

데이터분석, machine learning, deep-learning

Portfolio

지원자: 최은비

연락처 : 010-6341-1204

e-mail : p5653ceb@gmail.com

github : github.com/p5653ceb

Contents

1. Brunch 사용자를 위한 추천시스템
2. CCTV 인물 detection
3. 여성인력 임금 예측
4. 홈쇼핑 매출 예측 & 최적 편성 제안

1. Brunch 사용자를 위한 추천시스템

brunch



[machine learning] Brunch 사용자를 위한 추천시스템 개발

프로젝트 상세

프로젝트 내용

- Brunch 사용자의 구독 정보를 기반으로 새로운 글을 추천하는 추천 시스템 개발

활용 솔루션

- 3000명의 추천 대상 그룹에게 동일한 방법으로 추천을 해주는 것은 의미가 없을 것으로 판단
- 대상그룹을 세분화 하여 각자 다른 방식으로 추천
 - 1) following based 추천
 - 전체 독자의 98%는 follow하는 독자가 있음
 - 2) magazine based 추천
 - magazine의 글을 읽는 경향이 더 큼
 - 3) popularity based 추천
 - 등록되지 2주가 지난 후에는 글 소비 수가 거의 0에 가까워짐
 - brunch platform은 지식 전달의 목적보다는 일상이나 자신의 생각을 전달하는 글이 대부분으로, steady seller와 같은 글이 거의 없음
 - 4) interest based 추천 (collaborative filtering)
 - keyword list를 활용하여 독자가 읽은 글의 키워드와, 작가가 쓴 글의 키워드로 취향이 유사한 작가와 독자를 선별하여 읽은글 추천

주요 역할

- 1) 데이터 전처리 및 EDA
 - 최근 2주 이내 소비된 글을 보는 경향: 글 추천 기간 한정
 - 대상 그룹 구분: 단 한 편의 글만 읽은 사람과 여러편의 글을 읽은 사람에게 추천은 다른 방식으로 진행되어야 함
- 2) 협업필터링 모델 개발
 - 독자-독자, 독자-작가 간 취향 유사도 확인
 - tf-idf를 활용한 vector화 및 cosine 유사도 확인
 - Doc2vec을 활용한 유사도 확인
 - 취향 유사독자, 유사작가 10명씩을 뽑아 유사 독자가 읽었거나, 유사 작가가 쓴 최신/인기글 100개 추천

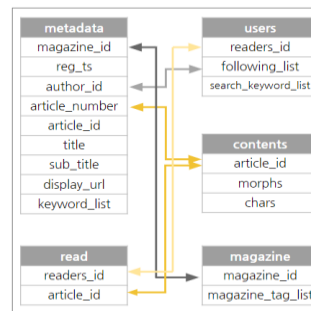
※ 깃헙 주소: https://github.com/p5653ceb/Regression_3_project

[machine learning] Brunch 사용자를 위한 추천시스템 개발

데이터 탐색

2 데이터 소개 _ 스키마 분석

제공된 브런치 데이터는 5개의 데이터 프레임으로 구성되어 있으며, 주석별자, 외래키를 사용해 데이터가 서로 연동되어 있음

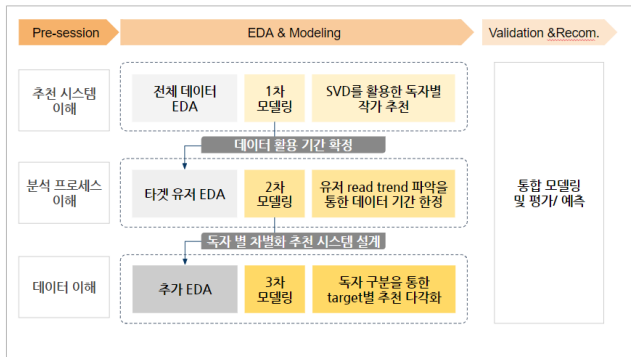


데이터 셋은 5개의 데이터 프레임으로 구성

metadata	글의 메타 데이터 (제목, 아이디 등)
users	브런치에 가입한 작가/독자의 정보
read	독자가 본 글의 정보
contents	글 본문 정보 (실체 컨텐츠, 원태소 분석 결과)
magazine	브런치 매거진 데이터

1 프로젝트 수행 프로세스

프로젝트 프로세스는 pre-session, EDA & Modeling, Validation & Recom 순서로 진행되며, EDA & Modeling 단계에선 1차, 2차, 3차 단계로 반복해서 구체화함.



2 EDA & Modeling _ 1차 모델링

1차 모델링 SVD를 활용한 독자별 작가 추천



[machine learning] Brunch 사용자를 위한 추천시스템 개발

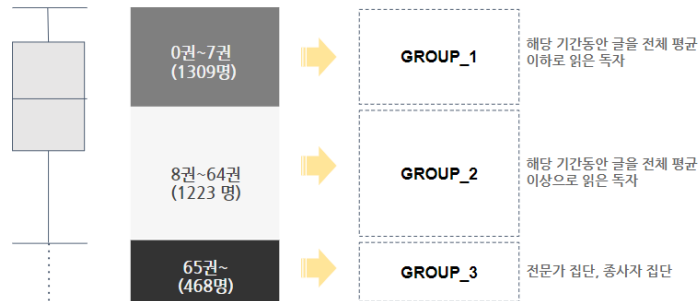
타겟유저 구분

2 EDA & Modeling _ 타겟유저 EDA

2차 모델링 독자 구분을 통한 target별 추천 다각화

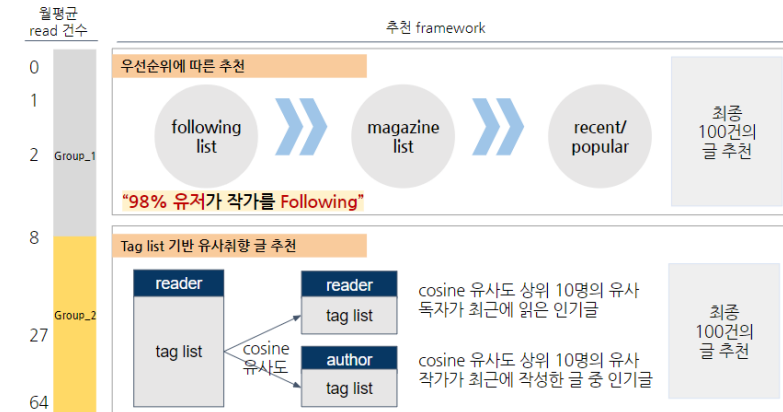
validation 기간(2월 7일~2월 22일)에 타겟 독자들이 읽은 글을 전체 데이터의 월별 최소값, 평균, upperfence 정보를 바탕으로 타겟 세그멘테이션 진행, 각 그룹별로 다른 추천 프레임워크 적용 예정

