

1 <처음보는 용어집>

1. pandas : 구조화 된 data 처리를 지원하는 파이썬 라이브러리로 python계의 엑셀이라 불림 (출처 : 부스트코스 2024 코칭스터디 강의자료)
2. 스프레드시트(Spreadsheet) : 표 형식의 데이터 조직, 분석, 저장을 가능케 하는 컴퓨터 애플리케이션으로 엑셀과 유사한 형태의 데이터 관리 도구 (출처 : 남동득, 『실전에서 바로 써먹는 업무자동화』, 클라우드 나인(2023))
3. csv 데이터 타입 : **comma-separated values**의 약자로 필드를 쉼표(comma)로 구분한 텍스트 파일 (출처 : wikipedia^[1])
4. 산점도(Scatter plot) : 좌표상의 점들을 표시한 도표를 이용해 두 변수 간의 관계를 나타내는 그래프 방법 (출처 : wikipedia^[2])
5. list 객체 : 여러 값을 순차적으로 저장할 수 있는 자료형으로 파이썬에서 가장 많이 사용되는 컬렉션 중 하나 (출처 : 부스트코스 <컬렉션의 구조와 활용> 5-1 강의 영상)
6. NaN : Not-A-Number(숫자가 아님)를 의미한다. 데이터 프레임 내에 빈 데이터를 표기하기 위해서 NaN이라는 용어를 사용한다. 0과는 다르다.(출처 : mdn web docs^[3])
7. lambda(람다) : $(\text{lambda } x,y: x + y)(10, 20)$ (출력 : 30)와 같이 함수를 한 줄로 간단하게 쓸 수 있는 기법이다. (출처 : wikidocs^[4])
8. adam optimizer : 딥러닝 최적화 기법 중 성능이 좋아 가장 많이 쓰는 최적화 기법이다. (출처 : 코칭스터디 2-4 강의 영상)
9. np.inner : 두 벡터의 내적을 계산할 수 있게 도와주는 파이썬 함수이다. (출처 : 코칭스터디 2-5 강의영상)
10. Proj(x) : 벡터 y로 정사영된 벡터x의 그림자를 의미한다. Proj(x)의 길이는 코사인법칙에 의해 $(x \text{의 노름}) * \cos(\text{theta})$ 로 계산할 수 있다. (출처 : 코칭스터디 2-5 강의영상)

11. ResNet: Layer를 늘리면 성능이 저하되는 문제를 해결하고자 Layer들 사이에 shortcut connection을 적용해 큰 성능 향상을 보였다. weight layer를 거치고 난 후인 $F(x)$ 에 본래의 입력값 x 를 다시 더해준다는 점에서 필요한 gradient가 사라지지 않는 효과를 얻을 수 있었고 그 결과 더 깊은 모델을 구성할 수 있게 되었다. (출처: 서울대학교 AI 연구원)

12. Loss function (손실함수) 딥러닝에서의 손실 함수(loss function) 또는 비용 함수(cost function)는 주어진 함수에 대한 최소화가 실제 결과값(정답) y 와 모델의 추정치 \hat{y} 의 차이를 최소화하도록 정의된 특수한 함수이다. (출처: 서울대학교 AI 연구원)

13. 파인 튜닝 딥 러닝에서 사전 훈련된 모델의 가중치가 새로운 데이터에 대해 훈련되는 전이학습에 대한 접근 방식이다. (출처: wikipedia^[5])

14. GAN(생성적 대립 신경망) 생성모델과 판별모델이 경쟁하면서 실제와 가까운 이미지, 동영상, 음성 등을 자동으로 만들어 내는 기계학습(ML: Machine Learning) 방식의 하나. (출처: 한국정보통신기술협회)

15. 정사영: 평면 α 밖의 점 P 에서 α 에 그은 수선의 발 P' 를 점 P 의 평면 α 위로의 정사영이라 한다. 또, 도형 F 에 속하는 모든 점의 평면 α 위로의 정사영으로 이루어지는 도형 F' 을 F 의 α 위로의 정사영이라 한다. (출처: 통합논술 개념어 사전, 한림학사)

