Chapter. 23

진짜 문제를 해결해보자 (1) 상점 신용카드 매출 예측

# 문제 소개

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

# I 대회 소개

- 출처: 상점 신용카드 매출 예측 경진대회
  - ➢ 문제 제공자: FUNDA (데이콘)
  - ➢ 소상공인 가맹점 신용카드 빅데이터와 AI로 매출 예측 분석
  - ➢ https://dacon.io/competitions/official/140472/overview/

- 문제 개요: 2019년 2월 28일까지의 카드 거래 데이터를 이용하여 2019년 3월 1일 ~ 5월 31일까지의 **상점별 3개월 총 매출** 예측

# I 사용 데이터

- funda_train.csv: 모델 학습용 데이터
  - store_id: 상점의 고유 id
  - card_id: 사용한 카드의 고유 아이디 // card_company: 비식별화된 카드 회사
  - transcated_date: 거래 날짜 // transacted_time: 거래 시간
  - installment_term: 할부 개월 수
  - region: 상점 지역 // type_of_business: 상점 업종
  - amount: 거래액

- submission.csv: 모델 적용 데이터
  - store_id: 상점의 고유 id

# I 문제의 핵심: 특징 및 라벨 추출을 위한 데이터 요약

- 제공된 데이터의 레코드의 단위는 **거래**이며, 예측하고자 하는 레코드의 단위는 3개월 간의 **상점 매출**임

| | store_id | card_id | card_company | transacted_date | transacted_time | installment_term | region | type_of_business | amount |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | b | 2016-06-01 | 13:13 | 0 | NaN | 기타 미용업 | 1857.142857 |
| 1 | 0 | 1 | h | 2016-06-01 | 18:12 | 0 | NaN | 기타 미용업 | 857.142857 |
| 2 | 0 | 2 | c | 2016-06-01 | 18:52 | 0 | NaN | 기타 미용업 | 2000.000000 |
| 3 | 0 | 3 | a | 2016-06-01 | 20:22 | 0 | NaN | 기타 미용업 | 7857.142857 |
| 4 | 0 | 4 | c | 2016-06-02 | 11:06 | 0 | NaN | 기타 미용업 | 2000.000000 |

**Y: store_id별 2019년 3월 1일 ~ 5월 31일의 매출 합계**

Chapter. 23

진짜 문제를 해결해보자 (1) 상점 신용카드 매출 예측

# 학습 데이터 구축

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

# I 기본 데이터 구조 설계



- 레코드가 수집된 시간 기준으로 3개월 이후의 총 매출을 예측하도록 구조를 설계해야 함

| 상점 ID | 시점 | 특징 | 라벨 |
|---|---|---|---|
| 1 | 4 | 시점 1 ~ 3까지의 상점 1의 특징 | 시점 5 ~ 7까지의 상점 1의 매출 합계 |
| 1 | 5 | 시점 2 ~ 4까지의 상점 1의 특징 | 시점 6 ~ 8까지의 상점 1의 매출 합계 |
| 1 | 6 | 시점 3 ~ 5까지의 상점 1의 특징 | 시점 7 ~ 9까지의 상점 1의 매출 합계 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 2136 | 38 | 시점 35 ~ 37까지의 상점 1의 특징 | 시점 38 ~ 40까지의 상점 1의 매출 합계 |

- 시점의 정의 = ((년 – 2016) * 12 + 월 )

# I 시점 변수 생성

1. 기존 시간 변수 (transacted_date) 에서 연도 (transacted_year) 와 월(transacted_month)을 추출

2. 시점 변수 생성: 시점 (t) = (연도 – 2016) * 12 + 월

3. 불필요한 변수 제거
   - transacted_year
   - transacted_month
   - transacted_date
   - transacted_time

# | 범주 변수 탐색

1. card_id, card_company는 특징으로 사용하기에는 도메인 지식 하에서 부적절하다고 판단하여 삭제

2. 업종 (type_of_business), 지역 (region), 할부 거래(installment_term)에 대한 value_counts 수행
   - ➢ 상태 공간이 매우 큰 범주 변수임을 확인하여, 더미화하기에는 부적절하다고 판단
   - ➢ 업종 및 지역에 따른 상점 매출 합계의 평균을 사용하기로 결정
   - ➢ 할부 값은 할부 거래인지 여부만 나타내도록 이진화
   - ➢ 이 과정에서 결측은 제거하지 않고 없음이라고 변환