전체 독자 30만명 평균 → 전체 타겟 독자 3000명 분포 확인 → 타겟 세그먼트 (3개 그룹으로 독자 세분화)



2 EDA & Modeling _ 2차 모델링/ 추천framework 재구성

2차 모델링 독자 구분을 통한 target별 추천 다각화



[machine learning] Brunch 사용자를 위한 추천시스템 개발

타겟별 추천 다각화

2 EDA & Modeling _ 2차 모델링/ 추천framework 재구성



2차 모델링 독자 구분을 통한 target별 추천 다각화

최근에 읽은 글이 8편 미만으로 취향 파악이 어려운 독자

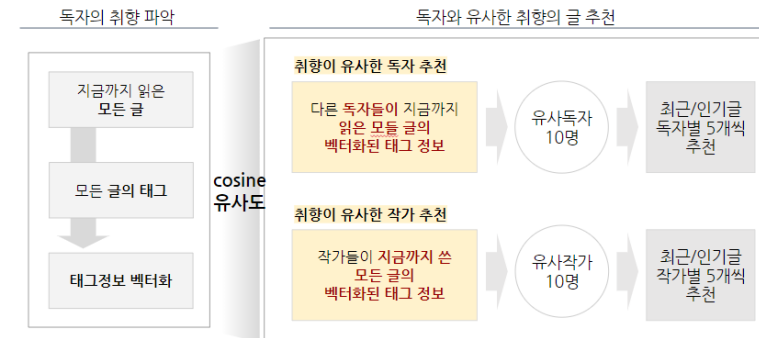


2 EDA & Modeling _ 2차 모델링/ 추천framework 재구성



2차 모델링 독자 구분을 통한 target별 추천 다각화

8편 이상 글을 읽어 취향 파악이 용이한 독자

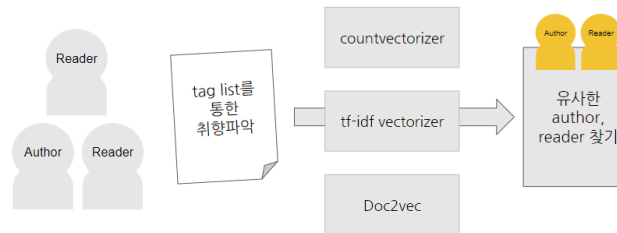


[machine learning] Brunch 사용자를 위한 추천시스템 개발

vectorizer활용

2 EDA & Modeling _ 3차 모델링

3차 모델링 다양한 vectorizer를 활용한 추천 시도



2 EDA & Modeling _ 3차 모델링

3차 모델링 다양한 vectorizer를 활용한 추천 시도

countvectorizer

텍스트 문서를 token count의 형태로 변환

tf-idf vectorizer

단어빈도와 역문서빈도를 활용하여 중요도가 표현되도록 벡터화

Doc2vec

주변 단어를 통한 단어의 벡터화

아래 글을 읽은 독자의 취향과 비슷한 글을 쓰는 작가를 추천해줘!

```
train_key_sum[train_key_sum['readers_id'] == '00013c35c709f903654ba06389ea75cb']
```

readers_id	keyword_list
0 00013c35c709f903654ba06389ea75cb	['욕아', '결혼', '시어머니', '연애', '욕아', '결혼', '시어머니', ...]

author_id	keyword_list
3931 @dotok	['욕아', '결혼', '시어머니', '연애', '욕아', '결혼', '시어머니', ...]
665 @ellnew	['욕아', '취향인', '결혼', ...]
11845 @morningyout	['시어머니', '결혼', '가족', '결혼', '우울증', '에세이', '결혼', ...]
17370 @yoonarch	['결혼', '친구이여가', '생각', '결혼', '결혼생활', '결혼', '사랑', ...]
2060 @brunchphw	['결혼', '결혼식', '시어머니', '사약', '머느리', '결혼', ...]
4087 @dust-universe	['연간관계', '욕아', '에세이', '연애', '결혼', '욕아', '욕아', ...]
12804 @suason08	['시어머니', '가족', '문화', ...]
15047 @suason08	['레고로프르즈', '결혼', '장남장녀', ...]
10931 @hongmang	['결혼', ...]
4540 @unsilnee	['결혼', ...]

2 EDA & Modeling _ 3차 모델링

3차 모델링 다양한 vectorizer를 활용한 추천 시도

countvectorizer

텍스트 문서를 token count의 형태로 변환

tf-idf vectorizer

단어빈도와 역문서빈도를 활용하여 중요도가 표현되도록 벡터화

Doc2vec

주변 단어를 통한 단어의 벡터화

```
# 한 독자의 관심사를 하나의 document로 보고, 독자 id를 문서 제목으로 봄
tags = reader_klist_df['keyword_list'].index
words = reader_klist_df['keyword_list'].values
```

2.1.3 Doc2Vec model build 2.1.4 Doc2Vec model 학습

```
max_epochs=10
model = Doc2Vec(window=10,
                 size=150,
                 alpha=0.025,
                 min_alpha=0.025,
                 min_count=2,
                 db=-1,
                 negative=3,
                 seed=9999)
model.build_vocab(tagged_data)

for epoch in range(max_epochs):
    print('iteration {}'.format(epoch))
    model.train(tagged_data,
                total_examples=model.corpus_count,
                epochs=model.iter)
    model.alpha -= 0.002
    model.min_alpha=model.alpha
    iteration 0
    iteration 1
    iteration 2
```

readers_id	keyword_list
0 00013c35c709f903654ba06389ea75cb	['욕아', '결혼', '시어머니', ...]
1 00013c35c709f903654ba06389ea75cb	['욕아', '결혼', '시어머니', ...]
2 00013c35c709f903654ba06389ea75cb	['욕아', '결혼', '시어머니', ...]
3 00013c35c709f903654ba06389ea75cb	['결혼', '욕아', '시어머니', ...]
4 00013c35c709f903654ba06389ea75cb	['욕아', '결혼', '시어머니', ...]