16. Pandas: Pandas는 Python 프로그래밍 언어로 작성된 고수준의 데이터 조작 도구입니다. 주로 데이터 분석을 위해 사용되며, 특히 숫자 테이블과 시계열을 다루는 데 탁월합니다. Pandas는 "Panel Data"의 약자에서 유래했으며, 이 라이브러리는 DataFrame 객체를 중심으로 구성되어 있으며, 이는 데이터를 처리하고 분석하는 데 있어 다양한 기능들을 제공합니다.

Pandas는 다양한 기능

- 다양한 형태의 데이터(텍스트, CSV, SQL 데이터베이스 테이블 등)의 읽기 및 쓰기
- 데이터 정제 및 전처리 (결측치 처리, 데이터 변환, 데이터 정렬)
- 데이터 분석 및 모델링을 위한 통계적 분석, 데이터 집계 및 요약
- 시계열 데이터 분석 (출처: [위키피디아](#))

17. 벡터의 노름(Vector Norm): 벡터의 크기나 길이를 측정하는 방법입니다. 벡터의 노름은 다양한 종류가 있으며, 각각의 노름은 벡터의 다른 성질을 나타냅니다. 가장 일반적인 노름은 유클리드 노름(Euclidean norm, L2 노름)이며, 이는 벡터의 각 성분의 제곱합의 제곱근으로 계산됩니다. 노름은 벡터 공간에서의 거리와 크기를 측정하는 기본적인 도구로, 선형대수학, 함수해석학, 기계학습 등 다양한 분야에서 중요하게 사용됩니다.(출처:

[위키피디아](#))

18. 자연어 처리(Natural Language Processing, NLP): 자연어 처리(NLP)는 인간의 언어를 해석, 조작 및 이해하는 능력을 컴퓨터에 부여하는 기계 학습 기술입니다. 오늘날 조직의 이메일, 문자 메시지, 소셜 미디어 뉴스 피드, 동영상, 오디오 등, 다양한 커뮤니케이션 채널에서 생성되는 대량의 음성 및 텍스트 데이터를 NLP 소프트웨어를 사용하여 이 데이터를 자동으로 처리하고 메시지의 의도나 감정을 분석하며 사람의 커뮤니케이션에 실시간으로 응답하는 것입니다. (출처 : [AWS AMAZON](#))

19. 코사인시밀러리티 (Cosine Similarity): 코사인 유사도는 두 벡터 간의 코사인 각도를 사용하여 두 벡터의 유사도를 측정하는 방법입니다. 이 방법은 두 벡터의 방향성이 얼마나 유사한지를 기준으로 하며, 벡터의 크기는 고려하지 않습니다. 이 방식은 주로 텍스트 문서의 유사도 측정, 추천 시스템, 정보 검색 등에 활용됩니다.(출처: [위키피디아](#))

20. Docker: Docker는 소프트웨어를 컨테이너 내에 패키징하여 응용 프로그램이 실행 환경에 구애받지 않고 동일하게 작동할 수 있도록 하는 오픈 소스 프로젝트입니다. 각 컨테이너는 소프트웨어, 런타임, 시스템 도구, 시스템 라이브러리 등 응용 프로그램을 실행하는 데 필요한 모든 것을 포함합니다. 이렇게 함으로써, 개발에서부터 스테이징, 프로덕션 환경까지 일관된 환경을 제공합니다.(출처: [위키피디아](#))

21. 표현 학습: 특징 학습으로도 부르며, 특징을 자동으로 추출할 수 있도록 학습하는 과정 (출처: 위키피디아)

22. kwargs: keyword arguments의 줄임말로, 키워드 형태로 들어오는 인수를 의미한다.