# I 학습 데이터 구조 작성

- 기존에 **정리되지 않은 데이터**를 바탕으로 학습 데이터를 생성해야 하는 경우에는 레코드의 단위를 고려하여 **학습 데이터의 구조를 먼저 작성**하는 것이 바람직함

- funda_train.csv (이하 train_df)에서 store_id, region, type_of_business, t를 기준으로 **중복을 제거**한 뒤, 해당 컬럼만 갖는 데이터프레임으로 학습 데이터(train_df)를 초기화함

# 평균 할부율 부착

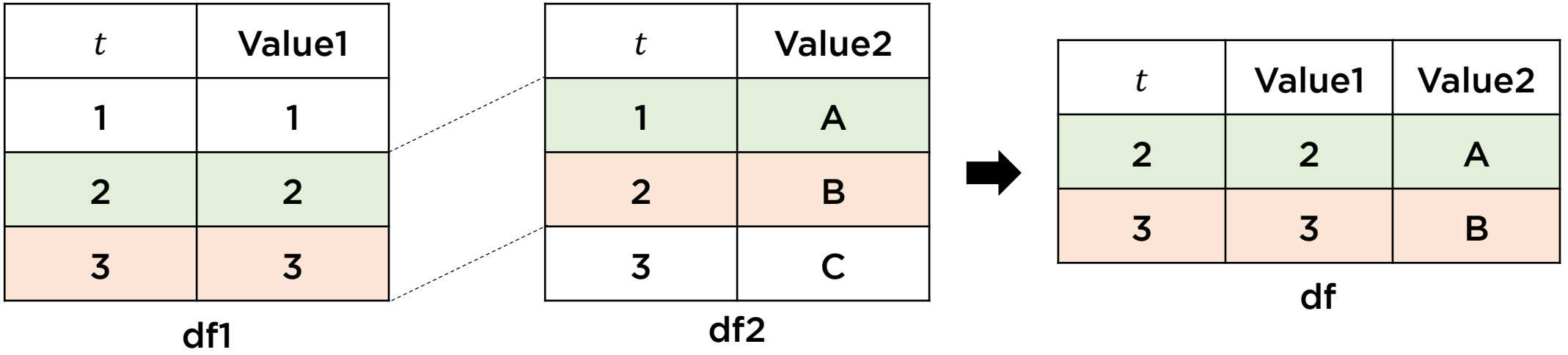
1. installment_term_per_store 생성
   - ➢ store_id에 따른 installment_term의 평균을 groupby를 이용하여 생성: installment_term_per_store

2. installment_term_per_store를 사전화: installment_term_per_store.to_dict()

3. train_df의 store_id를 replace하는 방식으로 평균 할부율 변수 생성

# I 기존 데이터 부착 테크닉

- 한 데이터에서는 시점 t를, 다른 데이터에서는 시점 t - 1을 붙여야 하는 경우



| $t$ | Value1 |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

df1

| $t$ | Value2 |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |

df2

| $t$ | Value1 | Value2 |
|---|---|---|
| 2 | 2 | A |
| 3 | 3 | B |

df

- Case 1. t가 유니크한 경우, 각 데이터를 정렬 후, 한 데이터에 대해 shift를 사용
- Case 2. t가 유니크하지 않은 경우, t_1 변수를 생성

# I 기존 데이터 부착 테크닉: Case 1

- Case 1. t가 유니크한 경우, 각 데이터를 정렬 후, 한 데이터에 대해 shift를 사용한 뒤 concat 수행

| $t$ | Value1 |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

df1

| $t$ | Value2 |
|---|---|
| NaN | NaN |
| 1 | A |
| 2 | B |

df2.shift(1)

concat

| $t$ | Value1 | Value2 |
|---|---|---|
| 2 | 2 | A |
| 3 | 3 | B |

df

# I 기존 데이터 부착 테크닉: Case 2

- Case 2. t가 유니크하지 않은 경우, t+1 변수를 생성하여 merge를 수행

| $t$ | Value1 |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

df1

| $t+1$ | Value2 |
|---|---|
| 2 | A |
| 3 | B |
| 4 | C |

df2

merge(df1, df2,
left_on = 't1',
right_on = 't+1')

| $t$ | Value1 | Value2 |
|---|---|---|
| 2 | 2 | A |
| 3 | 3 | B |

df

# 기존 매출 합계 부착

1. store_id와 t에 따른 amount의 합계 계산: amount_sum_per_t_and_sid

2. 다음 과정을 k = 1, 2, 3에 대해 반복
   1) amount_sum_per_t_and_sid에 t_k 변수 생성 (t_k = t + k)
   2) train_df와 amount_sum_per_t_and_sid 병합
      (단, amount_sum_per_t_and_sid에는 t 컬럼 삭제)
   3) 병합 후 train_df의 amount 변수명을 k_before_amount로 변경
   4) 불필요한 변수가 추가되는 것을 막기 위해, amount_sum_per_t_and_sid와 train_df에 t_k 변수 삭제

