

|  |
| --- |
| 추가 설명자료 |
|  |
| 2020 빅콘테스트  챔피언리그  팀 엄덕구 |

## **목차**

1. 전처리
2. 피쳐 엔지니어링
3. 내부 변수
4. 외부 변수
5. 변수 선택
6. 모델링
7. 예측 모델링
8. 최적화 모델링
9. 전처리

* Log Transfomation
  + 취급액

취급액 변수의 왜도가 심하여 이를 조정해주기 위해 로그화 적용

* Outlier
  + 취급액

1억원 이상의 값이 전체 데이터의 약 1%수준으로, 빈도가 적은 것에 비해 데이터의 대표성에 미치는 영향이 크므로 제거함

* Missing Value
  + 취급액

취급액의 결측치(전체 데이터의 2.5%)는 모두 0을 의미하므로 제거함

* + 노출(분)

다수의 Missing Value(전체 데이터의43.8%)들이 존재하였으며, 동일 시간에 함께 판매된 제품들의 노출(분)은 최상위 데이터에만 기록되고, 아래 데이터들은 기록되지 않은 것을 발견.

전 Row의 값을 끌어오는 Forward Fill 방식으로 채워넣음

* Time Adjustment
  + 방송일시

주어진 방송일시는 일별 방송 시간이 {당일 새벽 6시~익일 새벽2시}로 되어 있어 자정이 넘어가는 방송의 경우 {당일, 당월, 당해}가 아닌 {익일, 익월, 익년}으로 할당이 되는 문제가 발생함.

이를 해소하기 위해 모든 방송 시간대를 3시간씩 당겨주었음

1. 피쳐 엔지니어링
2. 내부/내부 파생 변수

|  |  |  |  |
| --- | --- | --- | --- |
|  | 변수명 | 설명 | 비고 |
| 1 | 노출(분) | 원 데이터의 노출 | 단위 : 분 |
| 2 | cast\_time | 해당 상품의 누적 연속 노출시간 | 단위 : 분 |
| 3 | cast\_count | 해당 상품의 누적 연속 편성수 |  |
| 4 | cast\_time\_sum | 해당 상품의 총 연속편성 시간 | 단위 : 분 |
| 5 | cast\_count\_sum | 해당 상품의 총 연속편성 횟수 |  |
| 6 | cast\_time\_ratio | cast\_time\_sum 대비 해당 편성의 비율 |  |
| 7 | 마더코드 | 원 데이터의 마더코드 |  |
| 8 | 상품코드 | 원 데이터의 상품코드 |  |
| 9 | 상품명 | 원 데이터의 상품명 |  |
| 10 | 상품명\_plan | 상품의 결제 관련 정보 | 무이자:0, 일시불:1 |
| 11 | 상품명\_add | 추가구성 여부 | T : 1, F : 0 |
| 12 | 상품명\_maker | 상품명에서 추출한 브랜드명 |  |
| 13 | 상품명\_set | 세트상품 여부 | T : 1, F : 0 |
| 14 | 상품명\_sex | 특정 성별 전용 상품 여부 | 여성:1 남성:2 구분없음:0 |
| 15 | 상품명\_kid | 아동용 상품 여부 | T:1, F:0 |
| 16 | 판매단가 | 원 데이터의 판매단가 |  |
| 17 | fake\_weight | 동시 편성된 여러 조건의 상품을 상품단가를 기준으로 역가중치 |  |
| 18 | fake\_weight2 | 동시편성 동일단가이면서 다른 상품인지 여부 |  |
| 19 | 가격\_9x | 가격 끝자리가 9로 끝나는지 여부 |  |
| 20 | 할인여부 | 판매단가에서 추가 할인이 있었는지 여부 |  |
| 21 | mean\_amt\_by\_hhmm | 시간당 평균 판매금액 |  |
| 22 | 방송일시 | 원 데이터의 방송일시 |  |
| 23 | 방송일시\_MM | 방송월 |  |
| 24 | 방송일시\_DD | 방송일 |  |
| 25 | 방송일시\_hh | 방송시 |  |
| 26 | 방송일시\_mm | 방송분 |  |
| 27 | 방송일시\_MMDD | 방송월,일 |  |
| 28 | 방송일시\_DDhh | 방송일,시 | (월:0~일:6) |
| 29 | 방송일시\_hhmm | 방송시,분 |  |
| 30 | 방송일시\_MMDDhh | 방송월,일,시 |  |
| 31 | 방송일시\_mmmm\_1 | 방송일시를 분 단위로 누적 | 일단위 리셋 |
| 32 | 방송일시\_mmmm\_2 | 방송일시를 분 단위로 누적 | 월단위 리셋 |
| 33 | 방송일시\_mmmm\_3 | 방송일시를 분 단위로 누적 | 년단위 리셋 |
| 34 | 방송일시\_dow | 방송 요일 |  |
| 35 | 방송일시\_dow2 | 주말 여부 |  |
| 36 | time\_cat1 | 구간화된 방송시간1 |  |
| 37 | time\_cat2 | 구간화된 방송시간2 |  |
| 38 | 판매단가\_cat | 구간화된 판매단가 |  |
| 39 | encoding\_상품명 | Label Encoding된 상품명 |  |
| 40 | encoding\_상품군 | Label Encoding된 상품군 |  |
| 41 | encoding\_상품명\_brand | Label Encoding된 브랜드명 |  |
| 42 | encoding\_new\_상품명 | Label Encoding된 불필요 요소가 제거된 상품명 |  |
| 43 | com | 동시판매 상품 여부 | T(1) / F(0) |
| 44 | fake\_weight3 | 동시판매 상품 중 몇번째인지 여부 |  |

