NCSOFT®

게임유저 이탈 예측 모델

강 태 형 심 재 헌 조 윤 기 차 호 진

MULLIN

INDEX

CHAPTER 1

주제 및 데이터

리니지 개요

〈리니지〉

리니지는 엔씨소프트가 제작한 중세 판타지를 배경으로 하는 **다중역할수행목적게임(MMORPG)**이다.

한국 최초의 인터넷 기반 온라인 게임으로서 온라인 게임 대중화 시대를 열었다.

지금도 한국을 대표하는 온라인 게임으로서 대중의 사랑을 받고있다.

리니지 게임 구조

〈 리니지의 게임 구조〉

대부분의 컨텐츠는 **PVP와 혈맹전에 집중**되어 있다.

캐릭터 레벨이 높아질 수록 레벨 상승이 어려워 **라이트 유저들의 진입장벽 존재**

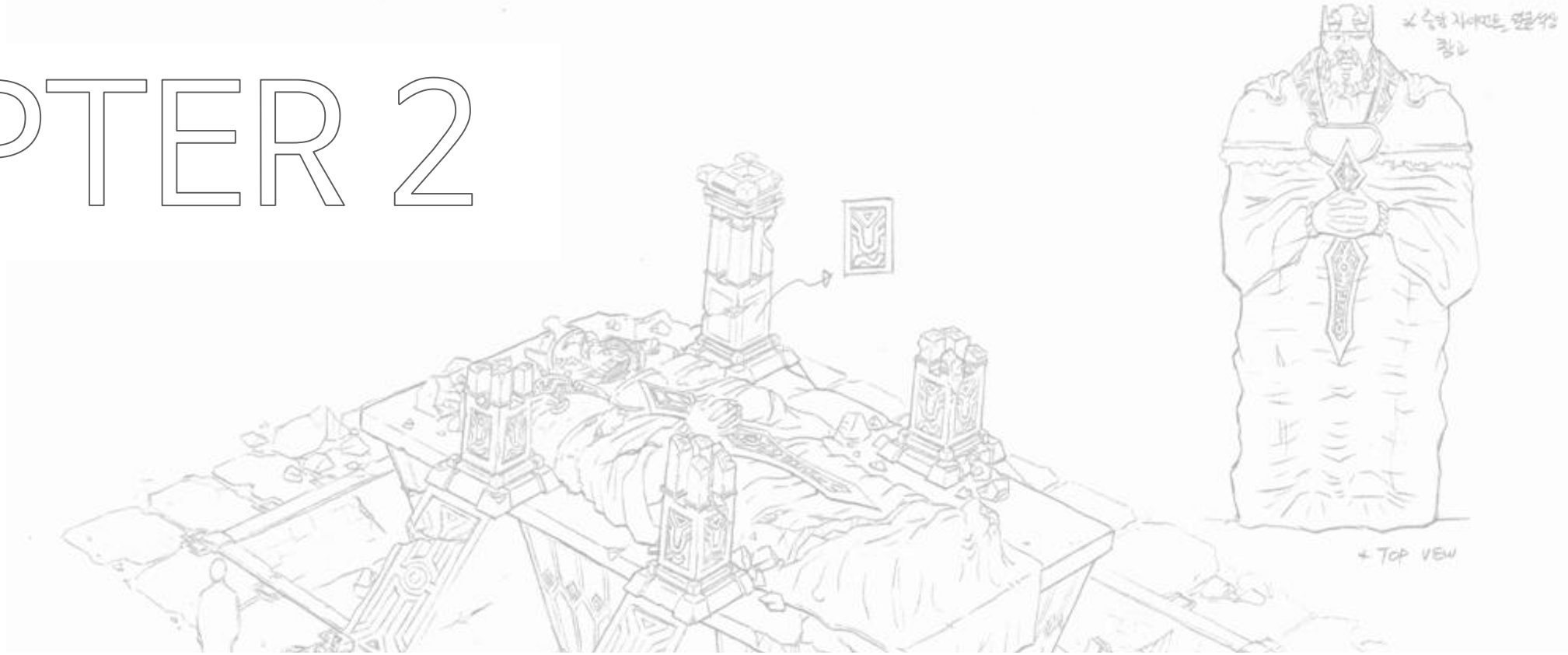
〈 리니지의 구조적 문제〉

상위 유저들의 게임 내 횡포

아이템 획득을 위한 과금문제

"게임 안해!"

