

한·미 양국간의 금리상관성 분석 및 개인별 소비패턴분석을 통한 예·적금추천앱

빅데이터 Final 프로젝트

4팀

류재현/박준영/서동준/심예은

목차

1. 프로젝트 개요
2. 프로세스 일정 및 분담
3. 요구사항 정의서, 테이블 정의서, ERD
4. 데이터 분석
 - 정기 예금 기반
 - 단기 적금 기반
 - 물품 추천 기반
5. 앱 구현
 - 전체 흐름도
 - 성별, 연령대 선택 화면
 - 목록 화면
 - 정기예금 선택
 - 단기적금 선택
 - 물품 선택
 - 선택한 목록에 따른 화면
 - 정기예금 상품 추천 화면
 - 단기적금 상품 추천 화면
 - 물품을 사기위한 적금 상품 추천 화면
 - 상품 상세 화면

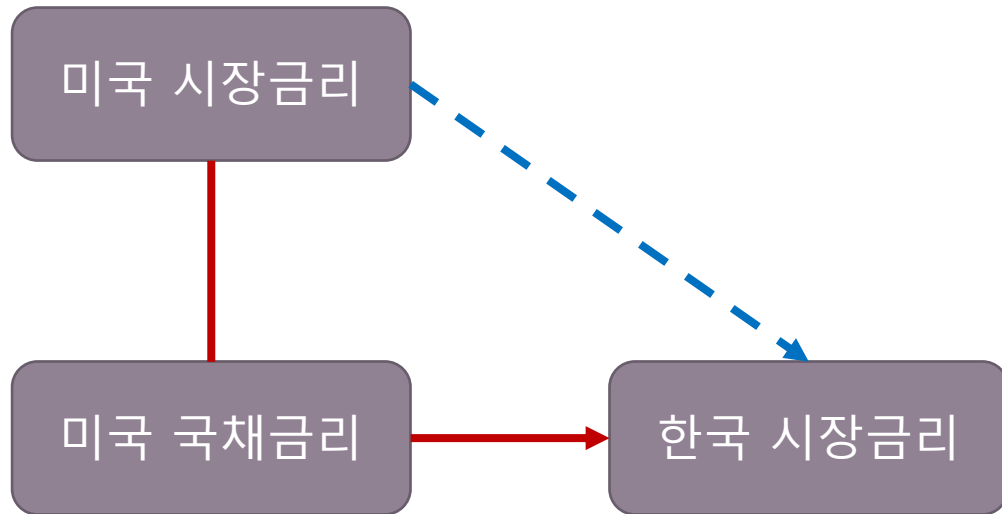


01

프로젝트 개요

프로젝트 개요

-개발 배경(or 목표)



스토리 술술 적어주세요~

-개발 환경

android
studio 

 **Kotlin**

 **spring
Boot**

 ANACONDA.

 **MySQL**

02
프로젝트 일정
및 분담

프로세스 일정 및 분담

[illegible]

03

요구사항 정의서

테이블 정의서

ERD

요구사항 정의서, ERD

[illegible]

작성일자	2021-12-18				
버전	1.2(성별 추가)				
No	구분	요구사항ID	요구사항 명	요구사항 상세 설명	비고
1	모바일	REQ-001	성별선택 화면	남녀(2개버튼)	
2	모바일	REQ-002	연령선택 화면	REQ-001 선택 후 10대 ~ 70대, 10대미만, 70대초과(7개버튼)	인적사항 테이블(성별컬럼, 나이컬럼)
3	모바일	REQ-003	연령별 상품리스트 화면	REQ-002 선택 후 연령별 희망하는 상품리스트	연령별 소비패턴 분석 DATA(비밀번호)
4	모바일	REQ-004	적금상품 추천하는 화면	REQ-003 선택 후 상품별 추천 적금 리스트	
5	모바일	REQ-005	적금상품 안내하는 화면	REQ-004 선택 후 나의 위치와 가장 가까운 지점 출력(카카오api), 상품정보, 지점 연락처 출력	

04

데이터 분석 - (1) 정기 예금

- 데이터 수집
- 데이터 전처리
- 데이터 분석 및 시각화

정기 예금 추천 타겟 :