23. 히스토그램: 표로 되어 있는 도수 분포를 정보 그림으로 나타낸 것. 즉, 도수분포표를 그림으로 나타낸 것. (출처: 위키피디아)

24. scikit-learn(사이킷런): 대표적인 파이썬의 머신러닝 분석 라이브러리. 여러 가지 머신러닝 모듈로 구성되어 있으며, 오픈 소스로 공개되어 있다.

25. correlation(상관관계): 인과 여부에 관계없이 2개의 확률 변수 또는 이변량 데이터 사이의 모든 통계적 관계.

2 < 팀 미션 - 2문제 >

문제 1. 파이썬의 대표적 시각화 도구 라이브러리인 matplotlib의 기본 사용법에 대해 숙지하고, matplotlib에서 사용되는 여러 graph들에 대해 조사해 보자.

또, 해당 graph들에 어느 데이터가 실무에서 적합하게 쓰일지 조사해 보자. [주제 : 파이썬]

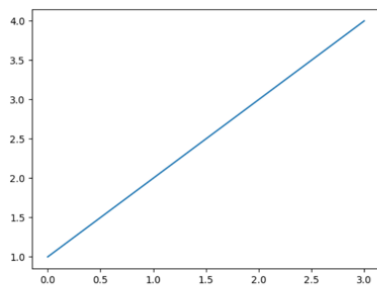
matplotlib 기본적인 사용법 숙지

1. 기본 그래프 그리기

matplotlib.pyplot 모듈을 이용하여 기본적인 그래프를 만들고 변형하는 작업을 할 수 있다.

```
import matplotlib.pyplot as plt

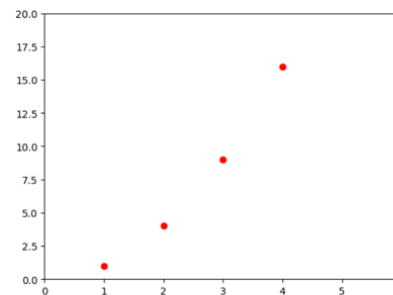
plt.plot([1, 2, 3, 4]) # 리스트의 값들이 y 값들이라고 가정하고, x 값 [0, 1, 2, 3]을 자동으로 만들어냄
plt.show() # 그래프를 보여주는 함수
```



2. 포맷 문자열 지정하기

선의 컬러 및 형태를 지정하는 “포맷 문자열”을 인자로 넣을 수 있다.

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro') # 'r'은 'red'를, 'o'는 '원형'을 의미함
plt.axis([0, 6, 0, 20])
plt.show()
```



axis() 함수는 축의 범위 [xMin, xMax, yMin, yMax]를 지정하는 함수이다.
참고로 컬러는 blue(b), green(g), red(r), yellow(y), black(k), white(w) 등이 있고,
형태는 circle(o), star(*), square(s), x(x), diamond(D) 등이 있다.

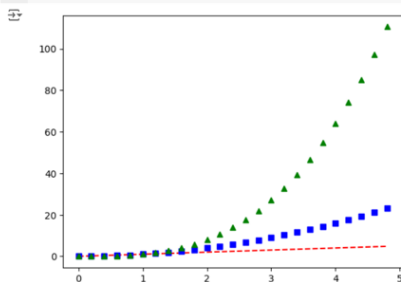
3. 그래프 여러 개 그리기

numpy array를 이용하면 여러 개의 그래프를 동시에 그릴 수 있다.