CHAPTER 2



EDA

EDA

〈목적〉

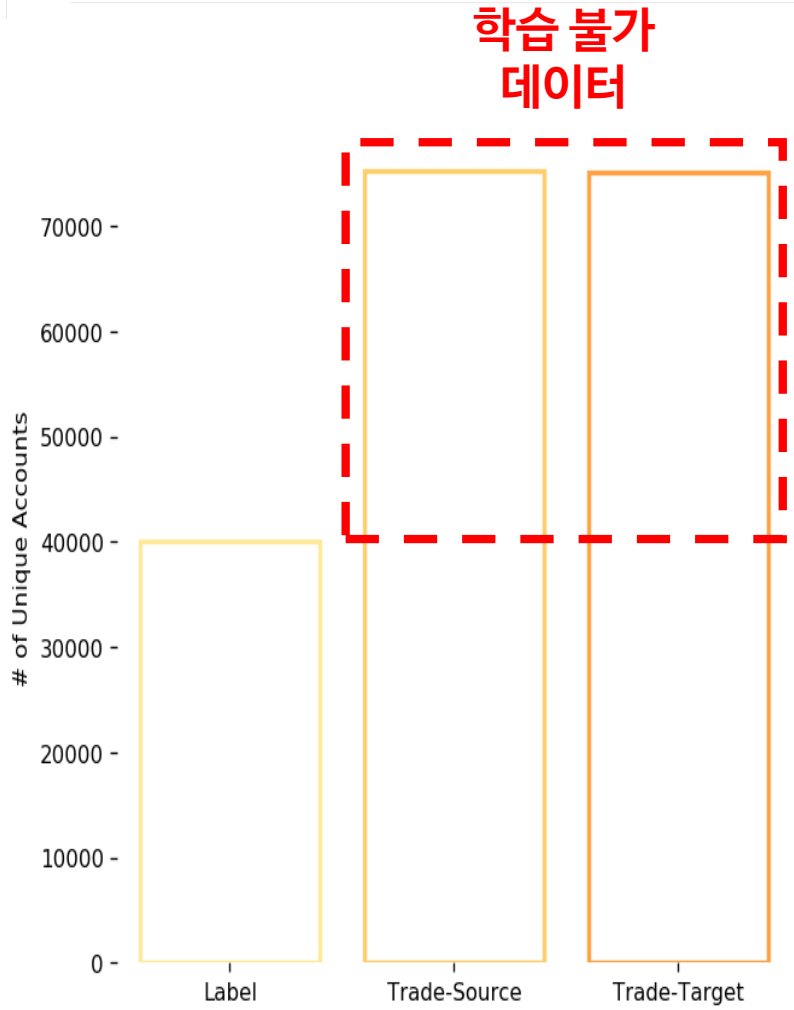
EDA를 통해 전반적인 **데이터 분석 과정에 대한 계획 수립**

〈Brainstorming〉

Activity, Combat, Payment, Pledge, Trade, Label의 6개 Train Dataset이 존재
Target Value를 가지고 있는 Label Data를 제외한 나머지는 Input Data이므로 모두 병합하여 하나의 Train Dataset으로 만들 필요가 있음

Activity	Combat	Payment	Pledge	Trade
캐릭터 일일 주요 활동. 게임 활동에 관한 전반적인 정보.	전투와 관련된 정보	일별로 결제한 금액에 대한 정보	캐릭터가 소속된 혈맹과 관련된 각종 정보	유저간 거래 기록에 대한 정보

EDA

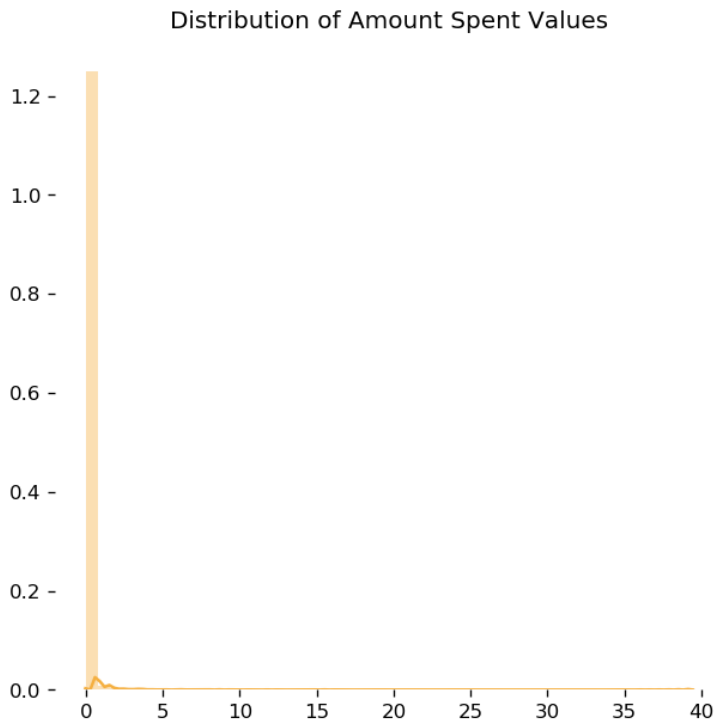


〈목적〉

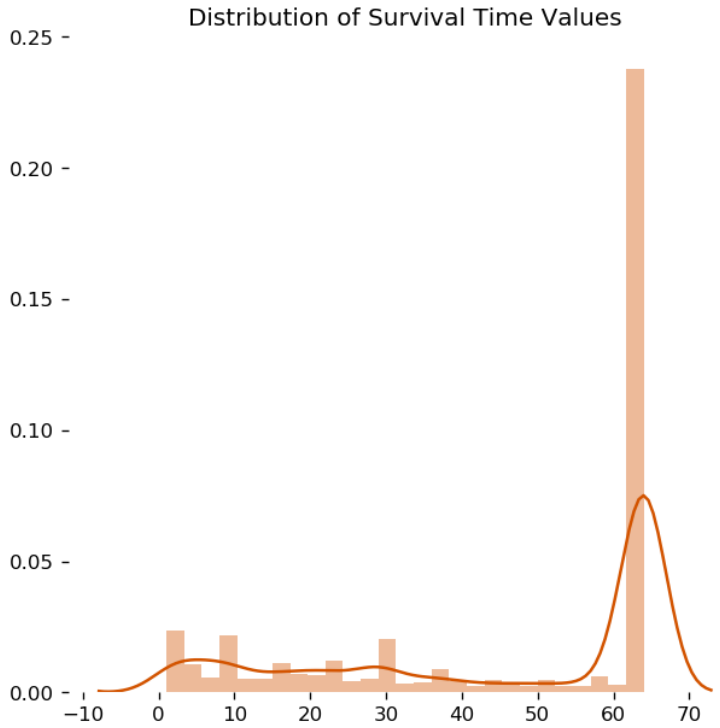
Label Data에 존재하지 않는 계정에 대해서는 어떠한 **학습을 할 수가 없으므로** 해당 **계정에 대한 정보는 삭제**