1. 미국 금리 1년 후의 변동을 예측 (LSTM 사용)
-1956~2021년 사이의 실질 이자율 데이터를 이용하여
2022년의 금리 예측

2. 미국과 한국 시장금리 간의 상관관계 분석
-1976~2020년 사이의 한국, 미국 시장금리 데이터를
이용하여 상관관계 분석

자료출처: kosis 주요국제금리(OECD)
주요국제금리(OECD) (kosis.kr)

FRED(St. Louis 연방준비은행의 연구
부서에서 관리하는 DB)

1. 미국 금리 예측 DATA

DATE	명목이자율	물가상승률	실질이자율
1956-01-01	2.9	1.5	1.4
1956-02-01	2.84	1.5	1.34
1956-03-01	2.96	1.5	1.46
1956-04-01	3.18	1.5	1.68
1956-05-01	3.07	1.5	1.57
1956-06-01	3	1.5	1.5
1956-07-01	3.11	1.5	1.61
1956-08-01	3.33	1.5	1.83
1956-09-01	3.38	1.5	1.88
1956-10-01	3.34	1.5	1.84
1956-11-01	3.49	1.5	1.99
1956-12-01	3.59	1.5	2.09
1957-01-01	3.46	3.3	0.16
1957-02-01	3.34	3.3	0.04
1957-03-01	3.41	3.3	0.11
1957-04-01	3.48	3.3	0.18
⋮			
2021-07-01	1.32	4.5	-3.18
2021-08-01	1.28	4.5	-3.22
2021-09-01	1.37	4.5	-3.13
2021-10-01	1.58	4.5	-2.92
2021-11-01	1.56	4.5	-2.94

2. 상관분석 DATA

DATE	한국 시장금리	미국 시장금리
1976-01-01	8.95	5.05
1977-01-01	18.09	5.54
1978-01-01	19.32	7.93
1979-01-01	18.86	11.19
1980-01-01	22.85	13.36
1981-01-01	18.14	16.38
1982-01-01	14.18	12.26
1983-01-01	13	9.09
1984-01-01	11.39	10.23
1985-01-01	9.35	8.1
1986-01-01	9.7	6.81
1987-01-01	8.93	6.66
1988-01-01	9.62	7.57
1989-01-01	13.28	9.22
1990-01-01	14.03	8.1
1991-01-01	17.03	5.69
⋮		
2016-01-01	1.34	0.4
2017-01-01	1.26	1
2018-01-01	1.52	1.83
2019-01-01	1.59	2.16
2020-01-01	0.7	0.38

04 정기예금(1)LSTM -데이터 전처리

	명목이자율	물가상승률	실질이자율
DATE			
1956-01-01	2.90	1.5	1.40
1956-02-01	2.84	1.5	1.34
1956-03-01	2.96	1.5	1.46
1956-04-01	3.18	1.5	1.68
1956-05-01	3.07	1.5	1.57
1956-06-01	3.00	1.5	1.50
1956-07-01	3.11	1.5	1.61
1956-08-01	3.33	1.5	1.83
1956-09-01	3.38	1.5	1.88
1956-10-01	3.34	1.5	1.84

	실질이자율
DATE	
1956-01-01	1.40
1956-02-01	1.34
1956-03-01	1.46
1956-04-01	1.68
1956-05-01	1.57
1956-06-01	1.50
1956-07-01	1.61
1956-08-01	1.83
1956-09-01	1.88
1956-10-01	1.84

정규화 - 분산 줄이기 (스케일링)

```
array([[0.41343284],  
       [0.40895522],  
       [0.41791045],  
       [0.43432836],  
       [0.4261194 ],  
       [0.42089552],  
       [0.42910448],  
       [0.44552239],  
       [0.44925373],  
       [0.44626866],  
       [0.45746269],  
       [0.46492537],
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
DatetimeIndex: 791 entries, 1956-01-01 to 2021-11-01  
Data columns (total 1 columns):  
 #   Column  Non-Null Count  Dtype  
---  ---  
 0   실질이자율    791 non-null    float64  
dtypes: float64(1)  
memory usage: 12.4 KB
```

Train data, Test data 8:2로 나누기 (65년 데이터 중, 52년 13년 분할)

```
x_train, x_test, y_train, y_test = train_test_split(df, df['실질이자율'],  
                                                    test_size=0.2, random_state=0, shuffle=False)
```

```
x_train.shape, x_test.shape
```

```
((632, 1), (159, 1))
```

```
y_train.shape, y_test.shape
```

```
((632,), (159,))
```

