석 사 학 위 논 문

머신러닝을 이용한 단체급식 다중코너 식수 예측 모델 연구

Forecasting the Number of Customers in Institutional Food Service having Multiple Menus by Applying Machine Learning Methods

고려대학교 컴퓨터정보통신대학원

빅데이터융합학과

조 예 주

2021년 2월

정 순 영 교 수 지 도

석 사 학 위 논 문

머신러닝을 이용한 단체급식 다중코너 식수 예측 모델 연구

Forecasting the Number of Customers in Institutional Food Service having Multiple Menus by Applying Machine Learning Methods

이 논문을 공학 석사학위 논문으로 제출함

2021 년 2 월

고려대학교 컴퓨터정보통신대학원

빅데이터융합학과

조 예 주

조예주의 공학 석사학위논문 심사를 완료함.

2021년 2월

위원장 정순영 (인)

위 원 서태원 (인)

위 원 유헌창 (인)

요 약

기존 단체급식의 취식 인원을 예측함에 있어, 과학적이고 통계적인 방법보다 급식종사자에 의한 경험을 바탕으로 취식 인원 예측이 진행되어 정확도가 다소 부정확하다.

고객만족도 제고, Loss 최소화를 통한 재무적 성과 달성, 음식물 쓰레기의 감소등의 효과를 위해 체계적, 과학적 기법을 기반으로 한 다중코너 취식 인원 예측이 필요한 것으로 파악되었다.

본 연구는 다중코너를 갖고 있는 A급식 사업장의 중식기준 10개 코너 데이터를 기반, 공공데이터를 이용하여 데이터셋을 구축하였으며 이를 통계적 기계학습 모델과 딥러닝 모델인 Linear Regression, Ridge, Lasso, XGBoost, LightGBM, Keras (Multilayer Perceptron), Stacking Ensemble을 적용하여 성능을 측정하였다.

연구 결과로 경험에 의한 다중코너 취식 인원 예측 오차율인 18~20%보다 정확한 예측율인 결정계수 89.1의 성능을 보인 모델을 확인할 수 있었다.

Abstract

Predicting the number of people eating meals in food service organization was not based on a scientific and statistical methods, but the experience of the people working in food service, which led to a slight decrease in accuracy.

In order to improving customer satisfaction, achieving financial performance through minimization of loss, and reducing food waste, it is necessary to predict the number of people eating meals in multiple corners based on systematic and scientific techniques

In this study, a data set was constructed using public data and the A company’ s lunch data which has 10 corners. Based on the dataset, we try to find a model that show high performance by applying statistical machine learning models and deep learning model such as Linear Regression, Ridge, Lasso, XGBoost, LightGBM, Keras (Multilayer Perceptron), and Stacking Ensemble.

As a result of the study, it was possible to create a model of 89.1 with a coefficient of determination that showed a more accurate prediction rate than the predicted error rate of 18-20% by experience.

목 차

그림 목차

.표 목차

1 서론

1.1 연구의 배경 및 필요성

국내 단체급식 시장규모는 '17년 기준 15조원 이상으로 성장 중에 있다.[1]

단체급식에서 취식 인원 예측이란 발주부터 배식까지 이어지는 과정에 있어 전반적으로 영향을 끼칠 수 있는 매우 중요한 의사결정요인이다. 정확한 취식 인원 예측은 조리과정을 효율화 시킬 뿐만 아니라 음식 잔반량을 감소시켜 식재료비를 절감할 수 있는 주요 요소가 된다. 또한 음식의 품절을 막을 수 있어 고객 만족도에도 영향을 줄 수 있다.

현재 단체급식에서는 취식 인원을 예상 및 계획하는데 있어 종사자의 경험을 기반으로 예측이 이루어지고 있다. 이때 제한된 예산 및 정보의 부족으로 예측의 정확성에 대한 신뢰도 문제가 야기되고 있다.

과거 급식은 주로 단일코너, 단일메뉴로 고객의 메뉴 선택권이 제한되어 분산도가 낮아 예측이 수월했던 것에 비해 현재는 프리미엄 급식으로 푸드코트와 같이 복수메뉴를 제공하는 형태로 트렌드가 진화 중에 있다. 이로 인해 취식 인원 예측의 정확성을 하락 시키며 이에 따라 미배식 잔반 및 품절현상이라는 비효율을 가중시키고 있다.

본 연구는 다중코너를 갖고 있는 단체급식 사업장에 도움이 되고자 중식 기준 10개 코너에서 10개의 메뉴를 제공하는 A급식 사업장의 1년 6개월치의 실제 데이터를 이용했다. 해당 데이터를 기반으로 공공데이터를 추가하여 기계학습과 딥러닝을 통해 다중코너 취식 인원을 예측하는데 있어 유의미한 변수를 파악하고 어떤 알고리즘이 취식 인원 예측 모델로 우수한지를 확인하고자 한다.

1.2 기존 연구 고찰

과거 Miller J J 외 2명(1991) “Forecasting production demand in school” [2]에서는 정확한 식수 예측 방법의 필요성을 강조했으나 충분한 연구가 이루어지지 않았다는 것을 확인할 수 있다.

정리나 외 2명(2003) “대학교 급식소의 식수예측 모델 개발”[3]에서는 2000년대 대학교 단체 급식소의 식수를 예측하는데 있어 영향을 미치는 요인을 분석하고, 해당 요인을 사용하여 대학교 단체 급식소에 맞는 수요 예측모델을 소개하였다.

정종식 외 2명(2019) “기계학습방법을 활용한 집단급식소의 식수 예측 S시청 구내직원식당의 실데이터를 기반으로”[4]에서는 8개 범주 카테고리를 선정하여 총 63개 변수들을 추출하고 6가지 기계학습 기법을 활용하여 식수 예측을 진행하였다. 그 결과 기존 예측 오차율이 10~11%대에서 약 6~7% 이내로 줄어 기계학습 방법의 유효성을 증명하였다.

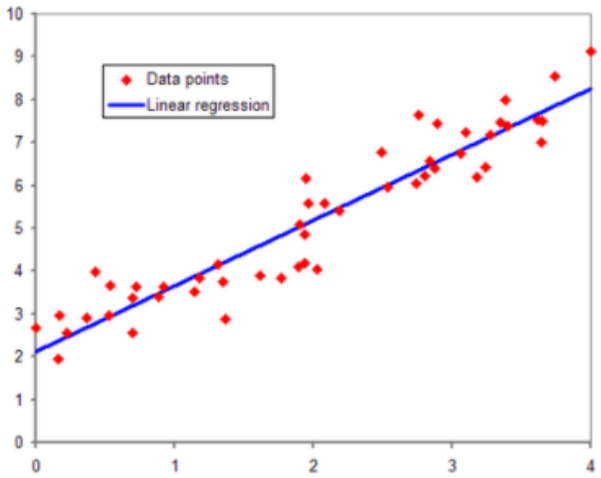
임재경(2015) “위탁급식 전문업체의 산업체 급식소 식수 예측 향상을 위한 식수 오차율 영향 요인 분석”[5]에서는 급식소의 메뉴 제공 형태가 단일 메뉴로 제공하는 경우가 복수 메뉴로 제공할 때보다 식수 오차율이 낮은 것으로 소개했다. 또한, 식수 예측시 고려 요인으로는 사무실형 급식소는 요일, 제공메뉴, 선호도, 날씨 순으로 중요성을 소개하였다.