1. 외부 변수

|  |  |  |  |
| --- | --- | --- | --- |
|  | 변수명 | 설명 | 비고 |
| 1 | review\_counts | 네이버쇼핑 기준 리뷰 개수 |  |
| 2 | internet\_price | 네이버 쇼핑 기준 최저가 |  |
| 3 | price\_minus | 최저가 대비 가격 |  |
| 4 | search\_naver | 네이버 쇼핑에 검색되는지 여부 |  |
| 5 | temperature | 기온 |  |
| 6 | search\_compare | 네이버 데이터랩 기준 NS홈쇼핑의 타 홈쇼핑 대비 상대적 검색량 |  |
| 7 | 변동 % | 해당 날짜의 전일대비 코스피 지수 상승률 |  |
| 8 | encoding\_cat1 | 네이버쇼핑API를 통해 재분류한 상품의 대분류 | 라벨인코딩 |
| 9 | encoding\_cat2 | 네이버쇼핑API를 통해 재분류한 상품의 중분류 | 라벨인코딩 |
| 10 | encoding\_cat3 | 네이버쇼핑API를 통해 재분류한 상품의 소분류 | 라벨인코딩 |

1. 변수 선택

* RFE(Recursive Feature Elimination)
* 모든 변수에서 시작해 사전 설정된 변수 개수에 다다를 때까지 가능한 모든 조합으로 변수를 제거하는 방법.

<변수 개수별 성능(MAPE) 비교>

|  |  |
| --- | --- |
| 변수 개수 | MAPE |
| 35 | 45.1 |
| 38 | 44.3 |
| 40 | 43.3 |
| 42 | 44.5 |
| 45 | 47.6 |

* 비교 결과 40개의 변수를 사용할 때 가장 좋은 성능을 보임.

<선택된 40개의 변수>

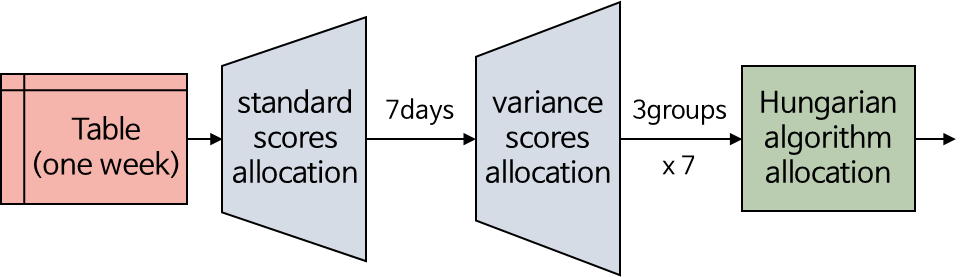
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 노출(분) | 마더코드 | 상품코드 | 판매단가 | 상품명\_kid |
| 상품명\_plan | 상품명\_add | 상품명\_maker | 상품명\_set | 상품명\_sex |
| fake\_weight | fake\_weight2 | fake\_weight3 | 가격\_9x | 할인여부 |
| price\_minus | search\_naver | review\_counts | internet\_price | mean\_amt\_by\_hhmm |
| encoding\_상품군: | encoding\_상품명\_brand: | encoding\_new\_상품명: | 판매단가\_cat | encoding\_상품명: |
| 'cast\_time | cast\_count | cast\_time\_sum | cast\_count\_sum | cast\_time\_ratio |
| 방송일시\_hh | 방송일시\_MMDD | 방송일시\_hhmm | 방송일시\_MMDDhh | 방송일시\_dow2 |
| time\_cat2 | com | encoding\_cat1: | encoding\_cat2: | encoding\_cat3: |