Trade에서 그러한 계정이 약 3만개 정도 발견됨

데이터 분포



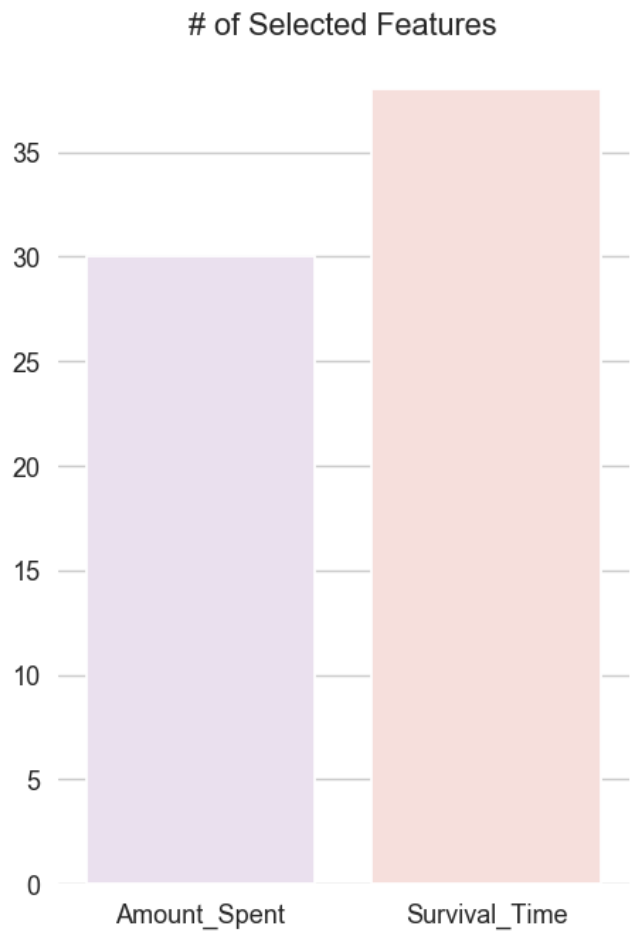
결제금액 분포



생존기간 분포

두 데이터 모두 값이 **편포**되어 있음

상관분석



〈목적〉

주요 Feature들을 추출

결제 금액과 생존기간을 분리하여 각각 상관분석을 진행

〈기준〉

결제금액의 **상관계수가 0.3보다 높은 Feature만 추출**

생존기간의 **상관계수가 0.1보다 높은 Feautre만 추출**

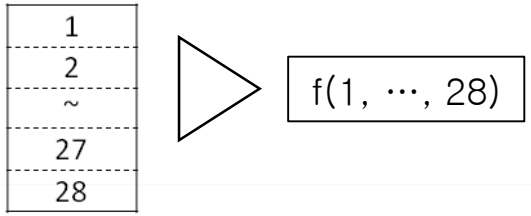
결론

〈EDA 결론〉

일반적인 상식에 비추어 봤을 때 필요하지 않은 데이터() 삭제

RNN 모델에 넣기 위해 데이터의 날짜별 정보를 유지할 지, 아니면 날짜별 정보를 모두 더하거나 평균을 내서 계정별 정보만 가지고 분석을 진행할지에 대한 고민

-〉 이탈 예측 모델의 경우 시간의 흐름에 따른 유저의 패턴 변화에 큰 영향을 받을 것이라 판단하여 RNN을 사용하기로 함



CHAPTER 3

데이터 전처리
파생변수 생성

PRIMARY KEY

⟨label⟩

⟨payment⟩

⟨trade⟩

{acc_id}

⟨pledge⟩

⟨combat⟩

⟨activity⟩

⟨Common Features⟩

acc_id : 각각의 계정

char_id: 계정 별 보유하고 있는 캐릭터의 수

⟨Primary Key⟩

제공받은 데이터셋을 **acc_id를 기준으로 결합**

파생변수 생성 〈activity〉

변수 이름	변수 설명
activity_playtime	일일 플레이시간
activity_quest_exp	퀘스트 획득 경험치
activity_solo_exp	솔로 획득 경험치
activity_party_exp	파티 획득 경험치

〈경험치〉

qexp_per_playtime : 플레이 시간당 퀘스트 경험치

sexp_per_playtime : 플레이 시간당 solo 경험치

pexp_per_playtime : 플레이 시간당 party 경험치

▶ **플레이 시간에 따른 경험치 획득**을 통해 **'액티브 유저'** 특성 파악

〈활동 이력〉

activity_logged_in : 해당 일 해당 계정당 활동 유무

▶ **해당 계정의 활동 이력**을 통해 **'액티브 유저'** 특성 파악

〈결제 이력〉

payment_logged_in : 해당 일 해당 계정당 결제 유무

▶ **해당 계정의 결제 이력**을 통해 **'액티브 유저'** 특성 파악

파생변수 생성 〈payment〉

변수 이름	변수 설명
Payment_day	날짜
Payment_acc_id	유저 아이디
Payment_amount_spent	결제 금액

〈결제액〉

min_spent : 해당 일 해당 계정 기준 가장 낮은 결제액

▶ **최소 결제 금액**을 통해 **'라이트 / 헤비 유저'** 판단

mean_spent : 해당 일 해당 계정 기준 평균 결제액

▶ **평균 결제 금액**을 통해 **'라이트 / 헤비 유저'** 판단

tot_spent : 해당 일 해당 계정 기준 총 결제액

▶ **총 결제 금액**을 통해 **직관적 비교**

Median_spent : 해당 일 해당 계정 기준 중간값 결제액

▶ **결제 금액의 중간값**을 통해 **기본적인 과금 단위** 도출

Max_spent : 해당 일 해당 계정 기준 가장 높은 결제액

▶ **최대 결제 금액**을 통해 **'라이트 / 헤비 유저'** 판단

파생변수 생성 〈trade〉

변수이름	변수설명
Day	거래발생일
source_acc_id	판매 유저 아이디
target_acc_id	구매 유저 아이디
item_amount	거래 아이템 수량
item_price	거래 가격

〈거래 규모_개수〉

sell_item_amount : 해당 일 해당 계정의 총 판매 아이템 개수

get_item_amount : 해당 일 해당 계정의 총 구매 아이템 개수

▶ **계정당 거래 아이템 개수를** 을 통해 **유저 활성화** 판단

〈거래 규모_가격〉

sell_item_price : 해당 일 해당 계정의 총 판매 아이템 값

get_item_price : 해당 일 해당 계정의 총 구매 아이템 값

▶ **거래 아이템 값**을 통해 **거래활동의 실질적인 가치** 산출

* 거래 아이템 값은 게임 내 통화단위인 아데나로 환산

파생변수 생성 〈trade〉

변수이름	변수설명
day	거래 발생 일
Trade_time_bin	거래 발생 시간 (00:00:00 ~ 23:59:59)
type	거래 구분 (교환창 = 1, 개인상점 = 0)

