머신러닝 기반 고객 재구매 상품 예측

남기백*, 박상원**
*강릉원주대학교 정보통신공학과
**SK 주식회사
e-mail : qweadg@naver.com

Prediction of Products Purchase Again Using Machine Learning.

Gibaek Nam*, Sangwon Park**
*Dept. of Information and Communication Engineering, Gangneung-Wonju National University
**SK HOLDINGS CO., LTD

요 약

본 연구의 목적은 머신러닝 기법을 활용하여 e-commerce 시장에서 고객의 구매패턴을 파악하여 고객이 필요로 하는 상품 추천 모델을 만들고 이를 검증한다. 일반적인 e-commerce 시장은 무분별한 정보의 제공으로 고객은 자신이 원하는 상품을 찾아 헤매야 하고 이는 기업들의 고객유지를 저해하여 기업 손실로 이어진다. 따라서 본 논문에서는 결정트리(Decision Tree)에 boosting 기법을 활용하여 고객의 주문내역과 상품정보 등을 분석하여 특징을 추출한 후 사용자에게 상품을 추천하는 모델을 만들어 검증한다. 그 결과 f1 score 가 0.3792 를 나타내었고 이는 고객이 다음에 구매하려는 목록의 30% 이상을 예측하는 결과이며 이는 기업이 고객에게 필요한 상품정보를 제공해주는 서비스임을 확인할 수 있었다.

1. 서론

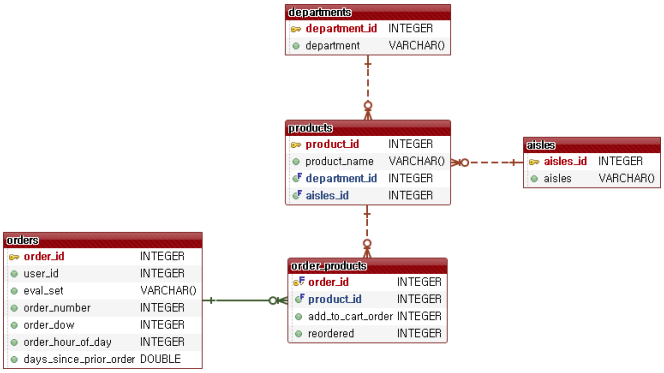
정보기술의 발전으로 우리의 생활은 빠르게 변화하고 있으며 이제 없어서는 안 될 중요한 영역이 되었다. 과거에는 대부분 매장에서 직접 상품을 보고 구매하였으나 최근에는 스마트폰이나 TV를 비롯한 각종 스마트 기기의 발달로 인터넷 쇼핑이 생활화되어 언제 어디서나 원하는 물건을 구매할 수 있게 되었다. 그러나 수많은 종류의 상품들 중 고객들은 자신이 원하는 상품을 구매하는데 어려움이 존재한다. 판매자는 이를 해결하기 위해 고객이 원하는 상품의 선호도를 고려하여 적절한 상품을 추천해준다. 이를 구매로 연결시키는 것은 이윤창출과 직결된다. 특히 e-Commerce 의 분야에서 특정 고객에게 관심을 가질만한 제품이나 서비스를 추천해주는 '추천 시스템(recommender system)'은 실제적으로 가장 널리 이용되는 수단으로 이미 아마존(amazon) 등 해외의 유수 사이트는 물론 국내에서도 널리 적용되고 있다[1].

사용자에게 재구매 상품을 예측하는 방법으로 본 논문은 결정트리(Decision Tree) 알고리즘을 사용하였다. 결정트리 알고리즘은 기계학습 에서 사용하는 예측 모델링 방법 중 하나로 시각적이고 명시적인 방법으로 의사 결정 과정과 결정된 의사를 보여주는데 사용된다. 하지만 다른 기계학습 알고리즘에 비해 성능이 낮아 이를 해결하기 위한 방법으로 gradient boosting 기법을 사용한다. 결정트리는 다른 통계모형과는 다르게 가정이 적고, 범주형이든 연속형이든 제약 없이 쉽게 만들 수 있는 모델이기 때문에 사용했다[2]. 이를 사용하기 위해 XGBoost(Extreme Gradient Boosting)를 사용하였다. XGBoost 는 오픈 소스 소프트웨어 라이브러리로 gradient boosting 프레임워크를 제공한다[3].