```
import matplotlib.pyplot as plt
import numpy as np

# 200ms 간격으로 균일하게 샘플된 시간
t = np.arange(0, 5, 0.2)

# 빨간 대위, 파란 사각형, 녹색 삼각형
plt.plot(t, t, 'r--', t, t**2, 'b*', t, t**3, 'g^')
plt.show()
```



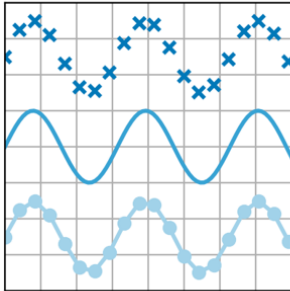
* numpy를 통해 어레이를 생성하고, 한 평면에 여러 개의 그래프를 그린 모습이다.
(예제 코드 출처 : 위키독스)

matplotlib graph들의 특징

1. 쌍별 데이터

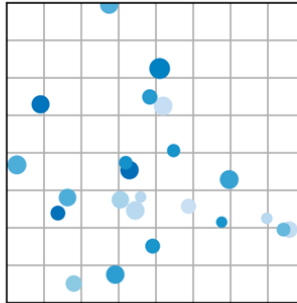
- plot(x,y)

: y 대 x를 선 및/또는 마커로 표시



- scatter(x,y)

: 마커 크기 및/또는 색상이 다양한 y 대 x의 산점도
좌표를 통해 그래프를 그리기에 두 변수 사이의 관계를 눈으로 확인 가능.

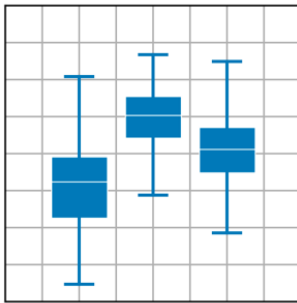


2. 통계분포

- boxplot(x)

: 데이터의 분포와 이상치(outlier)를 시각화.

박스플롯을 활용하면 이상치가 얼마나 포함되어 있는지를 쉽게 판단할 수 있음



실무에서 해당 graph에 사용되는 데이터 파악

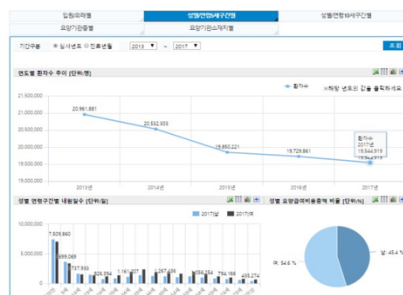
1. 금융산업

- 데이터: 주식 가격, 거래량, 금리 등
- 그래프 유형: 선 그래프, 캔들스틱 차트
- 결과 도출: 선 그래프를 통해 시간에 따른 주식 가격의 추세를 파악할 수 있으며, 캔들스틱 차트는 하루 동안의 가격 변동성을 상세히 보여줌으로써 단기적인 거래 전략 수립에 도움을 줄 수 있습니다.



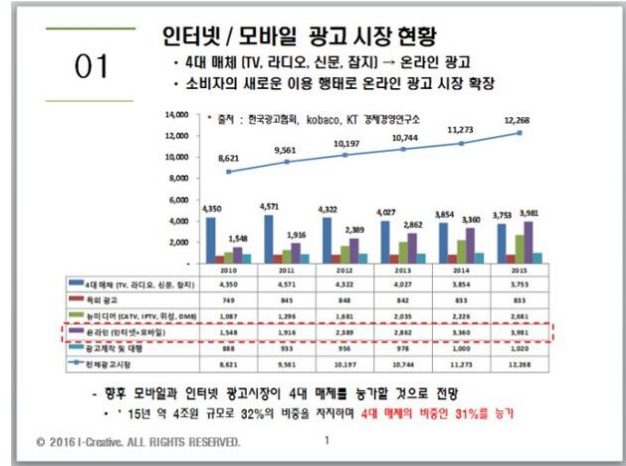
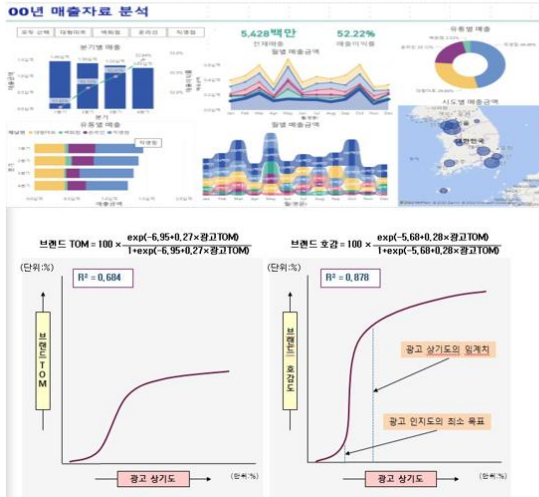
2. 헬스케어

- 데이터: 환자의 건강 지표(혈압, 체온 등), 질병 발생률
- 그래프 유형: 막대 그래프, 선 그래프
- 결과 도출: 막대 그래프를 통해 다양한 나이대나 지역별 질병 발생률을 비교할 수 있으며, 선 그래프는 시간에 따른 환자의 건강 지표 변화를 모니터링하여 치료 효과나 질병의 진행 상황을 평가할 수 있습니다.



3. 마케팅

- 데이터: 광고 클릭률, 방문자 수, 구매 전환율