이와 같이 다양한 연구에서 정확한 취식 인원 예측의 중요성을 강조했으나 기계학습을 통해 취식 인원 예측에 대한 연구가 적었으며, 기존 연구는 단일 코너, 단일 메뉴만을 가지고 예측했다는 한계점이 있다.

2. 이론적 배경

2.1 Linear Regression

Linear Regression은 선형 회귀분석으로 종속 변수 y와 한 개 이상의 독립 변수 x와의 선형 상관관계를 모델링하는 회귀분석 기법이다. 선형 회귀 분석에서는 실제 데이터와 함수와의 차이가 존재하는데 이를 손실이라고 하며 이러한 손실을 구하는 것을 평균 제곱 오차(mean squared error, MSE)라고 한다. 또한 손실을 최소화하기 위하여 경사하강법(Gradient Descent)를 이용하여 최적의 파라미터를 찾는다. 이를 바탕으로 수렴과 학습을 통하여 전체 데이터로부터 나오는 오차의 평균을 최소화할 수 있는 최적의 기울기와 절편을 찾는 것이다. 선형 회귀분석은 추세 분석 즉 데이터를 시간 축으로 놓았을 때 장기적으로 데이터의 값이 어떻게 변화하는지를 파악하며 역학 조사, 재무 관리 등에 많이 쓰인다. [6][7]



[그림 1] 선형 회귀 그래프

2.2 Ridge

Ridge는 다항 회귀분석모델로 [그림 2] Ridge 회귀식을 보면 잔차제곱합(Residual sum of squares, RSS)과 패널티항(베타 값)의 합으로 이루어져 있다. RSS를 최소화함으로 데이터에 잘 적합하는 동시에, 패널티항에 의해 계수들을 0에 가깝게 하는 Shrink 효과를 적용한다.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

[그림 2] Ridge 회귀식

Ridge는 변수들 간의 공선성이 있을 경우 변수가 많아도 변수간 상관관계로 인하여 사용가능한 정보가 적을 경우 이 독립변수들 간의 variance를 크게 줄여 least square보다 좋은 결과를 보여준다.

특징으로 변수 선택이 불가능, Closed form solution 존재, 변수 간 상관관계가 높을 경우 좋은 예측, 크기가 큰 변수를 우선적으로 줄이는 경향을 갖고 있다.[8][9]

2.3 Lasso

Lasso는 다항 회귀분석모델로 Linear Regression에 MSE가 최소가 되게 하는 가중치와 bias 향을 찾는 동시에 가중치의 절대값들이 최소가 되게 하는 추가적인 제약 조건을 부가한다. Lasso는 Ridge와 같이 약간의 bias를 희생하여 기존 least square보다 variance 측면에서 좋은 예측 값을 보이는 모델이다.

$$
\begin{aligned}
& MSE + penalty \\
&= MSE + \alpha \cdot L_1\text{-}norm \\
&= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\sum_{j=1}^{m}|w_j|
\end{aligned}
$$

$$
\underset{w,\ b}{argmin}\left\{ \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \alpha\sum_{j=1}^{m}|w_j| \right\}
$$

[그림 3] Lasso 회귀식

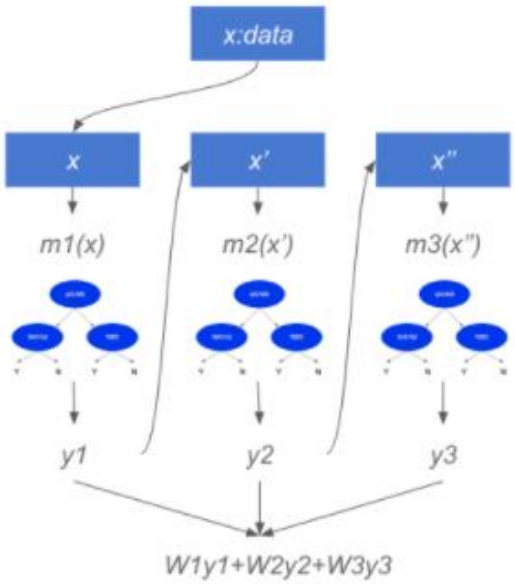
Lasso의 특징은 변수 선택이 가능하며, Closed from solution이 존재하지 않는다. 또한, 파라미터의 크기와 관계없이 정규화를 적용하여 작은 값의 파라미터를 0이 되게 함으로써 그에 해당하는 특성들을 제외해준다. 결과적으로 모델에서 가장 중요한 특성이 무엇인지 알게 되어 모델 해석력이 좋아진다.[10]

2.4 XGBoost

XGBoost는 Gradient Boosting 알고리즘을 분산환경에서도 실행할 수 있도록 구현해 놓은 라이브러리로, Regression, Classification 문제에 활용 가능하며, 성능과 자원 효율이 좋은 알고리즘이다.

여러 개의 Decision Tree를 조합해서 사용하는 Ensemble 알고리즘 종류 중 하나로 Decision Tree는 여러 개의 이진 노드를 겹쳐서 피처 별로 판단 후 최종 값을 뽑아내는 형태이다.

[그림 4]와 같이 Boosting은 m1~3의 모델이 있을 때, m1에는 x에서 샘플링된 데이터를 넣고 출력된 결과 중에서 예측이 잘못된 x중의 값들에 가중치를 반영하여 다음 모델인 m2에 넣는다. 마찬가지로 y2 결과에서 예측이 잘못된 X'값들에 가중치를 반영해서 m3에 넣고, 각 모델의 성능차이를 극복하기 위해 가중치 W를 반영한다.



[그림 4] Boosting 개념도

Boosting 기법을 이용하여 구현한 알고리즘은 Gradient Boost가 대표적이며 병렬 학습이 지원되도록 구현한 라이브러리가 XGBoost이다.[11]

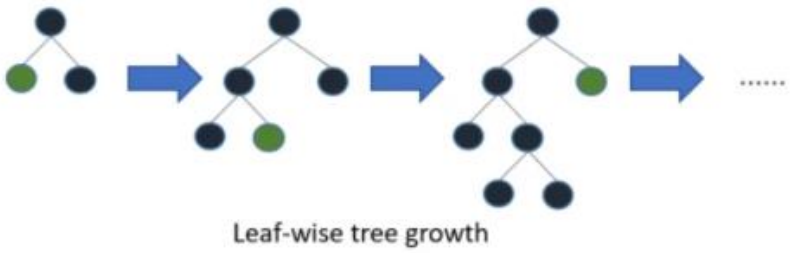
2.5 LightGBM

Light GBM역시 Gradient boost의 일종으로 Tree 기반의 학습 알고리즘이다.