1. 모델링
2. 예측 모델링

* 모델 선택
* 바닐라 모델링을 통해 다양한 모델의 기본 성능을 비교
* 모델은 트리 기반, 선형 기반, 부스팅 기반의 모델들을 골고루 사용
* 가장 기본 성능이 좋은 RF와 XGB를 튜닝 모델로 선택
* 모델 튜닝
* 하이퍼 파라미터 튜닝을 위한 베이즈 서치 활용
* 하이퍼 파라미터별 특정 범위를 입력하여 최적의 파라미터 값 출력
* RF={max\_depth=20, max\_features=’auto’, n\_estimators=1000, min\_samples\_leaf=1, min\_samples\_split=2}
* XGB={learning\_rate=0.247, max\_depth=7, n\_estimators=300, colample\_bytree=0.31}
* 최종 결과
* 위 모델 튜닝으로 출력된 최적의 하이퍼 파라미터 값을 통해 모델 학습 후 테스트 스코어 출력
* RF = 33.64
* XGB = 29.68

1. 최적화 모델링

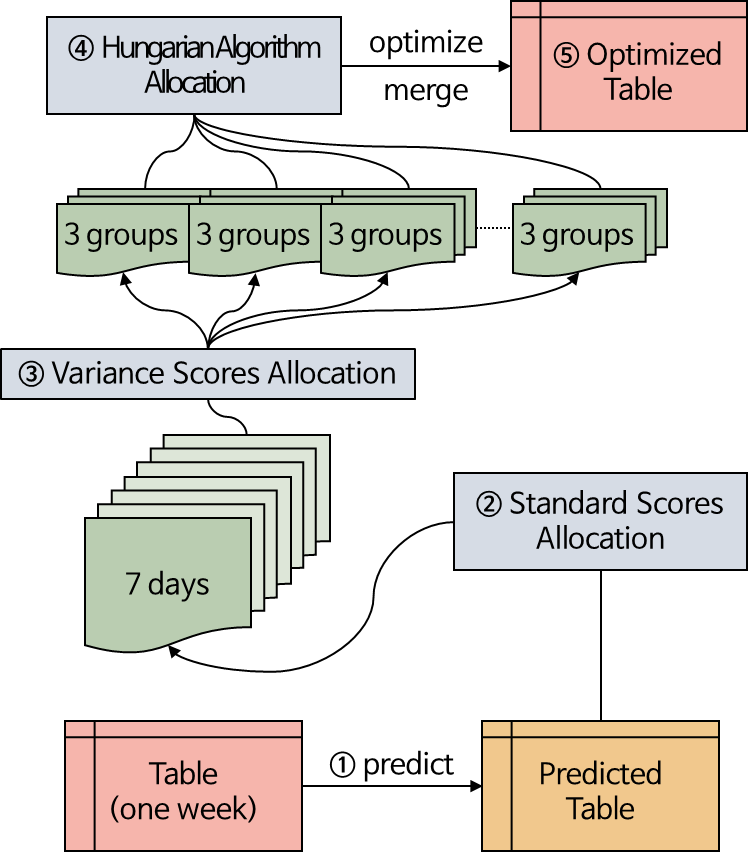
* 모델링 아이디어
* 가장 좋은 최적화 방안은 모든 경우의 수를 조합하여 전부 비교하는 것(모집단을 확인하려는 전수조사와 같음)
* 그러나 모든 경우의 수를 고려할 경우 연산량이 기하급수적으로 상승하여 현실적으로 불가능(하루치 편성표에 대한 연산량이 대략 20! 로 234경의 연산량이 듦)
* 이러한 연산량을 줄이기 위해 할당 알고리즘인 헝가리안 알고리즘을 활용하고자 함
* 그러나 헝가리안 알고리즘 또한 시간복잡도가 n의 4승이므로 행렬의 크기가 커질수록 연산량이 기하급수적으로 상승함
* 즉, 최대한 작은 단위로 나누어 작은 행렬을 만들고 병렬연산을 통해 연산량을 줄이고 결과값을 합치는 방식을 만들어야 함
* 따라서, ①일주일치의 상품을 하루 단위로 할당하고, ②하루 단위의 상품을 3그룹으로 할당하여, ③헝가리안 알고리즘을 병렬적으로 진행
* 모델 구조



