개인 KPI 정리

github 주소 : https://github.com/gilgim/BigdataStudy

cloneUrl : https://github.com/gilgim/BigdataStudy.git

성과목표 : 머신러닝 기술 습득

- **학습 개요 및 진행 순서**

    1. 미적분학

        - 교재 : 미분적분학 바이블 (한빛 아카데미)
        - 함수의 극한 ~ 매개변수 방정식과 극좌표
            - 벡터공간과 연산을 위해 기본적인 과정으로 학습.

    2. 선형대수학

        - 노트 정리 및 문제 풀이
        - 선형 대수학과 빅데이터의 연관성 학습

    3. 파이썬

        - PEP8 학습
        - Test Program 작성

    4. 머신러닝 이론 및 모델링 학습

        - 머신러닝 : 빅데이터 학습 과정.ipynb에 정리해두었다.

• 학습 상세 내용 및 기반 기술

1. **미적분학**

    (선형대수학을 위한 기초 선행 과정이다.)

    - **함수의 극한**
      수의 확장을 위해서 필요한 개념으로 실수 체(Field)의 증명과 후에 나오는 선형대수학에서의 벡터체를 이해하는데 선행되는 학습이다. 극한의 개념을 통해 점과 점 사이의 간격을 무한대로 좁혀 해당 지점의 기울기를 구하거나 그래프를 무한한 직사각형으로 채워 넓이를 구하는 개념에 사용되는 기초 개념이다.

    - **도함수 및 응용 & 적분 및 적분의 응용**
      벡터의 연산을 위한 행렬식을 위한 선행과정이다. 해당 부분은 매개변수를 이용한 단위접선 벡터를 활용할 수 있고, 곡률를 구하는 등 벡터 연산에서 필수적인 선행과정이다. 벡터의 기저를 구할 때 적분의 개념을 사용할 수 있으며, 벡터의 공간을 정의할 때 사용된다.

    - **매개변수**
      x,y 등 좌표계를 더욱 넓은 환경에서 접근할 수 있게 만들어 준다고 생각한다. 벡터의 방향수(벡터가 나아가야

할 방향)를 정할 때도 사용되고, x,y,z를 하나의 매개변수로 표현할 수 있기 때문에 세 값을 관련짓게 해주는 중요한 개념이다.

- **벡터와 공간기하**
  스칼라와 벡터를 구분짓고 벡터가 좌표계에서 가지는 기하적의미를 공부하고 위치벡터, 단위벡터, 접선벡터, 외적, 내적 등을 학습하여 선형대수학 전에 벡터의 기초를 다질 수 있다.

2. **선형대수학**

(머신러닝 학습 시 라이브러리인 sklearn, pandas를 이해하기 위한 선행 과정이다.)

- **벡터 공간**
  벡터 체를 정의하기 위해 실수 체에서의 항등원과 역원의 이해를 바탕으로 벡터체가 성립하기 위한 증명 과정을 담고있다. 벡터의 일차독립, 기저, 차원 등을 정의하여 벡터공간을 구성할 수 있는 최소집합을 구한다. 벡터공간 내에서 행렬를 함께 익히고 행렬은 pandas에서 DataFrame의 형태로 사용가능하다.

- **선형변환과 행렬**
  일반 식을 행렬로 변환하는 것을 벡터 공간을 통해 벡터공간의 기저와 차원의 개념을 통해 익혔다면 변환된 행렬 식을 선형화 할 수 있다. 식을 선형화하게 되면 데이터들의 값을 비교적 높은 확률로 예측할 수 있게 되고 해당 과정은 python의 sklearn 라이브러리의 함수를 통해 사용할 수 있다. 앞서 공부한 행렬을 희소 행렬, 밀집 행렬 등으로 변화시키는 함수 또한 라이브러리에 포함되어있다.

3. **파이썬**

- 다른 언어와 달리 tap을 이용해 구문을 구분하고 오픈 소스 기반으로 성장한 언어이기에 개발자들이 읽기 쉽게 하기위한 코딩 스타일 가이드인 PEP이 존재한다.