# 기존 지역별 매출 합계 부착

1. store_id를 키로 하고, region을 value로 하는 사전 생성

2. amount_sum_per_t_and_sid에서 region 변수 생성 및 region과 t에 따른 amount 평균 계산: amount_mean_per_t_and_region

3. 다음 과정을 k = 1, 2, 3에 대해 반복
   1) amount_mean_per_t_and_region에 t_k 변수 생성 (t_k = t + k)
   2) train_df와 amount_mean_per_t_and_region 병합

      (단, amount_mean_per_t_and_region에는 t 컬럼 삭제)
   3) 병합 후 train_df의 amount 변수명을 k_before_amount_of_region로 변경
   4) 불필요한 변수가 추가되는 것을 막기 위해, amount_sum_per_t_and_sid와 train_df에 t_k 변수 삭제

# 기존 업종별 매출 합계 부착

1. store_id를 키로 하고, type_of_business를 value로 하는 사전 생성

2. amount_sum_per_t_and_sid에서 type_of_business 변수 생성 및 type_of_business와 t에 따른 amount 평균 계산: amount_mean_per_t_and_type_of_business

3. 다음 과정을 k = 1, 2, 3에 대해 반복

   1) amount_mean_per_t_and_ type_of_business에 t_k 변수 생성 (t_k = t + k)

   2) train_df와 amount_mean_per_t_and_ type_of_business 병합

   (단, type_of_business에는 t 컬럼 삭제)

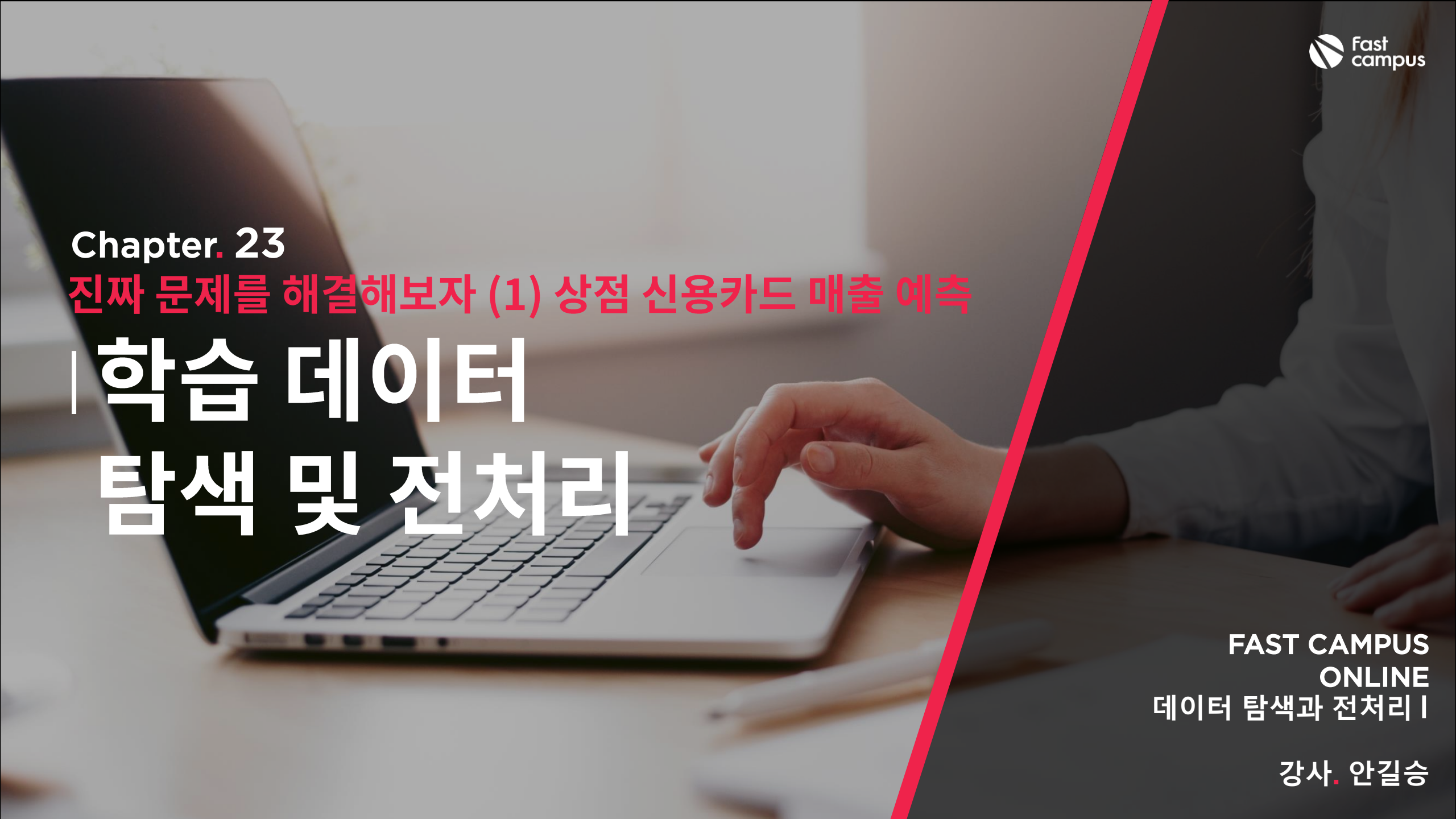
   1) 병합 후 train_df의 amount 변수명을 k_before_amount_of_region로 변경