2. YOLO v5를 이용한 CCTV 인물 detection

[deep learning] YOLO v5를 이용한 CCTV 인물 detection

프로젝트 상세

프로젝트 내용

- 입구에서 촬영된 CCTV영상으로 학습 후 실내 CCTV에서 동일인물을 detection 하도록 하는 프로젝트

활용 솔루션

- YOLO v5
 - YOLO의 가장 최신 버전인 v5 활용
 - 더 적은 양의 사진으로 더 정확하고 빠르게 학습이 필요하여 v5를 선택
 - 학습은 YOLO v5s를 사용하여 진행
- data augmentation
 - 선명하게 나온 사진을 많이 활용하기 위해 영상을 다시 자르지 않고 선명한 사진을 증강함
 - 1장의 사진을 12장의 사진으로 증강하여 학습에 사용 총 9장의 사진을 108장으로 증강하여 학습에 사용
- train
 - 1000 epoch, batch-size 64 설정
 - weights는 설정 없이 학습 진행 (학습 중 기존에 사용했던 best weight를 사용해 보았으나 성능차이가 거의 없었음)

주요 역할

- 1) 데이터 좌표 변경 및 라벨링
 - 기존의 json형식의 좌표를 YOLO v5에서 요구하는 좌표의 순서에 맞게 txt파일로 변경 (class, x, y, w, h)
 - BlackPink 데이터에 대해서는 YOLO mark를 사용하여 라벨링을 진행
- 2) 학습 및 augmentation
 - YOLO v5를 활용하여 학습
 - data augmentation 진행 : 기존에 가진 좌표도 함께 augmentation하기 위해 imgaug 모듈 사용

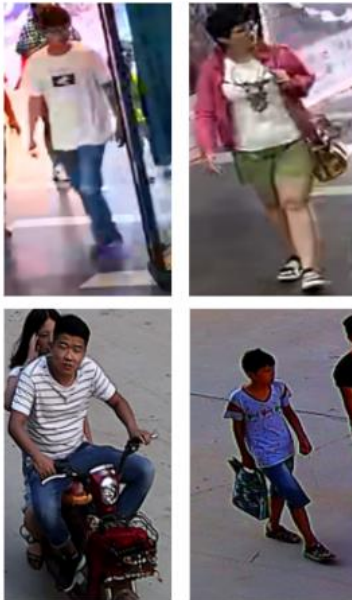
※ 깃헙 주소: <https://github.com/p5653ceb/deeplearning-repo-2>

[deep learning] YOLO v5를 이용한 CCTV 인물 detection

CCTV dataset

2. Custom Dataset with YOLO v5

prepare dataset



json

```
"mark": [
{
  "coordinates": [
    [
      830.4345999999999,
      298.59640000000013
    ],
    [
      920.9345999999999,
      298.59640000000013
    ],
    [
      920.9345999999999,
      567.5964000000001
    ],
    [
      830.4345999999999,
      567.5964000000001
    ]
  ],
  "label": {
    "Category": "pedestrian",
    "Clear face": "no",
    "Truncated": "non truncated",
    "Orientation": "left",
    "Occluded": "largely occluded"
  }
},
{
  "title": "10541_market/frames/01/00003.jpg"
}
```

```
1 # text file로 만들기
2 for filename in json_name_ls:
3     with open(filename, 'r') as file:
4         json_data = json.load(file)
5         with open(filename.split('.')[0] + '.txt', 'w') as f:
6             x = round(((json_data['mark'][0]['coordinates'][0][0]/1920) + (json_data['mark'][0]['coordinates'][1][0]/1920))/2, 3)
7             y = round(((json_data['mark'][0]['coordinates'][0][1]/1080) + (json_data['mark'][0]['coordinates'][1][1]/1080))/2, 3)
8             w = round((np.abs(json_data['mark'][0]['coordinates'][1][0]-json_data['mark'][0]['coordinates'][0][0]))/1920, 3)
9             h = round((np.abs(json_data['mark'][0]['coordinates'][1][1]-json_data['mark'][0]['coordinates'][0][1]))/1080, 3)
10            cl = filename.split('/')[1].split('-')[0] # market명에 번호를 class번호로 적어줌
11            a = [cl, x, y, w, h]
12            f.write(" ".join(map(str, a)))
```

01-00004.txt ✕

1 0 0.4095303125 0.43210259259254 0.06041666666666667 0.2537037037037037

txt

train.txt ✕ train.txt ✕ val.txt ✕

train.txt	train.txt	val.txt
1 /content/dr	1 /content/drive/MyDrive/Data/train_data/frame200.jpg	
2 /content/dr	2 /content/drive/MyDrive/Data/train_data/frame199.jpg	
3 /content/dr	3 /content/drive/MyDrive/Data/train_data/frame202.jpg	
4 /content/dr	4 /content/drive/MyDrive/Data/train_data/frame204.jpg	
5 /content/dr	5 /content/drive/MyDrive/Data/train_data/frame201.jpg	
6 /content/dr	6 /content/drive/MyDrive/Data/train_data/frame206.jpg	
7 /content/dr	7 /content/drive/MyDrive/Data/train_data/frame203.jpg	
8 /content/dr	8 /content/drive/MyDrive/Data/train_data/frame210.jpg	
9 /content/dr	9 /content/drive/MyDrive/Data/train_data/frame207.jpg	
10 /content/dr	10 /content/drive/MyDrive/Data/train_data/frame205.jpg	
11 /content/dr	11 /content/drive/MyDrive/Data/train_data/frame208.jpg	
12 /content/dr	12 /content/drive/MyDrive/Data/train_data/frame209.jpg	
13 /content/dr	13 /content/drive/MyDrive/Data/train_data/frame211.jpg	
14 /content/dr	14 /content/drive/MyDrive/Data/train_data/frame212.jpg	
15 /content/dr	15 /content/drive/MyDrive/Data/train_data/frame213.jpg	
16 /content/dr	16 /content/drive/MyDrive/Data/train_data/frame214.jpg	
17 /content/dr	17 /content/drive/MyDrive/Data/train_data/frame215.jpg	
18 /content/dr	18 /content/drive/MyDrive/Data/train_data/frame216.jpg	
19 /content/dr	19 /content/drive/MyDrive/Data/train_data/frame217.jpg	

data.yaml ✕

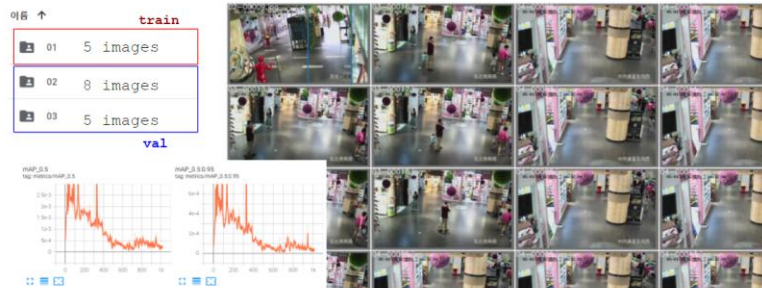
```
1 names: ['person']
2 nc: 1
3 train: /content/drive/MyDrive/Data/train.txt
4 val: /content/drive/MyDrive/Data/val.txt
```