〈거래 규모_횟수〉

trade_time_bin_0 : 해당 일 해당 계정의 0 - 6시의 거래 횟수

trade_time_bin_1 : 해당 일 해당 계정의 6 - 12시의 거래 횟수

trade_time_bin_2 : 해당 일 해당 계정의 12 - 18시의 거래 횟수

trade_time_bin_3 : 해당 일 해당 계정의 18 - 24시의 거래 횟수

▶ 접속하기 힘든 늦은 시간에 접속하는 유저들은 **'액티브 유저'일 가능성 이 높음**

count_sell : 해당 일 해당 계정 기준 평균 거래의 결제 횟수

Count_get : 해당 일 해당 계정 기준 평균 거래의 결제 횟수

Total_trade_count : 해당 일 해당 계정 기준 평균 거래의 결제 횟수

▶ 유저의 거래 횟수가 높을 수록 '액티브 유저'일 가능성이 높음

파생변수 생성 〈trade〉

변수이름	변수설명
day	거래 발생 일
Trade_time_bin	거래 발생 시간 (00:00:00 ~ 23:59:59)
type	거래 구분 (교환창 = 1, 개인상점 = 0)

〈거래 총체〉

tot_trade_amount : 거래한 아이템의 총량(아이템 총 물동량)

tot_get_money : 총 아데나 이익량

▶ **유저의 거래 이익량**을 통해 **이익률에 민감**한 **'액티브 유저'** 파악

trade_logged_in : 해당 일 해당 계정 거래 유무

▶ **해당일 거래 유무**는 **'액티브 유저'** 판단의 주요 척도

〈거래 채널〉

trade_type_0 : 자유시장 내에서 거래 횟수

trade_type_1 : 교환창 내에서 거래 횟수

▶ **거래 채널을 세분화**하여 거래 시스템 선호에 따른 **'유저 활성화'** 확인

파생변수 생성 〈pledge〉

변수이름	변수설명
Day	날짜
acc_id	유저 아이디
same_pledge_cnt	동일 혈맹 전투 횟수의 합
playtime	일일 플레이시간

〈혈맹 활성화〉

avg_play_rate_per_pledge : 혈맹당 평균 접속률

total_combat_cnt_per_pledge : 혈맹당 전체 전투횟수

pledge_num_people : 해당일 혈맹 관련 활동자 수

pledge_logged_in : 각각의 계정

▶ **혈맹은 강력한 게임 내 조직 문화**로서, 해당 **혈맹의 활성화 정도**는 **유저의 잔존가치를 산출**하는 중요 변수

파생변수 생성 〈combat〉

변수이름	변수설명
Day	날짜
Acc_id	유저 아이디
class	캐릭터 직업
temp_cnt	단발성 전투 횟수
etc_cnt	기타 전투 횟수
pledge_cnt	혈맹 전투 횟수

〈전투 횟수〉

combat_count : 해당 일 해당 계정의 전투 횟수

combat_logged_in : 해당 일 해당 계정의 전투 활동 유무

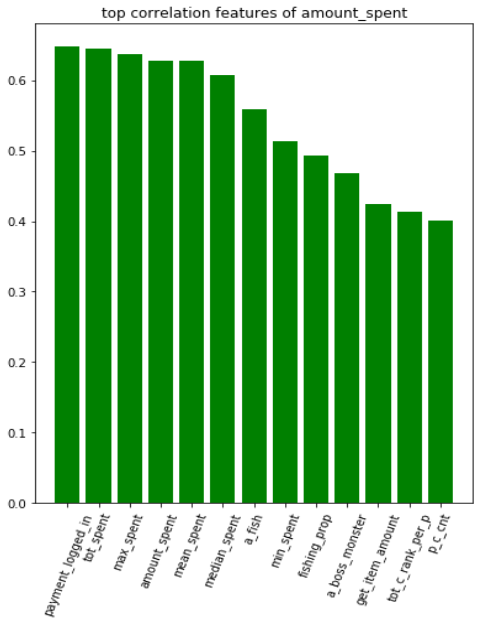
▶ 리니지 게임 특성상, 전투 유형의 관계 없이 전투 횟수의 총합은 액티브 유저를 판단하는 척도로 작용

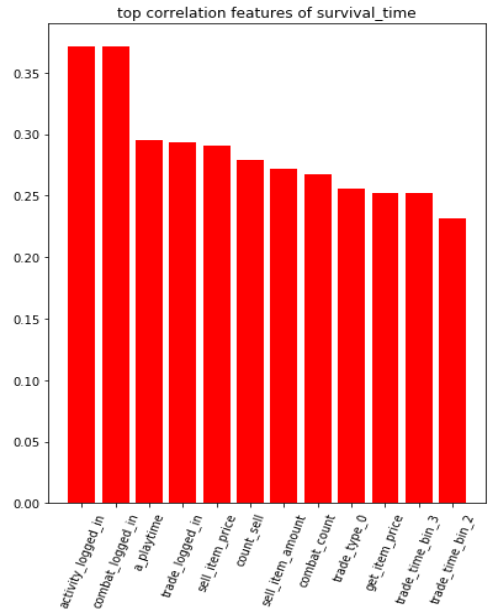
〈클래스〉

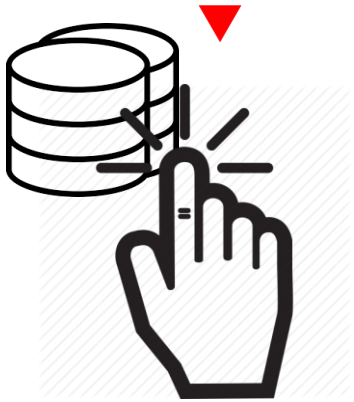
isMajorClass : 해당 일 해당 계정의 주류 클래스(기사,요정, 마법사,전사) 유무

