Log2Image

온라인 로그 데이터의 시각화 및 구매 시기 예측

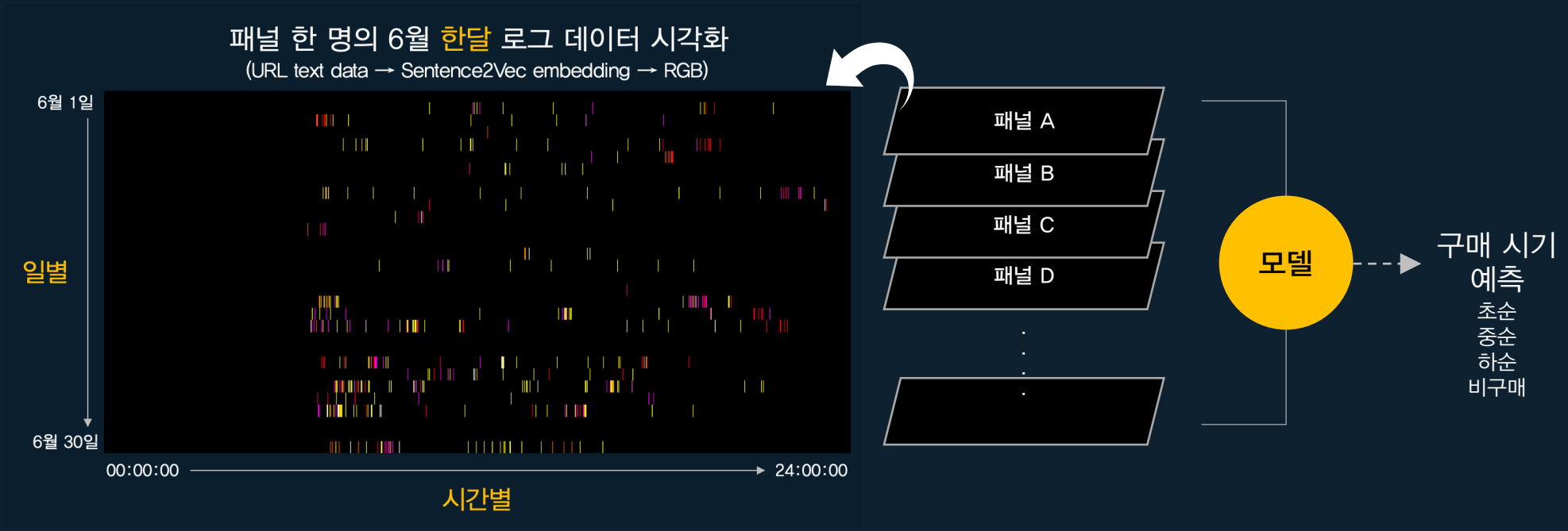
디지털애널리틱스 조예린

Index

01 Project Idea

비교적 큰 사이즈를 지니면서, URL이라는 비정형 텍스트 데이터를 지닌 로그 데이터
한 사람의 클릭스트림 패턴을 한 눈에 살펴볼 수 있도록 하는 방법

“Log2Image”



02 Data Selection

PC & Mobile 로그 데이터

- **수집기간**: 2014.04.01 ~ 2014.06.30
- **크기**: 34GB
- **주요 컬럼명**

UID: 패널 고유의 ID값
Time: 사이트의 클릭 날짜/시간
Full_URL: 패널이 방문한 사이트의 full url
Domain: 패널이 방문한 사이트의 도메인 정보
ownership_1: 사이트 오너십 기준 카테고리 대분류
ownership_2: 사이트 오너십 기준 카테고리 중분류

