**머신러닝 문제 만들기**

**21011928 전주혁**

1. 정확성 (5점): 얼마나 데이터를 정확히 파악하고, 그에 맞는 문제를 설계 하였는가?

2. 문제난이도 (5점): 데이터로부터 문제를 만드는 과정이 너무 쉽지는 않았는가? (기존에 공개된 내용/자료와 너무 유사하지는 않은가?)

3. 설명 (5점): 데이터셋으로부터 문제를 만드는 모든 과정에 대한 설명이 (시행착오, 여러 시도과정, 모델선정과정 등) 자세히 잘 되어 있는가?

4. 결과분석 (5점): 문제에 대한 코드를 사용한 결과에 대해 잘 분석 하였는가?

선택한 데이터 : 수산양식업 먹이생물 AI 학습 데이터

이번 중간고사 머신러닝 문제 만들기를 진행하기 위해 최대한 정형 데이터 위주인 데이터셋을 찾으려 노력하였다.

비정형 데이터를 사용한다면 현재 배우고 있는 머신러닝 기법들이 아닌 딥러닝 기법을 사용해야 할 수도 있기 때문이다.

그렇게 수산양식업 먹이생물 AI 학습데이터를 선택하게 되었다.

수산양식업 먹이생물 AI 학습 데이터는 총 95.85 GB의 용량으로 구성되어있다.

먼저 노트북의 쓰지 않는 파일들을 제거해 보았지만 약 100GB의 저장 공간은 부족하였다.

그래서 어떻게 하면 기존에 공개된 기계학습 방법과 유사하지 않게 최소한의 데이터를 사용하여 좋은 문제를 만들어 낼 수 있을지 데이터의 접근법을 먼저 생각해보았다.

수산양식업 먹이생물 AI 학습 데이터셋은 Training과 Validation set으로 구성되어 있었다.

Training Set은

1 - 1. 라벨링데이터

1 - 1 - 1 : 먹이생물

1 - 1 - 1 - 1 : 먹이생물\_이미지

1 - 1 - 1 - 2 : 먹이생물(수질+cell).csv

1 - 1 - 2 : 인공종묘

1 - 1 - 2 - 1 : 유생기 (성장, 행동패턴).csv : 11월13일~12월13일까지의 성장 데이터, 10월1일~12월13일까지의 행동패턴 데이터

1 - 1 - 2 - 2 : 부착지 (성장).csv : 12월8일~12월12일까지의 데이터

1 - 1 - 2 - 3 : 가이식기 (성장).csv : 12월15일, 12월17일 데이터

1 - 1 - 2 - 4 : 본이식기 (성장).csv : 10월14일~12월16일까지 7일 간격 데이터(불규칙)

1 - 2. 원천데이터

1 - 2 - 1 : 먹이생물

1 - 2 - 1 - 1 : 먹이생물

1 - 2 - 1 - 2 : 먹이생물(수질+cell).csv 날짜별 데이터(9월2일~11월12일까지의 데이터)

1 - 2 - 2 : 인공종묘

1 - 2 - 2 - 1 : 유생기(수질, 먹이급이).csv : 11월13~ 12월 13일까지의 데이터

1 - 2 - 2 - 2 : 부착기(수질, 먹이급이).csv : 12월8일~12월12일까지의 데이터

1 - 2 - 2 - 3 : 가이식기(유효플랑크톤).csv : 12월17일 데이터

1 - 2 - 2 - 4 : 본이식기(유효플랑크톤).csv : 10월14일~12월16일까지 7일간격 데이터(ex,14일 21일 28일)

이렇게 나누어져 있었고

Validation Set은

2. Validation

2 - 1. 라벨링데이터

1 - 1 - 1 - 1 : 먹이생물\_이미지(사용x)

1 - 1 - 2 : 인공종묘

1 - 1 - 2 - 1 : 유생기 (성장, 행동패턴).csv : 10월1일~10월22일까지의 성장 데이터, 10월1일~12월13일까지의 행동패턴 데이터

1 - 1 - 2 - 2 : 부착지 (성장).csv : 12월6일 성장 데이터

1 - 1 - 2 - 3 : 가이식기 (성장).csv : 12월13일 성장 데이터

1 - 1 - 2 - 4 : 본이식기 (성장).csv : 9월 15,17,23일 성장 데이터

2 - 2. 원천데이터

2 - 2 - 1 : 먹이생물

2 - 2 - 1 - 1 : 먹이생물\_이미지

2 - 2 - 1 - 2 : 먹이생물(수질+cell) : 8월19일~8월25일 데이터

2 - 2 - 2 : 인공종묘

1 - 2 - 2 - 1 : 유생기(수질, 먹이급이).csv : 10월1~22일까지의 데이터

1 - 2 - 2 - 2 : 부착기(수질, 먹이급이).csv : 12월6일 데이터

1 - 2 - 2 - 3 : 가이식기(유효플랑크톤).csv : 12월13일 데이터

1 - 2 - 2 - 4 : 본이식기(유효플랑크톤).csv : 9월15,17,23일 데이터

로 나누어져 있었다.