04 정기예금(1)LSTM -데이터 분석 -LSTM 모델 생성

```
def windowed_dataset(series, window_size, batch_size, shuffle):
    series = tf.expand_dims(series, axis=-1)
    ds = tf.data.Dataset.from_tensor_slices(series)
    ds = ds.window(window_size + 1, shift=1, drop_remainder=True)
    ds = ds.flat_map(lambda w: w.batch(window_size + 1))
    if shuffle:
        ds = ds.shuffle(1000)
    ds = ds.map(lambda w: (w[:-1], w[-1]))
    return ds.batch(batch_size).prefetch(1)
```

```
WINDOW_SIZE=20
BATCH_SIZE=32
```

WINDOW_SIZE와 BATCH_SIZE를 각 데이터셋에 적용

```
train_data = windowed_dataset(y_train, WINDOW_SIZE, BATCH_SIZE, True)
test_data = windowed_dataset(y_test, WINDOW_SIZE, BATCH_SIZE, False)
```

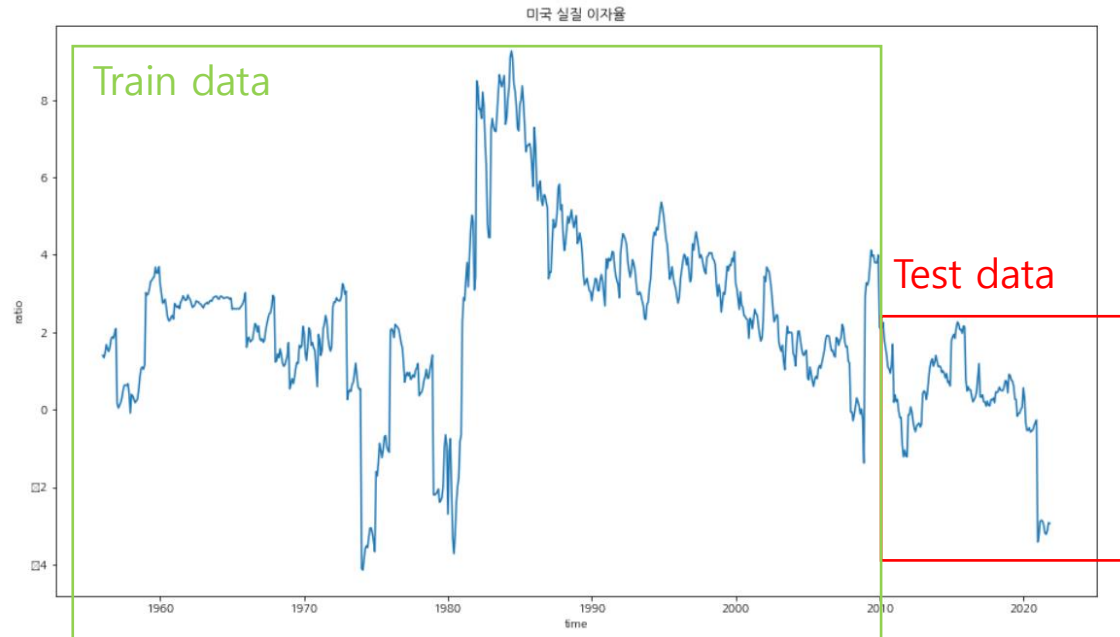
```
for data in train_data.take(1):
    print(f'데이터셋(X) 구성(batch_size, window_size, feature갯수): {data[0].shape}')
    print(f'데이터셋(Y) 구성(batch_size, window_size, feature갯수): {data[1].shape}')
```

데이터셋(X) 구성(batch_size, window_size, feature갯수): (32, 20, 1)
데이터셋(Y) 구성(batch_size, window_size, feature갯수): (32, 1)

