Yongtaek(Clark) Lim

Data Enthusiast, Planning Specialist

Master of Engineering
Dept. Bigdata Convergence
Korea University

clarklim@korea.ac.kr
http://github.com/clarklim

0. Project

https://github.com/clarklim/data_science_portfolio

본 포트폴리오에 수록한 프로젝트를 제외한
나머지 프로젝트는 모두 **GitHub**에서 확인하실 수 있습니다.

Kaggle Competitions ①

1. Titanic : Machine Learning from Disaster

◯ Data

- train.head() → (891, 11)
- test.head() → (418, 10)
- 컬럼 설명

① Survival : 생존 여부. 0이면 사망, 1이면 생존
② Pclass : 티켓 등급. 1등석(1), 2등석(2), 3등석(3)
③ Sex : 성별. 남자(male)와 여자(female)
④ Age : 나이
⑤ SibSp : 해당 승객과 같이 탑승한 형제/자매 (siblings)와 배우자(spouses)의 총 인원 수
⑥ Parch : 해당 승객과 같이 탑승한 부모(parents)와 자식(children)의 총 인원 수
⑦ Ticket : 티켓 번호
⑧ Fare : 운임 요금
⑨ Cabin : 객실 번호
⑩ Embarked : 선착장. C는 셰르부르(Cherbourg) 프랑스, Q는 퀸스타운 (Queenstown) 영국, S는 사우스햄튼(Southampton) 영국 지역

◯ Environment

- Python 3.73 with `jupyter==1.0.0`
- pandas, numpy, matplotlib, seaborn
- scikit-learn

➢ Perpose

어떤 승객이 생존하며, 또한 어떤 승객이 사망하는지를
예측하는 예측 모델을 만드는 것

➢ Methodology

- 알고리즘 : DecisionTree, RandomForest(Coarse & Fine Search)
- 모델평가 : Accuracy

➢ Solution

- Exploratory data analysis
  - 남자 승객의 생존률은, 18.9% 여성 승객의 생존률은 74.2%
  - Pclass 2등급인 경우 생존률이 1/2(50%), 3등급인 경우 1/4(25%)
  - Cherbourg(C)에서 탑승할수록 생존할 확률이 높으며, Southampton(S)에서 탑승할수록 사망할 확률이 높음
  - 나이가 15세 이하인 승객은 생존 확률이 높음
  - 핵가족(Nuclear)의 생존률이 높고, Single(독신), Big의 생존률이 낮음
- Feature : 등급(Pclass), 성별(Sex_encode), 운임요금(Fare_fillin), 선착장(Embarked), Age(Child 여부), 가족 수(Sibsp+Parch), 이름(Name 내 Master 포함여부)
- Label: 생존 여부(Survived)
- 스코어 : DecisionTree 0.78476, RandomForest 0.81339 예측(상위6% Rank)

2. Bike Sharing Demand

◯ Data

- train.head() → (10886, 12)
- test.head() → (6493, 9)
- 컬럼 설명
  - ① datetime : 시간. 연-월-일 시:분:초
  - ② season : 봄(1), 여름(2), 가을(3), 겨울(4)
  - ③ holiday : 1이면 공휴일, 0이면 공휴일이 아님
  - ④ workingday : 1이면 근무일, 0이면 근무일이 아님
  - ⑤ weather : 1 ~ 4 사이의 값
    1(깨끗한 날씨), 2(약간 안개와 구름이 끼어있는 날씨),
    3(약간 눈, 비가 오거나 천둥), 4(아주 많은 비 또는 우박)
  - ⑥ temp : 온도. 섭씨(Celsius)
  - ⑦ atemp : 체감 온도. 섭씨(Celsius)
  - ⑧ humidity – 습도.
  - ⑨ windspeed – 풍속.
  - ⑩ casual : 비회원(non-registered)의 자전거 대여량
  - ⑪ registered : 회원(registered)의 자전거 대여량.
  - ⑫ count : 총 자전거 대여량으로
    비회원(casual) + 회원(registered)

◯ Environment

- Python 3.73 with `jupyter==1.0.0`
- pandas, numpy, matplotlib, seaborn
- scikit-learn

➢ Perpose

정 시간대에 얼마나 많은 사람들이 자전거를 대여하는지 예측하는 것

➢ Methodology

- 알고리즘 : RandomForest, lightGBM(Coarse & Fine Search)
- 모델평가 : Root Mean Squared Logarithmic Error (RMSLE) $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$
  ※ 정답(aiai, actual)과 예측값(pipi, predict)의 차이가 크면 클수록 페널티를 덜 주는 방식

