

# 인터넷 고객 이탈 분석 및 예측

---



SKN20-2nd-3TEAM

BUSINESS PROPOSAL

# | 목 차

Chapter 1. Team Member 소개

Chapter 2. 데이터 전처리 & Feature Engineering

Chapter 3. 인터넷 고객 이탈(Churn) 분석 및 예측

Chapter 4. Modeling & Evaluation

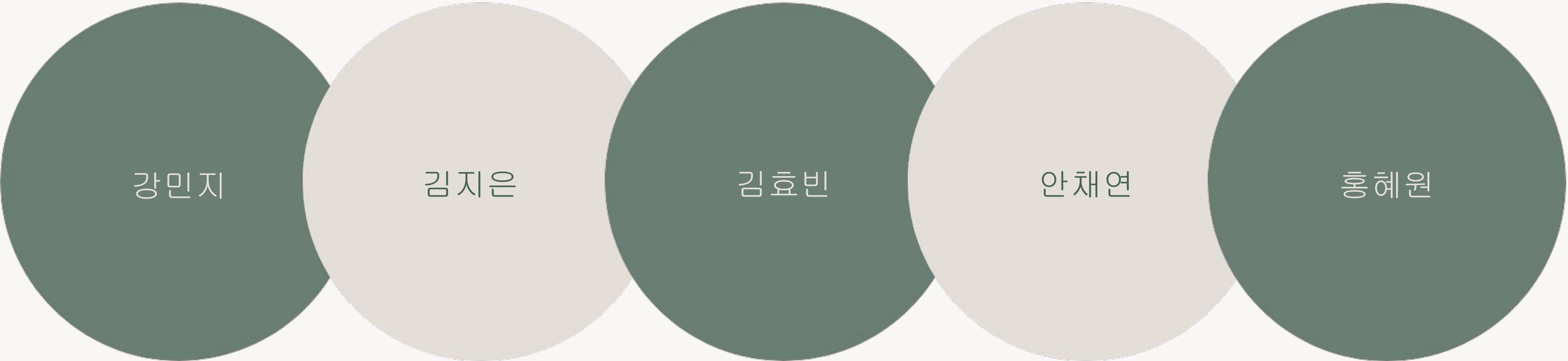
Chapter 5. 성능 지표 비교

# Chapter 1.

Team Member 소개

 SKN20-2nd-3TEAM

SKN 20기\_3TEAM



# Chapter 2.

## 분석배경

- 최근 전 세계적으로 인터넷 서비스 제공업체(ISP) 간 경쟁이 치열
- 신규 고객 확보보다 기존 고객 유지(Retention) 가 매출 성장과 장기적 수익성에 더 큰 영향
- 서비스 품질 저하, 요금 불만, 약정 만료 등으로 고객이 해지하는 현상을 “Churn(이탈)”이라 함
- 통신사는 이탈 징후를 조기에 파악하고 대응하는 것이 핵심

## 기대 효과

- 이탈 고객 사전 식별 및 맞춤형 전략
  - 예측 모델을 통해 이탈 가능성이 높은 고객을 미리 파악하고, 맞춤형 유지 전략을 수립할 수 있음.
- 비즈니스 전략 개선
  - 고객 행동 및 이탈 요인 분석을 통해 마케팅, 서비스, 제품 개선 등 전략적 의사결정에 활용 가능.

# Chapter 2.

기존 컬럼 수

11 개

기존 행 수

72,274 건

컬럼명	설명
id	userID
is_tv_subscriber	TV 구독 여부 [1:구독, 0:미구독]
is_movie_package_subscriber	영화 패키지(시네마) 구독 여부 [1:구독, 0:미구독]
subscription_age	서비스 이용 기간(년 단위), 고객이 서비스를 이용한 총 연수
bill_avg	최근 3개월 평균 청구 금액 (단위: \$)
remaining_contract	남은 계약 기간(년 단위), null이면 계약 없음. 계약 중 고객은 중도 해지 시 위약금 발생
service_failure_count	최근 3개월 동안 고객센터에 신고한 서비스 장애 건수 / 서비스 품질 관련 지표
download_avg	최근 3개월 평균 다운로드 사용량(GB) / 인터넷 사용량 지표
upload_avg	최근 3개월 평균 업로드 사용량(GB) / 인터넷 업로드 사용량
download_over_limit	지난 9개월 동안 다운로드 한도 초과 횟수, 한도 초과 시 추가 요금 발생
churn	이탈 여부(target) [1:서비스 해지, 0:서비스 유지]

# Chapter 2.

데이터 전처리 & Feature Engineering

## 결측치 확인

- Reaming\_contract : null -> ‘ 계약 없음 ‘
- Download\_avg 의 결측치 행 = upload\_avg 의 결측치 행

	0
id	0
is_tv_subscriber	0
is_movie_package_subscriber	0
subscription_age	0
bill_avg	0
reamining_contract	21572
service_failure_count	0
download_avg	381
upload_avg	381
download_over_limit	0
churn	0
dtype:	int64

# Chapter 2.

## 전처리 데이터셋 3개

트리기반 vs 비트리기반 모델의 데이터 처리 전략

구분	트리 기반 모델 (Tree-based Models)	비트리 기반 모델 (Non-tree Models)
주요 모델	Decision Tree, Random Forest, XGBoost	Logistic Regression, SVM
데이터셋 명	df_tree	df_re , df_re_log
범주형 변수 처리	Label Encoding (범주를 순서형 정수로 변환)	One-hot Encoding (더미 변수 생성)
연속형 변수	구간화(binning) 적용 ( subscription_age_group )	원본 값 유지 ( subscription_age )
로그 변환	불필요 (트리모델은 분포 형태 영향 적음)	bill_avg , download_avg , upload_avg 에 적용
목적	규칙 기반 의사결정 구조 파악	연속적 확률 예측 및 선형 분리 성능 향상

# Chapter 2.

## EDA/트리 기반 모델용 데이터셋 1개

•1. 불필요한 컬럼 제거 / 이상치 및 결측치 처리

id 삭제, subscription\_age = -0.02 제거, download\_avg & upload\_avg 결측치 제거

• 2. 파생 변수 생성 후 라벨링

• 계약 유형 (contract\_type)

조건	새로운 라벨	의미
<code>remaining_contract is null</code>	<code>no_contract</code>	무약정 (자유이용 고객)
<code>remaining_contract == 0</code>	<code>expired</code>	계약 종료 고객
<code>remaining_contract &gt; 0</code>	<code>active</code>	약정 유지 중인 고객

• 구독 연수 구간화 (subscription\_age\_group)

구간	라벨	고객군 정의
$0 \leq x < 1$	<code>0~1년</code>	신규 또는 초기 고객
$1 \leq x \leq 3$	<code>1~3년</code>	약정 중기 고객
$3 < x \leq 5$	<code>3~5년</code>	재계약 고객
$x > 5$	<code>5년 초과</code>	충성 고객

• 구독 유형 통합 (subscription\_label)

TV 구독	영화 구독	통합 라벨 ( <code>subscription_label</code> )	의미
0	0	<code>none</code>	구독 없음
1	0	<code>tv</code>	TV만 구독
0	1	<code>movie</code>	영화만 구독
1	1	<code>both</code>	둘 다 구독

# Chapter 2.

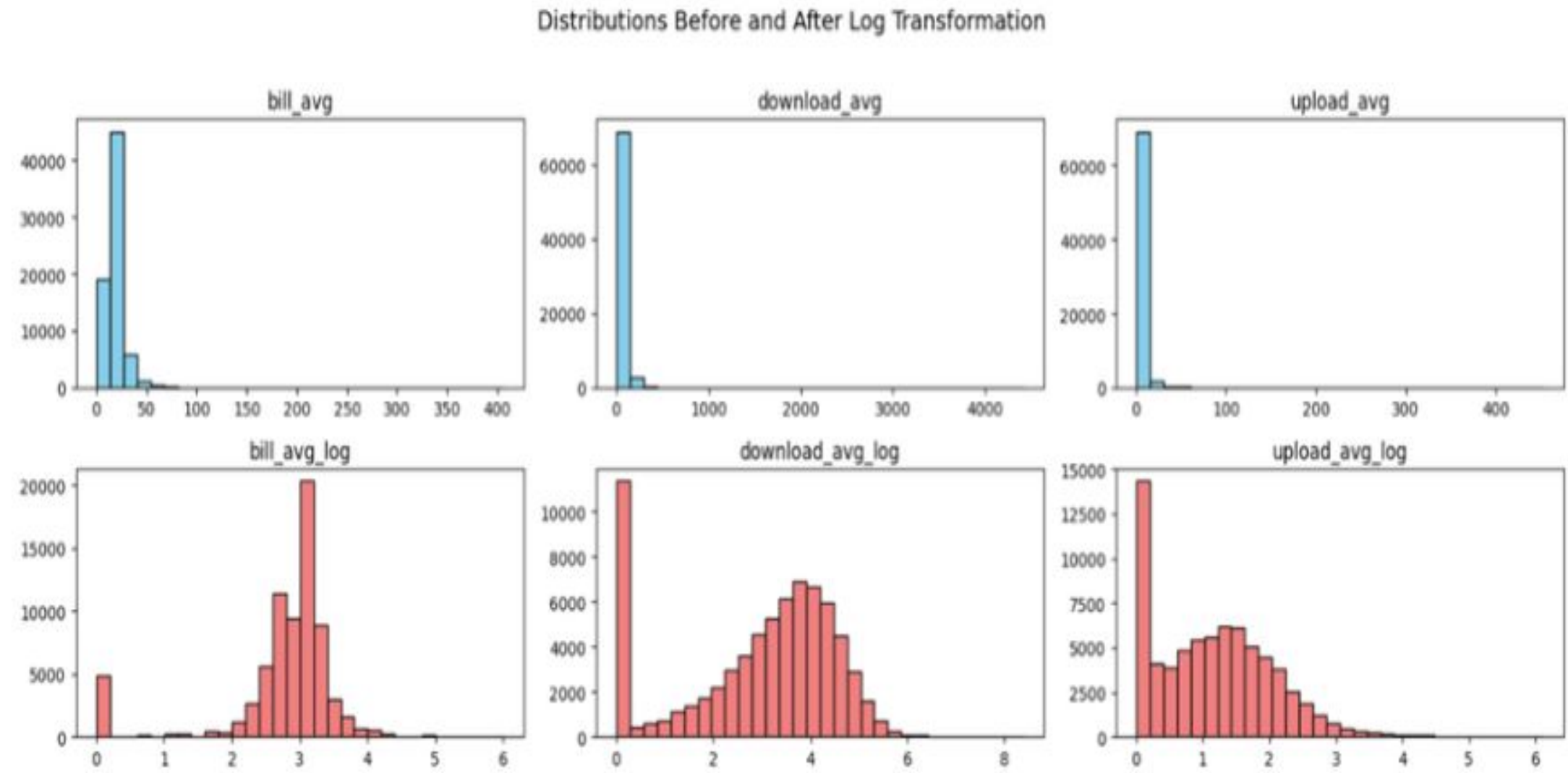
## 회귀 및 비트리 기반 모델용 데이터셋 2개

- 1. 불필요한 행 제거 / 이상치 및 결측치 처리
  - 2. 범주형 변수 원-핫 인코딩(계약 유형, 구독 유형)
- 3. 로그 변환 (평균 청구금액 / 다운로드량 / 업로드량)

원본 컬럼	인코딩 후 컬럼	처리 목적
contract_type	contract_no_contract , contract_expired , contract_active	계약 상태별 독립 변수 생성
subscription_label	sub_none , sub_tv , sub_movie , sub_both	구독 조합별 영향 학습 가능

- 3. 연속형 변수 유지(구독 연수)