- 그래프 유형: 막대 그래프, 파이 차트
- 결과 도출: 막대 그래프를 통해 다양한 광고 캠페인의 성능을 비교할 수 있으며, 파이 차트는 전체 방문자 중 특정 조치를 취한 비율을 시각적으로 표현하여 어떤 마케팅 전략이 더 효과적인지 평가할 수 있습니다.



4. 기후과학

- 데이터: 기온, 강수량, 해수면 상승률
- 그래프 유형: 선 그래프, 산점도
- 결과 도출: 선 그래프를 사용하여 장기적인 기후 변화 추세를 분석할 수 있으며, 산점도는 예를 들어 기온과 강수량 사이의 관계를 분석하여 특정 기후 조건이 강수 패턴에 미치는 영향을 파악할 수 있습니다.

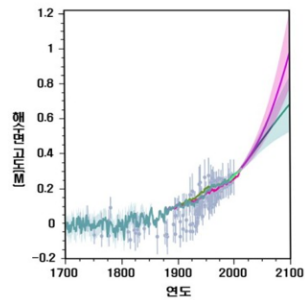
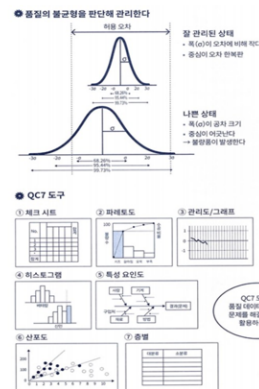


그림 1. IPCC 5차보고서에서 제시된 과거의 해수면 상승 분석 및 미래 전망 (출처: IPCC 5차 보고서)

5. 제조업

- 데이터: 생산량, 불량률, 기계 가동 시간
- 그래프 유형: 막대 그래프, 히스토그램
- 결과 도출: 막대 그래프를 통해 시간별 또는 제품별 생산량을 비교할 수 있으며, 히스토그램은 불량률이나 기계 가동 시간의 분포를 분석하여 공정 개선의 기회를 찾을 수 있습니다.



문제 2. 벡터의 노름은 데이터를 정제하는 방법 중 정규화(Normalization), 규제(Regularization) 등에서 유용하게 쓰여진다. 벡터의 노름 구조와 개념에 대해 더 자세히 알아보고 머신러닝 분야에서 어떻게 쓰이는지 자세히 알아보자.

또, L1 노름과 L2 노름이 머신러닝 분야에서 어떤 차이점을 가지고 사용되는지에 대해 비교 분석해보자. [주제 : 벡터]

노름의 개념 속지

벡터에서 노름(norm)이란 **원점에서부터의 거리**를 말한다.

벡터 노름을 계산하는 방법에는 여러 가지가 있으며, 각 방법은 고유한 표기법을 가지고 있다. 머신러닝에서 자주 사용되는 몇 가지 벡터 노름 계산 방법에 대한 정의를 살펴보면 다음과 같다.

1. 벡터 L1 노름

- L1 노름은 벡터에 변화량의 절대값을 모두 더한 값을 말한다.
- 모델의 복잡성을 줄이기 위해 머신러닝 알고리즘을 선택하는 경우 자주 사용된다.

2. 벡터 L2 노름

- L2 노름은 벡터 좌표가 벡터 공간의 원점에서부터 얼마나 떨어져 있는지를 나타내며, 유클리드 거리로도 알려져 있다. 벡터 값의 제곱합의 제곱근인 피타고라스의 정리를 이용해 계산된다.