먼저 training set의 라벨링 데이터의 인공종묘 -> 유생기 파일과 원천 데이터의 인공종묘 -> 유생기 파일이 11월13일~12월13일 데이터로 날짜가 중복되었고 용량도 적절한 데이터 개수일 것이라 생각했다.

그래서 일단 유생기 파일 안에 존재하는 성장, 행동패턴, 수질, 먹이급여 csv 파일을 살펴 보았다.

유생기 파일내의 각 csv 파일명은 4 -2 – 2 – x –x – 월일 – 1 or 2 로 시작하는 백만자리 숫자로 되어있었다.

먼저 성장 데이터는 ID CODE, Tank NO, Lenghth(um), Height(um), Weight(ug)의 피쳐가 존재하였고, 행동패턴 데이터는 ID CODE, Distance(mm) 피쳐가 존재하였고, 수질 데이터는 Time, Temperature, DO pH, salinity 피쳐가 존재하였고, 먹이급여 데이터는 Tank No, Plakton, Time, Feed(L)가 존재하였다.

먼저 이 네개의 파일을 어떻게 같이 사용할 지 고민을 하였고 공통 column을 찾아보았다.

성장 데이터와 행동 패턴 데이터는 ID CODE의 공통 column이 존재하였고 수질 데이터와 먹이급여 데이터는 Time이란 공통 column이 존재하였다.

또한 성장 데이터와 먹이급여 데이터의 Tank NO column도 동시에 존재하였다.

먹이급여와 성장은 밀접한 관계를 맺고 있다고 생각하였고 Tank NO는 각 물고기가 살고 있는 Tank NO로 생각하였다.

그렇게 나는 성장 데이터와 먹이급여 데이터를 합쳐보기로 하였다.

성장 데이터에는 4-2-2-4-월일-1or2로 데이터의 종류가 나누어졌는데 2로 시작하는 데이터는 먹이급여 데이터에도 맞지 않다 생각해 사용하지 않겠다 생각하였고 그렇게 성장 데이터 csv 29개와 먹이급여 데이터 29개로 숫자가 맞게 되었다.

하지만 이 두 csv 파일을 어떻게 합쳐야 할지 고민이 되었다.

Tank NO라는 공통 컬럼이 존재한다 하여도 성장 데이터는 없는 Tank NO열의 값이 먹이급여 Tank NO열의 값에 존재 하였고 그 역도 존재하였다.

또한 먹이급여 데이터에는 Tank NO의 각 고유값이 하나씩만 존재해야 성장 데이터와 합치기 쉬울것이라 생각하였지만 05시30분, 13시00분, 19시00분 하루에 3번 먹이를 줘서 Tank NO가 그때마다 중복돼서 나오는 문제가 있었고 또 중복도 랜덤으로 진행 되었다.

일단 concat 함수를 이용해 두 csv 파일을 합쳐보았지만 행, 열 두 조건을 주어도 NULL 값이 생기며 원하는 방향으로 두 파일이 합쳐지지 않았다.

그 다음 join 이라는 함수를 알게 되었다.

Join 함수로도 두 파일을 합쳐보았지만 피쳐들이 사라지는 문제가 생겨 join도 포기하게 되었다.

마지막으로 merge라는 함수를 사용해 보았다.

Merge는 left, right, outer, inner 그리고 on 이라는 조건을 줄 수 있어 공통된 열인 Tank NO를 on으로 설정해주고 grow를 기준으로 Merge 시켜주니 원하는 방향으로 맞춰지게 되었다.

하지만 이것도 문제가 존재하였다.

원래 성장 데이터의 행보다 행의 개수가 몇배 이상 늘어나게 된 것이다.

