

카드소비형태에 따른 보험종목별 사고율 분석

이제희, 이지호, 조민서

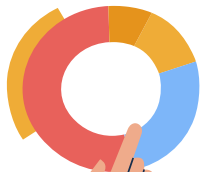
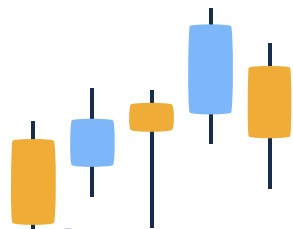
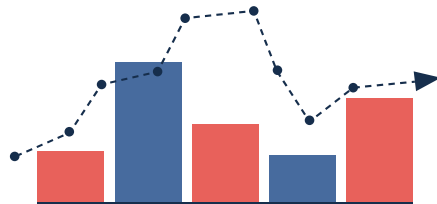


Table of contents



01

대회 소개

02

분석 기법 선택
배경

03

주요 기술 설명

04

PoC 코드 구현

05

앞으로의 계획



01

대회 소개

FSI Data Challenge 2023



01 대회 소개



참여자

전국 대학(원)생* (대1~3년) 단위로 참가

* 외국계, 통신회사 참가 가능. 1인, 졸업생은 참가 제한

공모주제

대회용 데이터 기반으로 지정된 주제에 대한 문제 해결 방법

또는 독창적인 분석 결과를 제안하고,

소스 코드 구현을 통해 PoC(Proof of Concept) 제시

(보험개발원과 삼성카드 데이터로 트랙 A 구성)

(보험개발원과 삼성카드, 삼성카드와 GranData 데이터로 트랙 B 구성)

(GranData 데이터로 트랙 C 구성)

트랙 A: 카드소비 형태에 따른 보험 종류별 사고율 분석

트랙 B: 지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

지역별 카드소비(보험개발원, 삼성카드, GranData 데이터로 분석)를 통한

참수기간

2023. 6. 12(목) ~ 7. 7(금)

시상내역

총 2,000만원의 예산으로 상금, 상장 및 기타 지원 제공

구분	트랙	순위	참수	상금	기타 지원
금융위원회 위원장상	-	1	1팀	500만원	
금융위원회 위원장상	B	2	1팀	400만원	
보험개발원 위원장상	A	2	1팀	400만원	
삼성카드 대표이사상	A	3	1팀	250만원	금융위원회
삼성카드 대표이사상	B	3	1팀	250만원	금융위원회
트랙A 우수상	A	4	1팀	100만원	
트랙B 우수상	B	4	1팀	100만원	

* 트랙 순위별 상금 지급 대상 제외 사유 시정할 수 있음

* 자세한 내용은 수상자 선정 후 발표

문의사항

[대회 운영 문의]
대외협력팀 | 문의사항: 02-3495-9934, makthread@fsec.or.kr

[지원기업 후원사비트 문의]
금융위원회 | 문의사항: 02-3495-9915, kiny@fsec.or.kr

주최·주관: 금융위원회, 금융보안원, 후원: 금융위원회, KCC, SK Telecom

공모주제

대회용 데이터 기반으로 지정된 주제에 대한 문제 해결 방법
또는 독창적인 분석 결과를 제안하고, 소스 코드 구현을 통해
PoC(Proof of Concept) 제시

+ 데이터: 보험 개발원과 삼성카드 데이터로 트랙 A 구성

+ 목표: 카드소비 형태에 따른 보험 종목별 사고율 분석

(주최·주관) 금융보안원

(후원기관) 금융위원회, 보험개발원, 삼성카드, GranData



01 대회 소개

데이터

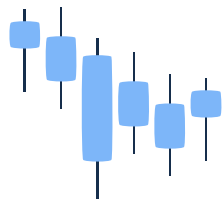
기준년도	성별	연령	보험종류	보장내용	계약건수	보험료	사고건수	지급금액	자동차국외산구분코드	자동차보험가입경력
2018	1	20	1	1	0.5	13,000	0	0		
2019	2	25	1	2	1	5,000	0	0		
2020	1	25	1	3	1	17,000	1	10,000,000		
2021	2	30	1	1	1	19,000	0	0		
2018	1	40	2	5	2	8,000	0	0		
2019	2	40	2	7	1	5,000	2	200,000		
2020	2	50	3	1	0.8	3,000	1	105,000	1	1
2021	1	60	3	2	1	8,000	1	350,000	1	2
2020	1	50	3	7	1	10,000	0	0	2	2
2021	1	50	1	1	1	21,000	0	0		