URL 클릭스트림 데이터 활용

구매행태 서베이 응답 데이터

- **수집시기**: 2014.07.14
- **크기**:1066명의 패널 대상, 151개의 질문 응답
- **주요 서베이 질문**

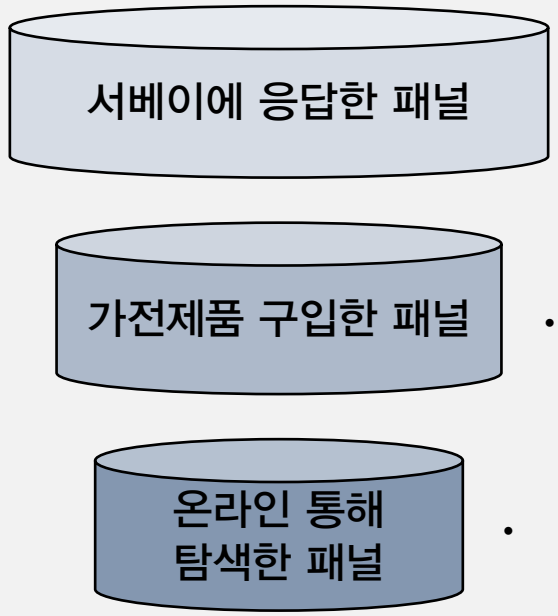
Q. 귀하께서 최근 3개월 이내에 구매한 제품을 모두 선택해주십시오.
A1-가전제품 A2-의류/잡화 A3-화장품 A4-금융상품
Q. 귀하께서는 해당 제품을 몇 월에 구매하셨습니까?
A1-4월 A2-5월 A3-6월 A4-7월
Q. 구매하신 시기는 해당 월 중 언제쯤 이십니까?
A1-초순 A2-중순 A3-하순

구매 시기 Label 추출

데이터 핸들링과 응답 데이터 신뢰도를 고려해,
서베이 응답 시기와 가장 가까우면서 한 달동안 완전히 수집된 달인 6월 로그 데이터 활용 *약 3495만 rows*
로그 데이터와 서베이 응답 데이터에 동시에 존재하는 UID(1063개)만을 고려

04 Data Preprocessing (1)Labeling

Labeling 과정



- 1063명 패널 대상
- 이 외: 패션의류/잡화, 금융상품, 화장품
- 이 외: 매장에서 제품 정보를 탐색함

구매 시기 응답 값(초순, 중순, 하순) 중
NaN(비구매) & 초순 = 0, 중순 = 1, 하순 = 2 부여

가전제품 구매 여부에 대한 label 생성

04 Data Preprocessing (2)전처리 및 URL 임베딩

전처리

- 노이즈 많은 Full_URL보다는 Domain 값 활용
- owership_1, ownership_2 결측치 수정
- owernship_1, ownership_2 결측 행 제거
- Domain .com, .co.kr 등 불용어 제거
- 11분 간 활동을 기준으로 Session_ID 부여

URL 임베딩

- Word2Vec Model
- 하나의 문헌 = 한 사람당 한 세션의 Domain들을 나열한 list

  Example. [‘auction’, ‘auction’, ‘kbstar’, ‘naver’]
- Dimension = 3, sg = 1, min_count = 1 로 학습
- 3차원 임베딩 값을 시각화에 이용할 RGB값으로 활용

6월 한 달, 매일, 매 분의 로그를 하나의 이미지로 나타내는
Log2Image 진행

04 Log2Image

패널 한 명의 6월 한달 로그 데이터 시각화

(URL text data → Word2Vec embedding → RGB)

05 Modeling

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense, Activation, BatchNormalization

model = Sequential()

model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(IMAGE_WIDTH, IMAGE_HEIGHT, IMAGE_CHANNELS)))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(3, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['acc'])

model.summary()
```

- 적은 이미지 수를 보완하기 위해, ImageDataGenerator 활용
- CNN(Convolutional Neural Network) 모델

05 Modeling

```
# imbalanced data를 위해 class weight 조정
class_weights = class_weight.compute_class_weight('balanced',
                                                  np.unique(train_data["category"]),
                                                  train_data["category"])
```

```
print(class_weights)
```

```
64166667 1.09219858 1.90123457]
```

```
epochs=3 if FAST_RUN else 50
history = model.fit_generator(
    train_generator,
    epochs=epochs,
    validation_data=test_generator,
    validation_steps=total_test//batch_size,
    steps_per_epoch=total_train//batch_size,
    callbacks=callbacks,
    class_weight=class_weights)
```

- Imbalanced data를 보완하기 위해 모델 학습 시,
  class_weight이라는 parameter를 통해
  데이터 수가 적은 label의 경우 가중해서 학습

06 Evaluation & Implication

Results

• Acc	0.89
• Loss	0.52
• Recall	0.8942
• F1 score	0.8579

- Metrics는 높은 편이나, 이는 Imbalanced data로 인한 결과
- 1063개의 극히 적은 이미지 수로 인한 한계 존재
- 구매 시기를 서베이에 기반하여 유추할 수 밖에 없었던 한계 존재
- 한 사람의 클릭스트림 및 온라인 행동 패턴을
  한 눈에 시각적으로 확인하며 추후 클러스터링, 구매 예측에 활용 가능