컬럼	설명	처리 방식
subscription_age	서비스 이용 기간(년 단위)	원본 값 유지



# Chapter 2.

항목	값
데이터 크기	(71892, 9)
주요 컬럼	bill_avg , service_failure_count , download_avg , upload_avg , download_over_limit , churn , contract_type , subscription_age_group , subscription_label
타겟 컬럼	churn (1=이탈,0=유지)

항목	df_ml_raw	df_ml_log
데이터 크기	(71892, 14)	(71892, 14)
주요 컬럼	bill_avg , download_avg , upload_avg , subscription_age , service_failure_count , download_over_limit , churn	동일
추가 더미 컬럼	contract_no_contract , contract_expired , contract_active , sub_none , sub_tv , sub_movie , sub_both (총 7개)	동일
로그 변환 컬럼	없음	bill_avg_log , download_avg_log , upload_avg_log
타겟 컬럼	churn (1=이탈, 0=유지)	동일

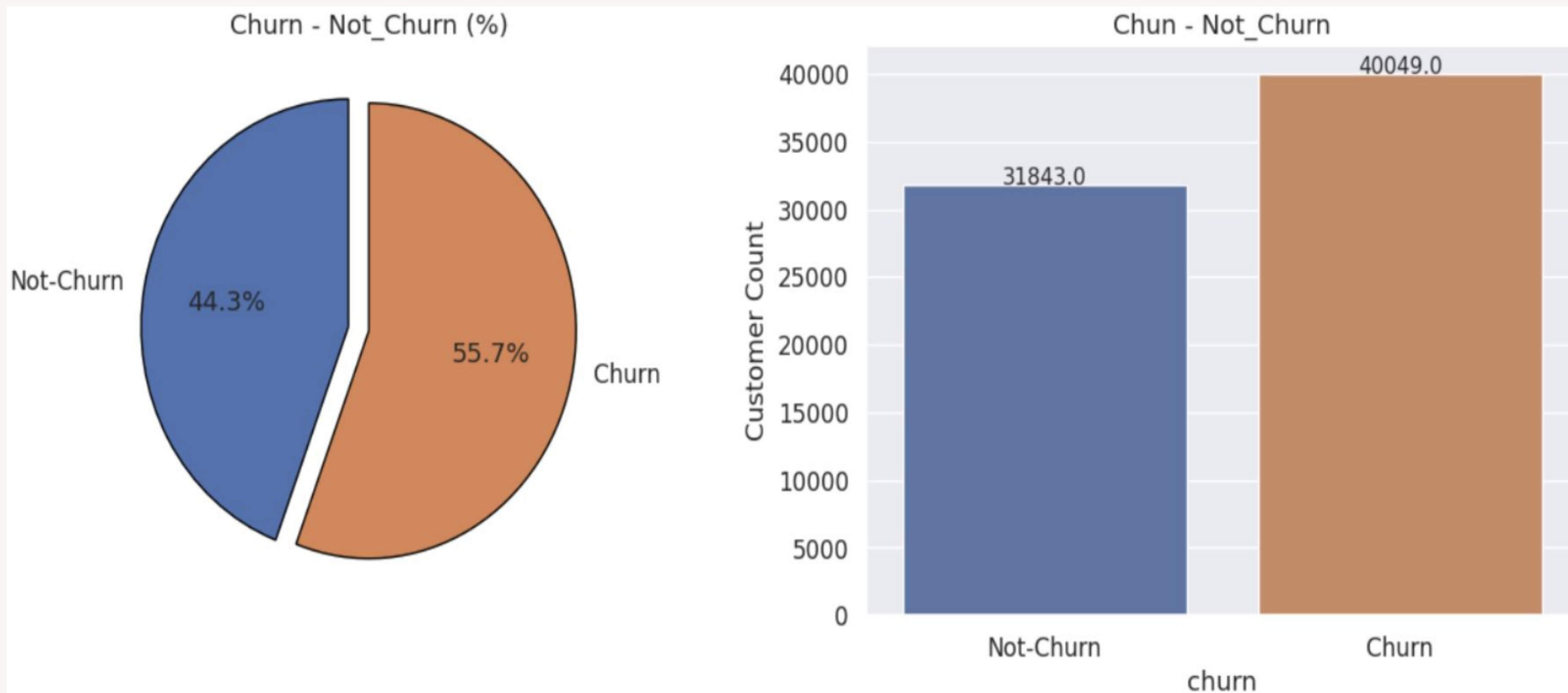
## Chapter 2.

데이터 탐색 및 분석

SKN20-2nd-3TEAM

### 고객 이탈(Churn) 분석

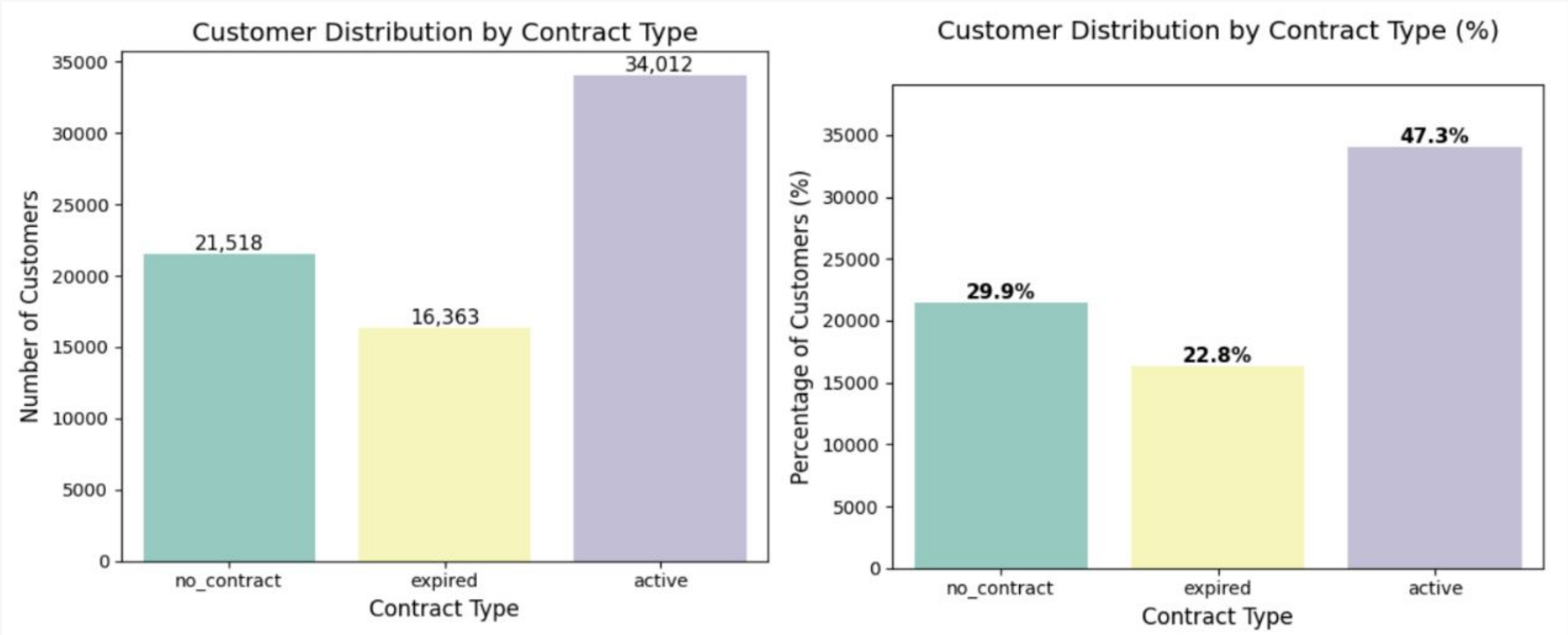
전체 고객 중 이탈 고객의 비율이 유지 고객의 비율보다 약 10% 더 높음



# Chapter 2.

데이터 탐색 및 분석

## 계약 유형 분포



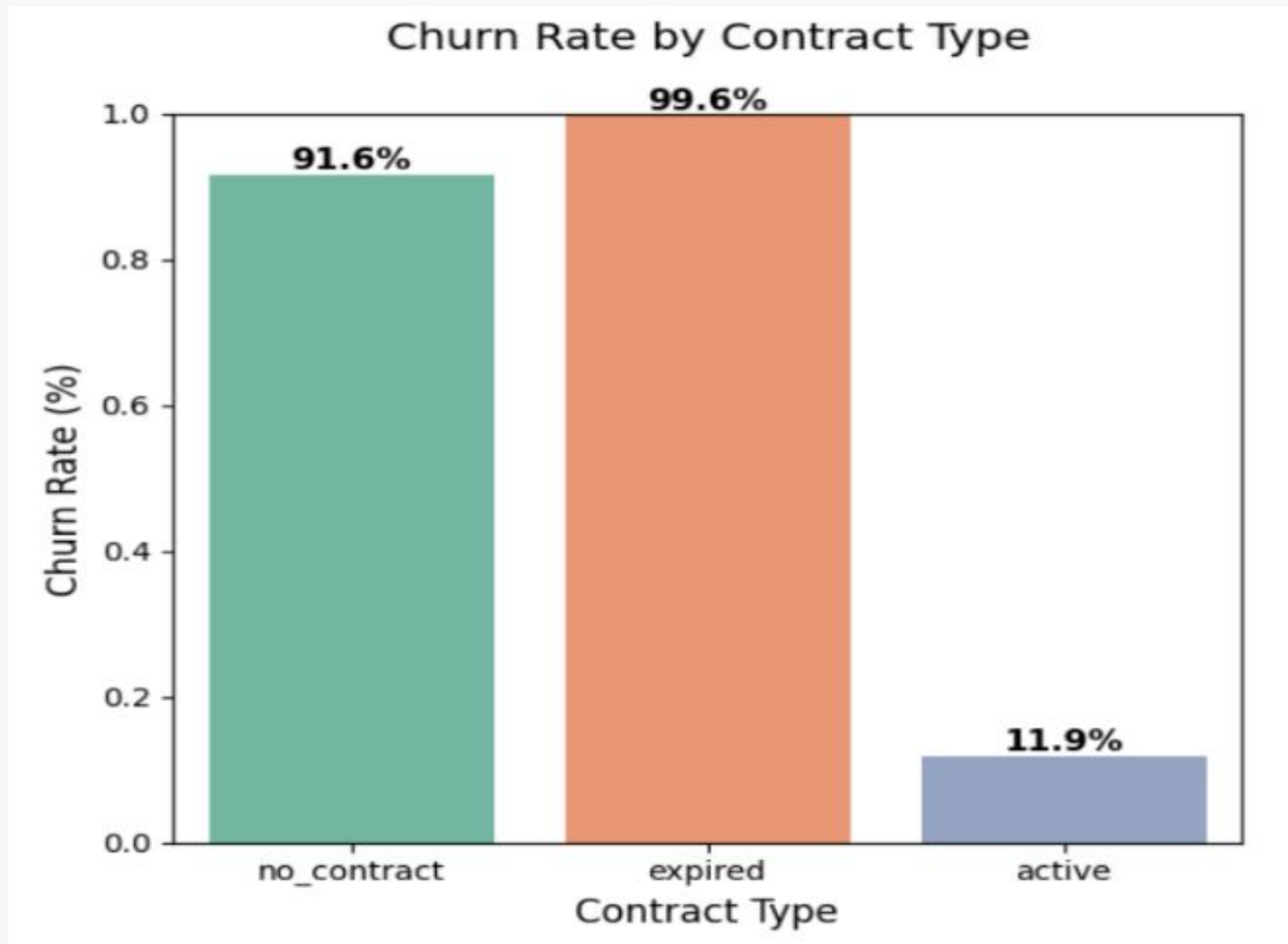
계약 유지(active) > 무계약(no\_contract) > 계약 만료(expired)