[deep learning] YOLO v5를 이용한 CCTV 인물 detection

augmentation

2. Custom Dataset with YOLO v5

train & validation

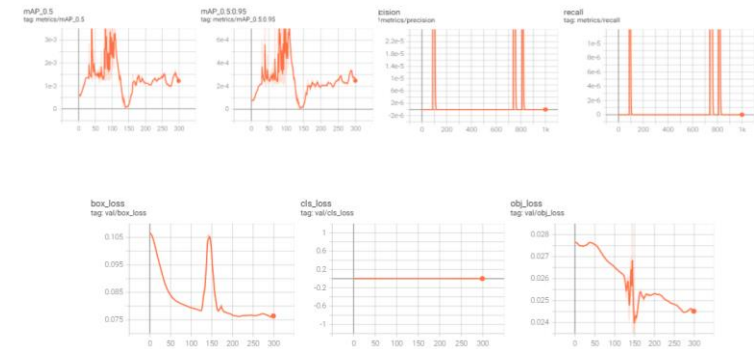


2. Custom Dataset with YOLO v5

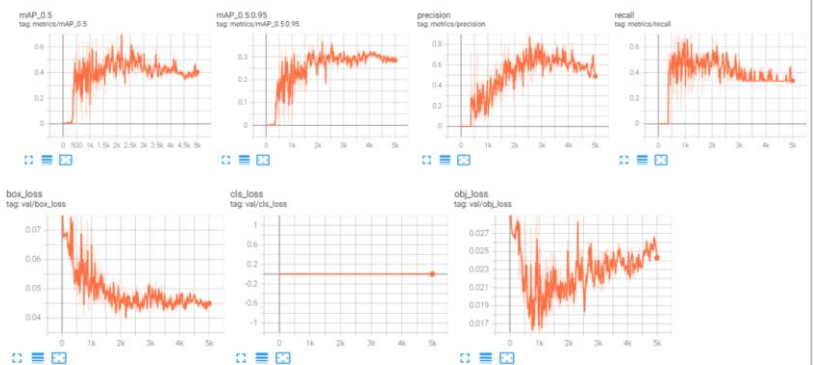
data augmentation



2. Custom Dataset with YOLO v5



2. Custom Dataset with YOLO v5



[deep learning] YOLO v5를 이용한 CCTV 인물 detection

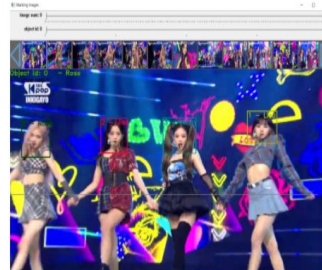
BlackPink dataset

2. Custom Dataset with YOLO v5

black pink video

1. labeling

- tool: yolomark



2. datasets

- Three versions of the same stage video

train



val

test



```
1 # 전체데이터를 train 시키기
2 !python train.py --data '/content/drive/MyDrive/Data/blackpink/data.yaml' --epochs 1000 --cfg '/content/drive/MyDrive/yolov5/models/yolov5s.yaml' --weights '' --batch-size 64 --name blackpink_detection
```


[deep learning] YOLO v5를 이용한 CCTV 인물 detection

결과



[deep learning] YOLO v5를 이용한 CCTV 인물 detection

결과





3. 여성 패널 데이터 기반 여성인력 임금 예측

[machine learning] 여성 인력 임금 예측

프로젝트 상세

프로젝트 내용

- 여성가족부 패널데이터를 활용, 여성인력의 임금을 예측하는 프로젝트

활용 솔루션

- RandomForest Regressor 사용 (max depth 3)
 - 1) test 결과값을 확인할 수 있는 machine learning model을 활용
 - R2 수치는 OLS에서 높게 나타났지만 train 결과에만 한정된 수치임을 확인하고 다른 모델 활용
 - 2) 다양한 모델, 파라미터를 grid search로 찾음
 - Linear, Decision Tree, Random Forest, GradientBoosting, XGBoost regressor를 모두 활용하고 grid search로 최적의 모델과 파라미터를 찾음
 - R2는 0.25, RMSE 52 : 월급의 단위가 만원, 예측의 오차는 52만원 정도

주요 역할

- | | |
|---|--|
| 1) 데이터 전처리 및 EDA <ul style="list-style-type: none">- 데이터 결측치 제거, 데이터 EDA 및 시각화- 대상 한정 (조사회사 및 응답자 첫 직장 입직년도 기준으로 한정) | 2) 예측 모델 개발 <ul style="list-style-type: none">- 변수 생성 (경력변수, 대졸여부 변수)- 모델 파이프라인 생성 및 하이퍼파라미터 튜닝 |
|---|--|

어려움

- 1) 패널데이터이다 보니 오래된 데이터가 많았음
- 2) 경력을 별도로 작성하지 않아 총 경력과 경력 단절에 대해 확인하기 어려웠음

해결방안

- 1) 2012년 이전 데이터는 분석에서 제외함
- 2) 경력데이터를 계산해서 삽입함 (첫 직장의 입직, 퇴직, 다음직장의 입직 퇴직 시기 등 고려)

※ 깃헙 주소: https://github.com/p5653ceb/Regression_3_project

[machine learning] 여성 인력 임금 예측 _ 데이터 전처리

데이터 전처리

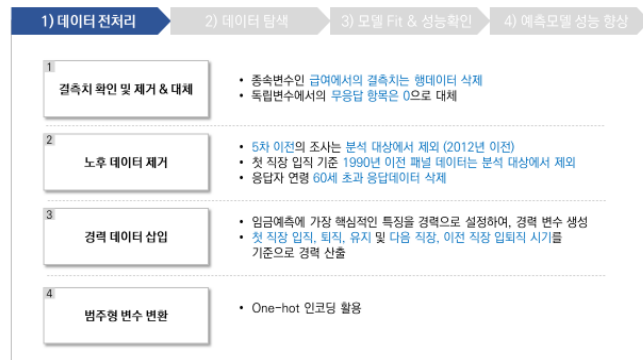
1. 분석 프로세스

다음 프로세스에 따라 분석을 진행함.



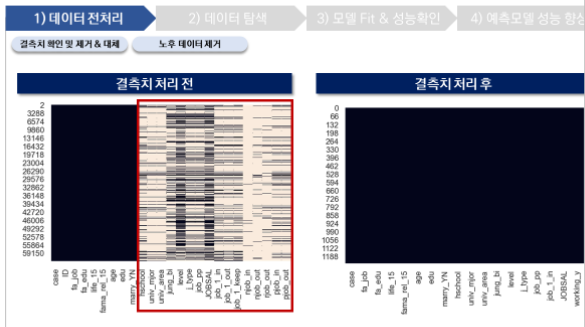
2. 분석 내용 _ 1) 데이터 전처리

원본 데이터에서 전처리를 위해 크게 결측치 확인 후 제거 및 대체, 노후 데이터 제거, 경력 데이터 생성 및 삽입, 범주형 데이터 변환의 네 가지 활동을 진행함.



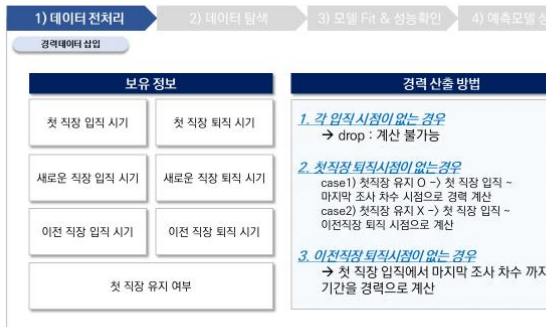
2. 분석 내용 _ 1) 데이터 전처리

전체 dataset에서 노후데이터를 제외하고 결측치를 처리하여 1245개의 최종 dataset을 도출함.



