**- 1주차 과제 (전수경) -**

[빅데이터 개념]

**0. 빅데이터란 단어의 정의(3V)를 기준으로 설명해주세요.**

Volume (큰 규모) , Velocity (빠른 속도) , Variety (다양성) 을 특징으로 갖는 데이터

기존의 관리 체계로는 감당할 수 없을 만큼의 거대한 데이터의 집합

**1. map-reduce model의 개념에 대해서 조사해주세요. map-reduce의 간단한 사용 예를 하나들어주세요. 인터넷에서 찾으셔도 되고, 코딩까지가 아닌 개념 적인 예를 본인이 직접 들어서 설명해주셔도 됩니다.**

map-reduce model = 한 컴퓨터에서 수행할 작업을 여러 컴퓨터에 분산하여 처리하는 프로그래밍 모델

map 함수 → 분할한 데이터를 가공하는 역할 (키, 값 쌍)

reduce 함수 → 앞서 가공한 데이터를 어떤 기준에 따라 다시 하나로 병합하는 역할 (키 기준)

- Word counting example

map 단계 = 입력 데이터를 키-값 쌍으로 변환 (단어=키, 빈도=값)

shuffle 단계 = 같은 키를 가진 모든 값 그룹핑

reduct 단계 = 각 키에 대해 그룹핑된 값을 합산하여 최종 결과 생성

**3.빅데이터는 보통 한 컴퓨터로 처리할수 없을 정도의 데이터를 말합니다. 하나의 컴퓨터에서 연산을 할때 장단점과, 여러대의 컴퓨터로 분산컴퓨팅을 통해 장단점을 조사하고 이둘을 비교해주세요. 결론적으로 어떤게 좋은지 의견도 내주세요.**

- 한 대의 컴퓨터 연산 : 설정과 유지보수 간단함, 데이터 일관성 유지, 하드웨어 관련 비용 절감 / CPU, 메모리 등 하드웨어의 자원에 한계가 있어 처리할 수 있는 데이터의 양과 속도가 제한적,

- 여러 대의 컴퓨터로 분산 컴퓨팅 : 확장에 제한이 없음, 장애 발생 시 허용 범위 넓음, 낮은 대기 시간, 비용의 효율성 / 하드웨어와 소프트웨어의 복잡성, 높은 초기 비용, 보안 문제

→ 빅데이터 처리할 때는 분산 컴퓨팅 방식이 더 유리하다. 용량과 시간 등의 모든 면에서 분산 컴퓨팅이 효율적으로 적용될 것이다.

**4. 기존 데이터를 다루는 기술이 아닌, 빅데이터를 다루는 기술을 사용해서 문제를 해결한 예를 하나 설명해주세요.**

- 의료 분야 : IBM Watson for Oncology → 수많은 환자들에게 각자 맞는 치료법을 찾는 데 한계 존재, 특히, 암 치료에서는 환자의 유전적 정보, 최신 연구 논문, 임상 시험 결과 등 방대한 양의 데이터를 분석해야 하는데, 이는 매우 시간이 많이 걸리고 복잡한 작업

→ Watson 은 인공지능과 빅데이터 기술을 결합하여 대량의 데이터를 분석하여 개인 맞춤형 치료법 제공 가능

- 데이터 수집 및 분석 : Watson은 수백만 개의 의료 문서, 연구 논문, 환자 기록 등을 분석

- 자연어 처리(NLP) : Watson은 자연어 처리 기술을 사용하여 비정형 데이터를 이해하고, 의료 문서를 읽고 해석

- 맞춤형 치료 추천: Watson은 개별 환자의 유전자 정보, 병력, 생활 습관 등을 고려하여 맞춤형 치료법을 추천

**5. 아래를 학습 후 정리 해주세요.**

- infinity-band : 무한대의 대역폭

고속 데이터 전송 기술로 데이터 센터, 고성능 컴퓨팅(HPC) 및 클러스터 컴퓨팅 환경에서 널리 사용됨 / 대역폭 뿐만 아니라 낮은 지연 시간을 갖기 때문에 슈퍼컴퓨터에 쓰이면서 데이터센터, 인공지능 등의 분야에서 활용됨

하지만 낮은 호환성과 비싼 가격이 단점

- throughput : 컴퓨터 시스템의 처리능력을 나타내는 개념

다양한 분야에서 사용될 수 있으며, 네트워크, 데이터베이스, 컴퓨터 시스템, 제조 공정 등에서 중요한 성능 지표로 사용됨

throughput의 일반적 정의

네트워크: 단위 시간당 성공적으로 전달된 데이터의 양을 의미 (일반적으로 초당 비트 수(bps)로 측정)