Merge 시킨 데이터를 살펴보니 위에서 생각했었던 05시 30분 13시 30분 19시00분 총 3번씩 Tank NO가 나와 시간만 바뀌고 다른 피쳐는 똑 같은 행이 여러 개 생기게 되는 문제였다.

다시 먹이급여 csv 파일로 돌아오게 되었다.

그리고 공부 끝에 groupby라는 함수를 알게 되었다.

groupby란 공통 행을 하나로 병합시켜주는 함수였다.

그렇게 groupby.mean()을 통해 중복되는 Tank NO를 하나로 합쳐주었고 그 후 다시 성장 데이터와 Merge 시키니 행의 변화 없이 완벽하게 원하는 방향으로 합쳐졌다.

하지만 또 문제가 발생하였다.

이것도 위에서 예상했었던 문제였다. 바로 성장 데이터에 있는 Tank NO의 값이 먹이급여 데이터에는 존재하지 않는 문제였다. 그렇게 존재하지 않는 Tank NO의 행의 새로 생긴 먹이급여 피쳐는 NULL값으로 채워져있었다.

하지만 기계학습 시간에 배웠던 NULL값 처리에 대해 생각하여 NULL 값은 먹이급여를 실시하지 않은 상황이라 생각하여 fillna(0) 으로 처리하였고 그렇게 무사히 두 데이터 파일을 합칠 수 있게 되었다.

학습한 모델로 예측을 진행할 Test 데이터셋 모델은 파일 중 Validation 파일에 들어있는 동일한 이름의 10월달 csv 파일을 사용하였고 위의 방법과 동일하게 merge를 진행하여 csv를 병합하였다.

문제1.(회귀)

전처리를 끝낸 후 각 Tank NO와 길이, 높이, 무게, 플랑크톤으로 평균 먹이의 양을 예측하는 문제를 만들어 주기 위해 train, test 셋으로 분리를 해주었다.

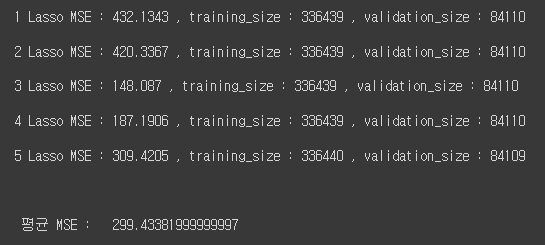
train\_x 데이터셋에는 학습에 필요없는 ID CODE와 target값으로 설정해줄 Feed(L)을 제외한 값을 넣어주었고 train\_y 데이터셋에는 타겟값인 Feed(L)을 넣어주었다.

회귀 예측 모델로는 Lasso, Ridge, Randomforest, Decision Tree를 사용하였다.

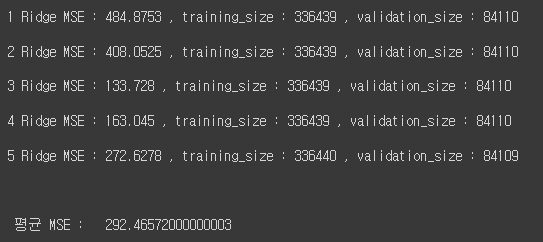
그리고 최종 모델 선택을 하기 위해 평가지표를 활용하려고 교차검증 방법을 사용하였는데 평가 매트릭스는 mean squared error를 사용하였고 fold 수는 5개로 설정하였다.

5폴드의 MSE 평가지표를 확인해보자

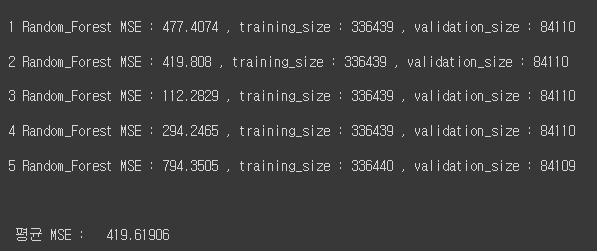
Lasso



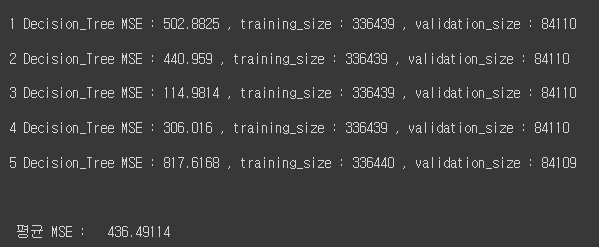
Ridge



Random Forest



Decision Tree

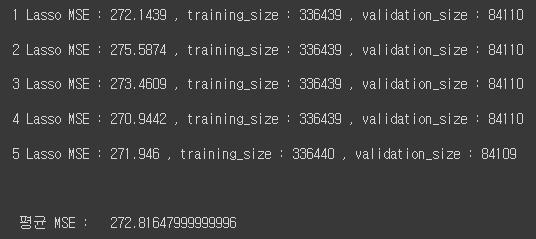


이러한 교차검증 결과를 본 후 폴드마다 MSE 값 차이가 심하다는 것을 알 수 있었고, 특히 폴드 3이 값이 좋게 나오는 것을 확인할 수 있었다.