➢ Solution

- Exploratory data analysis
  - overfitting을 피하기 위해 datetime 컬럼 중 year, hour만 사용
  - 알고리즘 성능향상을 위해 요일(dayofweek) feature 생성하여 학습시킴
    ※ 근무일(workingday) column과 시너지 효과가 난다고 판단
  - seaborn의 distplot을 통해 데이터 왜곡(skewed) 현상 발견
    ✓ 자전거 대여량이 1 ~ 20대인 비중이 굉장히 높음
    ✓ 자전거 대여량이 1,000대에 근접하는 경우도 있음 (977대)
    ■ 해결 Idea : Evaluation에 인사이트를 얻어 자전거 대여량(count) colum을 log transformation → 정규분포화 함으로써 예측 정확도를 높이고자 함
- Feature : "season", "holiday", "workingday", "weather", "temp", "atemp", "humidity", "windspeed", "datetime-year", "datetime-hour", "datetime-dayofweek"
- Label: log transformation한 자전거 대여량(log_count)
- 스코어 : RandomForest 0.51037, LightGBM 0.37502 예측(상위2.9% Rank)

데이터 분석 프로젝트 ①

3. Startup Data analysis (헬스케어 스타트업)

◯ Data

- user_data.head() → (10000, 14)
- 컬럼 설명

① Access Code : 고객식별코드
② Name : 고객 이름
③ Gender : 고객 성별
④ Age : 고객 나이
⑤ Height : 고객 키(cm)
⑥ Initial Weight : 회원 가입 시 몸무게(kg)
⑦ Lowest Weight : app을 이용하는 동안 가장 낮은 몸무게
⑧ Target Weight : 회원 가입 시 설정한 목표 몸무게
⑨ Product Name : 상세 제품명
⑩ Status : 고객의 유료 서비스 결제 현황
⑪ Price : 서비스 구입 가격
⑫ Purchased At : 서비스 구입 시간
⑬ Payment Type : 결제 방식
⑭ Channel : 서비스 구입 경로(구글, 페이스북, 네이버 등)

◯ Environment

- Python 3.73 with `jupyter==1.0.0`
- pandas, numpy, matplotlib

➢ Perpose

데이터 전처리 → 운영, 마케팅, 코칭 팀의 요청사항을 분석한 뒤 분석 결과 전달

- 마케팅 팀 : 데이터 분석 결과를 통해 페이스북 광고 채널의 예산 재조정
- 운영팀 : 데이터 분석 결과를 통해 찾아낸 VIP고객에게 혜택을 제공, 로얄티 제고
- 코칭 팀 : 데이터 분석 결과를 통해 고객의 코칭 만족도를 바탕으로 코칭 방침을 개선

➢ Question & Solution

- 무료 사용자 중, 유료 사용자로 전환할 확률이 가장 높은 연령/성별은 어디인가?
  - 여성 36 ~ 54세 CAC(Customer Acquision Cost)체크 후
    여성 25 ~ 35세와 동일하다면 마케팅 예산을 늘리는 것을 제안
  - 여성 36 ~ 54세 CAC가 상대적으로 높다면, 이 CAC을 낮추는 방안을 수립해야 함
- 무료 사용자들은 주로 어느 조건하에 유료 사용자로 전환하는가?
  - 월-수요일에 광고예산 집중해야 함. 구매를 유도하는 메일, 모바일 노티피케이션을 제안함
  - 페이스북 > 이메일 > 네이버 >기타채널 순으로 구매율이 나타났으며 ,
    구매량 및 결제율을 고려했을 때 네이버 검색채널을 집중적으로 튜닝해야 함
  - 내부에서 트래킹 코드나 데이터 클리닝 코드를 수정, 기타채널을 더 세분화 시켜야 함
- 유료 사용자를 코칭하는 코치 중, 어느 코치가 가장 만족도가 높은가?
  - 결제율↑ & 취소율↓ 코치를 sort하여 운영팀에 명단을 제공
  - 해당 코치의 노하우를 다른 코치들에게 전파하여 고객에게 만족도를 높일 수 있도록 개선요청

데이터 분석 프로젝트 ②

4. Startup Data analysis (프리랜서 오픈마켓 스타트업)

◯ Data

- conversion.head() → (134244, 19)
  - 웹사이트/모바일 서비스에서 활동한 모든 활동 기록(activity)을 포함한 데이터
  - app event category, eventdatetime, devicetype, osversion, params_campaign 등 컬럼으로 구성

- funnel.head() → (53, 6)
  - 사용자가 처음 유입(Acquistion) 되었을 때부터 구매(Revenue)를 할 때 까지의 과정을 구조화 한 데이터

- category.head() → (245, 9)
  - 디자인, 번역, 컨텐츠 제작 등 다양한 상품에 대하여 그룹화하고 정리된 카테고리를 포함한 데이터