컴퓨터 시스템: 시스템이 단위 시간당 처리할 수 있는 작업의 수를 의미

제조 공정: 단위 시간당 생산되는 제품의 수를 의미

- Challenges in handling bigdata : 빅데이터를 다루면서 마주하는 문제들

→ storage = 데이터 저장 비용과 저장 장치의 확장성 문제 / 빅데이터를 저장하기 위해서는 막대한 양의 스토리지 공간이 필요하며, 이 공간을 효과적으로 확장하는 것이 중요

→ network = 국제적으로 대규모 데이터셋을 공유할 때의 문제점 / 연구자들에게 데이터를 즉시 제공하기 위해 로컬 네트워크를 구축 필요

→ data integrity = 데이터 무결성을 유지하기 위해 여러 복사본을 만들거나 백업을 제공 / 데이터 손실을 방지하고 항상 정확한 데이터를 보장하는 것이 중요

→ Metadata, provenance, and ontologies = 데이터의 출처와 맥락을 이해하고 관리하는 것이 중요

→ Open access = 데이터에 대한 개방된 접근을 제공하면서도 보안 문제를 해결

→ Very long term data preservation = 데이터를 장기적으로 보존해야 하는 문제 / 데이터셋을 무제한 기간 동안 안전하게 보관하는 방법을 마련

→ Technology = 데이터를 효과적으로 처리하기 위해서는 새로운 아키텍처, 알고리즘, 기술이 필요 / 전문가가 필요

###파이썬 코드 숙제###

**1. 파이썬 기본문법 파일을 다 실행해보시고, 모르는 부분에 대해 질문을 남겨주세요**

**2. 아래 함수를 정의하고 사칙 연산에 대해 충분히 테스트 하는 코드 제시해주세요**

def add(num1, num2):

return num1 + num2

def minus(num1, num2):

return num1 - num2

def multiply(num1, num2):

return num1 \* num2

def div(num1, num2):

return num1 / num2 (일반적인 나눗셈, 결과는 소수점으로 나타냄)

def divint(num1, num2):

return num1 // num2 (나눗셈 결과를 정수형으로 나타냄)

def calc(num1, num2, op):

if op == '+' :

return add(num1,num2)

elif op == '-' :

return minus(num1,num2)

elif op == '\*' :

return multiply(num1,num2)

elif op == '/' :

return div(num1,num2)

elif op == '//' :

return divint(num1,num2)

print(calc(6,13,'+'))

print(calc(6,13,'-'))

print(calc(6,13,'\*'))

print(calc(6,13,'/'))

print(calc(6,13,'//'))

=> divide-by-zero 해결방법

1) try-except 활용하기

try:

result = a / 0

except ZeroDivisionError:

result = None # 또는 다른 적절한 값으로 초기화

print("Error: Division by zero is not allowed.")

print(result)

2) if denominator == 0:

result = None # 또는 다른 적절한 값으로 초기화

print("Error: Division by zero is not allowed.")

else:

result = numerator / denominator

print(result)

3) 함수 이용

def divide(x, y):

if denominator == 0:

print("Error: Division by zero is not allowed.")

return None # 또는 다른 적절한 값으로 초기화

else:

return numerator / denominator

**3. broading cast / list comprehension에 대한 개념**

broad casting = numpy에서 사용되는 개념으로 서로 다른 크기의 배열 간에 수학적 연산을 효율적으로 수행할 수 있도록 함 (브로드캐스팅이 메모리를 비효율적으로 사용하여 계산 속도가 느려지기 때문에 나쁜 생각인 경우 존재)

import numpy as np

a = np.array([1, 2, 3])

b = 2

result = a \* b

print(result)

import numpy as np

a = np.array([[1, 2, 3], [4, 5, 6]])

b = np.array([1, 2, 3])

result = a + b

print(result)

list comprehension = 파이썬에서 리스트를 간결하고 효율적으로 생성할 수 있는 방법, 기존의 리스트를 변경하거나 조건을 적용하여 새로운 리스트를 생성할 때 유용

**4. 1~100의 값이 들어있는 리스트를 생성하고 list comprehension을 활용해서 3의 배수만 골라 낸다.**

numbers = list(range(1, 101))

multiples\_of\_three = [x for x in numbers if x % 3 == 0]

print(multiples\_of\_three)