## Chapter 2.

데이터 탐색 및 분석

SKN20-2nd-3TEAM

### 계약 유형별 이탈률



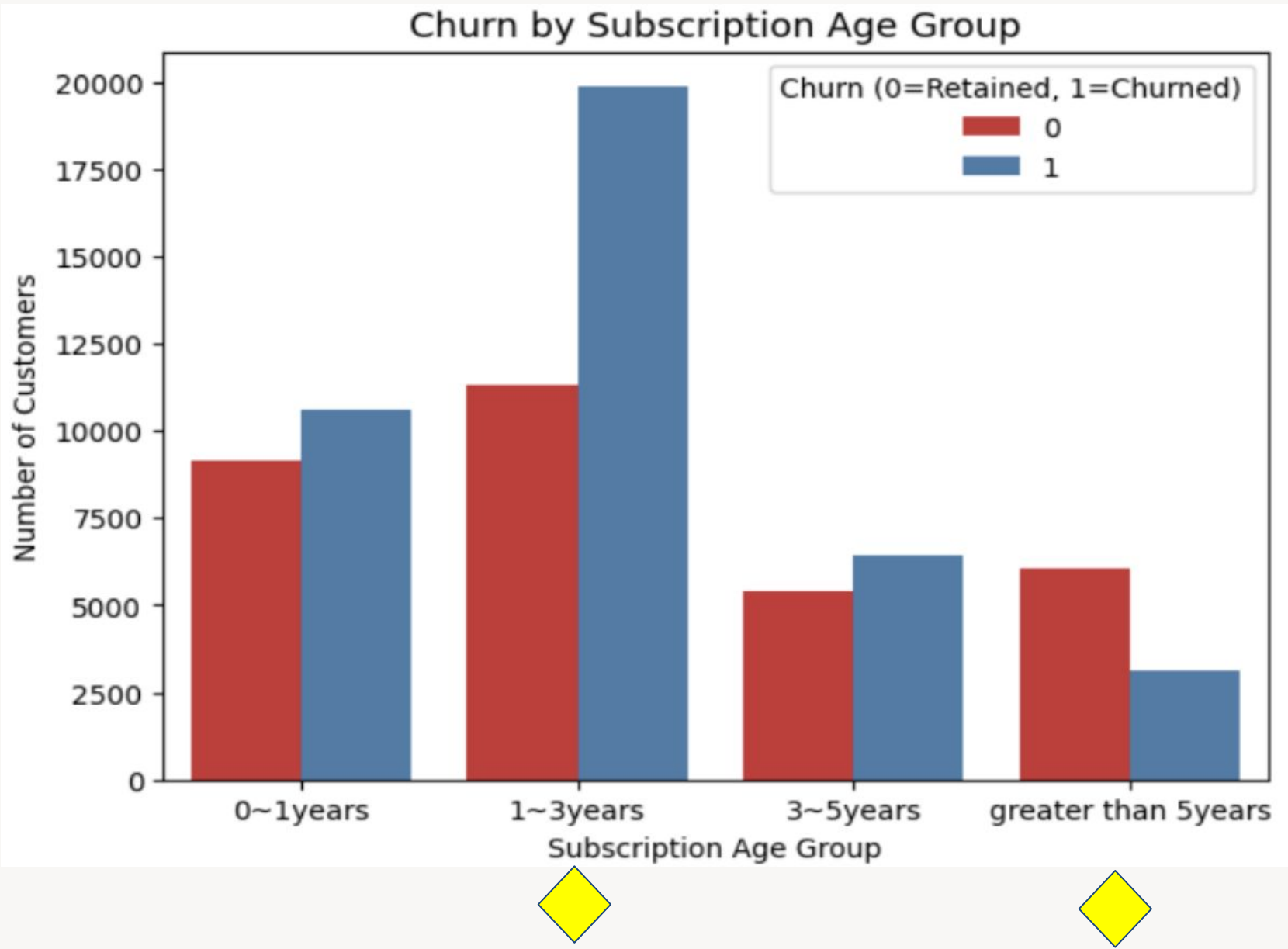
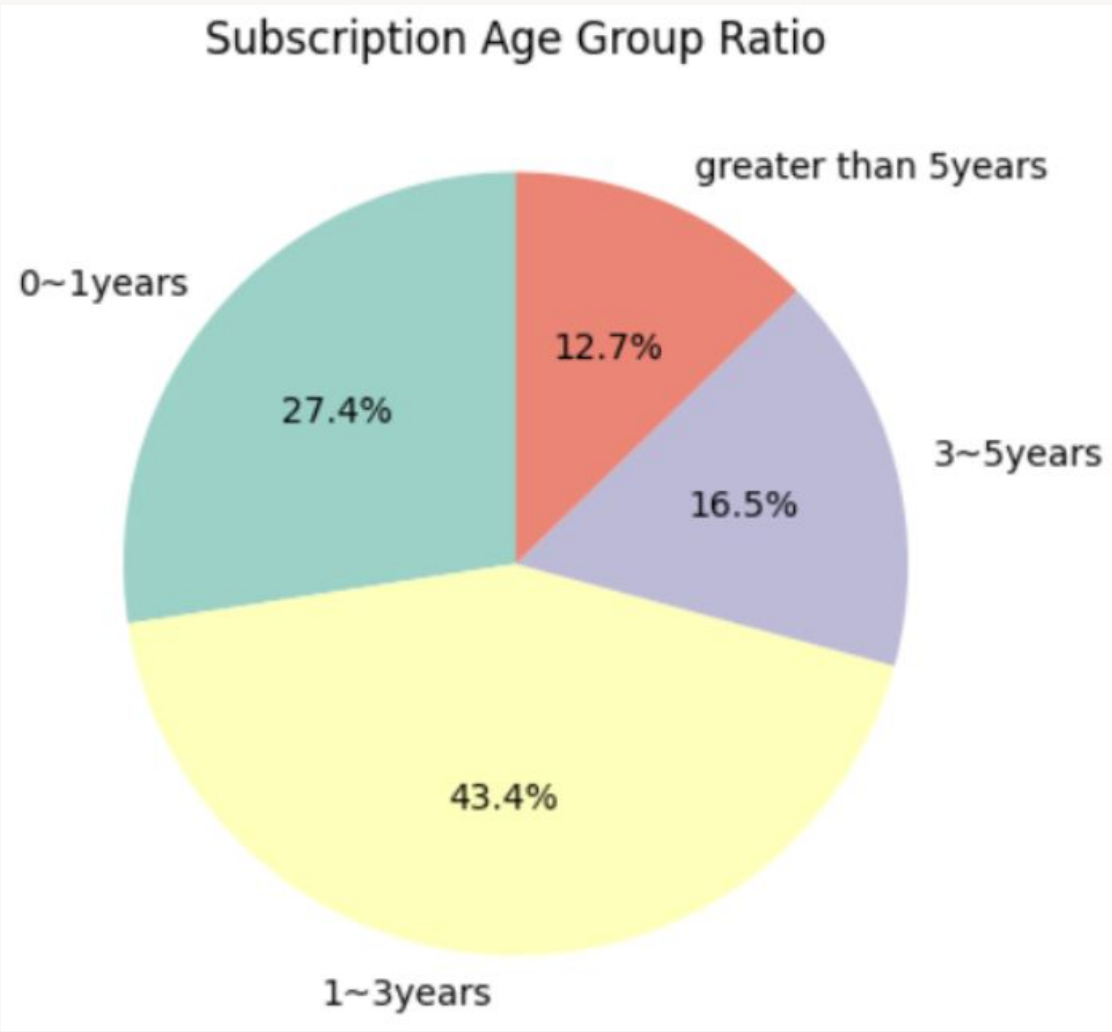
계약 없음(no\_contract), 계약 만료(expired) -> 대부분 이탈 / 계약 유지(active) -> 이탈률 낮음, 계약금

# Chapter 2.

데이터 탐색 및 분석

## 구독 연수 & 이탈

구간	고객 비율	이탈률	특징
0~1 years	27.4%	높음	초기 고객군 — 온보딩 실패 가능성
1~3 years	43.4%	매우 높음	약정 만료 구간, 이탈 집중
3~5 years	16.5%	보통	재계약 고객
greater than 5 years	12.7%	매우 낮음	충성 고객층 형성