(보험개발원)명세서

(삼성카드)명세서

기준년도	AGE	GENDER	JOB	INCOME	HOM_MGPO	HOM_SGG	OFFI_MGPO	OFFI_SGG	ONEPER_GD_C
2018	A	M	A	A	Seoul	Jung-gu	Seoul	Dongdaemun-gu	05
2019	B	F	B	B	Seoul	Seocho-gu	Seoul	Guro-gu	06
2020	C	M	C	C	Seoul	Songpa-gu	Gyeonggi	Ansan-si danwon-gu	07
2021	D	F	D	D	Seoul	Yeongdeungpo-gu	Gyeonggi	Seongnam-si bundang-gu	08
2018	E	M	E	E	Gyeonggi	Namyangju-si	Gyeonggi	Goyang-si Ilsandong-gu	09
2019	F	F	F	F	Gyeonggi	Siheung-si	Incheon	Michuhol-gu	10
2020	G	M	B	G	Gyeonggi	Uijeongbu-si	Seoul	Gwangjin-gu	11
2021	H	F	C	H	Gyeonggi	Ansan-si danwon-gu	Seoul	Yangcheon-gu	12
2020	I	M	D	I	Incheon	Namdong-gu	Incheon	Bupyeong-gu	13
2021	B	F	E	B	Incheon	Bupyeong-gu	Incheon	Seo-gu	





02

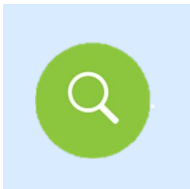


분석기법 선택 배경

EDA / Feature Engineering / Cluster analysis



분석기법

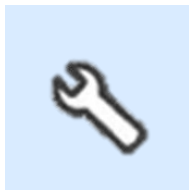


EDA

- 삼성카드 명세서 데이터에 대한 사전 지식 얻기
- 변수의 의미와 범위 파악
- 데이터의 분포, 이상치 등 파악



삼성카드 명세서 데이터에 대한
종합적인 이해 증가



Feature Engineering

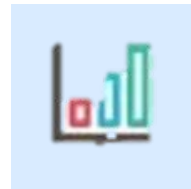
- 군집 형성에 불필요한 칼럼을 제거



Why?

Dimensionality가 증가하면 불필요한
feature로 인해 문제 발생.

K-means 모델을 만들기 위해서는
쓸모없는 feature들을 제거.



Cluster analysis

- k개의 군집의 특성을 파악.
- 시계열 분석을 통해 각 군집별 소비 패턴 확인.



군집별 소비패턴 파악과 시계열
분석을 통해 군집 특성을
파악하여 더욱 정확한 분석
결과를 도출.

03

주요 기술 설명

EDA / Feature Engineering / Cluster analysis



03 주요 기술 설명

EDA



: 데이터의 특성과 구조를 파악하는 과정.

Process

Dataset 특성과 전체적인 값
분포 파악

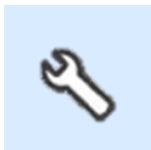
데이터셋에 대한 프로파일링을
수행하여 데이터의 특성과
문제점 파악

Column 특성 파악

데이터셋에 포함된 각 칼럼의
의미와 역할을 이해하고
이상치와 결측치의 파악 및 제거

데이터 시각화를 통한
특성 파악

어떤 연령대가 가장 많은 소비를
하는지, 어떤 연령대의 매출이
높은지 파악

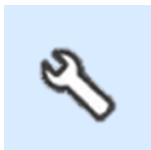


Feature Engineering

어떤 데이터를 다룰 것인가?

삼성 카드 명세서 데이터

- 1 나이 / 소득 / 라이프 스타이지에 관한 칼럼 정보로만만 군집의 특성과 소비패턴을 파악.
- 2 매출 건수, 매출 금액 데이터의 경우 각 업종마다의 각 시간대별(A~G) 칼럼으로 구성.
이 중 매출 건수의 데이터만 활용.
- 3 매출 건수 데이터를 다룰 때 업종 종류 대분류에 해당하는 데이터만 활용



Feature Engineering

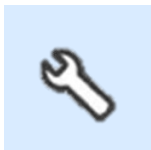
어떤 데이터를 다룰 것인가?