본 논문의 구성은 다음과 같다. 2 장에서는 먼저 데이터의 구조 및 특징에 대해 분석 후 3 장에서 데이터전처리 과정을 거친다. 이후 4 장에서 기계학습을 위한 특징을 추출하고 5 장에서 추출한 데이터를 알고리즘에 적용한다. 마지막으로 실험결과에 대해 서술한다.

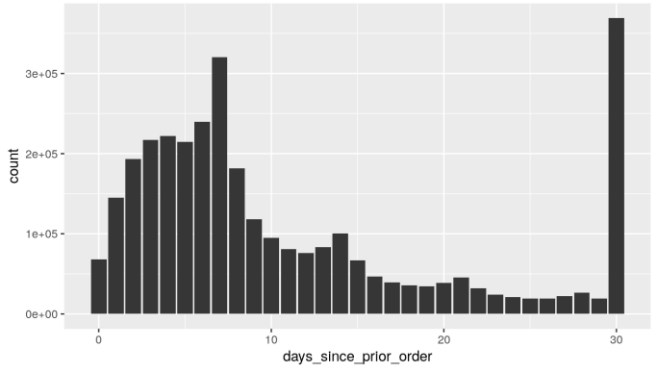
2. 데이터분석

분석에 사용된 데이터셋은 데이터분석 대회(Kaggle competition - Instacart Market Basket Analysis [4])에 참가하여 데이터를 제공받았다. 대회에서 제시한 문제는 미국의 식료품 배송 스타트업 회사 Instacart[5]의 고객 206,209 명 중 131,209 명의 마지막 구매내역 데이터를 학습데이터로 사용하여 모델을 만들고 75,000 명이 마지막에 주문한 상품의 예측이다. 구매한 이력으로 총 주문수는 3,421,083 건이며 총 주문된 상품의 개수는 32,434,489 개이다.

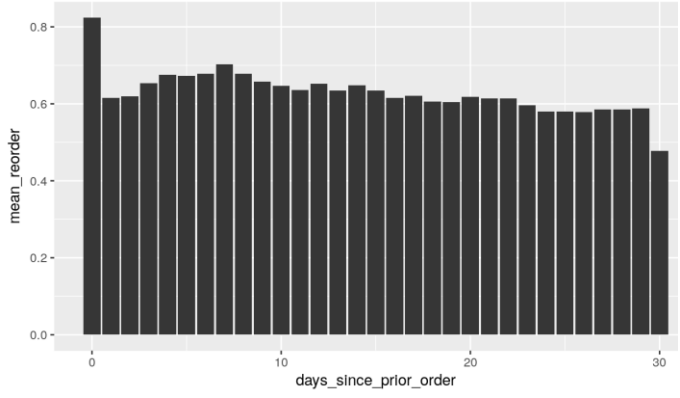


(그림 1) 데이터 구조

(그림 1)은 데이터의 테이블 구조를 나타낸다. Order_products 테이블은 주문번호와 주문한 상품이 포함되며 reordered 컬럼은 해당 상품의 재구매 여부를 나타낸다. Orders 테이블은 주문번호, 구매한 유저의 ID, 구매날짜 등의 데이터를 나타낸다. Products 테이블은 상품 ID, 상품명의 카테고리를 포함한다. departments, aisles 테이블은 상품의 카테고리 코드의 명칭을 나타낸다.