▶ 커뮤니티 분석 결과, 주류 클래스 해당 유무는 총체적인 게임의 흥미도에 영향을 주어 '액티브 유저' 판단의 척도로 작용

파생변수에 대한 상관분석







〈목적〉

상관분석을 통해 target변수에 영향을 끼치는 주요한 feature 선택

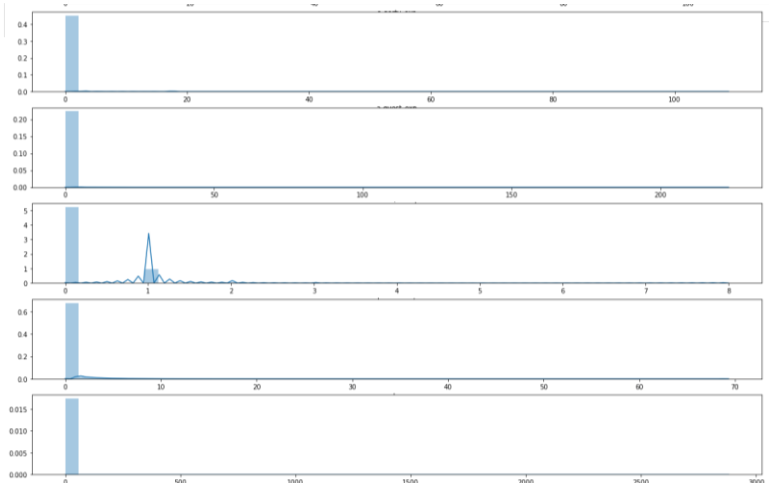
〈결과: amount_spent〉

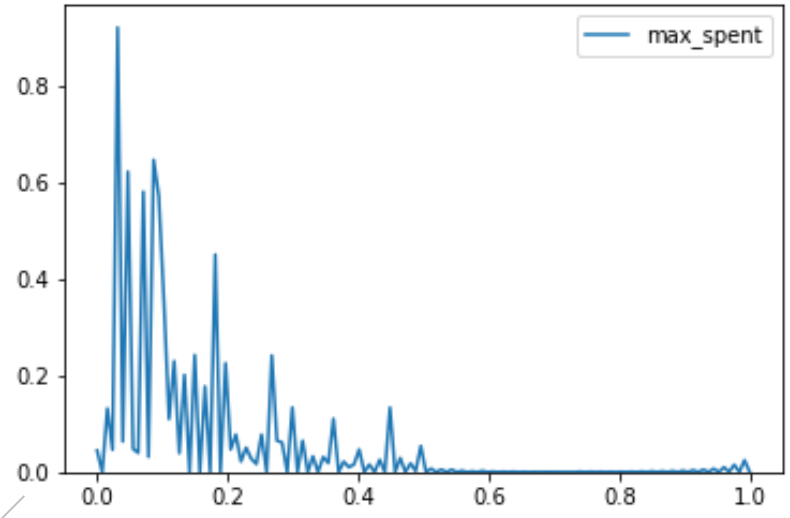
payment_logged_in
tot_spent
max_spent
amount_spent
(이하 생략)

〈결과: survival_time〉

상관분석을 통해 target변수에 영향을 끼치는 주요한 feature 선택
activity_logged_in
combat_logged_in
a_playtime
trade_logged_in
(이하 생략)

데이터 표준화





〈Standarad Scaling〉

$$\frac{x - \mu}{\sigma}$$

〈Min-Max Scaling〉

$$\frac{x - min}{\max - min}$$

〈목적(1)_ 평활화〉

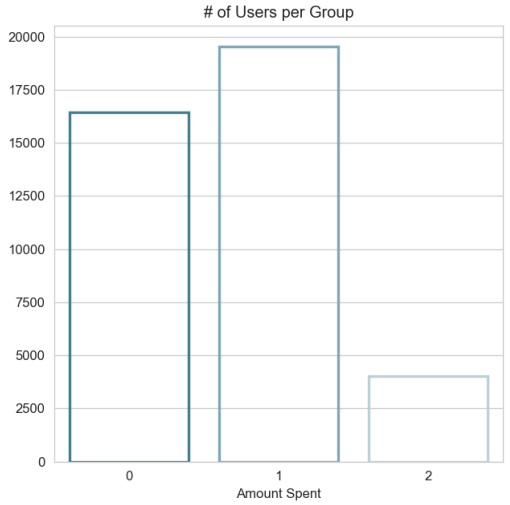
Feature의 특성을 반영하기 위하여 **편중된 데이터의 분포를 평활화**

〈목적(2)_ 성능〉

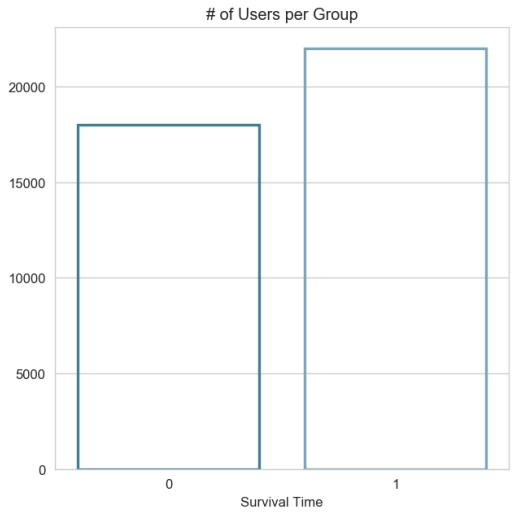
평활화된 데이터를 **신경망에 적합한 데이터**로 사용하기 위하여
0~1의 값을 갖게하는 Min-Max Scaler 활용

데이터 구간화(1)