단, 기존과는 다른 방식으로 Tree에 접근하는데 다른 알고리즘은 수평적 확장되는 반면, Light GBM은 Tree가 수직적으로 확장된다.

즉 다른 알고리즘은 level-wise개념인 반면, Light GBM은 leaf-wise으로 확장을 위한 max delta loss를 가진 leaf를 선택하게 되는 것이다.

따라서 동일한 leaf를 확장할 때, leaf-wise 알고리즘은 level-wise알고리즘보다 더 많은 손실을 줄일 수 있다.

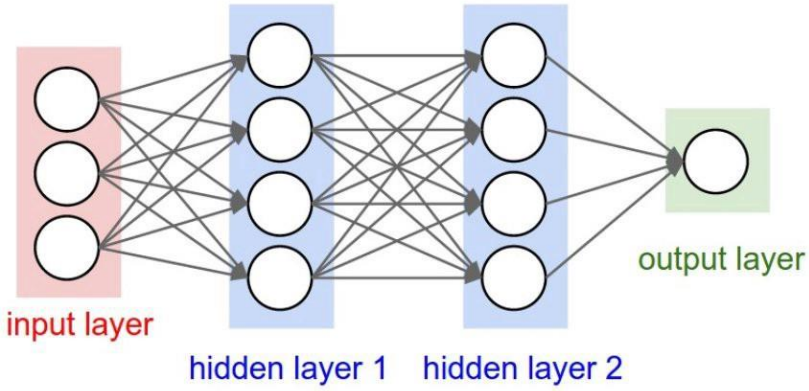


[그림 5] LightGBM 작동 방식

Light GBM은 다른 Tree구성으로 속도가 빠른 편이고, 큰 사이즈의 데이터를 실행할 때도 적은 메모리를 사용한다. 또한 결과의 정확도 역시 높은 편이다.[12][13]

2.6 Keras (Multilayer Perceptron)

Multilayer Perceptron 은 입력층과 출력층 사이에 하나 이상의 중간층이 존재하는 신경망으로 [그림 6] 같은 구조를 가진다.



[그림 6] Multilayer Perceptron 계층 구조

입력, 출력층 사이의 중간층을 은닉층(Hidden layer)으로 네트워크는 입력→은닉→출력층 방향으로 연결되는 전(全)방향 네트워크이다.

Multilayer Perceptron은 Single layer Perceptron과 유사한 구조를 가지고 있지만, 중간층과 각 unit의 입출력 특성을 비선형으로 함으로써 네트워크의 능력을 향상시켜 Single layer Perceptron의 여러가지 단점을 보완했다.

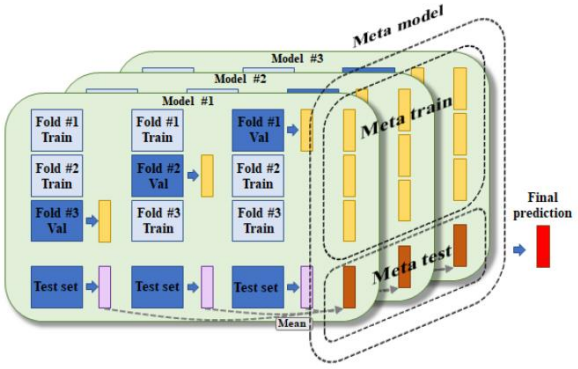
일반적인 Multilayer Perceptron의 학습 방법은 입력층의 각 unit에 입력 데이터를 전송하면 해당 신호는 각 unit에서 변환되어 중간층에 전달되고 최종적으로 출력층으로 나오게 된다. 이때 출력값과 기대 출력값을 비교하여 그 차이를 감소시키는 방향으로 연결 강도를 조정하는 것이다. 그러나 중간층이 많을수록 연결 강도가 어떤 오차를 유발하는지 알 수 없어 학습이 어려워진다는 단점이 있다.[14][15]

2.7 Stacking Ensemble

Ensemble은 머신러닝에 보편적으로 사용되는 알고리즘으로 Voting, bagging 위에서 살펴본 Boosting 그리고, Stacking으로 구성되어 있다.

[그림7]의 경우 CV Fold는 3으로 고정하고, Test set은 별도로 존재하는 것으로 가정 시, 해당 모델에 대해 Train Fold를 이용하여 최초 모델을 훈련하고, Validation Fold를 이용하여 모델 예측 값을 계산한다.

같이 각 Validation Fold마다 예측 값을 계산하여 이를 순차 시키고, Test Set을 이용한 예측 값 3개에 대한 평균을 산출하여 Meta 모델에 사용되는 Test Set으로 만든다. 즉 정렬된 3개의 Fold 결과 값은 결국 한 모델에서 계산된 모든 샘플에 대한 예측 값들이 된다.



[그림 7] Stacking 작동 구조

해당 과정을 다수의 모델에 대해서 반복하고 그 출력 값들을 순차하게 되면 Meta Train Data와 Meta Test Data를 얻게 되어 Meta 모델이라 불리는 2차 모델을 적용하여 훈련 및 최종 예측 값을 출력하는 것이다.

Stacking Ensemble의 특징은 모델의 예측 값을 토대로 학습하는 방식이며 다른 종류의 단일 모델을 Ensemble하는 방식으로 서로 다른 종류의 단일 모델들을 혼합한다. 이는 해당 기법의 본래 목적인 High Variance Problem을 해결하는데 크게 기여된다.[16]

2.8 평가 방법

실험결과의 성능을 평가하기 위하여 회귀 오류 지표인 RMSE(Root Mean Square Error), MAE(Mean Absolute Error)와 R^2(R-squared)를 사용하여 평가하였다. [17][18]