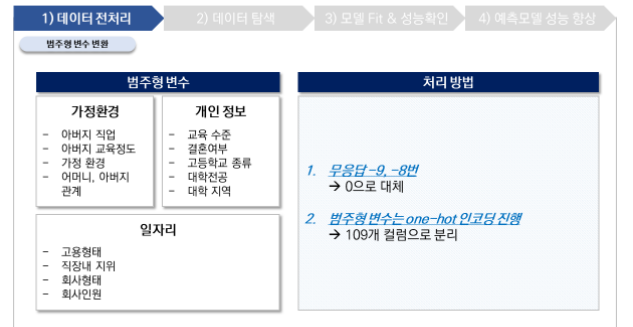
2. 분석 내용 _ 1) 데이터 전처리

경력정보 추가를 위해 기존 보유정보를 최대한 활용하여 경력을 산출함.



2. 분석 내용 _ 1) 데이터 전처리

무응답의 경우는 0으로 변경하고, 범주형 변수는 pandas의 pd.get_dummies를 이용하여 one-hot 인코딩 진행



[machine learning] 여성 인력 임금 예측 _ 데이터 탐색

데이터 탐색

2. 분석 내용 _ 2) 데이터 탐색

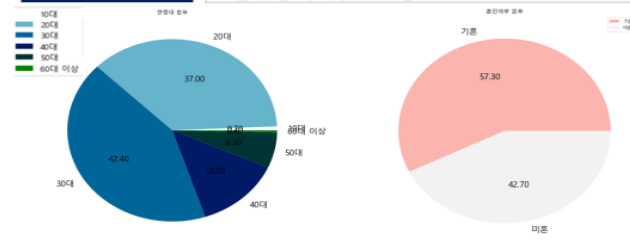
초기 62,426건의 데이터 중 전처리를 마친 1,877건의 데이터로 분석을 진행함.
연령에서 20~30대에 몰림 현상이 있으며, 혼인여부는 비교적 고르게 분포됨.

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

응답자 주요 특성 분석

분석데이터 요약

- 초기데이터: 62,426건
- 전처리 완료 후 데이터: 1,877건



15

2. 분석 내용 _ 2) 데이터 탐색

연령에 따른 급여의 변화를 보기 위해 두 변수의 상관 관계를 살펴봄

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

변수간 관계 분석



급여와 연령의 관계

- 여성의 임금은 30대까지는 연령 증가에 따라 증가하는 경향을 보이지만 30대에 들어서면서 정체 혹은 감소의 경향을 나타냄
- 30대 후반부터 급여 수준이 오르지만 40대로 진입하여 다시 감소추세를 나타냄

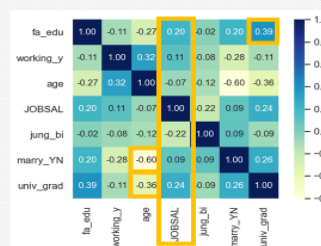
16

2. 분석 내용 _ 2) 데이터 탐색

변수간 관계분석을 위해 급여와 개인적 특성의 상관을 살펴봄.

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

변수간 관계 분석



급여와 상관관계

- 급여와 가장 상관이 높은 변수는 **대학 졸업 여부**(univ_grad, 0.24)
- 두 번째 상관이 높은 변수는 **아버지의 교육 수준**(fa_edu, 0.2)

변수간 상관관계

- 나이와 결혼여부가 가장 높음 (-0.6)
- 딸의 대졸여부는 아버지의 학력과 양의 상관**을 보임 (0.39)
- 나이와 대졸여부는 음의 상관을 보임 (-0.36)

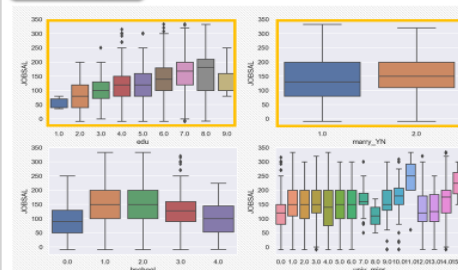
18

2. 분석 내용 _ 2) 데이터 탐색

급여에 영향을 미치는 요인들을 확인하기 위해 각 변수와 급여를 비교함

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

변수간 관계 분석



급여와 다른 변수와의 관계

- 급여와 가장 큰 상관을 보인 것은 본인의 교육 수준 (6년 전문대, 7년 4년제/5-6년제 포함), 8년 대학원 석사, 9년 대학원 박사)
- 본 분석에서는 대학졸업 여부로 구분하여 진행 (초대를 포함)
- 결혼여부는 1번이 유경험, 2번이 무경험. 중앙값은 미혼자가 조금 더 높음을 확인
- 대학전공과 급여에서 급격히 높아지는 전공은 (간호, 약학, 의학계열)

22

[machine learning] 여성 인력 임금 예측 _ 모델 fit & 성능확인

모델 fit & 성능확인

2. 분석 내용 _ 3) 모델 Fit & 성능확인

전처리 완료 후, 변수와 모델을 선정하고 fitting을 진행함.

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

최종 선정 변수

- 연령
- 경력
- 대학 소재지:서울
- 결혼여부: 미혼
- 교육수준: 4년제 대학졸업, 석/박사졸업
- 정규직 여부: 정규직

활용모델

- OLS
- LinearRegression()
- DecisionTreeRegressor(max_depth=3, random_state=13))
- RandomForestRegressor(n_jobs=-1, n_estimators=100, max_depth=3)
- GradientBoostingRegressor()
- XGBRegressor(max_depth=3)

26

2. 분석 내용 _ 3) 모델 Fit & 성능확인

OLS분석결과 R²값은 0.791로 모델의 설명력이 79%정도를 확인할 수 있었음.

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

OLS Regression Results

Dep. Variable:	WAGE	R-squared:	0.791						
Model:	OLS	Adj. R-squared:	0.789						
Method:	Least Squares	F-statistic:	1964.						
Date:	Wed, 26 Aug 2020	Prob (F-statistic):	0.00						
Time:	18:27:19	Log-Likelihood:	-11275.						
No. Observations:	1972	AIC:	2.256e+04						
Df Residuals:	1965	BIC:	2.260e+04						
Df Models:	7								
Covariance Type:	nonconstant								

	coef	std err	t	P> t	[0.025	0.975]
age	2.5818	0.885	27.286	0.000	2.488	2.772
working_y	3.4837	0.525	5.881	0.000	2.415	4.473
univ_area_1-8	13.8741	5.811	2.389	0.010	3.109	22.679
univ_area_2-8	47.4872	3.385	11.114	0.000	41.118	53.896
edu_1-8	41.0985	3.583	11.495	0.000	34.058	48.143
edu_9-0	16.1284	8.264	1.951	0.054	-1.152	33.409
edu_0-8	9.6672	26.226	0.369	0.712	-41.746	61.080

Omnibus:	5.357	Burkman-Watson:	1.931
Prob(Omnibus):	0.069	Jarque-Bera (JB):	6.830
Skew:	0.458	Prob(JB):	0.000
Kurtosis:	3.252	Cond. No.	551.