(그림 2) 재주문까지 소요된 날짜 분포



(그림 3) 재주문까지 소요된 날짜별 재주문율

(그림 2)의 재주문까지 소요된 날짜 분포에서 7 일마다 재주문하는 사람들이 가장 많은 것으로 나타난다. 마지막 30 일의 값은 30 일 이상 소요된 고객들을 전부 포함한 값이다. (그림 3)의 그래프에서 소요된 날짜가 0 일인 것도 확인할 수 있는데 이는 당일 재주문한 경우를 나타낸다. 0 일째는 재주문율이 0.8 을 넘는 높은 수치임을 확인할 수 있다.

3. 데이터 전처리

데이터가 재구매를 위한 뚜렷한 특징이 나타나지 않아 알고리즘의 성능 향상을 위해 데이터 전처리를 수행했다. 전처리는 가설을 세우고 이에 따른 데이터를 가공하는 방법을 사용했다. 사용한 가설은 가설 1:'한 번의 주문에 2 건이하의 주문을 한 경우 재구매율 예측과 관련이 없어 정확도에 영향을 미친다', 가설 2:'총 주문한 건수가 2 건 이하일 경우 재구매율 예측과 관련이 없어 정확도에 영향을 미친다'. 2 가지를 가설로 세우고 데이터 전처리를 진행했다.

4. 특징추출

예측에 사용될 특징들을 찾는 과정으로 필요한 데이터들만 합치고 기존 데이터셋에서 새로운 특징을 추출하는 작업을 진행했다.

<표 1> 추출한 특징 및 설명

Feature	Description
user_total_orders	고객이 총 주문한 횟수
user_total_items	고객이 총 주문한 상품 수
total_distinct_items	고객이 총 주문한 상품 종류의 개수
user_average_days_between_orders	고객이 주문까지 걸리는 평균기간
user_average_basket	고객의 장바구니 평균 상품개수
order_hour_of_day	고객이 상품을 구매한 시간
days_since_prior_order	고객의 재주문까지 기간
days_since_ratio	고객이 재주문까지 기간/고객의 주문까지 평균 기간
product_reorders	상품의 전체 재주문 수
product_reorder_rate	상품의 전체 재주문율
UP_orders	고객의 해당 상품 주문 횟수
UP_average_pos_in_cart	고객의 주문목록 평균 상품개수
UP_reorder_rate	고객의 해당 상품 주문 횟수 / 전체 주문 횟수
UP_orders_since_last	마지막 주문까지 걸린 기간
UP_hour_last	마지막 주문한 시간

각 컬럼들의 추출된 특징들의 ID 는 유저 ID*100000 + 상품 ID 으로 만들어 식별할 수 있게 하였고 ID 는 특징이 아니기 때문에 특징 추출이 끝난 후 소거하였다. 데이터들을 하나의 테이블로 합치는 과정에서 먼저 유저 관련된 특징, 주문 관련 특징, 상품 관련 특징을 정리한 후 각 특징을 추출하여 <표 1>과 같은 결과를 만들었다. 테이블 통합 과정은 order_id 를 키로 orders 테이블과 order_products 테이블을 합쳐 각

주문이 유저별, 상품별 어떤 특징을 갖는지 확인하였다. 그리고 마지막 주문에서 재주문한 상품을 예측해야 하므로 이에 대한 테이블을 만들었다.

departments 와 aisles 테이블은 각 ID 에 대한 카테고리의 명칭들만 있어 특징이 될 수 없으므로 포함시키지 않았다.

5. 예측기법

본 대회는 데이터를 분석하여 각 고객이 마지막에 구매한 상품을 예측해야 하므로 이를 위해 XGBoost 프레임워크를 활용한다. XGBoost 는 결정트리의 한 종류로써 Greddy 알고리즘을 사용하여 분류기를 찾고 분산처리를 사용하여 빠른(Extreme) 속도로 적합한 파라미터의 비중을 찾는 알고리즘으로 최근 많은 데이터 경진대회 우승자들이 사용하면서 성능이 입증되었다. 또한 Greedy 알고리즘은 자동으로 가지치기가 가능하여 과적합(Overfitting)이 잘 일어나지 않으며 빠르고 유연하여 다른 알고리즘과 연계 활용성이 좋아 함께 붙여서 앙상블(Ensemble) 학습이 가능하다[6].