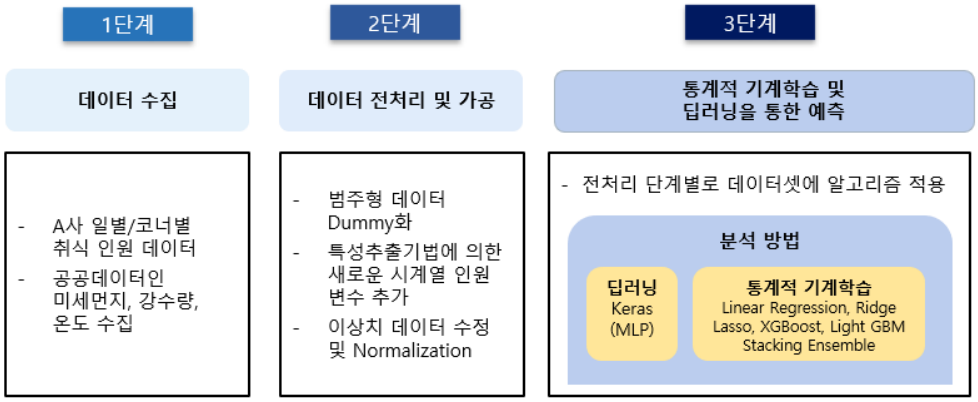
평가지표	수식	특징
RMSE (Root Mean Square Error)	$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$	- 모델의 예측값과 실제 값의 차이를 제곱해 평균한 것에 log 적용 - 예측 대상의 단위 크기에 주의해야함 - 특이치에 민감하다
MAE (Mean Absolute Error)	$MAE = \frac{\sum \lvert y - \hat{y} \rvert}{n}$	- 모델의 예측값과 실제값의 차이를 모두 더한다는 개념 - 절대값을 취하기 때문에 모델이 언더피팅인지 오피피팅인지 알 수 없음
R^2 (R Squared)	$R^2 = \frac{SSR}{SST}$	- 모델의 예측값과 실제값의 상관계수를 제곱한 값 - 1에 가까울수록 정확도가 높다

[그림 8] 평가 방법

3. 제안 내용

3.1 제안 방법 및 분석 절차

머신러닝을 활용한 단체급식 다중코너 식수예측은 [그림 9]와 같이 3단계에 걸쳐 예측율이 높은 모델을 찾고자 한다.



[그림 9] 제안 방법론 분석 절차

1단계는 데이터 수집으로 A급식 사업장의 취식인원 데이터와 공공데이터를 수집한다. 이를 바탕으로 판매일의 특징을 확인하여 파생변수로 계절, 요일, 휴일 등을 추가하여 기본 데이터를 완성시킨다.

2단계는 데이터 전처리 및 가공 단계로 총 4단계의 데이터셋을 만들었다. 기본 데이터에서 범주형 변수를 One-hot Encoding을 통해 Dummy 변수로 변경 및 추가했으며 취식인원데이터를 기반으로 새로운 시계열변수를 추출하여 파생변수로 포함하였다. 또한 Outliner와 같은 이상치 데이터를 수정 및 삭제했으며 데이터 log화 함에 따라 정규화를 진행하였다. 예측을 위해 사용한 변수들에 대한 자세한 설명은 4장인 사례 분석에서 확인 가능하다.

3단계는 2단계 전처리 및 가공과정을 거친 단계별 데이터셋을 기반으로 통계적 기계학습인 Linear Regression, Ridge, Lasso, XGBoost, LightGBM, Stacking Ensemble과 딥러닝 모델인 Keras (Multilayer Perceptron)을 적용하여 예측을 진행하고자 한다.

4. 사례 분석

4.1 데이터 수집 및 가공

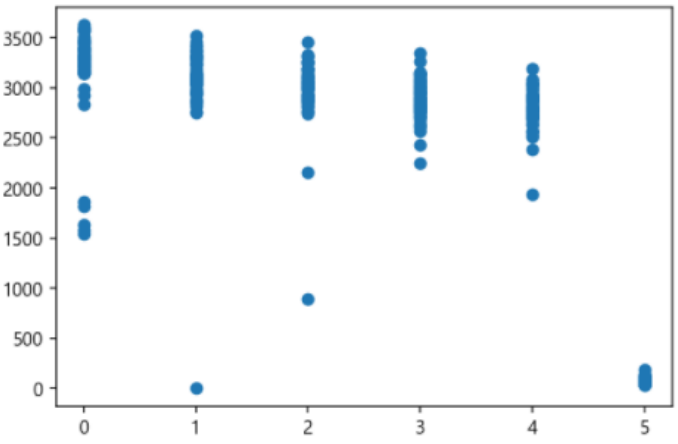
4.1.1 기본 데이터 수집 및 가공

본 논문에서 사용한 데이터로 A급식 사업장에서 제공하는 10개 코너명, 메뉴 제공날짜, 취식 인원수, 메뉴의 가격을 수집했으며, 기상청에서 제공하는 평균 기온, 강수량과 서울특별시 대기환경정보에서 미세먼지 평균농도를 수집하였다. 각각의 데이터는 2017년 1월 1일부터 2018년 6월 30일 기간의 데이터를 수집하였다.

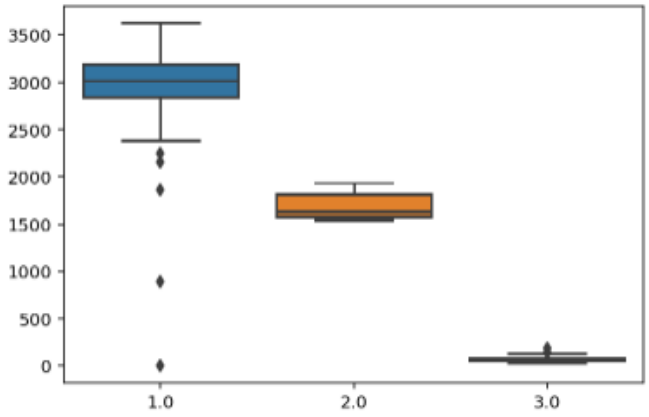
[표 1] 기본 수집 데이터

데이터 종류	추출날짜	출처	설명
A 급식 사업장 취식정보	2017 년 1 월 1 일 ~ 2018 년 6 월 30 일	A 급식 사업장	코너명, 메뉴제공날짜, 취식인원수, 메뉴가격
날씨정보	2017 년 1 월 1 일 ~ 2018 년 6 월 30 일	기상청	평균 기온, 강수량
미세먼지	2017 년 1 월 1 일 ~ 2018 년 6 월 30 일	서울특별시 대기환경정보	미세먼지 평균 농도

해당 수집 데이터를 기반으로 메뉴 제공날짜의 연도, 월, 요일, 계절, 휴일을 변수로 추가하였다. 요일(Dayofweek)의 경우 월(0), 화(1), 수(2), 목(3), 금(4), 토(5), 일(6)로 지정하였다. 계절(Season)은 봄(1), 여름(2), 가을(3), 겨울(4)로 표현하였다. 또한, 휴일(Holiday)는 일하는 날(1), 샌드위치 데이(2), 휴일(3)으로 지정했다. [그림10]은 요일(Dayofweek)와 취식 인원에 대한 시각화 그래프로 월요일부터 토요일까지 점차 감소하는 추세를 확인 가능하며 [그림11]는 휴일에 따른 취식 인원변화를 시각화한 그래프로 휴일(3)과 샌드위치 데이(2)에는 일하는 날(1)보다 취식 인원이 감소하는 경향을 보인다.



[그림 10] 요일과 취식인원



[그림 11] 휴일과 취식인원

공공데이터로부터 획득한 날씨정보와 미세먼지 정보는 아래와 같이 범주형으로 가공하였다.