**5. list과 tuple자료형에 대해 조사하고 둘의 다른점이 무엇인지 간단히 설명 후 아래코드 작성**

=> 문자열을 받고 각 알파벳이 몇개인지 출력하는 코드작성

list = 다양한 데이터 타입 표현 가능하며 순서 변경 가능 / [] 이용 / 데이터 추가와 삭제 가능

tuple = 리스트와 유사하지만 변경 불가능 / () 이용 / 데이터 추가와 삭제 불가능

@문자열을 받고 각 단어로 쪼갠후 이를 리스트에 저장해서 반환하는 함수

def toList(str):

print("toList Called!!")

print(" my ret is [{0}]",list(str))

return list(str)

@알파뱃이 들어있는 리스트를 받으면 알파벳이 몇개인지 저장하고 이를 반환하는 함수

def makeDict(elem):

print("makeDict Called!!")

print(" my ret is [{0}]",[elem[0],1])

return [elem[0],1]

def mapping(lst):

print("mapping Called!!")

print(" my lst is [{0}]",list(map(makeDict,lst)))

return list(map(makeDict,lst))

@ 데이터를 출력해주는 함수

def shuffle(mapped\_result):

shuffled\_result = {}

for key, val in mapped\_result:

shuffled\_result[key] = shuffled\_result.get(key, []) + [val]

print("shuffle Called!!")

print(" my ret is [{0}]",shuffled\_result)

return shuffled\_result

def reduce(shuffled\_data):

ret = {}

for key in shuffled\_data.keys():

ret[key] = sum(shuffled\_data[key])

print("reduce Called!!")

print(" my ret is [{0}]",ret)

return ret;

def repeatprint(reduced\_data):

var = ""

for key,value in reduced\_data.items():

var += (key + str(value))

#try var += (key \* value)

return var

repeatprint(reduce(shuffle(mapping(toList(“todayweather")))))

**6. numpy / pandas / matplot의 개념 설명**

=> 일반 list와 numpy를 활용해 크기가 100만인 데이터를 만든후, %time을 활용해서 리스트내에 모든 값을 random값으로 채운 후 2를 곱하는 연산을 1000번씩 수행한 후 실행결과를 출력한다 (list vs numpy가 구분되도록, 수행속도 비교가 목적)

numpy = 다차원 배열 객체와 이를 효율적으로 조작하기 위한 다양한 함수들을 제공하는 파이썬 라이브러리 / 배열 간의 빠른 수학 연산 가능

pandas = 데이터 조작 및 분석을 위한 고성능, 사용이 간편한 데이터 구조와 데이터 분석 도구를 제공하는 라이브러리입 / 특히 테이블 형태의 데이터를 다루는 데 유용

my\_array = np.arange(1000000)

my\_list = list(range(1000000))

print("np array : ")

%time for \_ in range(1000): my\_array2 = my\_array \* 2

print("list : ")

%time for \_ in range(1000): my\_list2 = my\_list \* 2

결과) np array :

Wall time: 3.91 s

list :

Wall time: 39.8 s

=> matplot에 대해서 조금더 알아봅니다

matplot = 데이터를 시각화하는 데 사용되는 포괄적인 라이브러리 / 다양한 종류의 플롯(라인 플롯, 바 플롯, 히스토그램 등)을 생성 가능

- Grouped bar chart with labels

import matplotlib.pyplot as plt

labels = ['G1', 'G2', 'G3', 'G4', 'G5']

men\_means = [20, 34, 30, 35, 27]

women\_means = [25, 32, 34, 20, 25]

x = np.arange(len(labels))

width = 0.35

fig, ax = plt.subplots()

ax.bar(x - width/2, men\_means, width, label='Men')

ax.bar(x + width/2, women\_means, width, label='Women')

ax.set\_xlabel('Groups')

ax.set\_ylabel('Scores')

ax.set\_title('Scores by group and gender')

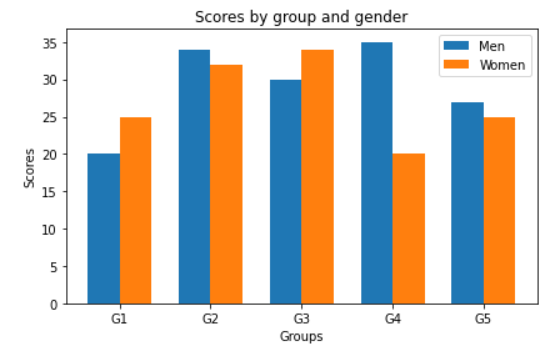
ax.set\_xticks(x)

ax.set\_xticklabels(labels)

ax.legend()

plt.tight\_layout()

plt.show()



- Horizontal bar chart

categories = ['A', 'B', 'C', 'D', 'E']

values = [3, 12, 5, 18, 45]