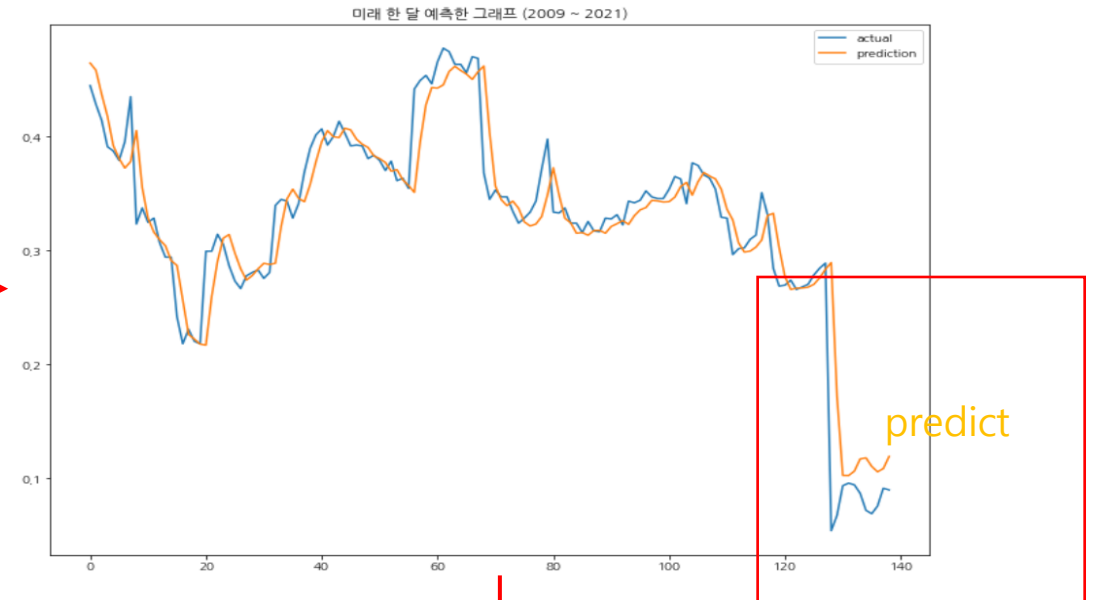
```
model = Sequential([
    Conv1D(filters=32, kernel_size=5,
           padding="causal",
           activation="relu",
           input_shape=[WINDOW_SIZE, 1]),
    LSTM(16, activation='relu'),
    Dense(16, activation="relu"),
    Dense(1),
])
```

```
19/20 [=====>...] - ETA: 0s - loss: 0.0010 - mse: 0.0020
Epoch 00046: val_loss did not improve from 0.00055
20/20 [=====] - 0s 14ms/step - loss: 0.0010 - mse: 0.0020 - val_loss: 5.7872e-04 - val_mse: 0.0012
Epoch 47/50
19/20 [=====>...] - ETA: 0s - loss: 9.9475e-04 - mse: 0.0020
Epoch 00047: val_loss did not improve from 0.00055
20/20 [=====] - 0s 12ms/step - loss: 0.0010 - mse: 0.0020 - val_loss: 0.0010 - val_mse: 0.0020
Epoch 48/50
19/20 [=====>...] - ETA: 0s - loss: 0.0013 - mse: 0.0026
Epoch 00048: val_loss did not improve from 0.00055
20/20 [=====] - 0s 17ms/step - loss: 0.0013 - mse: 0.0026 - val_loss: 7.3913e-04 - val_mse: 0.0015
Epoch 49/50
19/20 [=====>...] - ETA: 0s - loss: 9.6413e-04 - mse: 0.0019
Epoch 00049: val_loss did not improve from 0.00055
20/20 [=====] - 0s 12ms/step - loss: 9.6034e-04 - mse: 0.0019 - val_loss: 7.2019e-04 - val_mse: 0.0014
Epoch 50/50
16/20 [=====>.....] - ETA: 0s - loss: 0.0010 - mse: 0.0021
Epoch 00050: val_loss did not improve from 0.00055
20/20 [=====] - 0s 14ms/step - loss: 9.9507e-04 - mse: 0.0020 - val_loss: 6.2556e-04 - val_mse: 0.0013
```

04 정기예금(1)LSTM -데이터 분석 및 시각화

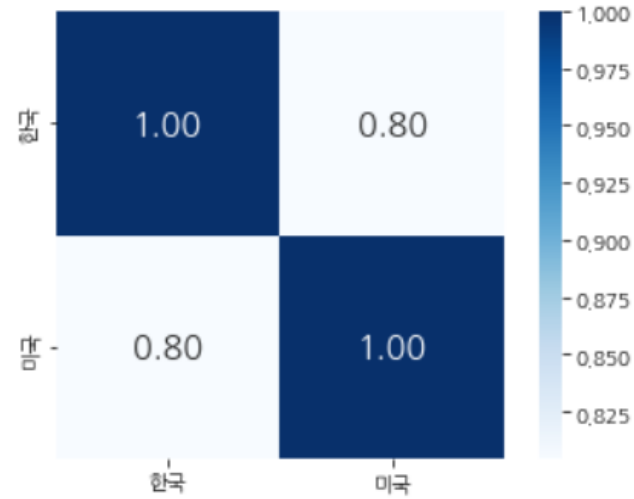


결론 - 2022년의 미국 실질 이자율은
상승할 것으로 예측되었다.