보험개발원의 데이터 명세서

- 1 보험 종류마다 보험 사고율의 정의가 다르기 때문에 보험 사고율 칼럼을 생성.
- 2 필요없는 칼럼 제거.

03 주요 기술 설명

Feature Engineering & PCA



PCA

: 기존 데이터의 최대의 분산을 갖는 축을 찾음으로써 정보를 최대한 보존하면서 고차원을 저차원 공간으로 변환시키는 방법 중 하나.

STEP

- 1 데이터 스케일링.
- 2 전체해서 해당 주성분(=고유벡터)의 고윳값이 차지하는 비율을 알아봄.
- 3 사이킷런의 `pca` 클래스의 `explained_variance_ratio_`를 사용하여 각 주성분에 대한 설명 분산량을 구해냄.

03 주요 기술 설명

Clustering analysis



K-means 모델 생성성

: 주어진 데이터를 k개의 클러스터로 묶는 알고리즘으로, 각 클러스터와 거리 차이의 분산을 최소화하는 방식으로 동작한다.

STEP

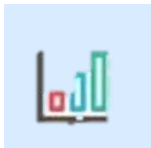
- 1 클러스터링의 수 k 정의.
- 2 각 측정값을 클러스터에 할당.
- 3 새로운 클러스터의 중심 계산.
- 4 클러스터 재분류.
- 5 경계가 변경되지 않으면 종료..



repeat

03 주요 기술 설명

Clustering analysis



군집별 소비패턴 파악

K-means clustering을 통해 만들어진 k개의 군집의 특성을 파악하는 단계.

예상 군집

- | | |
|----------------------------|----------------------------|
| 1 건강보조에 소비가 많은 집단. | 4 요식에 소비가 많은 집단. |
| 2 병원에 대한 소비가 많은 집단. | 5 자동차에 소비가 많은 집단. |
| 3 보험에 대한 소비가 많은 집단. | 6 여행_교통에 소비가 많은 집단. |

03 주요 기술 설명

Clustering analysis



군집별 소비패턴 파악

K-means clustering을 통해 만들어진 k개의 군집의 특성을 파악하는 단계.

시계열 분석

- 각 군집별 어느 시간대에 해당 업종에 대한 매출이 높은지 확인.

- 시간과 관련된 군집의 특징을 좀 더 구체화.

보험사고율 비교

- 군집별로 각각 생명보험, 장기보험, 자동차 보험에 대한 사고율의 평균 구하기.

- 도출된 결과는 차후에 개인의 소비패턴을 보고 보험을 추천해주는 마케팅 전략에 이용.



04

PoC 코드 구현

데이터 명세서와 샘플을 기반으로 예시 PoC
프로그램 구현





04 PoC 코드 구현

1. EDA

```
import pandas as pd
from pandas_profiling import ProfileReport

# 데이터셋 로드
df = pd.read_csv('dataset.csv')

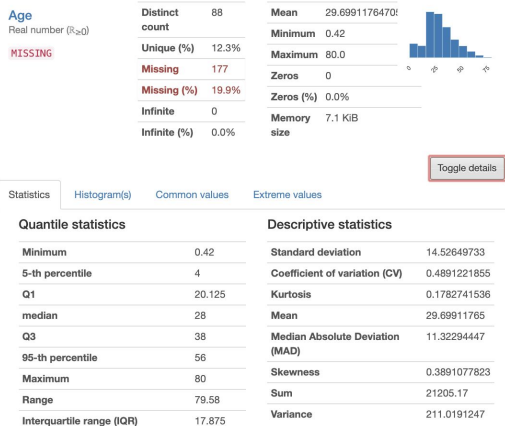
# 데이터셋 크기 확인
num_samples = len(df) # 샘플 개수
num_columns = len(df.columns) # 칼럼 개수
print(f"데이터셋 크기: {num_samples} rows, {num_columns} columns")
```

```
# 프로파일링 보고서 생성
profile = ProfileReport(df)

# 프로파일링 보고서 출력
profile.to_notebook_iframe()
```