- 해당 PEP8를 본 내용은 https://zerosheepmoo.github.io/pep8-in-korean/doc/introduction.html 에서 확인 할 수 있다.

- 내용 요약은 ipynb 파일로 정리해 깃허브에 올려두었다.

- 기초 문법은 익히기 쉬우므로 md 파일에 정리하진 않겠지만 문법 https://wikidocs.net/book/1 에서 익힐 수 있다.

- 다른언어와의 차이점을 간단하게 하면 파이썬은 타입이 정해지지 않았고 스위치문이 존재하지 않는다.

- C와 C++ 과의 연동 작업이 쉬워 속도가 느리지많은 않다.

- 머신러닝 공부 시 기초가되는 프로그램을 작성해 빅데이터 학습과정.ipynb 파일과 함께 익힐 수 있다. 시작 파일은 check_data.py 이다.

- **numpy**

  - 파이썬에 리스트가 존재함에도 불구하고 numpy를 사용하는 이유는 파이썬은 모든 타입을 리스트에 넣을 수 있기에 속도가 느리지만 넘파이는 C로 구현되어 있기 때문에 속도가 빠르다. 하지만 오로지 하나의 타입의 리스트만 구현이 가능하다. 해당 머신러닝에서 쓰이는 배열은 numpy를 as np로 하여 축약해서 사용한다.

- **pandas**

        - 데이터를 쉽게 접근하고 다루고 정제하기 위해 DataFrame이라는 틀을 제공하고 각종 분할 기능을 제공하는 함수이다. 작성한 프로그램에서는 첫 시작에 계층화를 허용하게되면 pandas를 이용하게된다.

    - **sklearn**

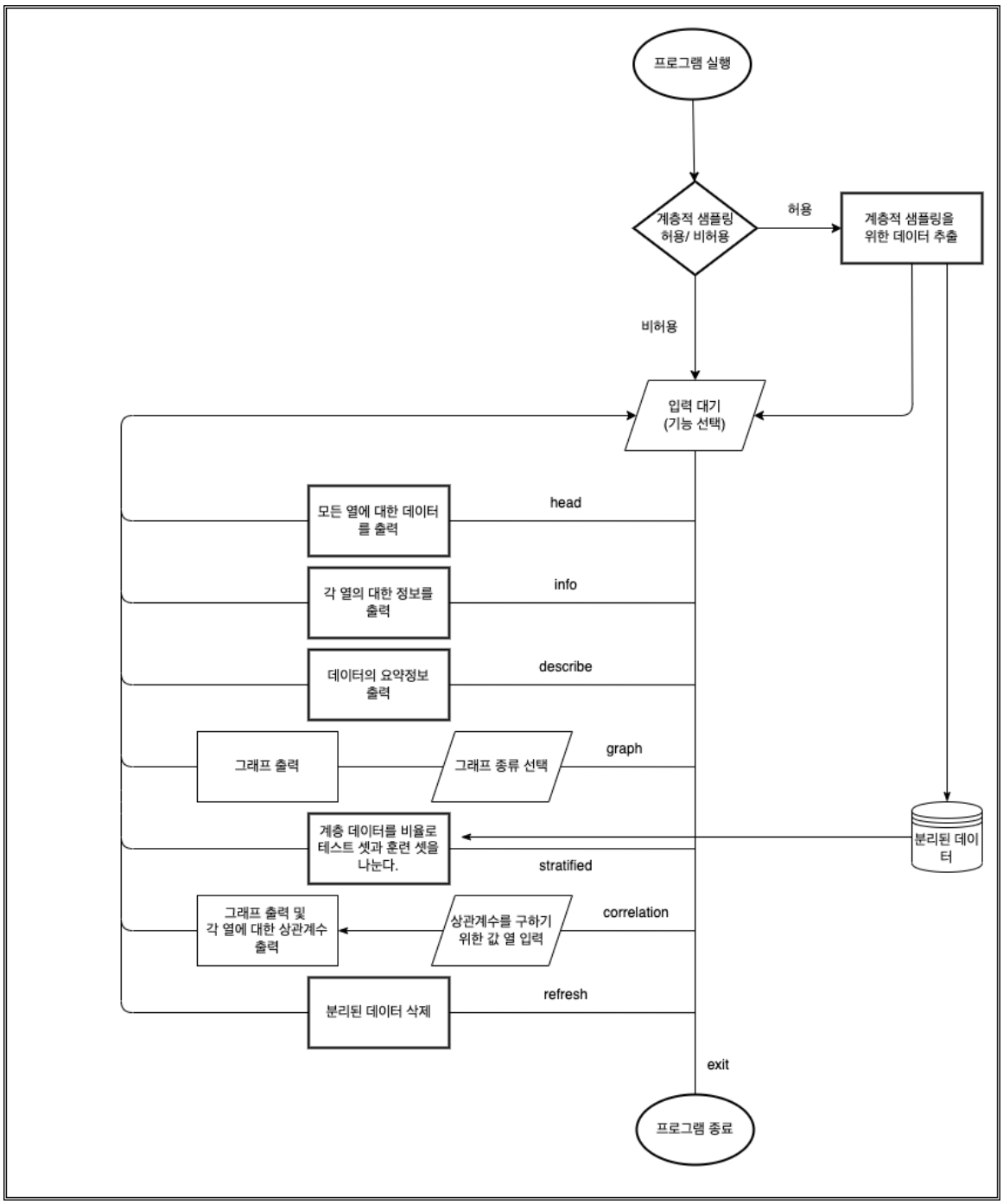
        - 계층화된 데이터를 통해 데이터를 훈련과 테스트에 맞게 정제하는 기능을 제공한다. 그리고 LinearRegression을 통해 선형화 작업을 진행하면 선형 회귀 모델을 만들 수 있다.