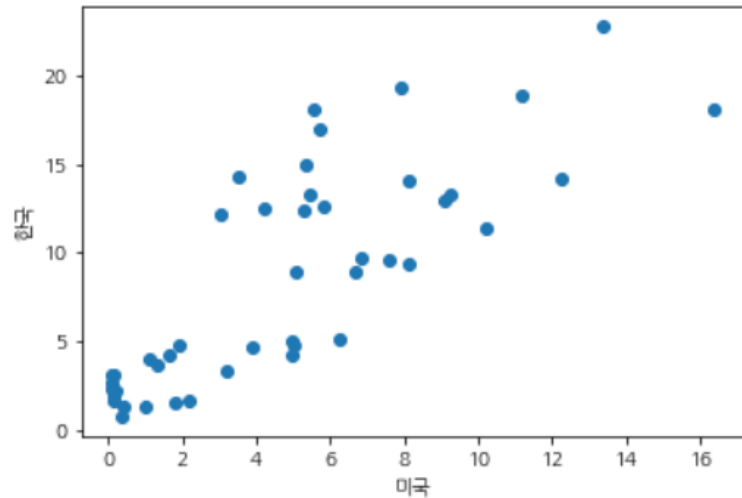


04 정기예금(2)상관관계 분석 - 데이터 분석 및 시각화

미국		한국	
DATE		DATE	
1976-01-01	5.05	1976-01-01	8.95
1977-01-01	5.54	1977-01-01	18.09
1978-01-01	7.93	1978-01-01	19.32
1979-01-01	11.19	1979-01-01	18.86
1980-01-01	13.36	1980-01-01	22.85
1981-01-01	16.38	1981-01-01	18.14
1982-01-01	12.26	1982-01-01	14.18
1983-01-01	9.09	1983-01-01	13.00
1984-01-01	10.23	1984-01-01	11.39
1985-01-01	8.10	1985-01-01	9.35



	한국	미국
한국	1.000000	0.804186
미국	0.804186	1.000000



결론 - 상관계수가 **0.804186**으로,
미국 금리와 한국 금리는
상관관계가 강하다.

04

데이터 분석 - (2)단기 적금

- 데이터 수집
- 데이터 전처리
- 데이터 분석 및 시각화

단기 적금 추천

타겟 :

- 1. 성별, 연령별 소비가 많은 카테고리 순으로 정렬하여 시각화 후 ,
- 2. 소비를 줄일 수 있는 카테고리 설문조사지를 돌려서 유추 -100명을 대상으로 조사함

자료출처 : 신한카드 카드사용내역 통계자료

1. 카드 소비패턴 DATA

업종분류	업종	성별	연령대	카드이용금액
가전/가구	가전/가구	남자	20대 미만	51072179
교육/학원	교육	남자	20대 미만	148918432
유통	기타유통	남자	20대 미만	6821781
유통	백화점	남자	20대 미만	157335767
의료	병원	남자	20대 미만	166770482
여행/교통	숙박	남자	20대 미만	53672060
스포츠/문	스포츠/레	남자	20대 미만	858875565
요식/유흥	식품	남자	20대 미만	3692217277
여행/교통	여행	남자	20대 미만	18296
온라인거래	온라인거래	남자	20대 미만	3175559
요식/유흥	유흥	남자	20대 미만	51776307
가정생활/인테리어	인테리어	남자	20대 미만	47857819
자동차	자동차	남자	20대 미만	68439289
요식/유흥	제과/커피	남자	20대 미만	388826271
의류/잡화	패션의류	남자	20대 미만	125336387
의류/잡화	패션잡화	남자	20대 미만	50293576