삼성카드 명세서 관련 데이터 셋 로드 후

데이터 내 분포, 결측값, 변수 간의 관계 등
데이터 파악을 위한 판다스 Profiling
라이브러리 사용예정





04 PoC 코드 구현

1. EDA

```
# 문자열 데이터 numerical로 변환  
df['categorical_column'] = pd.factorize(df['categorical_column'])[0]
```

```
# 이상치(outliers)와 결측치(missing values) 확인  
outliers = df[(df['column'] < lower_threshold) | (df['column'] > upper_threshold)]  
missing_values = df.isnull().sum()
```

```
# 이상치 제거 (예시)  
df = df[(df['column'] >= lower_threshold) & (df['column'] <= upper_threshold)]
```

```
# 평균값으로 결측치 대체 (예시)  
mean_value = df['column'].mean()  
df['column'].fillna(mean_value, inplace=True)
```

```
# 결측치 처리 (예시)  
df = df.dropna()
```

```
# 데이터셋 크기 확인 (처리 후)  
num_samples = len(df) # 샘플 개수  
num_columns = len(df.columns) # 칼럼 개수  
print(f"데이터셋 크기: {num_samples} rows, {num_columns} columns")
```

분석을 위해 나이, 성별, 직업,
라이프스테이지 등 문자열 데이터를
numerical로 변환

결측치, 이상치 제거 또는 대체

ex) 매출액 변수가 음수값이라면 환불을
의미하는 경우이므로 이상치로 판단하여
제거해줄 수 있음





04 PoC 코드 구현

1. EDA

연령대별 매출 건수 시각화

```
age_sales_count = df.groupby('AGE')['매출건수합계'].sum()
age_sales_count.plot(kind='bar')
plt.xlabel('연령대')
plt.ylabel('매출 건수')
plt.title('연령대별 매출 건수')
plt.show()
```



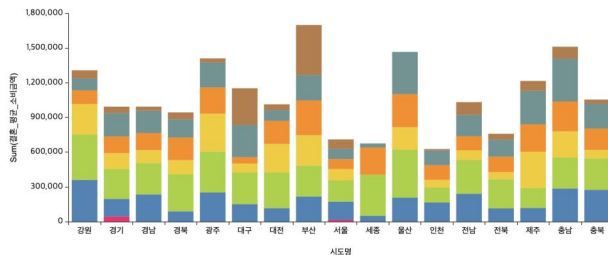
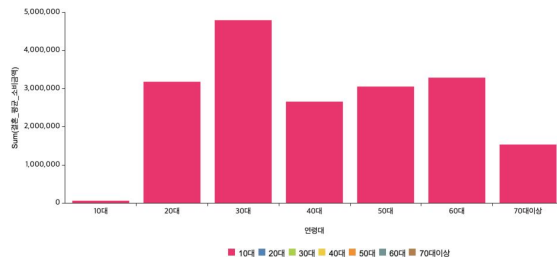
업종 분류별 매출 건수 시각화

```
industry_sales_count = df.groupby('업종분류')['매출건수합계'].sum()
industry_sales_count.plot(kind='bar')
plt.xlabel('업종 분류')
plt.ylabel('매출 건수')
plt.title('업종 분류별 매출 건수')
plt.show()
```

주소지별 매출 건수 시각화

```
address_sales_count = df.groupby('주소지')['매출건수합계'].sum()
address_sales_count.plot(kind='bar')
plt.xlabel('주소지')
plt.ylabel('매출 건수')
plt.title('주소지별 매출 건수')
plt.show()
```