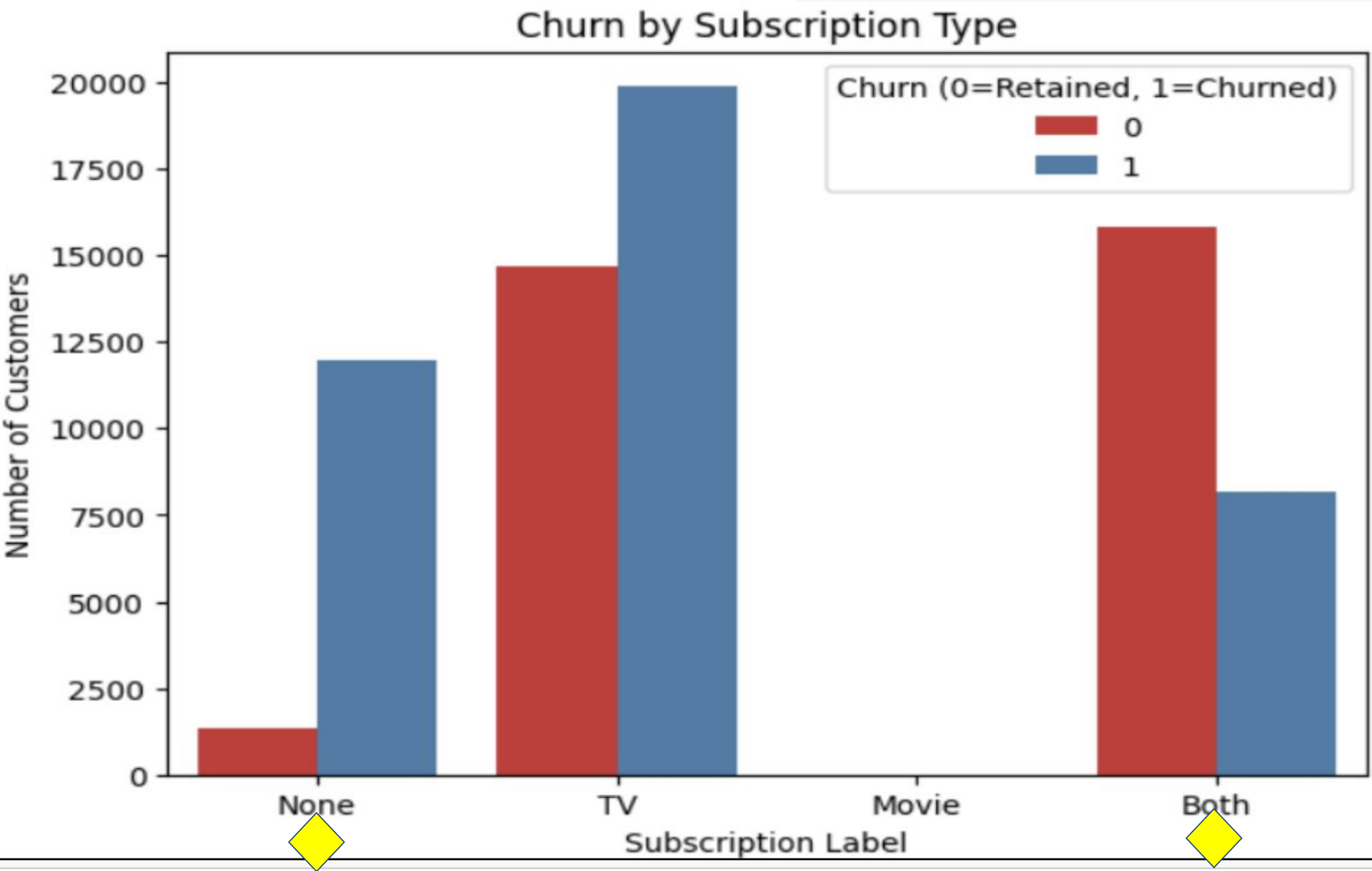
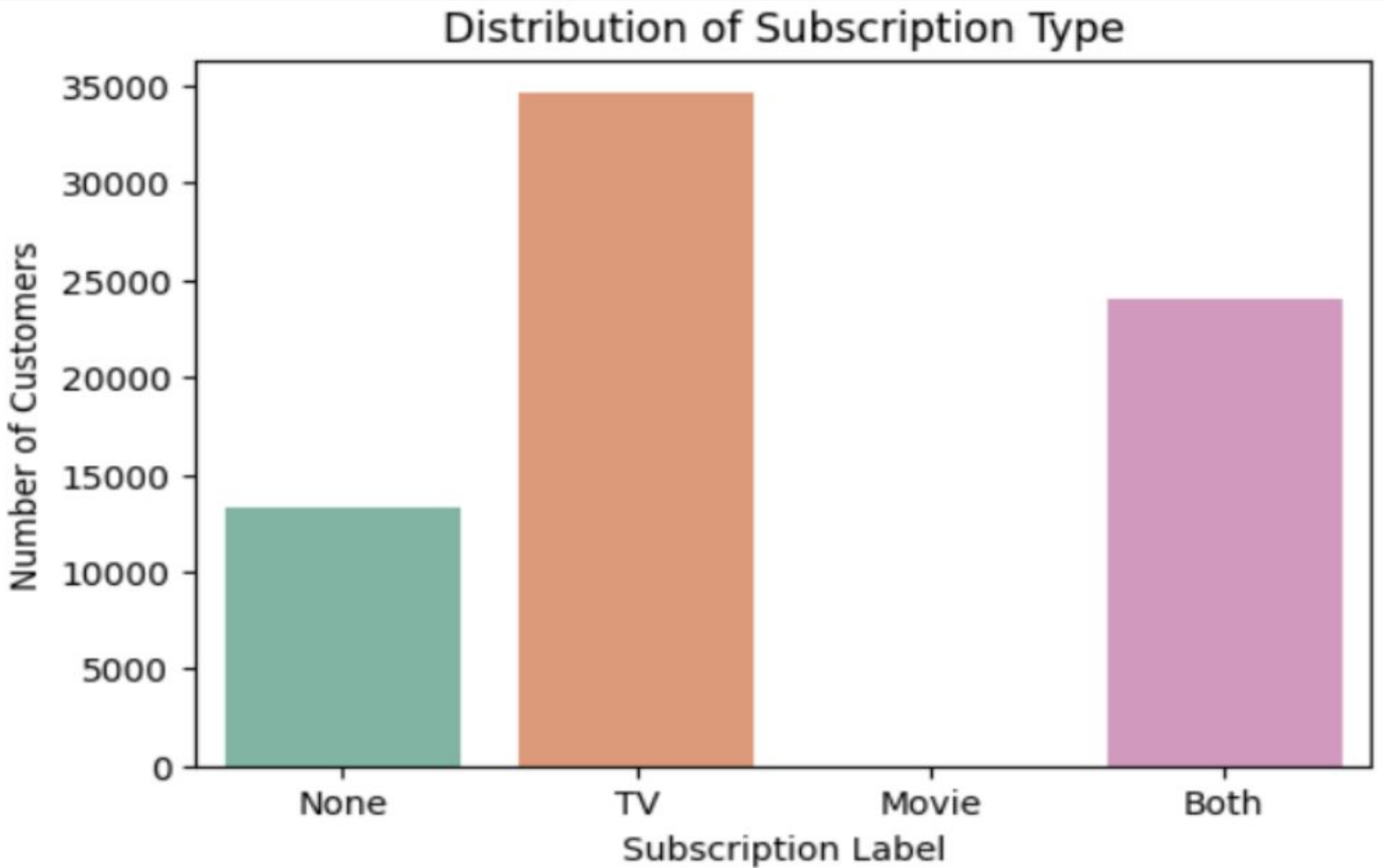
# Chapter 2.

데이터 탐색 및 분석

SKN20-2nd-3TEAM

## 구독 유형 & 이탈

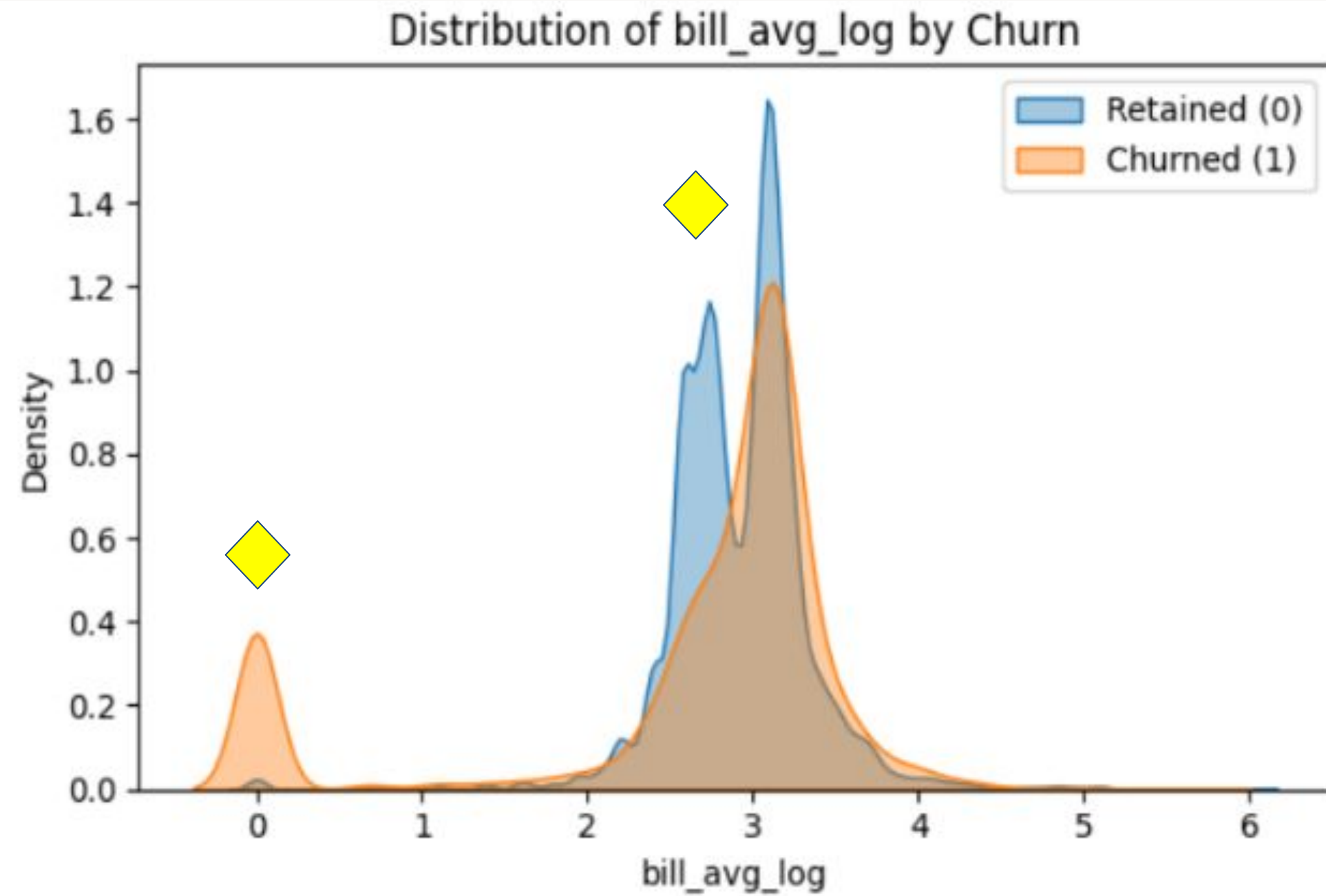
구독 유형	고객 수	특징	이탈률 추세
TV만 구독 (tv)	34,954명	전체 중 가장 많음	유지 > 이탈 (약간의 차이)
Both (tv+movie)	24,015명	복합 구독 고객군	유지 고객이 이탈보다 약 2배 많음
None (구독 없음)	13,281명	단일 서비스 이용	이탈 고객 비중 압도적
Movie만 구독	2명	표본 거의 없음	전부 이탈



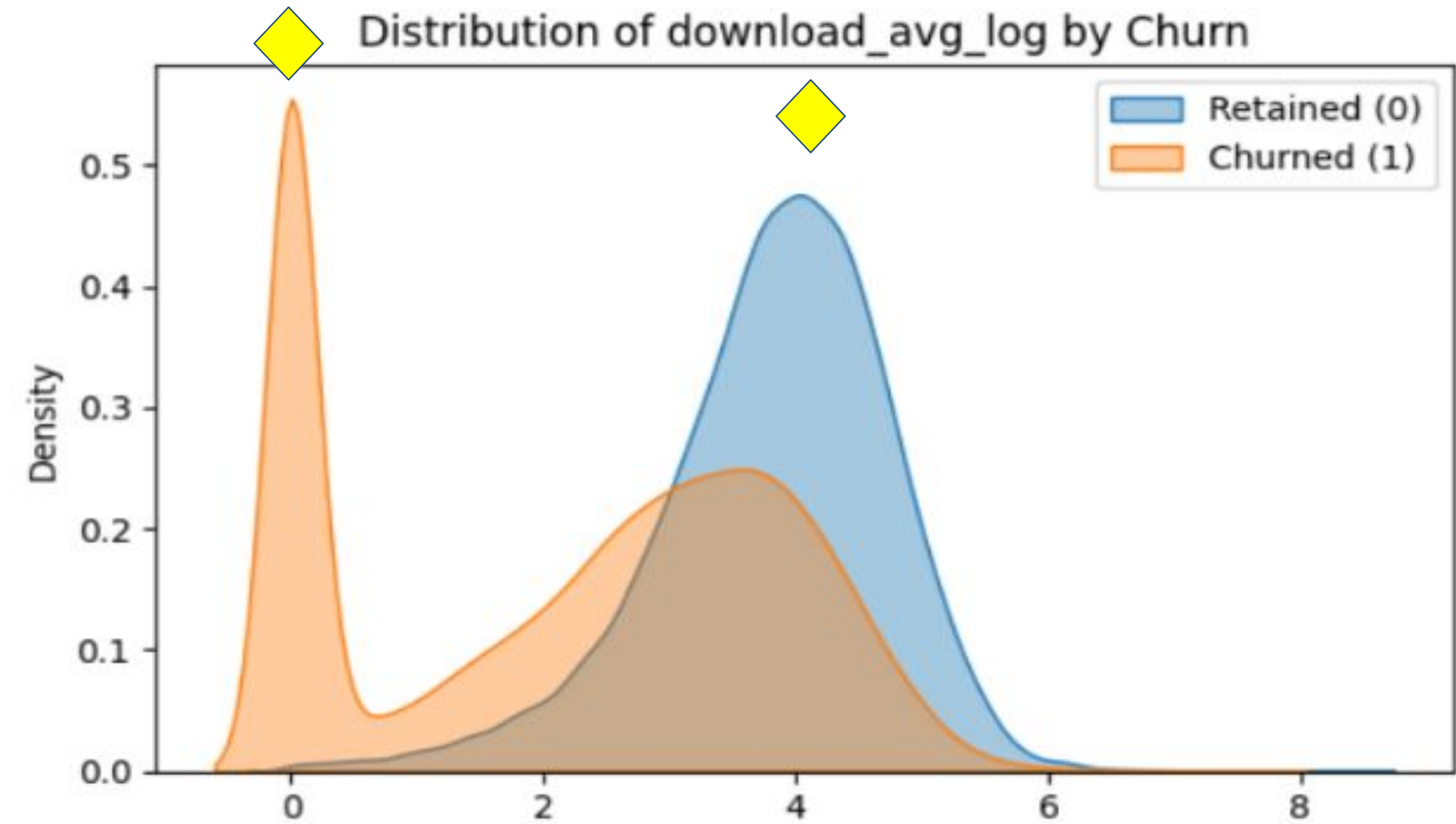
## Chapter 2.

