_on='subject_id')
```

	subject_id	test_score	first_name	last_name
0	4	61	Billy	Bonder
1	5	16	Brian	Black
2	7	14	Bryce	Brice
3	8	15	Betty	Btisan

   3. join

join method




```
pd.merge(df_a, df_b, on="subject_id", how="left")
```

	subject_id	test_score	first_name	last_name
0	1	51	NaN	NaN
1	2	15	NaN	NaN
2	3	15	NaN	NaN
3	4	61	Billy	Bonder
4	5	16	Brian	Black
5	7	14	Bryce	Brice
6	8	15	Betty	Btisan
7	9	1	NaN	NaN
8	10	61	NaN	NaN
9	11	16	NaN	NaN

        4. 레프트 조인 : 둘 다 있는 건 출력, 한쪽만 있는 건 난값
    4. concat, append
        1. 데이터프레임 뒤에 붙임
5. **확률론**
    1. 확률변수

이산확률변수 vs 연속확률변수

- 확률변수는 확률분포 $\mathscr{D}$ 에 따라 이산형(discrete)과 연속형(continuous) 확률변수로 구분하게 됩니다
- 이산형 확률변수는 확률변수가 가질 수 있는 경우의 수를 모두 고려하여 확률을 더해서 모델링한다

$$\mathbb{P}(X \in A) = \sum_{\mathbf{x} \in A} P(X = \mathbf{x})$$

- 연속형 확률변수는 데이터 공간에 정의된 확률변수의 밀도(density) 위에서의 적분을 통해 모델링한다

$$P(\mathbf{x}) = \lim_{h \to 0} \frac{\mathbb{P}(\mathbf{x} - h \leq X \leq \mathbf{x} + h)}{2h}$$

$$\mathbb{P}(X \in A) = \int_A P(\mathbf{x}) d\mathbf{x}$$

밀도는 누적확률분포의 변화율을 모델링하며 확률로 해석하면 안된다

1. 이산형과 연속형 확률변수 구분할 줄 알기
2. 모델링 하는 방법의 차이 알기
3. 딥러닝은 둘 다 쓰임

2. 조건부 확률

조건부확률과 기계학습

- 조건부확률 $P(y \mid \mathbf{x})$ 는 입력변수 $\mathbf{x}$ 에 대해 정답이 y 일 확률을 의미합니다
- 로지스틱 회귀에서 사용했던 선형모델과 소프트맥스 함수의 결합은 데이터에서 추출된 패턴을 기반으로 확률을 해석하는데 사용됩니다
- 분류 문제에서 softmax$(\mathbf{W}\phi + \mathbf{b})$은 데이터 $\mathbf{x}$ 로부터 추출된 특징패턴 $\phi(\mathbf{x})$ 과 가중치행렬 $\mathbf{W}$ 을 통해 조건부확률 $P(y \mid \mathbf{x})$ 을 계산합니다
- 회귀 문제의 경우 조건부기대값 $\mathbb{E}[y \mid \mathbf{x}]$ 을 추정합니다
- 딥러닝은 다층신경망을 사용하여 데이터로부터 특징패턴 ϕ 을 추출합니다

특징패턴을 학습하기 위해 어떤 손실함수를 사용할지는 기계학습 문제와 모델에 의해 결정된다

3. 기대값

기대값이 뭔가요?

- 확률분포가 주어지면 데이터를 분석하는 데 사용 가능한 여러 종류의 통계적 범함수(statistical functional)를 계산할 수 있습니다
- 기대값(expectation)은 데이터를 대표하는 통계량이면서 동시에 확률분포를 통해 다른 통계적 범함수를 계산하는데 사용됩니다

$$\mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[f(\mathbf{x})] = \int_{\mathcal{X}} f(\mathbf{x})P(\mathbf{x})\mathrm{d}\mathbf{x}, \quad \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[f(\mathbf{x})] = \sum_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})P(\mathbf{x})$$

- 기대값을 이용해 분산, 첨도, 공분산 등 여러 통계량을 계산할 수 있습니다

$$\mathbb{V}(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[(\mathbf{x}-\mathbb{E}[\mathbf{x}])^2] \quad \mathrm{Skewness}(\mathbf{x}) = \mathbb{E}\left[\left(\frac{\mathbf{x}-\mathbb{E}[\mathbf{x}]}{\sqrt{\mathbb{V}(\mathbf{x})}}\right)^3\right]$$

$$\mathrm{Cov}(\mathbf{x}_1,\mathbf{x}_2) = \mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2\sim P(\mathbf{x}_1,\mathbf{x}_2)}[(\mathbf{x}_1-\mathbb{E}[\mathbf{x}_1])(\mathbf{x}_2-\mathbb{E}[\mathbf{x}_2])]$$

위 수식에 f 대신 대입하면 통계량을 계산할 수 있다

4. 몬테카를로 생플링

몬테카를로 샘플링

- 기계학습의 많은 문제들은 확률분포를 명시적으로 모를 때가 대부분이다
- 확률분포를 모를 때 데이터를 이용하여 기대값을 계산하려면 몬테카를로(Monte Carlo) 샘플링 방법을 사용해야 한다

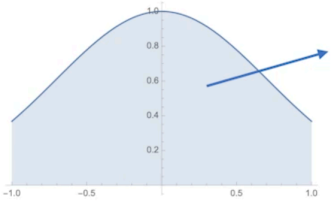
$$\mathbb{E}_{\mathbf{x}\sim P(\mathbf{x})}[f(\mathbf{x})] \approx \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}^{(i)}), \quad \mathbf{x}^{(i)} \overset{\text{i.i.d.}}{\sim} P(\mathbf{x})$$

- 몬테카를로 샘플링은 독립추출만 보장된다면 대수의 법칙(law of large number)에 의해 수렴성을 보장한다

몬테카를로 샘플링은 기계학습에서 매우 다양하게 응용되는 방법입니다

몬테카를로 예제: 적분 계산하기

- 함수 $f(x) = e^{-x^2}$ 의 $[-1,1]$ 상에서 적분값을 어떻게 구할까?



$$\int_{-1}^{1} e^{-x^2}\mathrm{d}x \approx 1.49364$$

$$\int_{-1}^{1} e^{-x^2}\mathrm{d}x \approx \frac{2}{N}\sum_{i=1}^{N} f(x^{(i)}), \quad x^{(i)} \sim U(-1,1)$$