- L2 노름은 머신러닝에서 모델의 복잡성을 줄이기 위한 정규화 방법으로 자주 사용된다.

3. 벡터 최대 노름 (Max norm)

- 최대 노름은 벡터 요소 중 가장 큰 절대값을 사용하는 방법이다.
- 최대 노름은 신경망 가중치에 대한 정규화 방법으로도 사용된다.

이처럼 각 노름은 머신러닝의 필요한 성질에 따라 다른 종류가 사용되기 때문에 계산 방식에 따라 다양한 종류로 분류한다.

머신러닝 분야에서의 쓰임

• L1 정규화 (L1 regularization)

$$J(w) + \alpha \sum_{i=1}^N |w_i|^1$$

• L2 정규화 (L2 regularization)

$$J(w) + \alpha \sum_{i=1}^N |w_i|^2$$

($J(w)$: 원래의 손실함수, α : 정규화의 강도를 조절하는 파라미터, w_i : 모델 파라미터)

정규화는 overfitting(과대적합)과 underfitting(과소적합)을 방지하기 위해 필요한 작업이다.

만약, 머신러닝 모델에서 주어진 데이터에 대해 최적의 그래프를 찾아야 하는데 다음 그림과 같이 Overfitting이 되면 예측의 노이즈가 발생할 수 있다.

이때 최적의 정규화를 수행하여 노이즈를 줄임으로서 원하는 예측값이 올바르게 잘 시행되게 해야한다. 따라서 적절한 정규화 기법을 사용해야 한다.



특성 스케일링 (feature scaling)

특성을 스케일링 하지 않으면 각자 다른 범위를 갖고 있는 특성에 대해 스케일링을 하였을 경우 편향이 발생할 수 있고, 스케일 차이로 인해 모델의 학습 속도가 느려질 수 있다. 따라서 모델의 성능을 최적화 하기 위해 특성 스케일링은 매우 중요하다. (모든 특성에 대해 0부터 1사이의 값으로 스케일링 할 수 있다.)

노름을 사용하는 feature scaling 에는 대표적인 방법으로 min-max normalization 이 있다.

$$x_{new} = \frac{x - \min(X)}{\max(X) - \min(X)}$$

모든 x집합에 대해 최대,최소값을 구한 후 각각의 x에 대해 위의 공식을 대입한다. 임의의 x값에 위 공식을 대입하면 x(new)가 나오는데 이는 0부터 1사이의 값으로 정규화 된 것이다.

이와같이 feature scaling을 하여 모델을 최적화하고 더 나은 학습결과와 더 빠른 연산속도를 기대할 수 있어 feature scaling은 데이터 전처리단계에서 매우 중요하다.