⋮

요식/유흥	제과/커피	여자	70대초과	2481238127
의류/잡화	패션의류	여자	70대초과	8008036109
의류/잡화	패션잡화	여자	70대초과	3948047251
유통	편의점/할	여자	70대초과	70991612788
미용	화장품/미	여자	70대초과	6911455194

2. 설문조사 DATA

연령대	화장품 / 미용 부분	제과 / 커피 / 패스트푸드	패션잡화	여행 / 숙박 / 스포츠	교육
20대	5	5	1	1	2
20대	5	5	5	5	5
30대	5	3	5	5	5
20대	3	5	5	3	4
20대	4	5	3	4	1
20대	5	5	5	5	5
20대	3	1	3	3	3
20대	3	2	2	4	4
20대	5	3	3	2	3
50대	3	4	5	4	2
20대	1	1	1	1	5
30대	2	2	2	2	4
20대	4	4	3	2	5
20대	4	2	1	4	3
20대	3	2	4	4	5
20대	4	4	2	2	3

⋮

30대	3	1	2	2	3
20대	1	2	1	2	4
30대	5	5	4	4	2
20대	4	2	4	3	4
60대 이상	5	5	4	3	3

04 단기적금(1)소비패턴 분석 -데이터 전처리

	업종분류	업종	성별	연령대	카드이용금액
0	가전/가구	가전/가구	남자	20대 미만	5.107218e+07
1	교육/학원	교육	남자	20대 미만	1.489184e+08
2	유통	기타유통	남자	20대 미만	6.821781e+06
3	유통	백화점	남자	20대 미만	1.573358e+08
4	의료	병원	남자	20대 미만	1.667705e+08
...
195	의류/잡화	패션잡화	여자	40대	1.372578e+10
196	유통	편의점/할인점	여자	40대	3.100000e+11
197	미용	화장품/미용	여자	40대	3.246565e+10
198	가전/가구	가전/가구	여자	50대	3.309343e+10
199	교육/학원	교육	여자	50대	4.694030e+10

	성연령	카테고리	소비금액
0	20대 미만 남자	가전/가구	5.107218e+07
16	20대 미만 남자	편의점/할인점	1.458310e+09
15	20대 미만 남자	패션잡화	5.029358e+07
14	20대 미만 남자	패션의류	1.253364e+08
13	20대 미만 남자	제과/커피/패스트푸드	3.888263e+08
...
172	30대 여자	유흥	1.386792e+09
171	30대 여자	온라인거래	3.962729e+09
170	30대 여자	여행	8.127085e+08
169	30대 여자	식품	4.110000e+11
168	30대 여자	스포츠/레저	4.037246e+10

```
df['카테고리'].unique()
```

```
array(['가전/가구', '편의점/할인점', '패션잡화', '패션의류', '제과/커피/패스트푸드', '자동차',  
      '인테리어/서비스', '유흥', '온라인거래', '화장품/미용', '식품', '스포츠/레저', '숙박', '병원',  
      '백화점', '기타유통', '교육', '여행'], dtype=object)
```

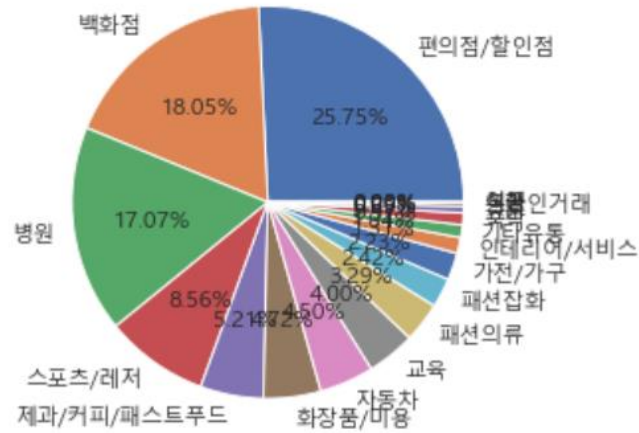
#20대 미만 남자 ~ 70대 초과 여자까지 있음

```
df['성연령'].unique()
```