결제금액	
0	라이트 유저
1	
2	헤비 유저



생존기간	
0	이탈
1	생존



〈목적〉

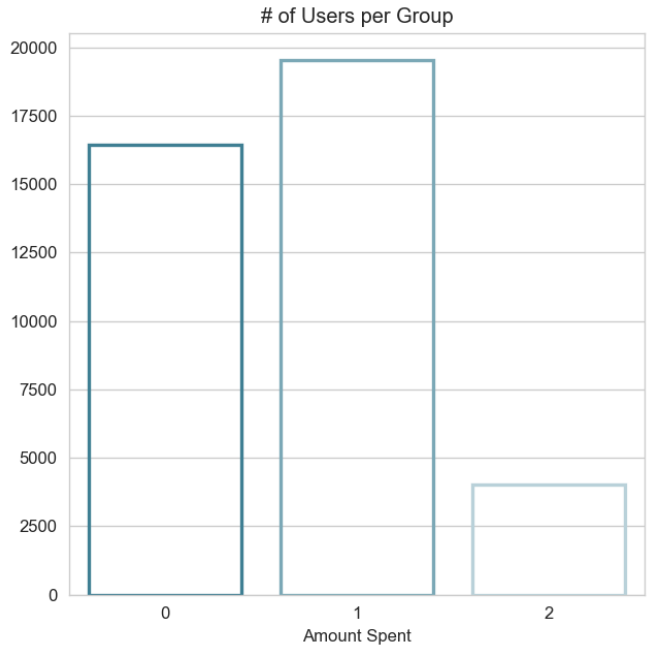
KNN-Clustering 분석을 위한 전처리

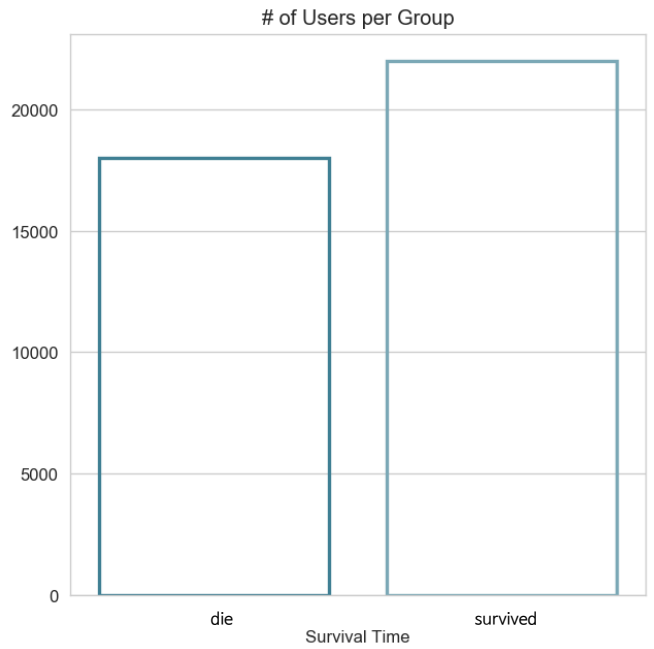
〈기준〉

결제금액: 금액에 따라 3단계로 구간화

생존기간: 64일 접속(생존) 여부

데이터 구간화(2)





