|  |  |  |  |
| --- | --- | --- | --- |
| **2023-1 머신러닝 및 실습** | | | |
| 학번 | 2021136150 | 이름 | 양동재 |

**[문제 1]**

1. **문제 정의**

이번 실습 내용은 Machine Learning에 대한 이해를 바탕으로 Machine Learning의 procedure을 이해하고 Linear regression algorithm을 구현해 봄으로써 Linear regression의 kernel algorithm의 동작 원리를 파악하는 것 이다.

* Case#1 composite frequency

5개의 frequecncy로 구성되어있는 composite frequency에 대해서 각 주파수가 몇 hz일때, composite frequency가 되는지 계산하는 알고리즘을 구현하는것이 문제이며, 기초적인 알고리즘에 대한 지식을 요구한다.

* Case#2 Predict Concrete Compressive strength

Linear regression algorithm을 구현하고, 구현한 algorithm을 통해 8개의 features를 활용하여 target value(concrete compressive strength)를 예측하는 모델을 개발하는것이 목표이며, 문제 해결을 위해 linear algebra, Loss function, gradient descent, data preprocessing, feature extraction, model evaluation method 등 Machine Learning에 전반적인 지식을 요구한다.

1. **학습 전략 (e.g., 데이터셋 분석, 가설 수립)  
     
   1) Case#1**
2. Greedy algorithm

Case#1 문제의 경우, 4개의 sine waves에서 composite frequency를 만드는 조합을 찾는 문제이다. 위 문제의 경우, composite frequency를 찾는 sine wave의 amplitude를 찾는 문제이기 때문에 가장 기본적인 greedy algorithm으로 접근해볼 수 있다. 단, Case#1의 경우 algorithm이 찾아야하는 조합이 무한대이기 때문에, computer resource를 최소화 하는 방향도 고려해야한다.

1. Random algorithm

Greedy algorithm의 경우, composite frequency를 찾기위해 0부터 ∞까지 1씩 값을 증가시키며 찾기 때문에, 찍었는데 맞추는 것 과 같은 ‘행운’을 기대할 수 없다. 따라서 랜덤하게 측정하는 방법 또한 computing resource를 효율적으로 활용할 수 있는 방법이 될 수 있음으로 random algorithm또한 고려를 해야한다.

1. **Case#2**

**A. Hypothesis**

Case#2 문제의 경우, Target Value (Concrete Compressive Strength)가 categorical data가 아닌, sequential value이기 때문에 문제를 Regression 문제로 정의하고, 교과목에서 공부한 Linea regression을 통해 문제를 해결하고자 한다.

a. gradient descent method

Case#2의 경우 cost function이 convex하지 않을 수 있기때문에 gradient descent method를 통해 local minimum이 아닌 global minimum value를 찾아야 한다. 따라서 gradient descent method 방법을 통한 linear regression 구현도 필요하다.

b. Normal Equation

Case#2의 데이터 셋은 데이터의 숫자와 feature가 크 지 않기 때문에, Normal Equation방법을 고려하여 gradient descent보다 더 빠르게 문제를 해결할 수 있을것으로 보인다.

**B. EDA(Exploratory Data Analysis)**

1. 데이터 결측치 확인

본 데이터의 경우, ‘0’이 갖는 의미또한 명확하다. 따라서 0은 의미있는 데이터로 인정하여야 하지만 데이터 중 결측치 NaN Value가 있을 수 있으므로, 이를 확인하고 만약 있다면 NaN Value는 Noise value가 될 수 있으므로, 해당 데이터를 제거하는 방식으로 처리를 계획했으나, 해당 데이터에 결측치는 존재하지 않았다.

1. 데이터 Noise 확인

Chart, box and whisker chart

Description automatically generated

Box plot을 통해 이상치를 확인하였다. Age를 제외하면 큰 이상치가 보이지 않으며, Age도 feature의 특성상 과도하게 높은 데이터 충분히 있을 수 있으므로 Noise가 아닌 의미를 갖는 데이터로 바라보아야 한다. 따라서 Noise 제거는 따로 진행하지 않았다.

1. Data Normalization

Case#2의 데이터는 feature간의 scale이 차이가 심하게 나기 때문에 normalization을 수행해주어야 한다. 해당 데이터의 경우 outlier를 보존하는 방식으로 data preprocessing 과정을 수행했기 때문에 outlier에 영향을 많이 받는 MinMax normalization이 아닌 Z-score Normalization을 수행했다.

1. **자신이 구현한 함수 및 클래스에 대한 설명 또는 유도 과정 작성  
   (MS equation 사용)**

* **MS equation**
  + **matrix** : 대문자 + bold + itelic
  + **vector** : bold + itelic
  + **scalar** : itelic

Case #1

Greedy search

1. 각 주파수에 순차적으로 +1씩 증가

2. mse 측정

Random search

1. frequency가 나올 수 있는 조합의 수로 전체로 2x n 배열 생성

2. 난수 생성을 통한 랜덤 주파수 조합 생성

3. mse 측정

4. 배열에서 랜덤 주파수 조합 제거 (랜덤 주파수 생성 경우의수 최소화를 위함)

Case #2

Data normalization

* + - Minmax scaling formular
    - z-score scaling formular

Model Evaluation

Gradient

Normal Equation

1. **결과 비교 및 분석**

Case #1

Random Search와 greedy search 알고리즘 모두 결과적으로 무한대에 가까운 연산을 진행하여야한다. 따라서 해결할 수 있는 문제이나, computing power에 따라서 해결할 수 있는 시점이 달라진다.

Case #2 Interation증가에 따른 성능분석

Chart, line chart

Description automatically generated

반복 횟수가 늘어날때마다 MSE값이 기하급수적으로 낮아지는 것을 확인할 수 있다. 이러한 현상이 발생하는 이유는 Linea regression algorithm이 편미분을 통해 feature x가 어떻게 변해야 y값에 근사할 수 있는지 계산하기 때문이다.

1. **고찰 및 느낀점**

* Machine learning의 경우, 모델 개발 뿐만 아니라, Feature Extraction, Data Preprocessing 과정 등이 좋은 모델을 만들때 훨씬 중요하다. 그 이유는 결국 Machine Learning도 Training Data를 활용하여 정답에 근사시키는 개념이기 때문에, Garbage in Garbage Out 적용된다.
* EDA를 통해 Data의 specific한 점들을 통찰하고, 그에 맞는 Preprocessing 기법들을 활용하는게 모델의 성능을 올리때 많은 영향을 준다.
* 일정 Iteration을 넘어가면 성능 향상의 폭이 기하급수적으로 줄어든다. 따라서 어느정도 목표치에 근사하였다면, 학습을 멈추는것이 Resource관리에 좋다.