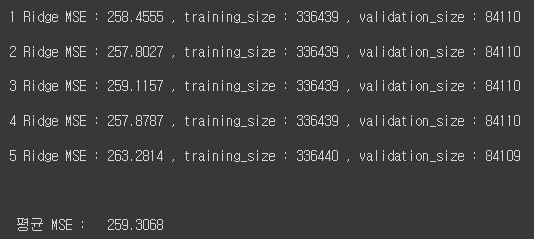
이러한 원인은 데이터가 정렬되어있는 상태에서 fold를 만들게 된다면 순서대로 5폴드씩 자르기 때문에 폴드별로 데이터 불균형 상태가 심해졌기 때문에 이러한 문제가 발생하였다고 생각하였고 KFold의 데이터를 셔플하는 하이퍼파라미터인 shuffle 을 True로 바꿔주고 다시 돌려보았다.

Shuffle =True 로 바꾼 후 평가지표

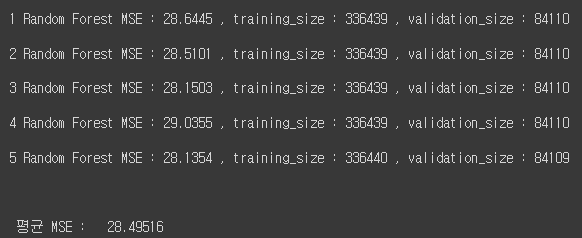
Lasso



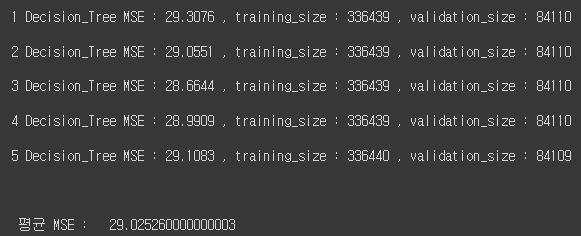
Ridge



Random Forest



Decision Tree



가설을 바탕으로 shuffle을 True로 변경한 후 평가지표를 다시 확인을 해 보니 폴드별 MSE 값의 차이가 거의 없다는 것을 확인할 수 있었고 특히 Random Forest, Decision Tree의 경우 평균 MSE 값이 400정도 하락한 것을 확인 할 수 있다.

평가지표가 좋은 모델인 Random Forest와 Decision Tree 중 성능은 비슷하지만 학습 시간이 Decision Tree가 압도적으로 짧아 모델은 Decision Tree로 사용하는게 효율적이라 생각을 하였고 grid search 방법으로 최적의 하이퍼파라미터를 찾고 학습 후 타겟값 예측까지 진행해 보았다.

Feed(L)의 타겟값을 머신러닝을 통해 예측을 하였고 이렇듯 어종의 원하는 크기, 높이, 무게를 Feed(L) 양 조절을 통해 어느정도 통제할 수 있을 것이라 생각한다.

문제2.(분류)

이번에는 회귀가 아닌 분류로 문제를 만들어보았다.

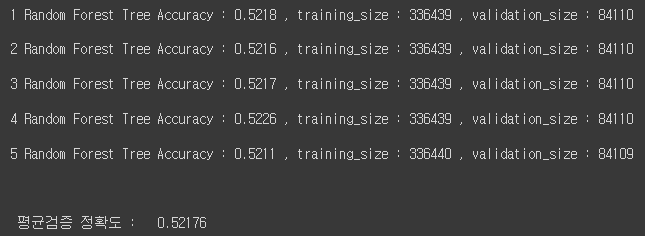
분류로 문제를 만들기 위해 target 값을 Feed(L)이 아닌 TanK NO로 정하였다. 즉, 물고기의 길이 높이, 무게, 플랑크톤, 평균 먹이를 보고 Tank NO를 예측하는 문제이다.

train\_x 데이터셋에는 학습에 필요없는 ID CODE와 target값으로 설정해줄 Tank No를 제외한 값을 넣어주었고 train\_y 데이터셋에는 타겟값인 Tank No를 넣어주었다.