날씨의 경우 기온(Temp)은 일평균기온으로 -15~-10(1), -10~-5(2), -5~0(3), -0~5(4), -5~10(5), 10~15(6), 16~20(7), 21~25(8), 25~30(9), 31이상(10)으로 구분하고, 강수량(Rainy)은 11시에서 13시까지의 강수량을 추출하여 우산 필요 유무에 따라 야외활동이 가능한지 판단되는 2mm를 기준으로 2mm미만(1), 2mm이상(2)로 구분하였다. 또한 미세먼지(PM10)는 농도별로 0~30(1), 31~80(2), 81~150(3), 151이상(4)로 나누어 데이터셋을 완성시켰다.

[표 2] 데이터셋 변수

순번	변수명	변수유형	변수설명
1	Corner	명목형	코너명
2	Price	연속형	메뉴 금액
3	Year	명목형	취식날짜 연도
4	Month	명목형	취식날짜 월
5	Day	명목형	취식날짜 일
6	Dayofweek	명목형	취식날짜 요일 월(0), 화(1), 수(2), 목(3), 금(4), 토(6), 일(7)
7	Holiday	명목형	휴일 일하는날(1), 샌드위치데일(2), 휴일(3)
8	Season	명목형	계절 spring(1), summer(2), fall(3), winter(4)
9	PM10	명목형	미세먼지 0~30(1), 31~80(2), 81~150(3), 151이상(4)
10	Temp	명목형	온도 -15~-10(1), -10~-5(2), -5~0(3), -0~5(4), -5~10(5), 10~15(6), 16~20(7), 21~25(8), 25~30(9), 31이상(10)
11	Rainy	명목형	강수량(11시~13시) 2mm 미만 (1), 2mm 이상 (2)

4.1.2 데이터 범주형 속성 추가

연속형이 아닌 범주형 속성변수인 "Rainy", "PM10", "Season", "Holiday", "Year", "Day", "Month", "Temp", "Dayofweek", "Corner"을 One-hot Encoding 기법을 적용하여 [표3]와 같이 총 84개의 Dummy 변수로 추가하였다.

[표3] 생성된 Dummy 변수들

순번	변수명	변수유형
1	Rainy_1	범주형
2	Rainy_2	범주형
3	PM10_1	범주형
4	PM10_2	범주형
5	PM10_3	범주형
6	PM10_4	범주형
7	Season_1.0	범주형
8	Season_2.0	범주형
9	Season_3.0	범주형
10	Season_4.0	범주형
11	Holiday_1.0	범주형
12	Holiday_2.0	범주형
:		
75	Corner_1	범주형
76	Corner_2	범주형
77	Corner_3	범주형
78	Corner_4	범주형
79	Corner_5	범주형
80	Corner_6	범주형
81	Corner_7	범주형
82	Corner_8	범주형
83	Corner_9	범주형
84	Corner_10	범주형

4.1.3 데이터 파생 변수 추가

취식 인원 예측하는데 있어 실제 사업장에서는 과거 식수데이터를 참고하는 경우가 많은 것을 파악되었다[4]. 그리하여 A급식 사업장 취식데이터에서 [그림 10]과 같이 요일별로 식수 차이가 있는 것을 확인하여 코너별, 요일별 전주까지의 평균 식수 데이터를 변수로 생성하였다. 또한, 요일을 제외한 코너별 전주까지의 평균 식수 데이터를 추가하였다. 즉 식수예측 모델링의 정확도를 높이고자 [표4]와 같이 시계열 인원 변수를 새로 생성해서 예측 모델링에 적용하였다.

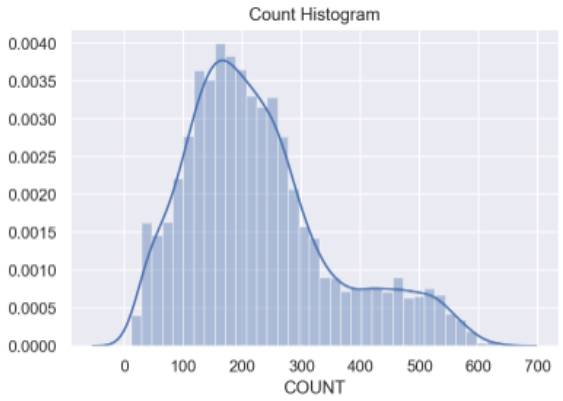
[표4] 데이터 파생 변수

순번	변수명	변수유형	변수설명
1	corner_theday_ave	연속형	코너별 요일별 전주까지의 평균 취식 인원
2	corner_week_before_ave	연속형	코너별 전주까지의 평균 취식 인원

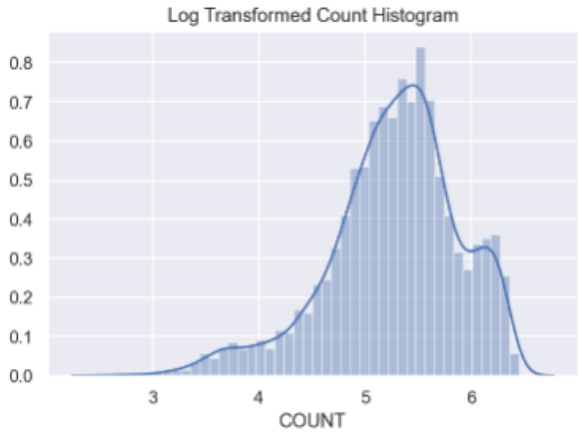
4.1.4 데이터 클리닝 및 정규화

데이터 클리닝을 위해 탐색적 데이터분석을 이용하여 휴일, 요일등 변수에 따른 취식 인원을 확인하였으며 이를 바탕으로 이상치인 Outliner를 제거하는 작업을 수행하여 4,498개의 데이터 중 10개로 제거하여 총 4,488개가 남았다.

또한 데이터의 log화를 진행하여 데이터간 편차를 줄임으로 정규성을 높여 예측의 정확한 값을 얻고자 하였다..



[그림 12] Log 진행 전



[그림 13] Log 진행 후

4.2 실험방법

실험은 총 2차로 나누어 진행하였다.

1차 실험은 성능 비교 분석으로 아래 6가지 통계적 기계학습과 딥러닝 모델로

1. Linear Regression
2. Ridge
3. Lasso
4. XGBoost
5. LightGBM
6. Keras (Multilayer Perceptron)

4 .1에서 설명한 데이터 전처리를 단계적으로 추가하여 총 4단계의 데이터셋으로 분석하였다.

1단계 범주형 변수 변경

2단계 데이터 파생 변수 추가

3단계 데이터 Outliner제거 및 정규화

4단계 K-Fold 진행

2차 실험은 Stacking Ensemble 모델을 이용하여 [표 5]와 같이 진행하였다.