   2) 불필요한 변수가 추가되는 것을 막기 위해, type_of_business와 train_df에 t_k 변수 삭제

# I 라벨 부착하기

1. 다음 과정을 k = 1, 2, 3에 대해 반복
   1) amount_sum_per_t_and_sid에 t_k (t_k = t - k)변수 생성
   2) train_df와 amount_sum_per_t_and_sid를 병합
   3) 병합 후, train_df의 amount 변수명을 Y_k로 변경

2. 라벨 생성: Y = Y_1 + Y_2 + Y_3



Chapter. 23

진짜 문제를 해결해보자 (1) 상점 신용카드 매출 예측

# 학습 데이터 탐색 및 전처리

FAST CAMPUS ONLINE
데이터 탐색과 전처리 I

강사. 안길승

# I 학습 데이터 기초 탐색 및 전처리

1. 특징과 라벨 분리
2. 학습 데이터와 평가 데이터로 데이터 분할
3. 학습 데이터 구조 및 기초 통계 분석
4. 이상치 제거
5. 치우침 제거
6. 스케일링 수행

Chapter. 23

진짜 문제를 해결해보자 (1) 상점 신용카드 매출 예측

# 모델 학습

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

# | 모델 선택

- 샘플 대비 특징이 적고, 특징의 타입이 전부 연속형으로 같음

- 따라서 아래 세 개의 모델 및 특징 선택 기준을 고려
  - ➢ 모델 1. kNN
  - ➢ 모델 2. RandomForestRegressor
  - ➢ 모델 3. LightGBM
  - ➢ 특징 선택: 3 ~ 10개 (기준: f_regression)

# | 파라미터 범위 선정 및 튜닝 수행

- k-NN
  - n_neighbors: [1, 3, 5, 7]
  - metric: [‘Euclidean’, ‘cosine’]

- Random Forest
  - max_depth: [1, 2, 3, 4]
  - n_estimators: [100, 200]
  - max_samples: [0.5, 0.6, 0.7, None]

- Light GBM
  - max_depth: [1, 2, 3, 4]
  - n_estimators: [100, 200]
  - learning_rate: [0.05, 0.1, 0.15]

# 최종 모델 학습

- 파라미터 튜닝을 통해 찾은 최적의 파라미터로 전체 데이터에 대해 재학습 수행

- 이때, 새로 들어온 데이터에 대해서도 동일한 전처리를 하기 위해, pipeline을 함수화함

Chapter. 23

진짜 문제를 해결해보자 (1) 상점 신용카드 매출 예측

# 모델 적용

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승

# 모델 적용

- 새로 들어온 데이터인 submission_df에 대해서도 모델의 입력으로 들어갈 수 있도록 전처리 수행
- 전처리된 데이터를 모델에 투입하여 출력값을 얻고, 이를 데이터프레임화하여 정리

Chapter.

진짜 문제를 해결해보자 (1) 상점 신용카드 매출 예측

# 감사합니다

FAST CAMPUS
ONLINE
데이터 탐색과 전처리 I

강사. 안길승