차트 시각화를 통해 연령대별, 업종
분류별, 주소지 별 매출 금액에 대한
분포 확인





04 PoC 코드 구현

2. Feature Engineering

2.1 삼성카드 데이터 전처리

총 394개의 칼럼 중 군집별 소비 성향 분석에 불필요한 칼럼 313개 삭제

군집 분석에서 시간에 따른 매출건수는 고려하지 않기 때문에 업종별 총 매출건수로

groupby 처리

```
#삼성카드데이터 불필요한 고객정보 칼럼삭제
delete_card_info = ['GENDER', 'JOB', 'HOM_MGPO', 'HOM_SGG', 'OFFI_MGPO', 'OFFI_SGG', 'ONEPER_GD_C',
                    'UMRD_SCORE', 'WEDD_EXP_SCORE', 'CHDB_EXP_SCORE', 'PSCH_OCH_SCORE', 'ESTUD_OCH_SCORE',
                    'MHSSTD_OCH_SCORE', 'CSTUD_OCH_SCORE', 'OCH_WEDD_EXP_SCORE', 'HOUSEWF_SCORE',
                    'ONL_PRSN', 'ERADP_TNDC', 'PRMM_CSM_TNDC', 'MDLV_PRSN', 'PET_PRSN',
                    'ODS_ACTI_INRT_TNDC', 'FRN_TRV_INRT_TNDC', 'INTERI_SLNG_TNDC', 'INTERI_REMDL_TNDC',

#매출금액 관련 칼럼삭제
sales_amount_columns = [sales_amount for sales_amount in df_card.columns if '매출금액' in sales_amount]
#중, 소분류 관련 칼럼 삭제
middle_small_category = ['매출건수합계_대중교통_A', '매출건수합계_대중교통_B', '매출건수합계_대중교통_C', '매출건수합계_대중교통_D',
                          '매출건수합계_대중교통_B', '매출건수합계_대중교통_F', '매출건수합계_대중교통_G', '매출건수합계_차량구매_A',
                          '매출건수합계_차량구매_B', '매출건수합계_차량구매_C', '매출건수합계_차량구매_D', '매출건수합계_차량구매_E',
                          '매출건수합계_차량구매_F', '매출건수합계_차량구매_G', '매출건수합계_주차_A', '매출건수합계_주차_B',
                          '매출건수합계_주차_C', '매출건수합계_주차_D', '매출건수합계_주차_E', '매출건수합계_주차_F', '매출건수합계_주차_G',
                          '매출건수합계_주유_A', '매출건수합계_주유_B', '매출건수합계_주유_C', '매출건수합계_주유_D', '매출건수합계_주유_E',
                          '매출건수합계_주유_F', '매출건수합계_주유_G', '매출건수합계_병원_A', '매출건수합계_병원_B', '매출건수합계_병원_C',
                          '매출건수합계_병원_D', '매출건수합계_병원_E', '매출건수합계_병원_F', '매출건수합계_병원_G', '매출건수합계_건강보조_A']
```

#각 업종별로 시간대 매출건수 통합 칼럼생성

```
df_card['총매출건수합계_요식_A'] = df_card[['매출건수합계_요식_A', '매출건수합계_요식_B',
                                              '매출건수합계_요식_C', '매출건수합계_요식_D', '매출건수합계_요식_E',
                                              '매출건수합계_요식_F', '매출건수합계_요식_G']].sum(axis=1)
df_card['총매출건수합계_자동차'] = df_card[['매출건수합계_자동차_A', '매출건수합계_자동차_B',
                                              '매출건수합계_자동차_C', '매출건수합계_자동차_D', '매출건수합계_자동차_E',
                                              '매출건수합계_자동차_F', '매출건수합계_자동차_G']].sum(axis=1)
df_card['총매출건수합계_교육'] = df_card[['매출건수합계_교육_A', '매출건수합계_교육_B',
                                              '매출건수합계_교육_C', '매출건수합계_교육_D', '매출건수합계_교육_E',
                                              '매출건수합계_교육_F', '매출건수합계_교육_G']].sum(axis=1)
df_card['총매출건수합계_건강'] = df_card[['매출건수합계_건강_A', '매출건수합계_건강_B',
                                              '매출건수합계_건강_C', '매출건수합계_건강_D', '매출건수합계_건강_E',
                                              '매출건수합계_건강_F', '매출건수합계_건강_G']].sum(axis=1)
df_card['총매출건수합계_여가'] = df_card[['매출건수합계_여가_A', '매출건수합계_여가_B',
                                              '매출건수합계_여가_C', '매출건수합계_여가_D', '매출건수합계_여가_E',
                                              '매출건수합계_여가_F', '매출건수합계_여가_G']].sum(axis=1)
df_card['총매출건수합계_생활'] = df_card[['매출건수합계_생활_A', '매출건수합계_생활_B',
                                              '매출건수합계_생활_C', '매출건수합계_생활_D', '매출건수합계_생활_E',
                                              '매출건수합계_생활_F', '매출건수합계_생활_G']].sum(axis=1)
df_card['총매출건수합계_유통'] = df_card[['매출건수합계_유통_A', '매출건수합계_유통_B',
                                              '매출건수합계_유통_C', '매출건수합계_유통_D', '매출건수합계_유통_E',
                                              '매출건수합계_유통_F', '매출건수합계_유통_G']].sum(axis=1)
df_card['총매출건수합계_쇼핑'] = df_card[['매출건수합계_쇼핑_A', '매출건수합계_쇼핑_B',
                                              '매출건수합계_쇼핑_C', '매출건수합계_쇼핑_D', '매출건수합계_쇼핑_E',
                                              '매출건수합계_쇼핑_F', '매출건수합계_쇼핑_G']].sum(axis=1)
df_card['총매출건수합계_여행'] = df_card[['매출건수합계_여행_A', '매출건수합계_여행_B',
                                              '매출건수합계_여행_C', '매출건수합계_여행_D', '매출건수합계_여행_E',
                                              '매출건수합계_여행_F', '매출건수합계_여행_G']].sum(axis=1)
```