Warnings: [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

OLS분석결과

- 분석 변수
 - 나이, 경력, 인서울 대학 여부, 미혼여부, 4년제 대학졸업, 석사, 박사 여부
- 모델의 R²값: 0.791
- 변수별 유의수준을 보면 박사 졸업(edu_9.0)은 급여에 영향을 주지 못함

26

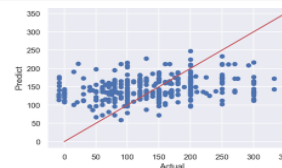
2. 분석 내용 _ 3) 모델 Fit & 성능확인

머신러닝 모델을 이용하여 동일 데이터를 분석하였을 때, R²값은 0.09, rmse는 71만원 정도로 분석됨. 실제 값과 예측값의 차이를 확인하기 위해 그래프로 나타냄

1) 데이터 전처리 2) 데이터 탐색 3) 모델 Fit & 성능확인 4) 예측모델 성능 향상

Model Fit 결과

	model_name	train_r2_score	test_r2_score	train_rmse	test_rmse
0	LinearRegression	0.08	0.09	70.72	71.68
1	DecisionTreeRegressor	0.11	0.05	69.52	73.25
2	RandomForestRegressor	0.13	0.06	68.90	72.63
3	GradientBoostingRegressor	0.21	0.05	65.28	73.89
4	XGBRegressor	0.20	0.06	65.87	72.96



Model Fit 결과

- 모델 분석 결과
 - Linear Regression가 가장 높은 R²값과 가장 낮은 rmse를 보임
- 예측값과 실제값 비교 그래프
 - 기울기 1인 직선상에 있는 데이터가 별로 없음
 - 데이터의 퍼짐이 심함

27

[machine learning] 여성 인력 임금 예측 _ 예측모델 성능 향상

예측모델 성능 향상

2. 분석 내용 _ 4) 예측모델 성능 향상

OLS분석 결과 R²값은 이전보다 개선된 0.892로 나타났고, 모든 변수는 본 모델에서 유의한 것으로 확인됨.

- 1) 데이터 전처리
- 2) 데이터 탐색
- 3) 모델 Fit & 성능확인
- 4) 예측모델 성능 향상

OLS Regression Results

Dep. Variable:	log_wage	Model:	OLS	R-squared (uncentered):	0.892	
Method:	Least Squares	Date:	Fri, 20 Aug 2020	Log-Likelihood:	-3881.8	
Time:	14:42:04	DF Residuals:	680	AIC:	7638.	
No. Observations:	685	DF Model:	5	BIC:	7648.	
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
age	2.7194	0.130	20.840	0.000	2.463	2.976
working_y	2.3486	0.730	3.216	0.002	0.755	3.942
marry_yn_2_0	36.4148	4.864	7.487	0.000	26.464	45.964
log_ill_1_0	46.7656	4.790	9.760	0.000	37.163	56.369
unhr_grat	33.3862	4.980	6.704	0.000	23.545	42.787
Const.	38.246					
Prob(Observed):	0.000	Jarque-Bera (JB):			33.486	
Skew:	0.496	Prob(JB):			5.35e-08	
Kurtosis:	3.426	Cond. No.			93.4	

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

OLS분석결과

- 분석 변수
- 나이, 경력, 미혼여부, 정규직 여부, 대학졸업여부(전문대 포함)
- 모델의 R²값: 0.892
- 모든 변수는 p값이 0.01 이하로 99% 수준에서 유의한 것으로 확인

2. 분석 내용 _ 4) 예측모델 성능 향상

MinMaxScaler를 적용하여 다시 분석한 결과 R²값이 다소 떨어졌지만 개별 변수의 유의확률은 높아짐.

- 1) 데이터 전처리
- 2) 데이터 탐색
- 3) 모델 Fit & 성능확인
- 4) 예측모델 성능 향상

OLS Regression Results

Dep. Variable:	log_wage	R-squared (uncentered):				
Model:	OLS	Adjusted R-squared:				
Method:	Least Squares	F-statistic:				
Date:	Fri, 20 Aug 2020	Prob (F-statistic):				
Time:	14:53:43	Log Likelihood:				
No. Observations:	685	AIC:				
DF Residuals:	680	BIC:				
DF Model:	5					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
x1	168.2864	8.648	19.458	0.000	151.226	185.346
x2	39.0867	10.112	3.868	0.000	19.154	58.019
x3	58.9375	5.292	11.140	0.000	48.350	69.525
x4	55.9985	5.871	9.537	0.000	44.434	65.567
x5	46.9117	5.087	9.223	0.000	36.824	56.899
Const.	29.138		Durbin-Watson:		1.953	
Prob(Observed):	0.000		Jarque-Bera (JB):		33.454	
Skew:	0.468		Prob(JB):		5.34e-08	
Kurtosis:	3.537		Cond. No.		5.83	

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

OLS분석결과 - MinMaxScaler적용

- 데이터는 범도의 스케일링 없이 사용
- R²값은 0.876으로 scaler 적용 전보다 떨어짐
- 각 변수의 유의확률은 모두 0.01 이하로 99% 수준에서 유의한 것으로 확인

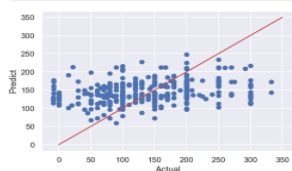
2. 분석 내용 _ 4) 예측모델 성능 향상

성능 향상 전, 후 예측값과 실제값의 비교 그래프 확인

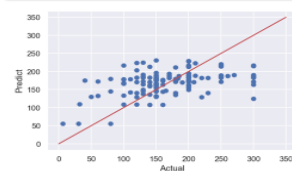
- 1) 데이터 전처리
- 2) 데이터 탐색
- 3) 모델 Fit & 성능확인
- 4) 예측모델 성능 향상

성능 향상 전후 비교

성능향상 전



성능향상 후



2. 분석 내용 _ 4) 예측모델 성능 향상

실제 임금 예측을 위해 가상의 인물을 선정하여 예측함.