[표 5] 2차 실험 과정

1단계	1차 실험에서 성능이 우수한 기록을 가진 데이터셋 + 결정계수가 높았던 상위 4개 모델
2단계	2차 실험의 1단계의 예측값 + 1차 실험에서 전반적으로 성능이 우수했던 1개 모델

모든 실험은 데이터를 8:2의 비율로 Training과 Test를 나누어 수행하였다.

4.3 실험조건

데이터셋은 [표 6]과 같이 속성값들로 분석에 활용하여 실험을 진행하였다.

[표 6] 전체 데이터셋 속성정보

기본 속성 (1개)	연속형 변수를 활용하였다..
파생 속성 (2개)	기존 속성에서 시계열 정보들 추출하였다.
Dummy변수 (84개)	범주형 변수를 Dummy 변수로 조정하였다.

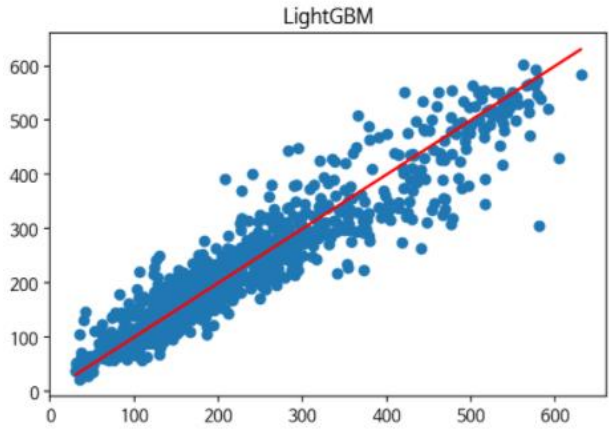
4.4 실험결과

4.4.1 1차 실험 성능비교

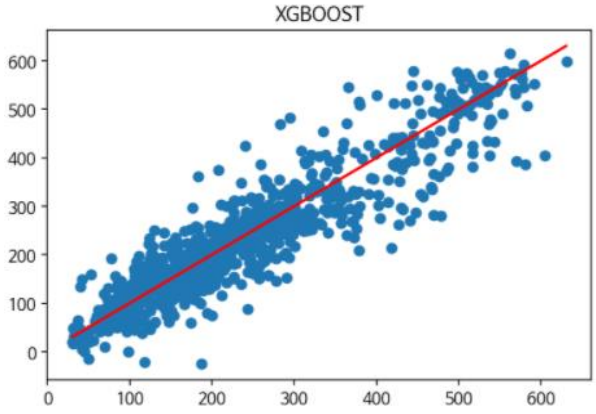
기본 데이터가 범주형으로 주로 이루어져 있어 1단계는 기본 데이터셋의 범주형 변수를 One-hot encoding을 적용하여 Dummy로 변수를 변경 및 추가하여 6개 모델의 분석을 수행하였다. 모델별로 [표 7] 과 같은 결과값을 얻었으며, R^2으로 확인해 본 결과 LightGBM이 87.7로 가장 성능이 높게 나왔으며, XGBoost가 83.1로 가장 낮게 나왔다

[표 7] 1단계 데이터 모델 성능 결과

분석모델	RMSE	MAE	R^2
Linear Regression	51.25	40.13	84.6
Ridge	50.67	39.43	85.0
Lasso	49.84	38.76	85.5
XGBoost	53.70	38.78	83.1
LightGBM	45.87	33.29	87.7
Keras (Multilayer Perceptron)	47.73	35.30	86.7



[그림 14] 1단계 실험시 성능 가장 높은 LightGBM 그래프

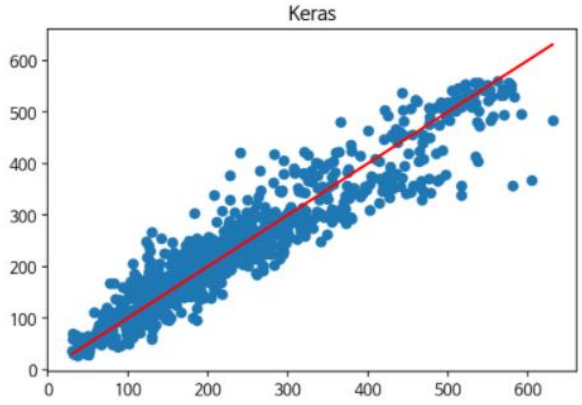


[그림 15] 1단계 실험시 성능 가장 낮은 XGBoost 그래프

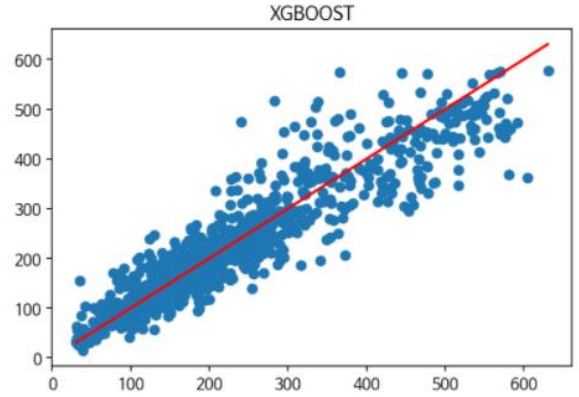
2단계에서는 기본 데이터의 범주형을 Dummy 변수화한 1단계 데이터셋에 시계열인원 파생 변수를 추가하여 6개 모델의 분석을 수행하였다. 모델별로 [표 8]과 같은 결과값을 얻었으며 R^2으로 확인해 본 결과 1단계에 비해 전반적으로 성능이 향상된 것을 확인할 수 있었다. Keras (Multilayer Perceptron)가 88.5로 가장 성능이 높게 나왔으며 XGBoost가 84.4로 가장 낮게 나왔다

[표 8] 2단계 데이터 모델 성능 결과

분석모델	RMSE	MAE	R^2
Linear Regression	47.62	36.91	86.7
Ridge	47.26	36.57	86.9
Lasso	47.46	36.76	86.8
XGBoost	51.53	37.29	84.4
LightGBM	47.51	34.17	86.8
Keras (Multilayer Perceptron)	44.36	32.61	88.5