데이터 탐색 및 분석

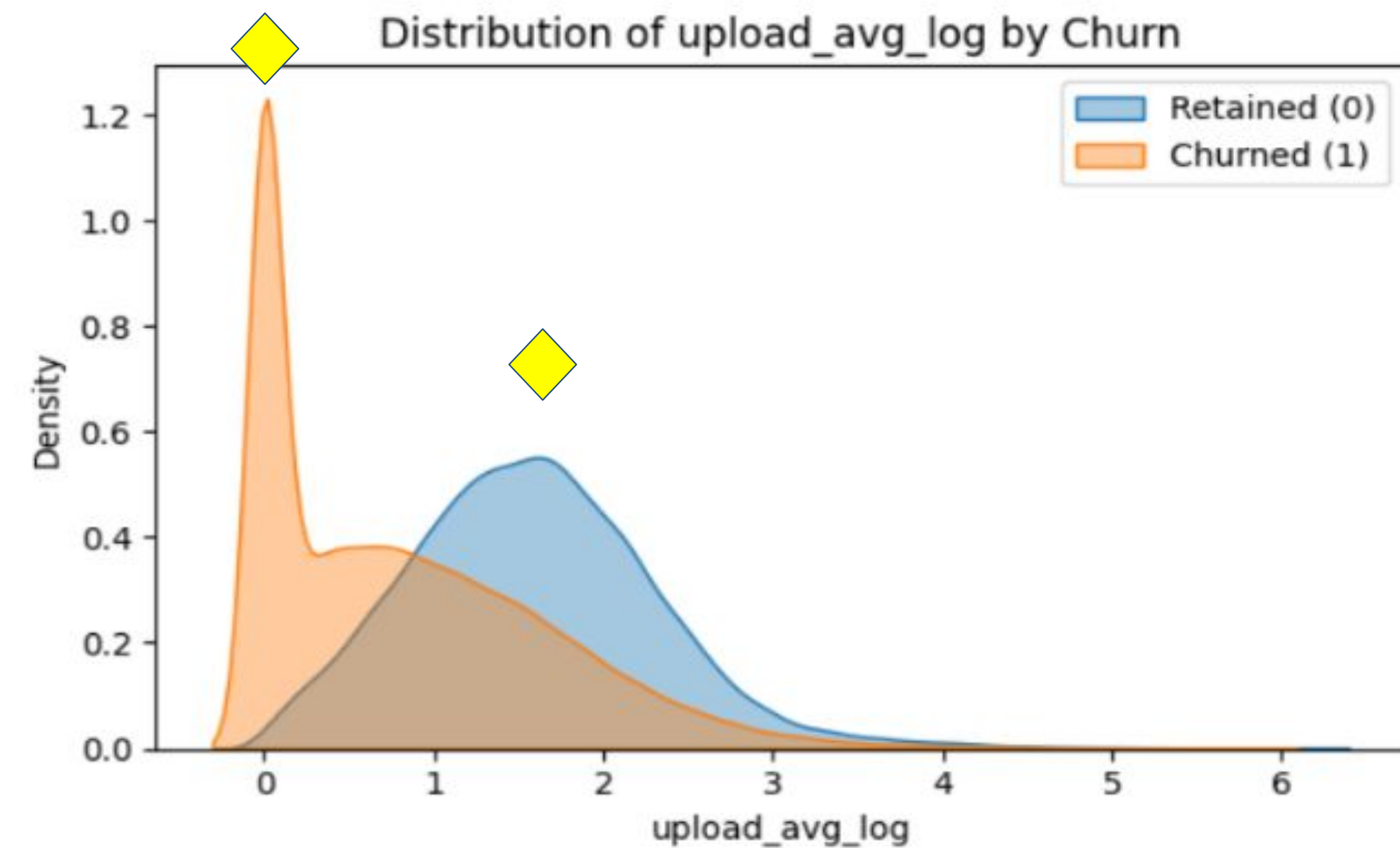
### 평균 청구금액 / 다운로드 / 업로드 & 이탈



- 청구금액 평균이 적을수록 유지보다 이탈을 더욱 많이 한다.



- 다운로드 평균이 낮을수록 유지보다 이탈을 더욱 많이 한다.



- 업로드 평균이 낮을수록 유지보다 이탈을 더욱 많이 한다.

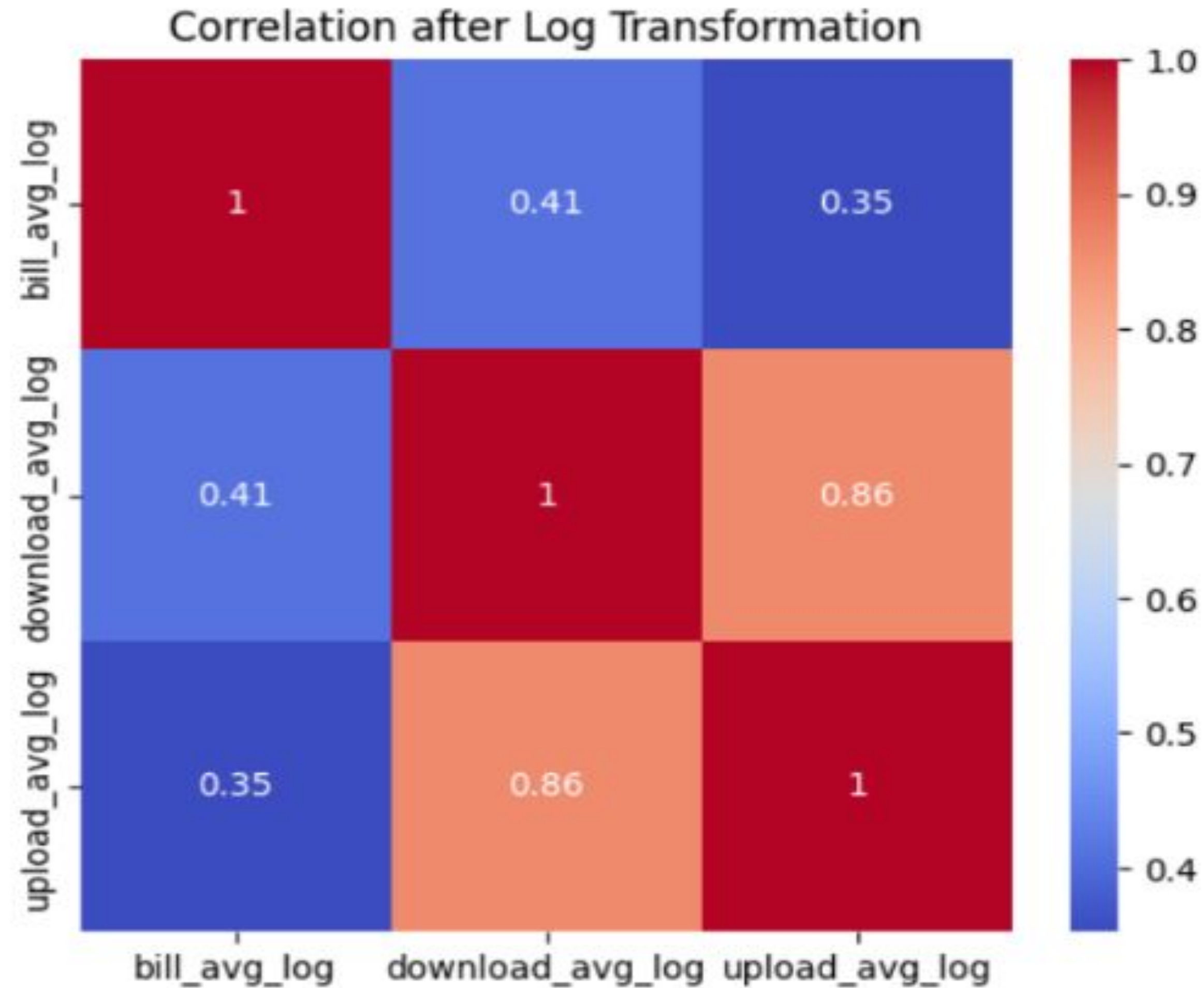
## Chapter 2.

데이터 탐색 및 분석 - 교차분석

SKN20-2nd-3TEAM

평균 청구금액 / 다운로드 / 업로드  
상관관계

### 6.1. bill\_avg & download\_avg & upload\_avg



- download\_avg와 upload\_avg는 강한 양의 상관관계를 갖는다.
- 따라서, download\_사용량이 많을 수록 upload\_사용량도 많다.

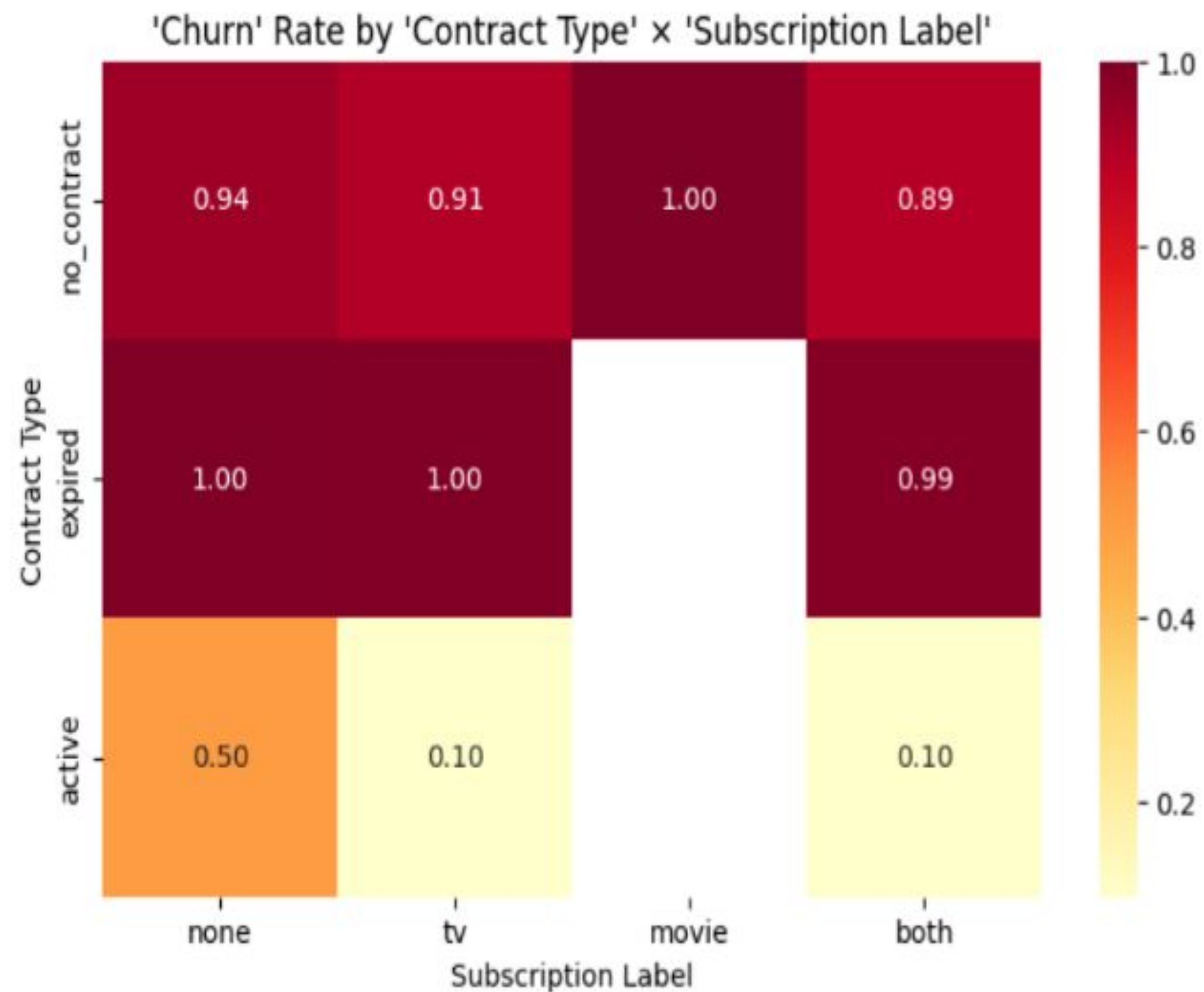
# Chapter 3.