2. Feature Engineering

2.1 삼성카드 데이터 전처리

기준년 도	AGE	INCOME	EST_LFSTG	총매출건수합 계_요식	총매출건수합 계_자동차	총매출건수합 계_교육	총매출건수합 계_건강	총매출건수합 계_여가	총매출건수합 계_생활	총매출건수합 계_유흥	총매출건수합 계_쇼핑	총매출건수합 계_여행	
0	2018	A	A	A	73	57	89	133	68	142	127	160	142
1	2019	B	B	B	98	116	96	85	70	102	82	92	90
2	2020	C	C	C	106	109	156	69	84	114	85	76	109
3	2021	D	D	D	117	104	122	120	99	83	96	145	76
4	2018	E	E	E	94	50	116	113	114	129	114	80	94

※ 실제 데이터가 아닌 예시용
데이터임

나이, 수입, 라이프스타이지, 업종별 총 매출건수로 구성된 데이터로 군집 분석
수행 예정





3. Cluster Analysis

2.2 보험개발원 데이터 전처리

총 11개의 칼럼 중 보험 사고율 분석에 불필요한 5개의 칼럼 제거

보험 사고율을 구하기 위해 문자형의 데이터를 float형으로 변환

```
# 보험 데이터 분석에 불필요한 칼럼 삭제
```

```
delete_insurance_columns = ['성별', '연령', '보장내용', '자동차국외산구분코드', '자동차보험가입경력']  
df_insurance.drop(delete_insurance_columns, axis=1, inplace=True)
```

```
# 보험료, 지급금액 숫자형으로 변환
```

```
df_insurance['보험료'] = df_insurance['보험료'].str.replace(',', '').astype(float)  
df_insurance['지급금액'] = df_insurance['지급금액'].str.replace(',', '').astype(float)
```





2. Feature Engineering

2.2 보험 데이터 전처리

보험 종목별 사고율을 계산하는 함수를 통해 보험 사고율 칼럼 생성

사고율을 구하는 데 사용된 칼럼들은 Redundant한 칼럼으로 처리되므로 제거

```
#보험 사고율 칼럼 생성하는 함수

def calculate_insurance_ratio(df):
    if df['보험종류'] == 1:
        return df['사고건수']/df['계약건수']
    elif df['보험종류'] in [2,3]:
        return df['지급금액']/df['보험료']
    else:
        return None

df_insurance['보험사고율'] = df_insurance.apply(calculate_insurance_ratio, axis=1)

# 계약건수, 보험료, 사고건수, 지급금액 칼럼 삭제
delete_columns = ['계약건수', '보험료', '사고건수', '지급금액']
df_insurance.drop(delete_columns, axis=1, inplace=True)
```