[그림 16] 2단계 실험시 성능 가장 높은 Keras (Multilayer Perceptron)그래프

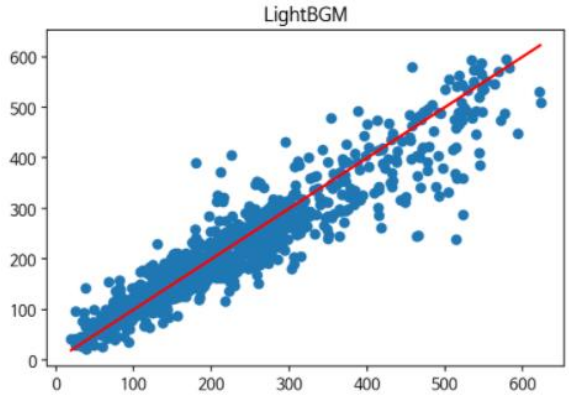


[그림 17] 2단계 실험시 성능 가장 높은 XGBoost그래프

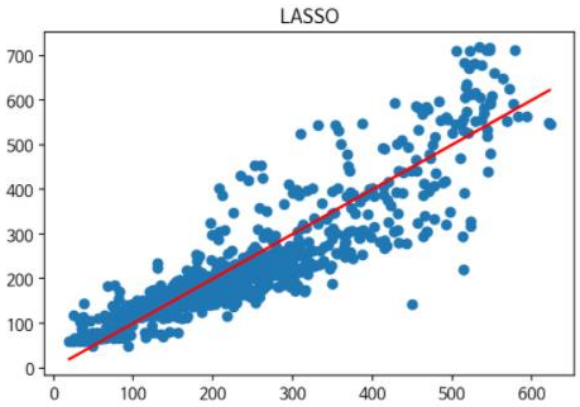
3단계는 2단계 데이터셋에서 Outliner 제거 및 log화로 정규화를 진행한 데이터셋을 기반으로 6개 모델의 분석을 수행하였다. 모델별로 [표 9]과 같은 결과값을 얻었으며 R^2으로 확인해 본 결과 2단계에 비해 XGBoost만 제외하고 전체적으로 성능이 감소하는 것을 확인하였다. LightGBM이 86.42로 가장 성능이 높게 나왔으며 Lasso가 76.59로 가장 낮게 나왔다

[표 9] 3단계 데이터 모델 성능 결과

분석모델	RMSE	MAE	R^2
Linear Regression	51.62	38.04	83.76
Ridge	52.53	38.67	83.19
Lasso	61.98	44.52	76.59
XGBoost	49.42	34.48	85.12
LightGBM	47.20	32.93	86.42
Keras (Multilayer Perceptron)	48.48	35.34	85.68



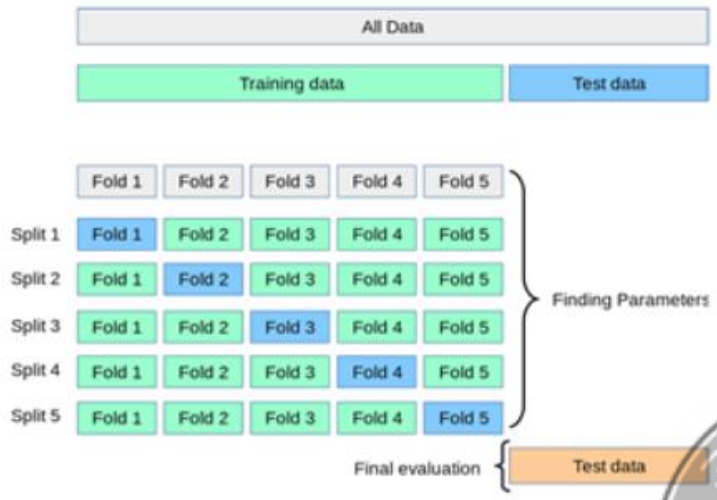
[그림 18] 3단계 실험시 성능 가장 높은 LightGBM 그래프



[그림 19] 3단계 실험시 성능 가장 낮은 Lasso 그래프

4단계에서는 3단계까지 진행된 데이터셋을 기반으로 [그림 20]와 같이 Training data를 5번 K-fold 진행한 파라미터 값으로 6개 모델의 분석을 수행하였다.

모델별로 [표 10]과 같은 결과값을 얻었으며 R^2으로 확인해 본 결과 Ridge를 제외하고 전반적으로 성능이 감소하는 것을 확인하였다. Keras (Multilayer Perceptron)가 85.50으로 가장 성능이 높게 나왔으며 Lasso가 74.08로 가장 낮게 나왔다



[그림 20] K-Fold[19]

[표 10] 4단계 데이터 모델 성능 결과

분석모델	RMSE	MAE	R^2
Linear Regression	56.28	40.95	81.03
Ridge	51.02	37.76	84.73
Lasso	65.69	46.65	74.08
XGBoost	56.32	40.33	81.33
LightGBM	55.18	39.11	82.03
Keras (Multilayer Perceptron)	48.81	35.09	85.50

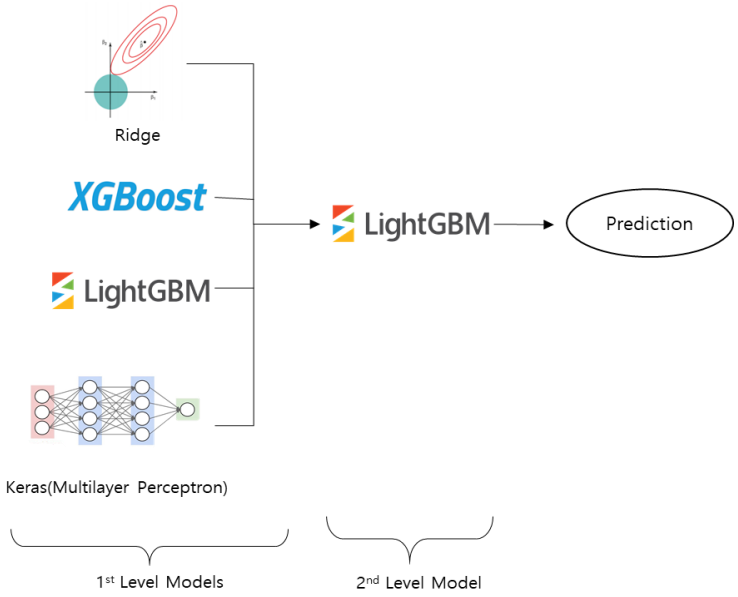
4.4.2 2차 실험결과

1차 실험의 모델 성능과 데이터셋 성능을 바탕으로 2차 실험에서는 [그림 21]과 같은 프로세스로 Stacking Ensemble 모델을 적용하여 분석하였다.

1^{st} Level 모델들과 2^{nd} Level 모델을 지정하여 진행하였다. 1^{st} Level 모델들은 1차 실험 결과값에서 R^2이 높았던 상위 4개 모델인 Ridge, XGBoost, LightGBM, Keras (Multilayer Perceptron)로 동일 실험 결과값에서 성능이 우수한 기록을 가진 2단계 데이터셋을 기반으로 예측을 진행하였다. 2^{nd} Level 모델은 1차 실험에서 전반적으로 우수한 성능을 보인 LightGBM을 적용하여 1^{st} Level 모델에서 나온 결과값으로 예측을 진행하였다. 그 결과 [표 11]과 같은 결과값을 얻었으며, R^2은 89.1로 단일 알고리즘을 적용한 결과값보다 높으며, RMSE, MAE도 낮아 우수한 성능을 보여주었다.