데이터 탐색 및 분석 - 교차분석

SKN20-2nd-3TEAM

## 계약 유형, 구독 유형 & 이탈률

### 6.2. contract\_type, subscription\_label & churn



- 구독 유형 & 계약이 만료된 고객(**expired**) -> 거의 대부분 이탈
- 구독 유형 & 무약정 고객(**no\_contract**) -> 거의 이탈,  
복합구독(**both**) -> 이탈률이 조금 감소
- 계약 유지 고객(**active**) & 구독 없음(**none**) -> 반 정도 이탈  
구독 있음(**tv/both**) -> 이탈률 매우 낮음

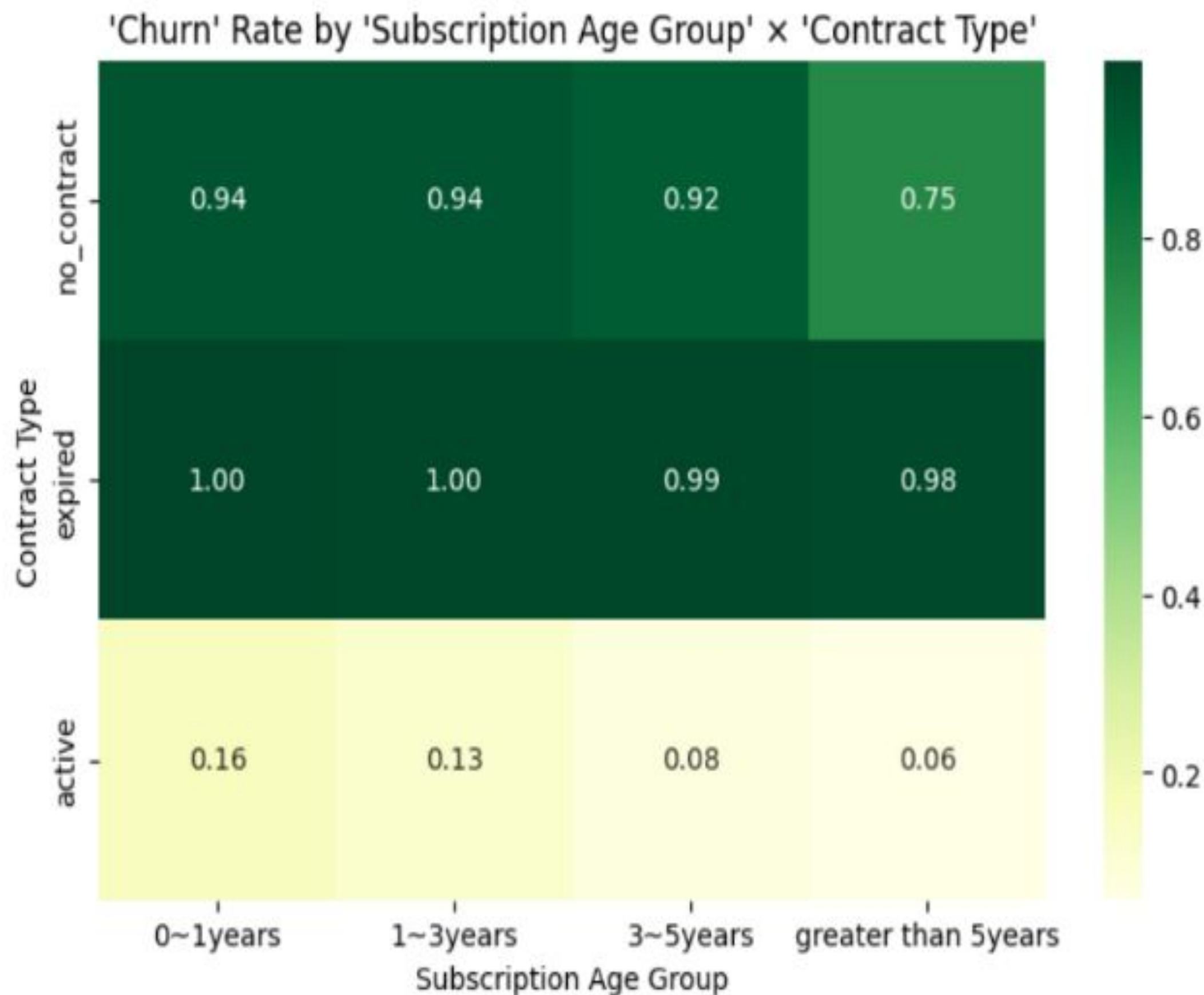
## Chapter 3.

데이터 탐색 및 분석 - 교차분석

SKN20-2nd-3TEAM

### 계약 유형, 구독 연수 & 이탈률

#### 6.3. contract\_type, subscription\_age\_group & chhrn



- 구독 연수 & 계약이 만료된 고객(**expired**) -> 거의 대부분 이탈
- 구독 연수 & 계약 유지 고객(**active**) -> 이탈률 10% 내외로 낮음
- 구독 연수 & 무계약 고객(**no\_contract**) -> 0~5년은 대부분 이탈  
5년+도 이탈률이 높음

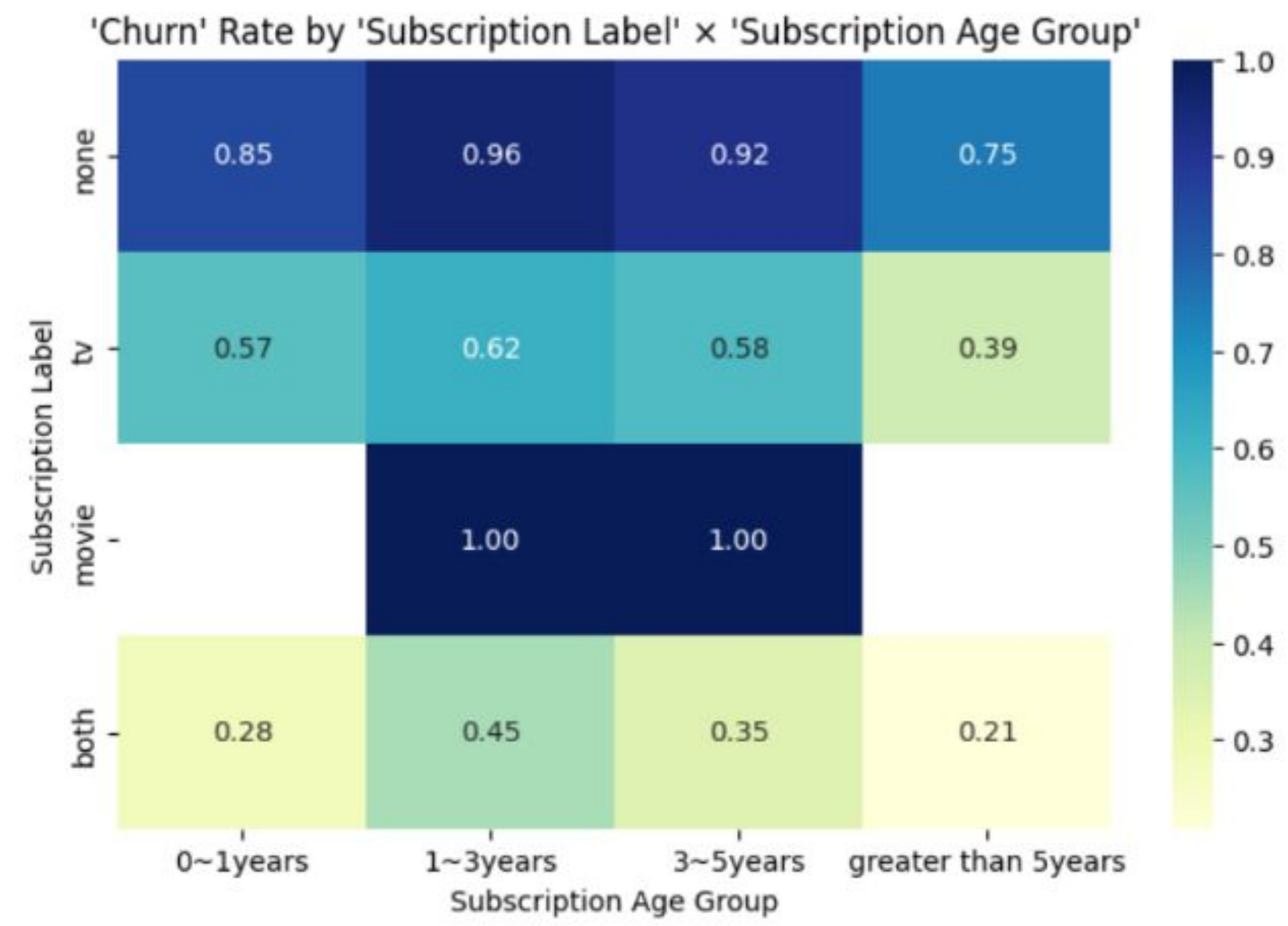
# Chapter 3.