4. **프로그램 구성**

(프로그램의 목적은 데이터를 분석하는 것에 목적이 있으면 학습은 주피터 노트북에서 학습한다.)

- 첫 시작 시 계층적 분할의 여부를 묻고 분할 시 테스트 케이스의 분리는 랜덤하게 된다.
- head 를 입력 시 각 데이터의 컬럼 값을 확인할 수 있다.
- info 를 입력 시 추후 희소행렬 사용 여부를 위한 컬럼 타입을 확인 할 수 있다.
- describe 를 입력 시 데이터 전반의 요약 정보를 확인 할 수 있다.
- graph 를 입력하면 계층화에 필요한 데이터 정보와, 전체적인 데이터 정보를 그래프로 확인할 수 있다.
- stratified 를 입력하면 계층화 데이터가 테스트 셋과 훈련 셋으로 분할 가능할 시 분할 시킨다.
- correlation 를 입력하면 서로 컬럼과의 상관계수를 출력한다.
- refresh 를 입력하면 분할된 테스트 셋과 훈련 셋을 다시 하나로 합친다.
- exit 프로그램을 종요한다.

해당 프로그램의 프로세스 흐름도



마치며

- 머신러닝 모델 하나하나가 깊은 지식이 필요하기에 많은 시간이 걸린다.
- 머신러닝이 필요한 데이터의 예측값을 실무에서 사용하게 되더라도 이론을 모르면 해당 결과 값을 사용하는데 이해도가 떨어지게 된다.
- 머신러닝 학습 중 DB의 값을 이용한 선형화가 선형대수학 이론을 사용하는 것을 선형화 과정으로써 체감하였고, 모델의 적용을 위해서 이론 공부가 선행되어야한다고 생각한다.
- 파이썬은 다른 언어들 보다 타입를 생략하고 키워드 언어가 많이 없어 쉽지만 코딩 가이드가 필요하고 타입을 명시하지 않아 분석에 어려움이 있을 것 같다.
- 미적분학은 미분과 적분은 기초개념만 습득하고 매개변수와 벡터를 탄탄히 공부하여야 선형대수학을 이해하기에 쉬울 것으로 예측된다.
- 추후에 신경망을 공부할 예정이다.