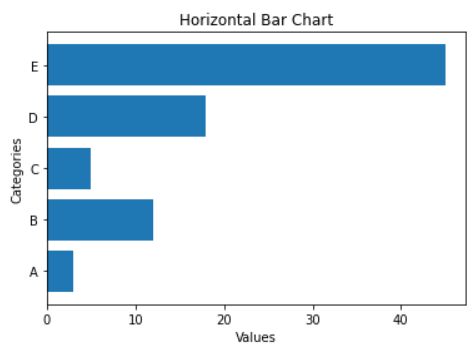
plt.barh(categories, values)

plt.xlabel('Values')

plt.ylabel('Categories')

plt.title('Horizontal Bar Chart')

plt.show()



- Plotting categorical variables

categories = ['cat1', 'cat2', 'cat3', 'cat4']

values = [5, 7, 8, 2]

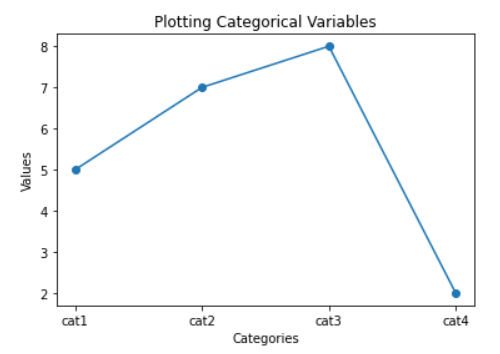
plt.plot(categories, values, 'o-')

plt.xlabel('Categories')

plt.ylabel('Values')

plt.title('Plotting Categorical Variables')

plt.show()



- Histograms

data = np.random.randn(1000)

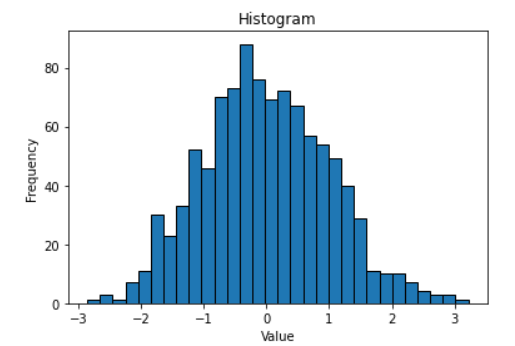
plt.hist(data, bins=30, edgecolor='black')

plt.xlabel('Value')

plt.ylabel('Frequency')

plt.title('Histogram')

plt.show()



- Plotting with keywords

x = np.linspace(0, 10, 100)

y1 = np.sin(x)

y2 = np.cos(x)

plt.plot(x, y1, label='Sine wave', color='blue', linewidth=2)

plt.plot(x, y2, label='Cosine wave', color='red', linestyle='--')

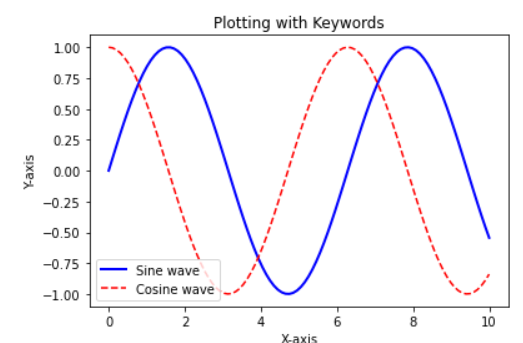
plt.xlabel('X-axis')

plt.ylabel('Y-axis')

plt.title('Plotting with Keywords')

plt.legend()

plt.show()



**7. 가트너 그룹은 "The Big Data Value Model"을 통해 빅데이터 분석의 주요 활용 목적을 아래와 같이 분류했습니다. 아래에 대해서 사례를 기반으로 정리 해주세요**

\* Customer Insight = 고객의 행동, 선호도, 요구 사항 등을 깊이 이해하여 맞춤형 서비스 및 제품을 제공하고, 고객 만족도와 충성도를 향상시키는 것

예) 넷플릭스, 유튜브 - 개인별 맞춤 콘텐츠 제공

\* Process Efficiency = 비즈니스 운영 프로세스를 최적화하고 비용을 절감하며 생산성을 향상시키는 것

예) 에어비앤비 - P2P 숙박 모델은 보다 개인화되고 저렴한 숙박 옵션을 제공

\* Digital Products & Service = 디지털 기술을 활용하여 새로운 제품과 서비스를 개발하고, 기존 제품과 서비스를 혁신하는 것