데이터 탐색 및 분석 - 교차분석

SKN20-2nd-3TEAM

## 구독 유형, 구독 연수 & 이탈률

### 6.4. subscription\_label, subscription\_age\_group & churn



- 구독 연수 & 복합구독 (both) -> 이탈률 낮은편
- 구독 연수 & 구독없음 (none) -> 대부분 이탈
- tv만 구독 & 0~5년 -> 이탈률 약 60%로 유지 불안전,  
5년+ -> 이탈률 39%로 낮아짐

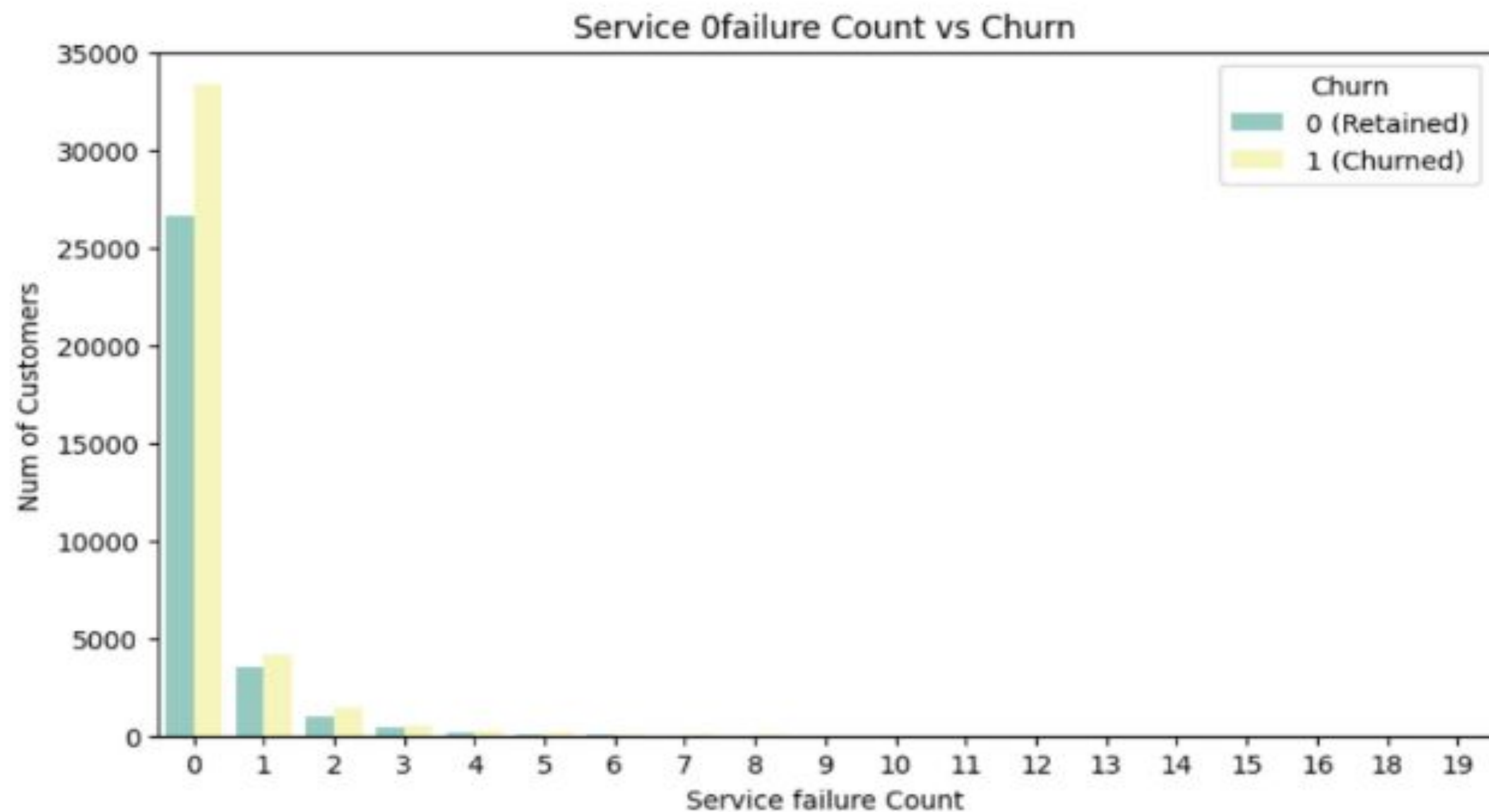
## Chapter 3.

데이터 탐색 및 분석 - 서비스 품질과 이탈 관계

SKN20-2nd-3TEAM

### 서비스 장애 신고 횟수 & 이탈률

#### 7.1. service\_failure\_cout & churn



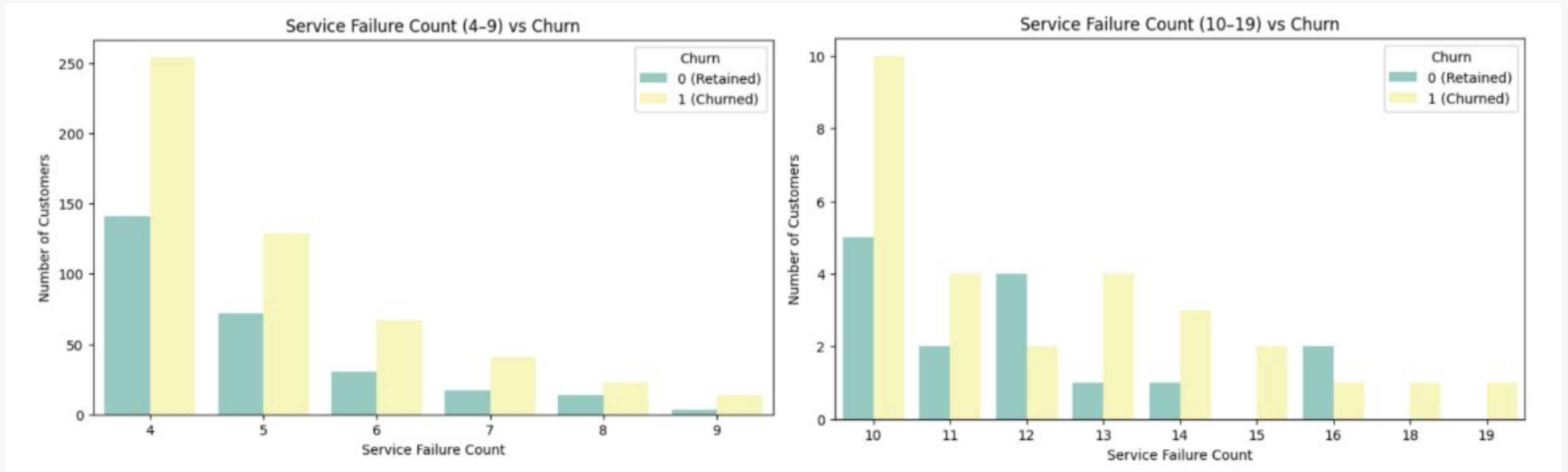
두 그룹 모두 0에 집중

-> 대부분의 고객은 서비스 장애 신고 경험이 없다!

# Chapter 3.

데이터 탐색 및 분석 - 서비스 품질과 이탈 관계

SKN20-2nd-3TEAM



- 서비스 장애 신고 건수가 많아질수록 이탈 고객이 더 많은
- 서비스 장애 신고 건수 = 12,16 -> 유지>이탈 (예외)

	service_failure_count	contract_type	churn
25046	12	2	0
50335	12	2	0
61656	12	2	0
66445	12	2	0
36674	16	2	0
65046	16	2	0

- 계약 기간이 남았을 때만(active) 서비스 장애 신고 건수가 많아도 이탈하지 않았다.

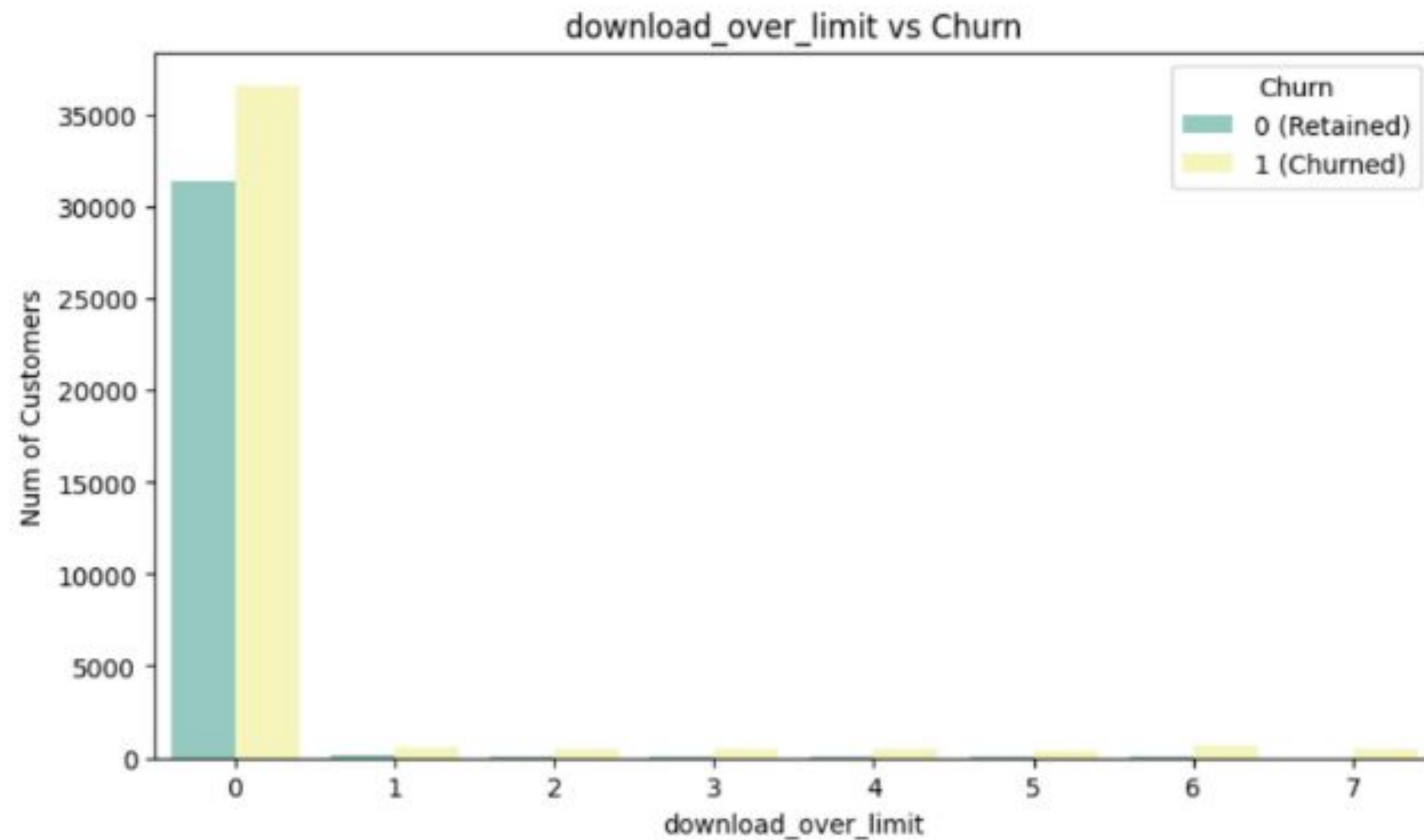
## Chapter 3.

데이터 탐색 및 분석 - 서비스 품질과 이탈 관계

SKN20-2nd-3TEAM

### 다운로드 초과 횟수 & 이탈률

#### 7.2. download\_over\_limit & churn



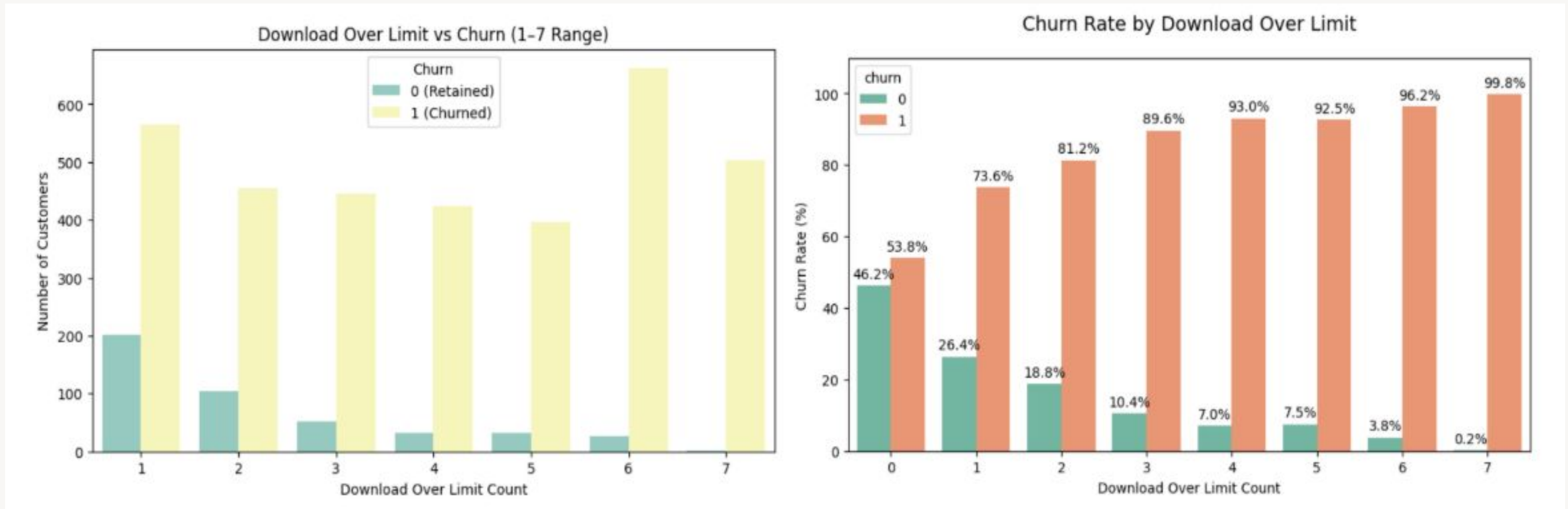
두 그룹 모두 0에 집중

-> 대부분의 고객은 다운로드  
한도를 초과한 경험이 없다!