[표 11] 2차 실험결과

분석모델	RMSE	MAE	R^2
Stacking Ensemble	43.50	30.23	89.1



[그림 21] 2차 실험 모델

4.4.3 최종 실험결과 정리

1, 2차 실험의 모델 별 성능을 정리해본 결과 [표 12]과 같이 R^2기준으로 Stacking Ensemble이 가장 우수한 성능을 보였으며 Keras, LightGBM, Ridge, Lasso, Linear Regression, XGBoost순으로 성능이 감소하는 것을 확인할 수 있었다.

실험조건으로 데이터셋을 4단계로 구분하여 진행한 결과 [표 12]과 같이 2단계 데이터셋이 가장 우수한 결과값을 도출하였고, 1단계, 3단계, 4단계순으로 모델의 성능이 하락되었다. 시계열 인원 파생 변수를 추가한 2단계 데이터셋의 회귀오차지표인 RMSE, MAE 역시 전반적으로 낮은 것으로 보아 취식 인원 예측 시 시계열 데이터가 정확한 값을 예측하는데 중요한 요인인 것을 확인할 수 있다. 3단계 데이터셋의 경우 Outliner제외 및 log화로 데이터 정규화를 진행하였는데 XGBoost를 제외한 나머지 모델에서는 성능이 감소하는 것으로 보아 데이터 정규화는 큰 의미가 없는 것으로 보인다. 최종 3단계 데이터셋에서 K-Fold를 진행한 4단계 데이터셋의 성능은 전반으로 하락한 것으로 보아 주요변수인

Corner가 Fold마다 불균등하게 나눠져 정확율을 낮추는 요인으로 작용했다고 추정할 수 있다.

따라서 식수 예측 모델을 구축하기 위해서는 데이터셋에 시계열 인원 파생 변수가 필수적으로 포함되어야 하며 분석 모델로는 단일 알고리즘의 경우 통계적 기계학습보다 딥러닝 모델 Keras가 우수했고, 그보다 다중 알고리즘을 이용한 Stacking Ensemble의 성능이 다소 우수하다는 것을 확인할 수 있었다.

[표 12] 1, 2차 실험 성능 R^2 결과

분석모델	1단계 데이터셋	2단계 데이터셋	3단계 데이터셋	4단계 데이터셋
Linear Regression	84.61	86.72	83.76	81.03
Ridge	84.96	86.92	83.19	84.73
Lasso	85.45	86.80	76.59	74.08
XGBoost	83.11	84.44	85.12	81.33
LightGBM	87.67	86.78	86.42	82.03
Keras (Multilayer Perceptron)	86.65	88.47	85.68	85.50
Stacking Ensemble		89.10		

5. 결론 및 향후연구

본 연구는 단체급식의 정확한 취식 인원 예측 중요성을 감안, 실제 다중코너를 갖고 있는 A급식 사업장의 데이터를 기반으로 다중코너의 식수 예측 방법론을 제안하였다.

이 과정에서 과거 식수뿐만 아니라 코너명, 취식날짜의 날씨, 요일과 같은 데이터를 포함하여 총 87개의 변수를 추출하여 4단계의 데이터셋을 구축하였다. 해당 데이터셋에 통계적 기계학습 모델인 Linear Regression, Ridge, Lasso, XGBoost, Stacking Ensemble과 딥러닝 모델인 Keras (Multilayer Perceptron)를 적용했으며 평가지표인 RMSE, MAE, R^2로 성능을 검증하였다.

그 결과 복수 메뉴를 제공 사업장에서 경험치에 의한 취식 예측 오차율인 18~20%보다 정확한 예측율을 보인 R^2 결과값 89.1을 갖은 다중 알고리즘을 사용한 Stacking Ensemble기법을 찾을 수 있었다. 이를 실제 사업장에 적용시 오차율이 확연히 줄어들어 고객만족도 상승 및 식재료비 절감, 음식물 쓰레기 감소 등의 효과를 보일 것으로 예상된다.

해당 논문은 2가지 측면에서 타논문과 차별성을 두고 있다. 첫째, 단일코너가 아닌 다중코너의 취식 인원 예측에 대한 모델링을 진행함에 따라 다양한 메뉴를 제공하는 현재 급식 트렌드에 맞는 실험을 진행하였다. 둘째, A 급식 사업장을 대상으로 실데이터를 이용하였고 다양한 알고리즘 모델을 사용하여 기존보다 낮은 오차율을 가진 성능이 우수한 알고리즘을 도출할 수 있었다.

향후에는 다년간 추적된 데이터 획득하고 식수 변화에 주요 원인이 될 수 있는 메뉴명, 사업장 이벤트 날짜와 같은 변수를 수집하여 현재보다 예측 성능을 향상시킬 예정이다.

참고 문헌

[1] aT 농식품유통교육원(2017). 농식품&유통 심층연구.

[2] Miller JJ, McCahon CS, & Miller JL, (1991), “Forecasting production demand in school food service. School Food Serv Res Rev” 15(2):117-121, 1991

[3] 정라나, 양일선, & 백승희. (2003, December 1). 大學 給食所의 食數豫測 모델 開發. 대한지역사회영양학회지, 8(6), 910-918.

[4] 전종식, 박은주, & 권오병. (2019). 기계학습방법을 활용한 대형 집단급식소의 식수 예측: S 시청 구내직원식당의 실데이터를 기반으로. 대한영양사협회학술지, 25(1), 44-58

[5] 임재영. (2016). "위탁급식 전문업체의 산업체 급식소 식수 예측 향상을 위한 식수 오차율 영향 요인 분석". 국내석사학위논문 연세대학교 생활환경대학원

[6] Linear Regression, 위키백과, https://ko.wikipedia.org/wiki/선형_회귀

[7] Linear Regression, http://hleecaster.com/ml-linear-regression-concept

[8] Ridge, https://soobarkbar.tistory.com/30

[9] Ridge, https://godongyoung.github.io/머신러닝/2018/02/07/ISL-Linear-Model-Selection-and-Regularization_ch6.html

[10] Lasso, https://bskyvision.com/193

[11] XGBoost, https://bcho.tistory.com/1354

[12] LightGBM, https://nurilee.com/lightgbm-definition-parameter-tuning/

[13] LightGBM, https://velog.io/@dbj2000/ML

[14] keras, http://www.aistudy.co.kr/neural/multilayer_perceptron.htm

[15] Yu, M. S. (2020). Application of Machine Learning in Rhinology: A State-of-the-Art Review. Korean Journal of Otorhinolaryngology-Head and Neck Surgery, 63(8), 341-349.

[16] Stacking Ensemble,
https://blog.naver.com/PostView.nhn?blogId=ckdgus1433&logNo=221588139765
[17] 평가 방법, https://bkshin.tistory.com/entry/머신러닝-17-회귀-평가-지표
[18] 회귀평가지표, https://partrita.github.io/posts/regression-error/
[19] K-Fold, https://scikit-learn.org/stable/modules/cross_validation.html