예) 스포티파이 - 빅데이터 분석을 통해 사용자별로 추천 플레이리스트 제공

\* Operational Excellence, Digital Marketing = 운영 효율성을 극대화하고, 디지털 마케팅을 통해 고객에게 더 효과적으로 다가가는 것

예) 스타벅스 - 더 좋은 제품과 서비스를 제공하기 위해 직원 교육 프로그램과 sns마케팅을 활발히 진행 중

**8. GE의 산업 인터넷(Industrial Internet)에 대해서 조사해 주세요**

GE (General Electric) 은 미국의 회사로 사물 인터넷의 산업적 활용에 주목하고 있음

항공기 엔진에 센서를 부착해 엔진의 상태와 연료 소비량 등을 측정하고, 기존과 비교해 더 효과적인 운행이 가능하도록 하는 서비스 제공

콴타스 항공과 협력해 만든 플라이트펄스(FlightPulse)(링크) 앱이 있다. 이 앱을 통해 콴타스 항공 조종사들은 개별 비행에 대한 데이터를 보고, 이를 통해 연료 절감과 연비 개선 여지 등을 알 수 있음

클라우드 기반의 사물인터넷 플랫폼 = 프레딕스 → 소프트웨어 개발을 가능하게 하는 서비스형 플랫폼 (PaaS) 기계에서 발생하는 대규모의 데이터들을 분석 및 수집하는 클라우드 솔루션

위스의 식품 포장 기계 제조 기업 SIG(링크)는 프레딕스 플랫폼의 프레딕스APM과 서비스맥스(ServiceMax) 애플리케이션을 결합하여 설비의 성능을 추적하여 최적의 유지보수 전략을 실행

**9. 아래는 정부에서 발간한 2021 데이터산업 현황조사 결과보고서 입니다.**

데이터 산업 시장의 경우 지속적으로 성장하고 있으며 특히 데이터 판매 및 제공 서비스업의 성장이 돋보인다. 또한 데이터를 활용하는 산업 분야가 매우 다양하며 공기업과 사기업 모두 증가하는 추세를 보였다. 국내 뿐만 아니라 세계적으로도 크게 계속 성장할 것으로 예측되며 미국이 가장 주도적인 역할을 할 것으로 보인다.

이에 따라 데이터 산업의 종사자 역시 필요 인력이 증가하고 있다. 데이터 개발자, 데이터엔지니어, 데이터베이스 관리자, 데이터 기획자, 데이터분석가, 데이터 컨설턴트 등의 직업들이 있으며 수요와 공급이 모두 높게 나타나고 있다. 보통 남성이 주를 이루지만 데이터 개발자 직무에서는 여성이 상대적으로 높게 나타났다. 직무별로 보면 이 중 데이터 개발자와 데이터베이스 관리자가 비중이 가장 크다. 또한 인력 현황의 경우 중급, 고급, 초급 순으로 존재하며 수요가 매우 높은 직군이다. 특히 데이터 과학자의 인력이 많이 부족하다. 또한 2021년 기준 5년 이내에 추가로 필요한 기술등급의 경우 중급이 가장 높게 나타났다. 그러나 2021년의 경우 채용이 굉장히 많았으나 현재는 많이 감소한 상태이다. 실무 역량 및 경험을 갖춘 사람을 원하기 때문이며 분석 프로그래밍, 통계적 분석 능력 등을 기본적으로 요구하고 있다. 우대 자격증의 경우 정보처리기사, ADP, SQLD 등이 있다.

해외 진출, 매출의 경우를 보면 일본과 베트남, 중국, 미국 순으로 나타나고 있다. 데이터 산업의 경우 지속적으로 수요가 필요한 직업이므로 인력은 계속 필요할 것으로 예상된다.

**10. 데이터 엔지니어와 데이터 사이언티스트의 차이점은 무엇인지 조사해주세요**

- 데이터 엔지니어 = 주로 데이터 파이프라인을 설계, 구축, 관리하는 업무에 집중

대규모 데이터를 수집하고 저장하기 위한 시스템을 구축하며, 데이터 웨어하우스나 데이터 레이크를 관리하여 데이터가 안정적으로 수집, 저장, 접근될 수 있도록 함. 주로 데이터베이스 기술, 데이터 처리 기술에 중점

- 데이터 사이언티스트 = 데이터를 분석하여 인사이트를 도출하고, 예측 모델을 개발하여 비즈니스 문제를 해결하는 데 집중

데이터를 탐색하고 시각화하여 패턴이나 트렌드를 발견하며, 머신러닝 알고리즘을 사용하여 예측 모델을 구축하고 평가. 주로 통계, 머신러닝, 데이터 분석 기술에 중점