## Chapter 3.

데이터 탐색 및 분석 - 서비스 품질과 이탈 관계

SKN20-2nd-3TEAM



- 다운로드 초과 횟수가 0회인 고객은 이탈과 유지율이 비슷한 반면,
- 다운로드 초과 횟수가 1회 이상인 고객부터 명확한 차이 발생 -> 초과 횟수↑ 이탈률↑

# Chapter 3.

- (계약 만료 or 무계약) × (구독없음 or 단일구독) -> 이탈률 높음
- (계약:1년~3년) × (계약 만료) -> 이탈 집중 구간
- (청구금액 낮음) × (저-사용량) -> 이탈률 높음

- (계약 유지) × (복합구독) -> 이탈률 가장 낮음, 가장 안정적인 고객군
- (청구금액 높음) × (고-사용량) -> 이탈률 낮음

구분	주요 요인	특징 및 시사점
이탈 고객 (churn=1)	계약 만료·무약정, 구독없음&단일 구독, 저사용량, 서비스 장애 다발	재계약 유도 및 품질 안정화 필요, 저이용 고객 대상 체험·혜택 제공 필요
유지 고객 (churn=0)	계약 유지상태, 복합구독, 고사용량, 서비스 장애 없음	장기 복합 구독 고객 중심의 리워드·혜택 강화, 신규 고객 대상 복합구독 유인

## Chapter 3.

데이터 탐색 및 분석 - 총정리

SKN20-2nd-3TEAM

### 8.4. 고객 유지 전략

- (단기) 계약 만료 고객 방어: 이탈률이 100%에 달하는 계약만료( expired ) 고객과 1~3년 차 도래 고객을 대상으로 한 자동 재계약 프로모션 도입
- (중기) 상품 가치 제고: 이탈률이 가장 낮은 '복합 구독(Both) 을 주력 상품으로 하여, TV 단일 또는 구독없음( none ) 고객의 전환을 유도하는 업셀링 (Up-selling) 전략
- (장기) 충성도 관리: active 계약에만 의존하는 현재 구조에서 벗어나, 서비스 사용량을 높일 수 있는 콘텐츠 추천(e.g., Movie 추천) 이나 충성고객 ( 5년+ ) 대상 리워드 프로그램을 도입하여 계약이 없어도 유지될 수 있도록 관리



# Chapter 4.

Modeling & Evaluation

베이스 모델 구상



머신러닝 : RandomForest  
딥러닝 : MLP

# Chapter 4.

## 머신러닝 하이퍼파라미터 기준

파라미터명	탐색 값(범위)
max_depth	4, 6, 8
learning_rate	0.01, 0.05, 0.1
n_estimators	300, 500, 800



이상적인 파라미터 조합을 도출하여,  
해당 기준을 기반으로 모 든 모델에 동일한 탐색 범위 지정

# Chapter 4.

Modeling & Evaluation

## RandomForest

Baseline (기본 모델)	지표 ▶	After (하이퍼튜닝+앙상블)
0.9344	Accuracy	0.9372
0.9401	F1-score	0.9429
0.9663	ROC-AUC	0.9673

# Chapter 4.

Modeling & Evaluation

## XGBoost

Baseline (기본 모델)	지표 ▶	After (하이퍼튜닝+앙상블)
0.9376	Accuracy	0.9380
0.9429	F1-score	0.9434
0.9705	ROC-AUC	0.9716

# Chapter 4.

Modeling & Evaluation

## LightGBM

Baseline (기본 모델)	지표 ▶	After (하이퍼튜닝+앙상블)
0.9383	Accuracy	0.9384
0.9436	F1-score	0.9437
0.9720	ROC-AUC	0.9724

# Chapter 4.

Modeling & Evaluation

## CatBoost

Baseline (기본 모델)	지표 ▶	After (하이퍼튜닝+앙상블)
0.9381	Accuracy	0.9381
0.9434	F1-score	0.9434
0.9712	ROC-AUC	0.9720

# Chapter 4.

Modeling & Evaluation

## LogisticRegression

Baseline (기본 모델)	지표 ▶	After (하이퍼튜닝+앙상블)
0.9324	Accuracy	0.9325
0.9384	F1-score	0.9385
0.9578	ROC-AUC	0.9579

딥러닝 MLP 최적 하이퍼파라미터

● 최적의 파라미터 후보 탐색

파라미터명

탐색 값(범위)

learning\_rate

0.1, 0.01, 0.001

Layer

(16-8-1), (32-16-1), (64-32-1)  
(8-16-8-1), (16-32-16-1), (32-64-32-1)  
(16-8-16-1), (64-32-64-1)  
(16-8-4-1), (32-16-8-1), (64-32-16-1),  
(128-64-32-1)

● RandomSearch

- Hidden Layer Size =  
**(16-8), (64-32-64), (16-8-16), (16-32-16), (16-8-4)**  
**(128- 64 -32) (64 - 32) (64 - 32 - 16)**
- Activation Function = **ReLU, Tanh**
- Optimizer = **Adam, SGD**
- Learning Rate (LR) = **0.001, 0.01, 0.1**
- Weight Decay = **0.1, 0.0001, 0.001**
- Batch Size = **32, 64, 128**
- Training Epochs = **10, 20, 30**
- Feature Scaling Method = **StandardScaler, MinMaxScaler, RobustScaler**

# Chapter 4.

## Learning Rate 비교

- 모델 구성

- o EarlyStopping 적용
- o Loss: BCEWithLogitsLoss(sigmoid 포함 정리)
- o Optimizer: Adam
- o Batch\_size : 32
- o Epoch : 100
- o 은닉층 3개 (128 – 64 -32)

(Pytorch)

Layer	Operation	Units
Input	Linear	Cin → 128
Hidden 1	Linear → BatchNorm → ReLU → Dropout(0.2)	128
Hidden 2	Linear → BatchNorm → ReLU → Dropout(0.2)	64
Hidden 3	Linear → BatchNorm → ReLU → Dropout(0.2)	32
Output	Linear	32 → 1

\* BatchNorm1d: 학습 중 각 배치의 feature 분포를 정규화하여 훈련 안정성과 수렴 속도를 높임.  
\* Dropout(0.2): 뉴런의 20%를 랜덤으로 비활성화하여 과적합을 방지하고 일반화 성능을 향상시킴.

- Learning rate 성능 비교

Learning Rate	Accuracy	F1 Score	Precision	Recall
0.001	93.76	0.9432	0.9570	0.9298
0.01	93.75	0.9431	0.9572	0.9293
0.1	93.63	0.9419	0.9577	0.9266

=> 성능이 거의 동일함.

# Chapter 4.

## Layer 비교

- 모델 구성
  - EarlyStopping 적용
  - Loss: Binary Crossentropy
  - Optimizer: Adam
  - Batch\_size : 16
  - Epoch : 50

(TensorFlow)

Layer	Operation	Units	Activation
Input	Dense	64	ReLU
Hidden	Dense	32	ReLU
Output	Dense	1	Sigmoid

- Layer 성능 비교

Layer	Accuracy	F1 Score	Precision	Recall
(16-8-1)	0.94	0.94	0.95	0.94
(64-32-64-1)	0.94	0.94	0.95	0.94
(16-8-16-1)	0.94	0.94	0.95	0.93
(16-32-16-1)	0.94	0.94	0.95	0.93
(16-8-4-1)	0.94	0.94	0.95	0.93
(128-64-32-1)	0.93	0.94	0.95	0.93
(64-32-1)	0.93	0.94	0.95	0.93
(64-32-16-1)	0.94	0.94	0.95	0.93
(32-16-1)	0.93	0.94	0.95	0.93
(8-16-8-1)	0.94	0.94	0.95	0.93
(32-64-32-1)	0.94	0.94	0.95	0.93
(32-16-32-1)	0.94	0.94	0.95	0.93

=> 성능이 거의 동일함. 상위 8개 RandomSearch 파라미터로 활용

## RandomSearch

- RandomSearch 파라미터

파라미터	Values
Hidden Layer Size	(16-8), (64-32-64), (16-8-16), (16-32-16), (16-8-4), (128-64-32), (64-32), (64-32-16)
Activation Function	ReLU, Tanh
Optimizer	Adam, SGD
Learning Rate (LR)	0.001, 0.01, 0.1
Weight Decay	0.0, 0.0001, 0.001
Batch Size	32, 64, 128
Training Epochs	10, 20, 30
Scaler / Feature Scaling Method	StandardScaler, MinMaxScaler, RobustScaler

### 최적의 파라미터 탐색

- CV = [2, 3] 변경
- n\_iter = [150, 300, 600] 변경

### 최적의 파라미터

- CV = 3 , n\_iter = 150
- Optimizer = Adam
- layers = [128-64-32]
- activation = Tanh
- epoch = 30
- lr = 0.001
- batch\_size = 64

# Chapter 5.

## 성능 지표 비교

### 딥러닝 성능 지표

모델명	Accuracy	F1-score	ROC-AUC
MLP	0.937	0.9427	0.9661



다양한 Layer 구조 및 학습률 변경에도 성능 편차가 거의 없었으며,  
최종적으로 안정적인 MLP 성능(Accuracy 0.937 / F1 0.9427 / ROC-AUC 0.9661)을  
[확인함](#)

# Chapter 5.

## 성능 지표 비교

### 머신러닝 성능 지표 종합 비교 요약

모델명	Accuracy	F1-score	ROC-AUC	모든 모델을 동일한 데이터셋과 전처리 기준으로 비교한 결과, F1-score 수치가 동일하여 ROC-AUC 기준으로 보았을때 LightGBM이 가장 뛰어난 예측 성능을 보임.  ▼ 최종 분석 모델로 LightGBM 을 채택
Logistic Regression	0.9325	0.9385	0.9579	
Random Forest	0.9372	0.9429	0.9673	
XGBoost	0.9380	0.9434	0.9716	
LightGBM	0.9384	0.9437	0.9724 🏆	
CatBoost	0.9381	0.9434	0.9720	

# Chapter 5.

## 성능 지표 비교

### 머신러닝 & 딥러닝 성능 지표 비교

모델명	Accuracy	F1-score	ROC-AUC
LightGBM	0.9384	0.9437	0.9724 🏆
MLP	0.9370	0.9427	0.9661

두 모델의 정확도와 F1-score는 거의 유사하지만,  
ROC-AUC 기준으로 LightGBM이 약간 더 높은 예측 성능을 보여 최종 모델로 선정



최종 분석 모델로 LightGBM 으로 선정

감사합니다.

GitHub | 강민지

GitHub | 김지은

GitHub | 김효빈

GitHub | 안채연

GitHub | 홍혜원