유사도 측정 (similarity measurement)

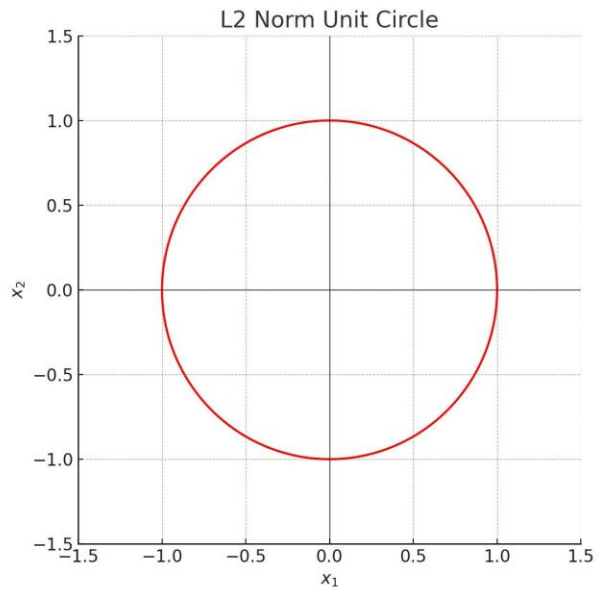
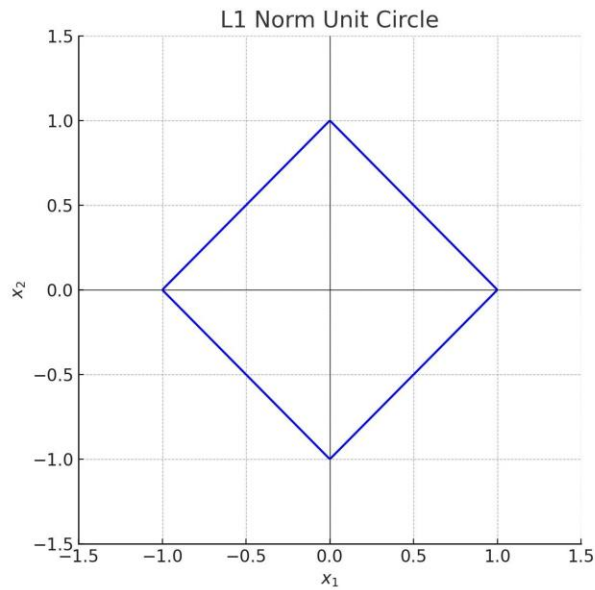
FastText 모델은 텍스트를 벡터로 바꾸는 가장 최신 모델이다. FastText를 통해 단어를 벡터로 유사도 측정을 통해 단어와 단어 사이에 얼마나 밀접한 관계가 있는지에 대해 연구한다.

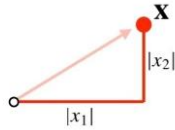
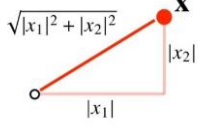
이때 바뀌어진 벡터가 얼마나 가깝게 위치하는지 알기 위해 cos유사도를 통해 벡터 사이의 각도를 분석한다.

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} = \frac{a_1 b_1 + a_2 b_2}{\sqrt{a_1^2 + a_2^2} \sqrt{b_1^2 + b_2^2}}$$

위 공식을 통해 각각의 벡터에 대해 노름을 사용하여 계산하고 theta를 계산하여 각각의 벡터에 대해 얼마나 가깝거나 멀리있는지 조사한다.

L1 노름과 L2 노름의 차이점 비교 분석



	L1 Norm	L2 Norm
정의	<p>모든 벡터 변화량의 절댓값을 더한 값</p> <p>=맨허튼 거리(Manhattan Distance)</p>	<p>모든 벡터의 제곱의 합을 제곱근한 값</p> <p>=유클리드 거리(Euclidean Distance)</p>
공식	$\ \mathbf{x}\ _1 = \sum_{i=1}^d x_i $ 	$\ \mathbf{x}\ _2 = \sqrt{\sum_{i=1}^d x_i ^2}$ 
특징	<ul style="list-style-type: none"> • 벡터간의 거리를 직관적으로 측정 가능 • 작은 변화에 민감하게 반응 • 벡터 사이의 거리를 계산할 수 있음 	<ul style="list-style-type: none"> • 큰 값에 가중치를 더 부여하여 큰 값의 변동에 민감하게 반응 • 벡터 사이의 거리뿐만 아니라 각도도 계산할 수 있음 ($\cos(\theta)$ 법칙 사용)

응용 분야	<ul style="list-style-type: none">• Robust 학습• Lasso 회귀	<ul style="list-style-type: none">• Laplace 근사• Ridge 회귀
-------	--	---