- 1) 데이터 전처리
- 2) 데이터 탐색
- 3) 모델 Fit & 성능확인
- 4) 예측모델 성능 향상

실제 값 예측하기



- 나이: 21세
- 경력: 1년
- 결혼여부: 미혼
- 정규직여부: 비정규직
- 대출여부: 고졸

```
test_data = [[21, 1, 1, 0, 0]]
reg = RandomForestRegressor(max_depth=5, n_estimators=1000)
reg.fit(X_train, y_train)
reg.predict(test_data)
array([[108.06521622]])
```

예측임금

월 108만원



- 나이: 30세
- 경력: 5년
- 결혼여부: 기혼
- 정규직여부: 정규직
- 대출여부: 대출

```
test_data = [[30, 5, 0, 1, 1]]
reg = RandomForestRegressor(max_depth=5, n_estimators=1000)
reg.fit(X_train, y_train)
reg.predict(test_data)
array([[228.46811862]])
```

예측임금

월 228만원



4. 홈쇼핑 매출 예측 & 최적 편성 제안

[machine learning] 홈쇼핑 매출 예측 & 최적 편성 제안

프로젝트 상세

프로젝트 내용

- 2019년 1년 상품 매출 데이터를 기반으로 2020년 6월의 매출을 예측하고 최적의 편성 제안

활용 솔루션

- 예측: LGBM과 catboost 모델을 활용하여 최종 예측 진행
 - 모델 선정 기준: 높은 정확도와 짧은 학습시간/데이터의 비선형성을 효과적으로 반영할 필요
 - 앙상블 계열 알고리즘 중 많은 양의 데이터에 적합하고 속도가 빠른 LGBM과 범주형 변수 처리 속도가 빠른 catboost 사용
- 편성 제안: 통계적 예측기를 활용한 예상 매출 예측 및 헵타리안 알고리즘으로 최적 배치
 - 프로젝트의 예측 데이터를 기반으로 최대값을 산출하는데 활용
 - 1주일을 168시간, 방송이 없는 시간을 제외하여 147시간으로 보고 1주일 편성 제안
 - 카테고리별 평균 취급액과 시간대별 평균 취급액의 가중치를 조정하여 실제값과 예측값의 RMSE를 최소로 만드는 가중치의 조합 도출
 - 147개 시간대 별로 최적 가중치를 적용하여 매출을 예측
 - 매출이 최대가 되는 최적 편성을 위해 헵타리안 알고리즘을 활용

주요 역할

- | | |
|---|---|
| <ul style="list-style-type: none">1) 데이터 전처리 및 EDA/ 예측모델 개발<ul style="list-style-type: none">- 결측치 제거/ 경향성 EDA- 변수 생성: 카테고리 관련 파생변수 생성- 외부변수 탐색: 마스크 검색량/ 기온/ 상대기온 추가/ 타방송 시청률 크롤링 및 추가 | <ul style="list-style-type: none">2) 편성 모델 개발<ul style="list-style-type: none">- 카테고리별 시간대별 평균 매출액을 활용하여 예상매출 matrix 생성 고안- 시간대/ 카테고리별 차별화된 가중치 적용- 헵타리안 알고리즘을 활용한 최적 편성 제안 |
|---|---|

[machine learning] 홈쇼핑 매출 예측 & 최적 편성 제안

매출예측 모델링

2-3. 데이터 모델링 학습알고리즘 선정 및 분석

상품 편성의 반복 주기를 활용하여 19년 6월 데이터를 기준으로 모델을 생성하였으며, 학습데이터와 예측 데이터의 비연속성을 효과적으로 반영하기 위해 시계열 모델이 아닌 모델을 선정하여 예측에 활용하였습니다.

매출 예측

예측 목표

- ns 홈쇼핑 20년 6월 매출 예측
- 제공 데이터 : 19년 12개월 데이터

상품 편성 반복 주기 활용 예측



- ▶ 월별 평균 매출액
- 월별, 계절별 매출액의 평균값이 다름
- 월 단위의 주기성은 보이지 않음

계절에 민감한 홈쇼핑 산업의 특성상, 1년 단위로 상품 주기성을 발견할 수 있을 것

⇒ 20년 6월과 가장 유사한 편성을 가질 것으로 예상되는 19년 6월을 기준으로 모델 생성

학습 데이터와 예측 데이터의 비연속성 반영



시계열 데이터 분석 방법으로 접근 시, 예측값 이전 5개월 데이터의 공백을 예측값으로 채워서 분석을 진행하게 됨

⇒ 학습/예측 데이터 간 비연속성을 반영할 수 있는 모델 선정

2-3. 데이터 모델링 학습알고리즘 선정 및 분석

데이터의 오차 값을 낮추기 위하여 데이터 셋을 5개 그룹으로 분리하여 분석을 진행하였으며, 각 카테고리에 적합한 변수선정과 하이퍼 파라미터 튜닝을 거쳐 1차 예측값을 도출하였습니다.

매출 예측

모델 활용

1차 모델링

기계학습모델

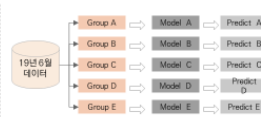
- 데이터의 비선형성을 반영할 수 있으며, 데이터 셋 당 MAPE 값을 가장 낮게 하는 모델을 선정해 1차적으로 매출을 예측한다.

1. 데이터셋 분리 및 모델 선정



주간 매출 장성이 비슷한 상품군들을 묶어 5개의 데이터셋을 생성한 후, 그룹별 mape 가 가장 낮은 모델을 선정한다.

2. 모델 통한 1차 매출 예측(19년 6월)



생성한 모델에 19년 6월 데이터를 대입해 19년 6월 매출을 예측한다.

2-3. 데이터 모델링 학습알고리즘 선정 및 분석

시간대별 매출액 평균값과 세부 상품군별 매출액 평균값을 특정 가중치로 반영하여 실제값과 가장 가까운 값을 도출하도록 세부카테고리별로 다른 가중치를 도출하였습니다.

매출 예측

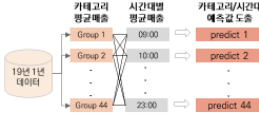
모델 활용

2차 모델링

통계적 예측

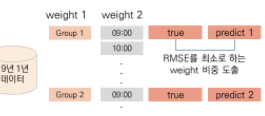
- RMSE값을 가장 낮게 하는 가중치를 선정하여 통계 모델을 생성한다
- 예측 모델과 통계 모델에 가중치를 부여하여 예측&통계 모델을 생성한다

3. 통계 모델 생성



시간대별 매출액 평균값과 상품군 평균값을 반영하여 통계적 모델을 생성하고 한 번도 판매한 적 없는 시간대*상품군 조합에 예측값 추가

4. 통계적 접근 최적 가중치 설정



실제 값과 예측값의 오차가 최소가 되는 상품군과 시간대의 가중치 설정

2-3. 데이터 모델링 학습알고리즘 선정 및 분석

학습알고리즘은 모델의 측면에서는 예측 정확도와 학습 시간을 고려하였고, 데이터의 측면에서는 보유한 데이터의 비선형성을 효과적으로 반영할 수 있는 모델로 선정하였습니다.