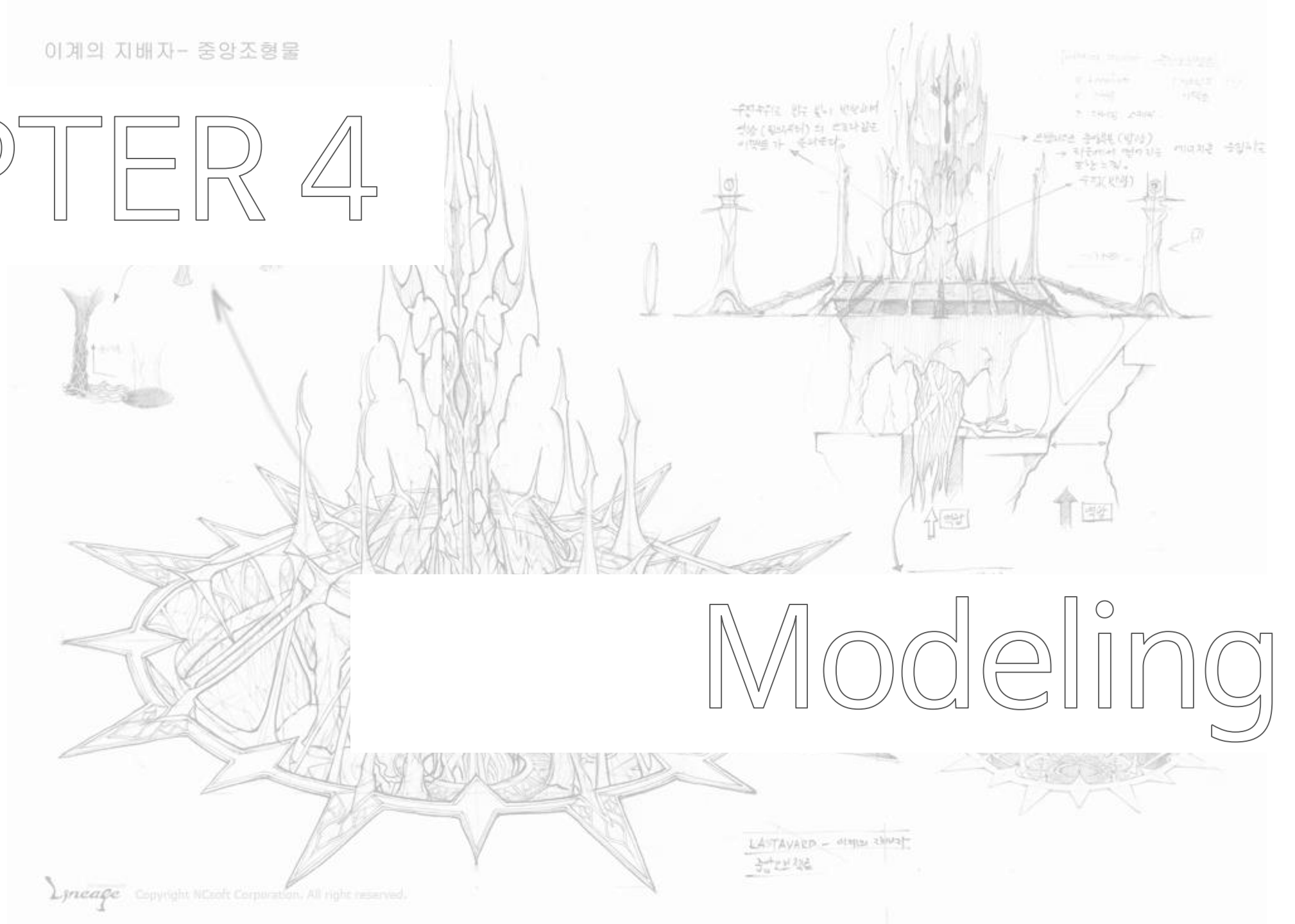
〈목적〉

KNN-Clustering 분석을 위한 전처리

〈기준〉

결제금액: 금액에 따라 3단계로 구간화

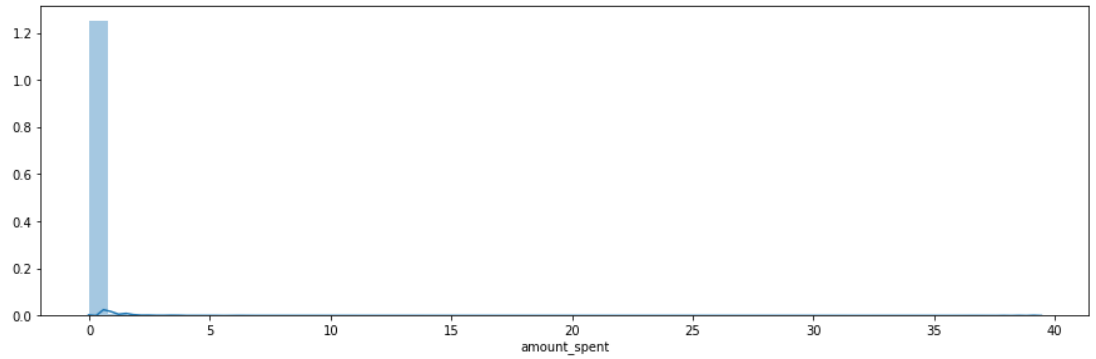
생존기간: 64일 접속(생존) 여부

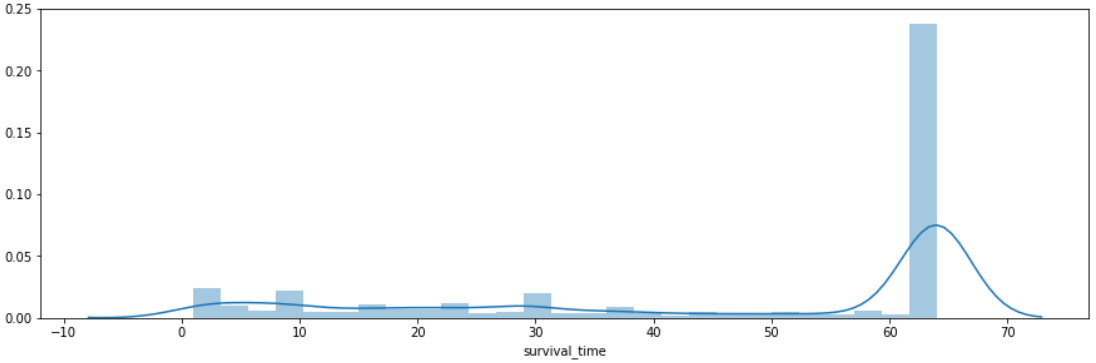


CHAPTER 4

Modeling

데이터 준비





〈목적〉

train_label의 amount_spent는 대부분의 값이 0에 분포
train_label의 survival_time은 대부분 값이 64에 분포

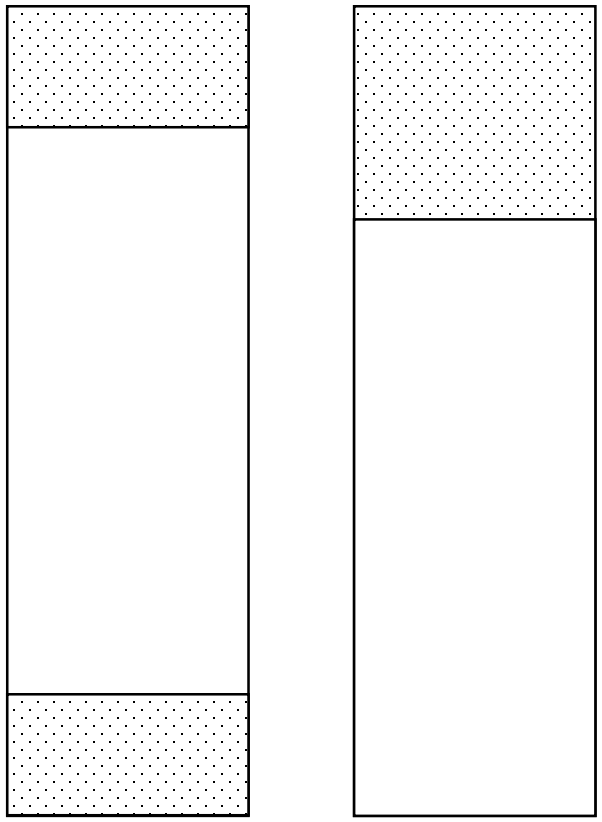
▶ 0과 64를 제외한 label은 예측 정확도가 낮음

〈결론〉

amount_spent(0), survival_time(64)로 분류되지 않을 데이터 선별

▶ 선별된 데이터를 분리하여 학습하여 정확도 상승

train_label 구간화



{amount} {survival}

⟨amo_group⟩

amount_spent의 값이 0

▶ 무과금 유저(0)

amount_spent의 값이 상위 10% 미만

▶ 라이트 유저(1)

amount_spent의 값이 상위 10% 이상

▶ 헤비 유저(2)