◯ Environment

- Python 3.73 with `jupyter==1.0.0`
- pandas, numpy, matplotlib

➢ Perpose

데이터 클린징(Cleansing)

- 데이터를 분석하기 전 가장 중요한 과정인 데이터 클리닝(Data Cleaning)을 맡아 진행함
- 웹, 아이폰, 안드로이드등 다양한 디바이스와 OS로 서비스를 이용하고 있는 사용자의 웹,앱 행동 데이터를 정리한 뒤, pandas 의 merge나 concat등을 활용해 하나로 묶어주는 작업을 진행함

➢ Solution

- Event datetime Column 연/월/일/시/분/초 추출
- App Os version Column Cleansing : IOS, Android 등
- Device manufacturer Column Cleansing : Samsung, LG, Apple 등
- Event channel Column Cleansing : google, naver, daum 등
- App event label Column Cleansing
- conversion-funnel-category Column 삭제, 변경, 정렬, 병합 등

데이터 분석 프로젝트 ③

5. Startup Data analysis (패션 쇼핑몰 추천 스타트업)

◯ Data

- order.head() → (867, 5)
  - 주문이 일어난 로그데이터
  - timestamp(주문시각), user_id(주문을 한 유저ID), goods_id(상품의 id), shop_id(쇼핑몰의 id), price(상품의 가격)
- user.head() → (10000, 3)
  - user_id, os, age
- shop.head() → (200, 4)
  - shop_id, name(고객이름), category (쇼핑몰 분류), age(타겟 연령), style(쇼핑몰 스타일 태그)
- log.head() → (105815, 6)
  - timestamp, event_origin (이벤트 발생한 앱 위치), event_name (발생한 이벤트 명), event_goods_id (이벤트 발생한 상품 고유 식별자). event_shop_id 등

※ 컬럼별 상세명세는 쥬피터 파일을 참고 바랍니다

◯ Environment

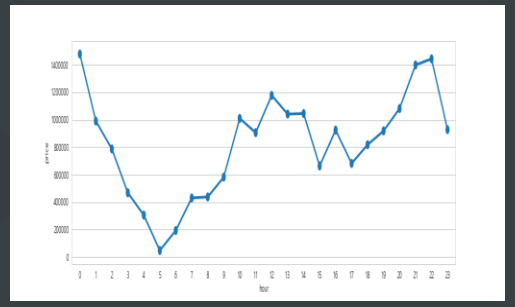
- Python 3.73 with `jupyter==1.0.0`
- pandas, numpy, matplotlib

➢ Perpose

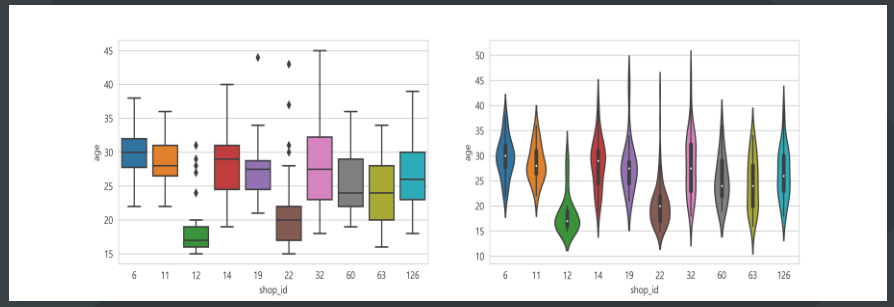
DB 접근(SQL) → 고객 행동데이터 데이터 분석 및 시각화 → 로그 데이터 분석

- 데이터베이스에서 SQL을 통해 고객 정보, 거래 정보, 상품 정보, 그리고 고객의 행동 정보를 로드하여, 데이터 시각화를 통해 데이터를 이해하고 분석하는데 중점을 둠
- 로그 데이터분석의 핵심 수치(page duration, session, 체류 시간)를 구하고 이를 위한 전처리를 진행 후 매출개선을 위한 인사이트를 얻는데 목적을 두고 데이터 분석을 진행