매출 예측

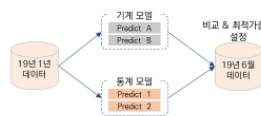
모델 활용

3차 모델링

기계 & 통계 모델

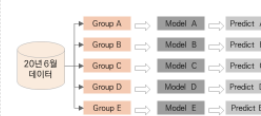
- RMSE값을 가장 낮게 하는 가중치를 예측&통계 모델을 통한 예측값과 모델 예측값에 부여하여 19년 6월 매출액을 예측한다

5. 기계&통계 모델을 통한 2차 매출 예측(19년 6월)



19년 1년 데이터로 구한 통계적 예측값과 모델 예측값이 19년 6월의 데이터와 가장 가까운지는 가중치 도출

6. 기계 & 통계 모델을 통한 20년 6월 매출 예측



19년 6월 편성표에 최적화된 가중치를 기준으로 20년 6월 매출 예측

[machine learning] 홈쇼핑 매출 예측 & 최적 편성 제안

편성 제안

2-3. 데이터 모델링 학습알고리즘 선정 및 분석

통계적 예측의 정확도를 높이기 위해 카테고리별 평균 취급액과 시간대별 평균 취급액의 가중치를 카테고리별로 다르게 적용하였습니다.

매출 예측

통계적 예측

실제 값과 예측값의 RMSE를 최적으로 만드는 가중치 조합 도출

$$Sales = w_1 * SalesPerCategory + w_2 * SalesPerHour$$

* SalesPerCategory : 카테고리별 평균 취급액, * SalesPerHour : 시간대별 평균 취급액 (147개 시간, 21h*7일)

구분	category1	...	44개 카테고리	sales/hour
06:00				
07:00				
...				
147개 시간대 (21h*7일)				
sales/category				

판매기록이 있는
Category와 시간대 조합의 **실제값**

오차 최소화

0부터 1까지 0.1 단위로 w1, w2를
조성해서 얻은 **예측값**

2020 빅콘테스트

38

NS홈쇼핑

2-3. 데이터 모델링 학습알고리즘 선정 및 분석

예측한 매출을 바탕으로 1일 최대 매출 달성을 위한 방송 시간대별 최적의 카테고리를 할가리안 알고리즘을 활용하여 할당하였습니다.

편성 방안 도출

편성 알고리즘

활용 알고리즘	할가리안 알고리즘
알고리즘 특징	할당문제 최적화에 활용하는 알고리즘 - 문제 해결을 위한 최소 비용을 구하는 알고리즘 - 본 프로젝트에서는 예측 데이터를 기반으로 최대값을 산출
편성단위	1주일 주간 편성 (147개 시간대별 추천 상품군 제시)
도출 방법	• 주간 편성과 일간 편성 동시에 진행 • 주간, 일간 편성에서 시간대별 중복 상품 우선 배치 • 중복되지 않는 시간대의 경우, 높은 예상매출액의 상품 우선 배치

2020 빅콘테스트

39

NS홈쇼핑

02. 프로젝트 세부내용 최종 예측 결과

모형을 활용한 예측에서는 앙상블계열 모델과 딥러닝 모델을 최종비교 하였고, 앙상블계열의 CatBoost Regressor가 가장 좋은 성능을 보였습니다. 2020년 6월의 예측 정확도와 오차는 구할 수 없어 2019년 6월의 예측치로 정확도와 오차를 확인하였습니다.

최종 결과

그림별 모델 예측

모든 예측 성능 비교	<ul style="list-style-type: none"> 앙상블 모델과 딥러닝 모델을 비교함 다섯 가지 모델 중 앙상블 모델인 catboost와 lgbm 성능이 가장 좋았음 															
기계 모델 예측 성능	<table> <tr> <th>model_name</th><th>R-squared</th><th>MAPE</th></tr> <tr> <td>XGBRegressor</td><td>0.61</td><td>68.80</td></tr> <tr> <td>LGBMRegressor</td><td>0.68</td><td>57.88</td></tr> <tr> <td>CatBoostRegressor</td><td>0.69</td><td>53.52</td></tr> <tr> <td>DNN</td><td>0.49</td><td>91</td></tr> </table>	model_name	R-squared	MAPE	XGBRegressor	0.61	68.80	LGBMRegressor	0.68	57.88	CatBoostRegressor	0.69	53.52	DNN	0.49	91
model_name	R-squared	MAPE														
XGBRegressor	0.61	68.80														
LGBMRegressor	0.68	57.88														
CatBoostRegressor	0.69	53.52														
DNN	0.49	91														
기계 모델 19년 6월 예측	<table> <tr> <th>model_name</th><th>R-squared</th><th>MAPE</th></tr> <tr> <td>XGBRegressor</td><td>-</td><td>-</td></tr> <tr> <td>LGBMRegressor</td><td rowspan="2">0.73</td><td rowspan="2">80.22</td></tr> <tr> <td>CatBoostRegressor</td></tr> <tr> <td>DNN</td><td>-</td><td>-</td></tr> </table>	model_name	R-squared	MAPE	XGBRegressor	-	-	LGBMRegressor	0.73	80.22	CatBoostRegressor	DNN	-	-		
model_name	R-squared	MAPE														
XGBRegressor	-	-														
LGBMRegressor	0.73	80.22														
CatBoostRegressor																
DNN	-	-														

모델 자체의 성능으로 확인하였을 때, CatBoostRegressor의 R²값이 0.69로 가장 높았으며, MAPE값 또한 53.52로 가장 낮게 분석됨

2019년 6월 데이터의 예측치와 실제 값을 비교하였을 때 기계모델을 종합하여 사용한 모델에서는 R²가 0.73, MAPE가 80.22로 분석됨

2020 빅콘테스트

41

NS홈쇼핑

02. 프로젝트 세부내용 최종 예측 결과

기계학습 모델만으로는 정확한 예측이 어렵다고 판단하여 통계적 예측기를 함께 사용하여 최종 예측을 진행하였습니다.

최종 결과

모델 예측 연계

모든 예측 성능 비교

- 통계적 접근으로 2019년 6월의 매출을 예측함
- 기계학습과 통계적 예측기를 혼합하여 2019년 6월의 매출을 예측한 결과

통계적 예측 결과

model_name	R-squared	MAPE
통계적 예측기	0.51	85.47

기계&통계 결과

model_name	R-squared	MAPE
기계 & 통계 예측기	0.25	52.89

통계적 예측기로 2019년 6월의 예측값을 도출하였고, 실제값과 비교하였을 때, R²값은 0.51, MAPE값은 85.47로 기계학습 모델보다는 다소 성능이 떨어짐

기계학습과 통계적 예측기를 함께 사용한 모델의 결과는 R²값이 0.25, MAPE값은 52.89로 도출되었음

2020 빅콘테스트

43

NS홈쇼핑

End of the document