2. Feature Engineering

2.2 보험 데이터 전처리

	기준년도	보험종류	보험사고율
0	2018	1	0.0
1	2019	1	0.0
2	2020	1	1.0
3	2021	1	0.0
4	2018	2	0.0

보험종류, 보험사고율로 구성된 데이터로
군집별 소비성향에 따른 보험 사고율 분석
예정

※ 실제 데이터가 아닌 예시용
데이터임





2. Feature Engineering

2.3 삼성카드, 보험개발원 데이터 결합

PCA 분석을 위해 기준년도를 기준으로 데이터 결합

	AGE	INCOME	EST_LFSTG	총매출건수합 계_요식	총매출건수합 계_자동차	총매출건수합 계_교육	총매출건수합 계_건강	총매출건수합 계_여가	총매출건수합 계_생활	총매출건수합 계_유흥	총매출건수합 계_쇼핑	총매출건수합 계_여행	보험 종류	보험사고 율
0	A	A	A	73	57	89	133	68	142	127	160	142	1	0.00
1	B	B	B	98	116	96	85	70	102	82	92	90	1	0.00
2	C	C	C	106	109	156	69	84	114	85	76	109	1	1.00
3	D	D	D	117	104	122	120	99	83	96	145	76	1	0.00
4	E	E	E	94	50	116	113	114	129	114	80	94	2	0.00
5	F	F	A	126	91	112	150	136	89	124	105	96	2	40.00
6	G	G	B	154	113	82	140	75	119	62	136	108	3	35.00
7	H	H	C	65	140	131	107	133	88	145	81	72	3	43.75
8	I	I	D	85	75	80	101	86	107	119	108	117	3	0.00
9	B	B	E	64	93	112	100	82	92	152	153	77	1	0.00

※ 실제 데이터가 아닌 예시용
데이터임





2. Feature Engineering

2.4 PCA

PCA 분석을 위해 삼성 카드 데이터 문자열 카테고리 칼럼(나이, 수입, 라이프 스타이지)을 숫자형으로 변환

보험사고율(종속변수)을 제외한 나머지 칼럼 스케일링 처리

```
# 스케일링을 위해서 속성값이 A~I인 칼럼을 숫자로 변환
mapping = {
    'A': 1,
    'B': 2,
    'C': 3,
    'D': 4,
    'E': 5,
    'F': 6,
    'G': 7,
    'H': 8,
    'I': 9}

df[['AGE', 'INCOME', 'EST_LFSTG']] = df[['AGE', 'INCOME', 'EST_LFSTG']].replace(mapping)

#데이터 스케일링
from sklearn.preprocessing import StandardScaler
x = df.drop(['보험사고율'], axis=1).values
x= StandardScaler().fit_transform(x)
features = ['AGE', 'INCOME', 'EST_LFSTG', '총매출건수합계_요식', '총매출건수합계_자동차',
'총매출건수합계_교육', '총매출건수합계_건강', '총매출건수합계_여가', '총매출건수합계_생활', '총매출건수합계_유흥',
'총매출건수합계_쇼핑', '총매출건수합계_여행', '보험종류']
```





2. Feature Engineering

2.4 PCA

주성분(n_components)를 5,6,7로 설정했을 때 각 누적 분산량을 비교

주성분이 6개일 때 전체 분산의 약 96%를 설명

7개일 때부터는 설명가능한 분산량이 약 0.02밖에 증가하지 않기 때문에 6개로 결정

```
#PCA 실행
from sklearn.decomposition import PCA
pca = PCA(n_components=6) #6개로 설정
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data=principalComponents, columns = ['principal component1', 'principal component2',
                                                                'principal component3', 'principal component4',
                                                                'principal component5', 'principal component6'])

#누적 설명 분산량 확인
print(pca.explained_variance_ratio_)
print(sum(pca.explained_variance_ratio_))
```

```
[0.30201501 0.2381027  0.15267139 0.11726854 0.08421189] → 주성분 5개, 분산량 약 0.89
0.8942695317031315
```

```
[0.30201501 0.2381027  0.15267139 0.11726854 0.08421189 0.06375508] → 주성분 6개, 분산량 약 0.96
0.9580246146936846
```

```
[0.30201501 0.2381027  0.15267139 0.11726854 0.08421189 0.06375508
 0.02273667] → 주성분 7개, 분산량 약 0.98
0.9807612881161143
```