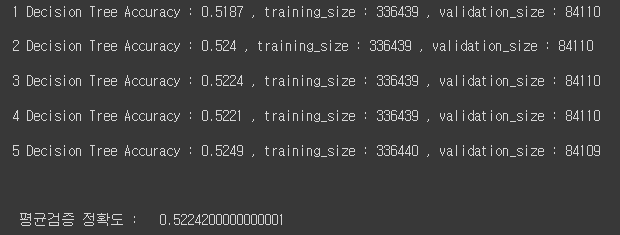
분류 예측 모델로는 Randomforest, Decision Tree를 사용하였다.

그리고 최종 모델 선택을 위해 평가지표를 활용하려고 교차검증 방법을 사용하였는데 평가 매트릭스는 Accuracy를 사용하였고 fold 수는 5개로 설정하였고 shuffle을 True로 설정해 주었다.

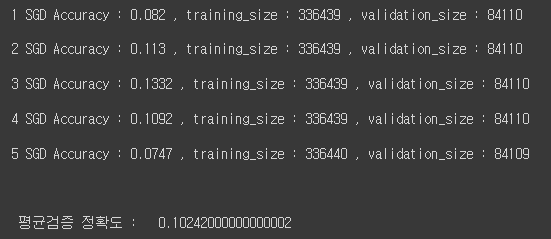
Random Forest



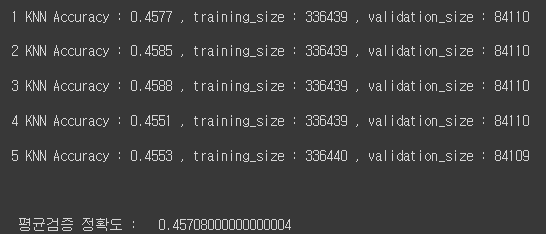
Decision Tree



SGD



KNN

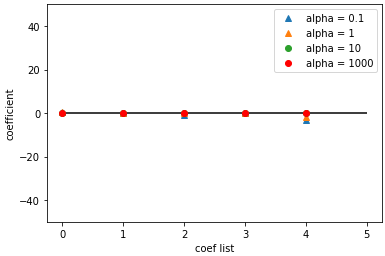


이번 평가지표도 회귀 문제때와 동일하게 RandomForest와 Decision Tree가 평가지표가 가장 좋게 나왔다.

분류문제도 회귀문제와 동일하게 학습시간이 짧은 Decision Tree를 사용하기로 하였고 grid search 방법으로 최적의 하이퍼파라미터를 찾고 학습 후 타겟값 예측까지 진행해 보았다.

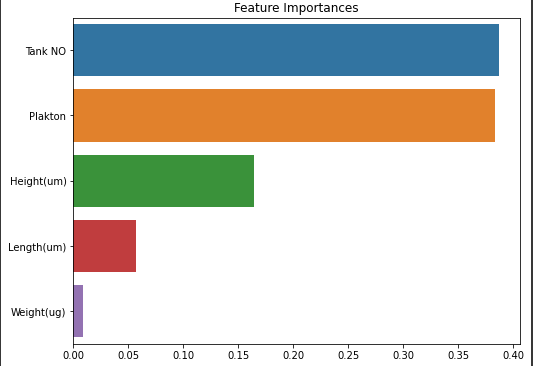
이번 머신러닝 문제 만들기를 통해 종류가 다른 csv를 병합하는 방법과 NULL 값을 처리하는 방법, 교차검증에도 다양한 하이퍼파라미터가 존재한다는 사실을 알게 되었고 각 모델의 하이퍼파라미터에 따른 모델 성질을 더욱 자세히 알게 되었다.

Lasso 모델로 예를 들어 보겠다.



Lasso 모델의 하이퍼파라미터에는 alpha가 존재하는데 alpha값이 커짐에 따라 L1 제약조건이 커지고 alpha의 값이 작아진다면 L1의 제약조건이 작아지는데 위 그래프를 보면서 alpha 크기에 따라 coefficient의 변화도 검증해보았다.

또한 Decision Tree 모델의 Feature Importance도 확인해보았고



모델 학습에 중요한 Feature들도 확인할 수 있다는 사실까지 알 수 있었다.

감사합니다