6. 결과

앞서 설명한 기법들을 바탕으로 실험결과를 측정하기 위한 도구로 F1 score[7]를 사용하였다. F1 score 는 정확도 계산을 위한 예측오류를 고려한 척도로 이용한다.

<표 2> 예측값과 실제값 비교 용어표

Actual / Predict	True	False
True	True Positive(TP)	False Positive(FP)
False	False Negative(FN)	True Negative(TN)

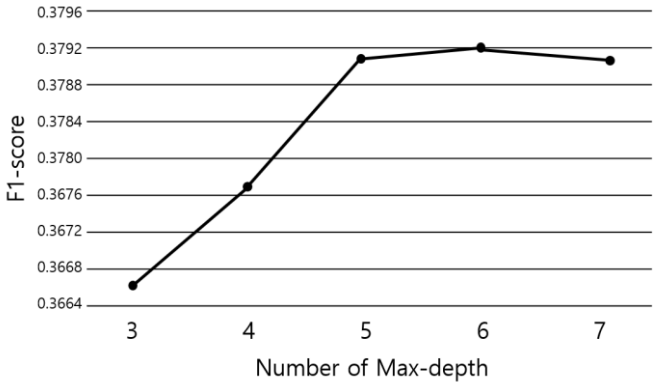
F1 score 는 precision 과 recall 두 가지 지수를 통계적으로 종합하여 주어진다. Recall 과 Precision 에 대한 식은 다음과 같으며 식의 용어 및 약어들은 <표 2>에 정의한다.

$$\text{Precision} = \frac{tp}{tp + fp} \quad (1)$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad (2)$$

F1 score 에 대한 식은 다음과 같다.

$$\text{F1 score} = 2 * \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} \quad (3)$$



(그림 4) Max-depth 개수 별 F1-score

알고리즘의 파라미터 최적화를 입증하기 위해 결정트리의 가지수를 결정하는 Max-depth 를 3~7 로 변화시킨 후 값을 비교하였으며 결과는 (그림 4)와 같다. 파라미터 값을 증가시킬수록 F1 score 는 점점 상승하다가 6 에서 가장 높았고 7 부터는 다시 하락하는 것을 확인하였다. 이를 Max-depth 최적의 값으로 확인하였으며 알고리즘의 성능 F1-score 는 0.3792 로 측정되었다.

본 논문에서는 결정트리 알고리즘과 boosting 기법을 활용한 고객의 재구매 상품을 예측하는 진행하였다. 그 결과 F1 score 의 Max-depth 의 개수가 6 일 때 가장 적합한 0.3792 를 나타냈다. 이는 고객에게 상품을 추천해주는 3 개의 상품 중 1 개 이상은 실제 고객이 구매하려는 상품임을 말한다. 또한 다른 상품들도 고객의 구매 패턴에 근거하여 고객에게 추천함으로써 고객은 필요한 상품을 구매할 수 있게 되므로 실제 e-commerce 시장의 활용가능성을 확인할 수 있었다.

※본 논문은 2017 년 한이음 ICT 멘토링 프로젝트의 결과물입니다.

참고문헌

[1] Roger S. Pressman. “Software Engineering, A
[2] Olaru, Cristina, and Louis Wehenkel. "A complete fuzzy decision tree technique." Fuzzy sets and systems 138.2 (2003): 221-254.
[3] XGBoost http://xgboost.readthedocs.io/
[4] kaggle competition – Instacart Market Basket Analysis https://www.kaggle.com/c/instacart-market-basket-analysis
[5] Instacart : Groceries Delivered From Local Stores https://www.instacart.com/
[6] xgboost 사용하기 https://brunch.co.kr/@snobberys/137
[7] f1 score wiki https://en.wikipedia.org/wiki/F1_score