⟨sur_group⟩

survival_time의 값이 64일 미만 ▶ 이탈 (0)

survival_time의 값이 64일 이상 ▶ 생존 (1)

test_dataset 분류(1)_K-NN 알고리즘

〈목표〉

구간화된 Target을 예측 =〉 이후 잔존가치 산출

〈과정〉

K-NN알고리즘을 활용하여 train_dataset에 대해 학습

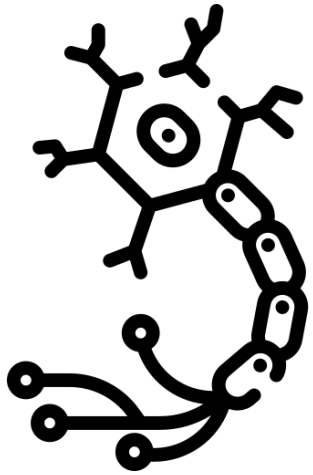
〈결과: amount_spent〉

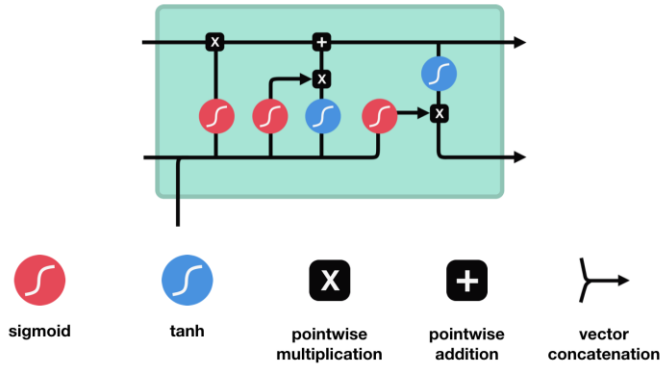
test_amo0 : 분류된 무과금 유저 그룹
test_amo1 : 분류된 라이트 유저 그룹
test_amo2 : 분류된 헤비유저 그룹

〈결과: survival_time〉

test_sur0 : 분류된 이탈 유저 그룹
test_sur1 : 분류된 생존 유저 그룹

test_dataset 분류(2)_RNN(with LSTM)

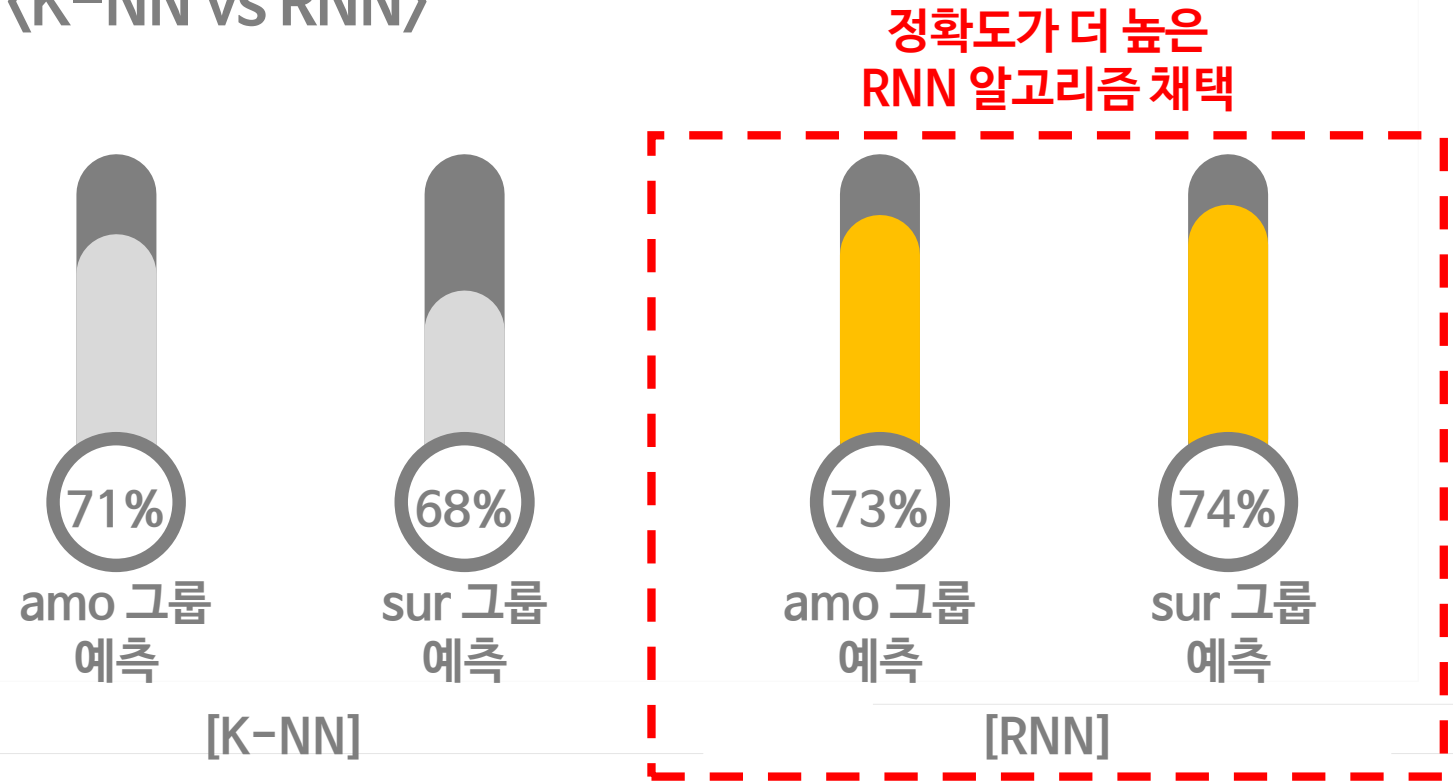




〈과정〉

LSTM을 활용한 인공신경망을 train_dataset에 대해 학습

〈K-NN vs RNN〉



참고자료

템플릿 〉 https://adstorepost.com
이미지 〉 https://lineage.plaync.com
아이콘 〉 https://www.flaticon.com

핸즈온 머신러닝
모두의 딥러닝
케라스 창시자에게 배우는 딥러닝