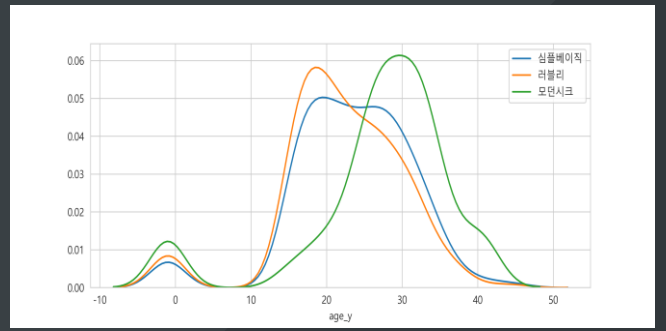
➢ Visualization



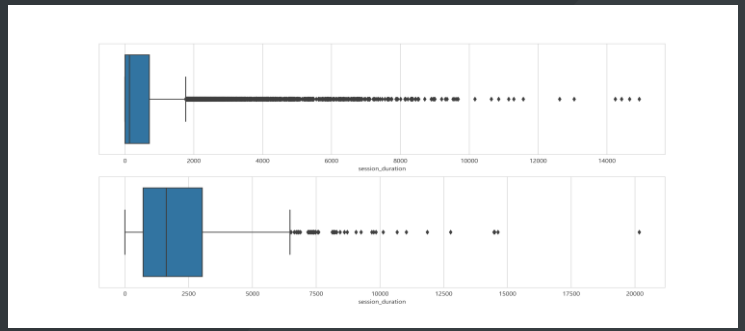
【시간별 Price 구간화(binning)】



【매출 Top 10 쇼핑몰 구매자들의 연령대를 쇼핑몰별로 시각화】



【매출 Top 3 스타일의 구매 연령대 분포 시각화】



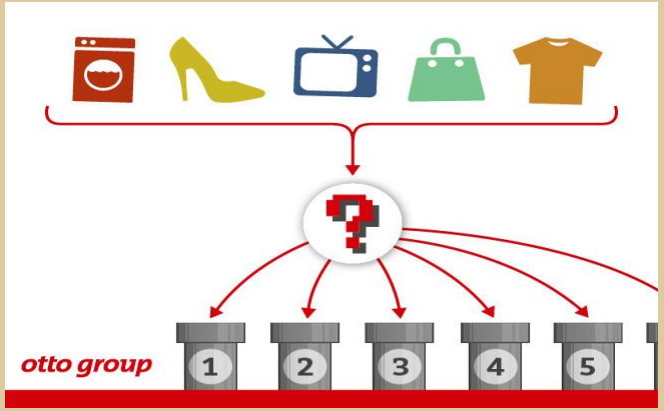
【구매/비구매 session별 평균 체류시간 시각화】

Kaggle Competitions ③

6. Otto group product classification challenge

◯ Data

- train.head() → (61878, 94)
- test.head() → (14368, 93)
- 데이터 설명
  - feature는 1부터 93까지 존재
  - category(target)은 Class1부터 9까지 존재
  - category는 가장 중요한 제품 범주 (패션, 전자제품 등) 중 하나를 나타냄



◯ Environment

- Python 3.73 with `jupyter==1.0.0`
- pandas, numpy, matplotlib, seaborn
- scikit-learn

➢ Perpose

익명화(anonymization)된 상품 정보 데이터를 통해 주어진 상품 카테고리 (target) Class 1~9에 대하여 Classification

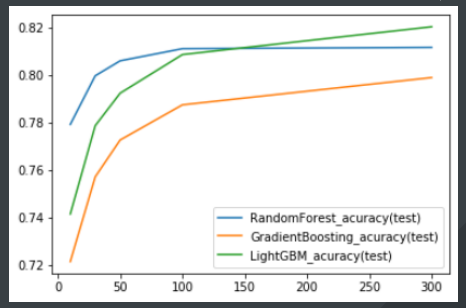
➢ Methodology

- 알고리즘 : RandomForest, GradientBoosting, lightGBM 성능 비교
- 모델평가 : multi-class logarithmic loss
  - N : number of products
  - M : number of class labels

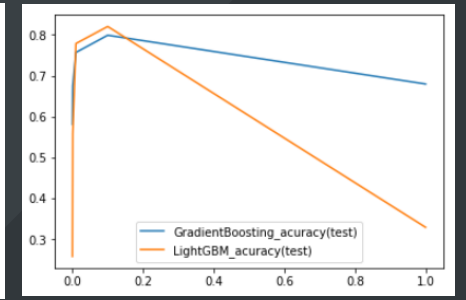
$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

➢ Solution

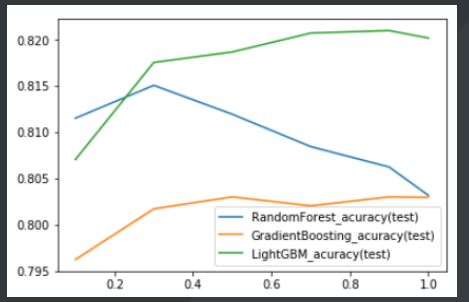
- 머신러닝 모델 성능 비교(Hold-Out Vallidation : test set의 Accuracy 사용)



【x1 = n_estimators】



【x2 = learning rate】



【x3 = max features】

- 스코어 : LightGBM 0.43259 예측(상위9.9% Rank)