* 일주일 단위의 편성표를 최적화하는 것을 기준으로 함
* 제공되는 편성표는 예측 모델을 통해 특정 시간(예측 취급액의 분산이 가장 큰 시간)을 기준으로 모든 요일(월~일)의 예측 취급액을 추출함
* Standard Scores Allocation(표준점수 할당 알고리즘)
  + - 본 단계는 일주일 간 판매할 상품을 하루 단위로 할당하는 과정임
    - 상품별로 최적의 요일에 할당하는 것이 목적임
    - 본 단계에 헝가리안 알고리즘이 아닌 표준점수 알고리즘을 활용한 이유는 헝가리안 알고리즘은 무조건 정방행렬을 기준으로 연산을 하기 때문에 이 경우 약 140!의 연산을 해야함. 그러나 표준점수 알고리즘은 압도적으로 적은 연산량으로 헝가리안 알고리즘에 버금가는 성능을 냄.
    - 알고리즘 로직은 다음과 같음
    1. 각 상품은 추출된 요일별 취급액을 표준점수화 시킴(상품별로 진행)
    2. 각 상품은 가장 높은 표준점수를 기록한 요일로 1차 할당
    3. 각 요일은 할당된 상품들의 표준점수를 내림차순하여 n개를 채우고 나머지는 탈락시킴(각 요일의 할당량은 균등함)
    4. 탈락한 상품들은 다음으로 높은 표준점수를 기록한 요일로 할당
    5. 모든 요일에 상품이 균등하게 할당될 때까지 ④~⑤번을 반복
* Variance Scores Allocation(분산점수 할당 알고리즘)
  + - 본 단계는 하루 동안 판매할 상품을 3그룹으로 할당하는 과정임
    - 상품별, 시간대별 예측 취급액의 분산의 크기에 맞춰 최적의 그룹에 할당하는 것이 목적임
    - 본 단계를 거치는 이유는 여전히 헝가리안 알고리즘이 작동하기에 너무 큰 연산량(약 20!)이 필요하기에 더 적은 연산량(6!~7!)로 줄여 최적화를 시킬 수 있도록 하는 것임

1. (방송 시간, 판매 상품)행렬을 만들어 모든 경우의 예측 취급액 추출
2. ①단계를 통해 추출된 취급액을 기준으로 각 방송 시간, 각 판매 상품에 대한 분산 값을 구함
3. ②를 통해 구해진 분산 값을 방송 시간, 판매 상품별로 내림차순하여 순서대로 1, 2, 3그룹으로 할당
4. 각 방송 시간 n그룹, 판매 상품 n그룹을 1:1로 매칭(방송1그룹:상품1그룹)
5. (방송 시간, 판매 상품)행렬은 (6~7, 6~7)행렬의 크기가 됨

* 최종 편성 최적화 모형



1. 일주일간 편성되어 있는 상품을 특정 시간 기준으로 예측모델을 활용하여 예상 취급액 추출
2. 표준점수 할당 알고리즘을 통해 각 상품을 최적의 요일로 균등 할당
3. 각 요일별 상품 후보를 분산점수 할당 알고리즘을 통해 3개의 그룹으로 할당
4. 시간그룹과 상품그룹을 1:1 매칭하여 헝가리안 알고리즘을 통해 최적의 조합 할당
5. ④번의 결과를 합쳐서 주간 편성표 확정