※ 샘플 데이터 기반이기 때문에 실제 데이터 내에서는 결과가 달라질 수 있음





3. Cluster Analysis

3. K-means clustering

실제 데이터 분석 후 `elbow_point`로 군집의 개수(`n_cluster`)를 결정한 후 각각의 군집을 나눠 데이터 저장

```
#데이터 로드
df = pd.read_csv('dataset.csv')

#dataset.csv에 대해서 클러스터링 진행
kmeans = KMeans(n_cluster=4, random_state=0) #여기서 n_cluster는 elbow_point를 이용해 결정할 것이다.
clusters = kmeans.fit(df)

#클러스터링 변수인 clusters 값을 원본 데이터인 'dataset' 내에 넣기
dataset['cluster'] = clusters.labels_ #어떤 클러스터가 만들어졌는지는 labels_ 메서드를 통해 알 수 있다.
df.head()

#클러스터를 기준으로 데이터 갯수 세기
df.groupby('cluster').count()

#군집 나눠서 저장하기
df_0 = df[df['cluster']==0]
df_1 = df[df['cluster']==1]
df_2 = df[df['cluster']==2]
df_3 = df[df['cluster']==3]
```

※ 샘플 데이터 기반이기 때문에 실제 데이터 내에서는 결과가 달라질 수 있음





04 PoC 코드 구현

3. Cluster Analysis

3. K-means clustering

시각화를 통해 군집별 연령대, 지역, 라이프스타일에 대한 분포를 확인

군집별 보험종목당 사고율의 평균을 구해 해당 군집 내에서 가장 사고율이 높은 보험이 무엇인지 분석할 예정

```
#군집별 연령대 분포 확인하기  
#군집 1  
sns.countplot(df_0["AGE"])  
plt.show
```

```
#군집2  
sns.countplot(df_1["AGE"])  
plt.show
```

```
#군집3  
sns.countplot(df_2["AGE"])  
plt.show
```

```
#군집4  
sns.countplot(df_3["AGE"])  
plt.show
```

```
#군집별 지역 확인하기  
#군집 1  
sns.countplot(df_0["HOM_MGPO"])  
plt.show
```

```
#군집2  
sns.countplot(df_1["HOM_MGPO"])  
plt.show
```

```
#군집3  
sns.countplot(df_2["HOM_MGPO"])  
plt.show
```

```
#군집4  
sns.countplot(df_3["HOM_MGPO"])  
plt.show
```

```
#군집별 라이프스타일 확인하기  
#군집 1  
sns.countplot(df_0["EST_LFSTG"])  
plt.show
```

```
#군집2  
sns.countplot(df_1["EST_LFSTG"])  
plt.show
```

```
#군집3  
sns.countplot(df_2["EST_LFSTG"])  
plt.show
```

```
#군집4  
sns.countplot(df_3["EST_LFSTG"])  
plt.show
```

```
#그룹별 특징을 알아보자 -- 그룹별 평균값  
df.groupby('cluster').mean()
```



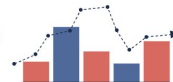
05

앞으로의 계획

본선진출 시 후원기관 제공 데이터를
원격분석환경에서 다룰 예정




카드소비형태에 따른 보험 종목별 사고율 분석

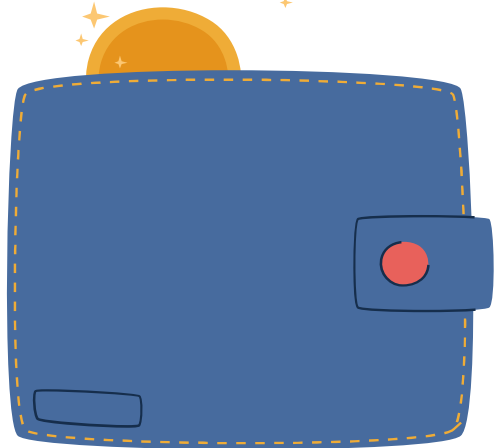
이제희, 이지호, 조민서



Planning for the final competition in August

7월 마지막 주 발표되는 예선 결과 확인 후, 8월 동안 본선 1차 심사물 준비 예정

Steps / Week	1	2	3	4	비고
1. 데이터 분석 및 EDA	 1주차~2주차 #1,2				Profiling 툴 사용
2. 데이터 전처리 및 PCA					이상치, 결측치 제거 및 Feature engineering
3. 군집 분석		 2주차~3주차 #3,4			K-means 모델링
4. 군집별 소비 성향 분석					Tableau 툴 사용
5. 군집별 보험 종목 사고율 분석			 3주차~4주차 #5,6		
6. 소비 성향에 따른 보험 추천 서비스 기획					도출된 인사이트를 바탕으로 최종 결과물 제출



Thanks!



CREDITS: This presentation template was created by Slidesgo, and includes icons by Flaticon, and infographics & images by Freepik