```python
import numpy as np

def mc_int(fun, low, high, sample_size=100, repeat=10):
    int_len = np.abs(high - low)
    stat = []
    for _ in range(repeat):
        x = np.random.uniform(low=low, high=high, size=sample_size)
        fun_x = fun(x)
        int_val = int_len * np.mean(fun_x)
        stat.append(int_val)
    return np.mean(stat), np.std(stat)

def f_x(x):
    return np.exp(-x**2)

print(mc_int(f_x, low=-1, high=1, sample_size=10000, repeat=100))
```

```
(1.4938754306231912, 0.00391398451303653)
```

1.49387±0.0039 이므로 오차 범위 안에 참값이 있다

        1. 위 함수는 적분이 어려우름로 몬테카를로 샘플링으로 근사값을 구할 수 있음
        2. 균등분포로 샘플링 후 평균값을 구해서 구간 크기를 곱하면 됨
        3.

1. **피어세션**
    1. 강의 리뷰
        1. 내용이 많은데 빨리 지나갔다
        2. 강의에 나온 내용을 실습하며 하려니 지체된다
        3. 분류 문제에서 소프트 맥스 함수가 사용되는 이유에 대해
            1. 여러 클래스에 대한 점수를 한번에 찾기 위해
            2. exp하는 이유는 좀 더 정확히 찾으려고