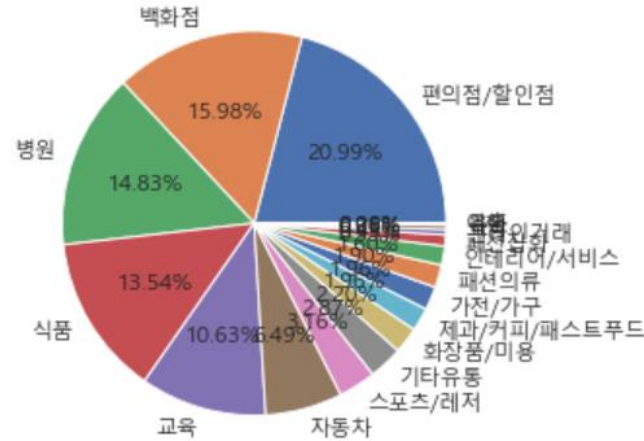
```
array([' 20대 미만 남자', ' 20대 미만 여자', '20대 남자', '20대 여자', '30대 남자', '30대 여자',  
      '40대 남자', '40대 여자', '50대 남자', '50대 여자', '60대 남자', '60대 여자',  
      '70대초과 남자', '70대초과 여자'], dtype=object)
```

04 단기적금(1)소비패턴 분석 – 데이터 분석 및 시각화 (성별, 연령에 따라)

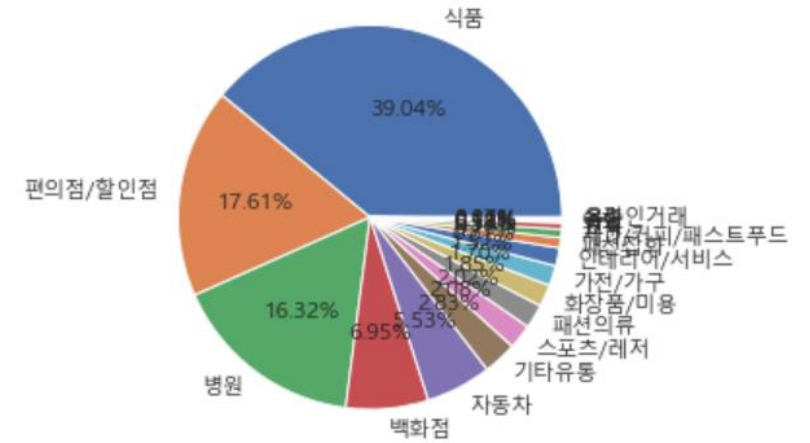
20대 여자의 소비 카테고리 순위



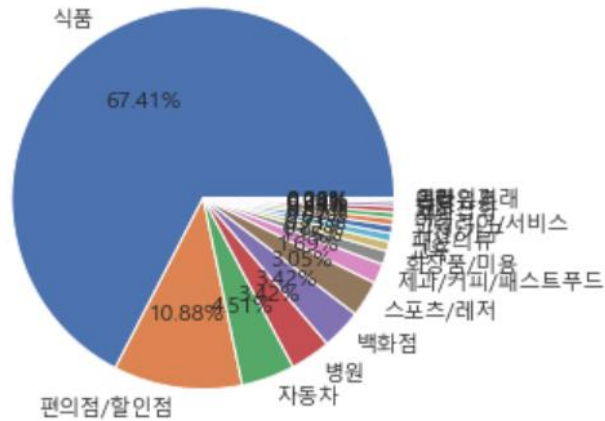
40대 여자의 소비 카테고리 순위



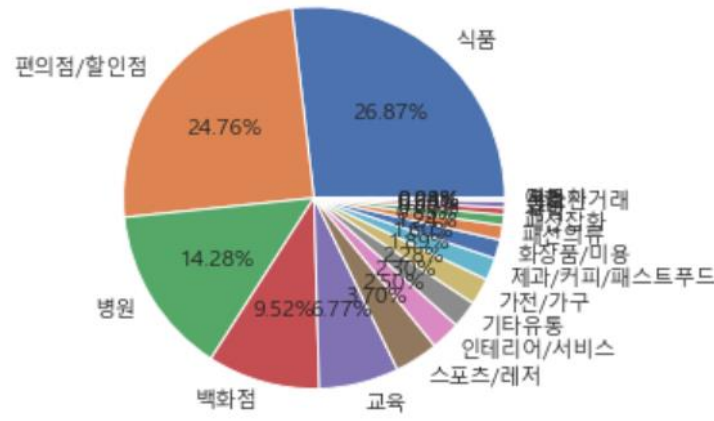
60대 여자의 소비 카테고리 순위



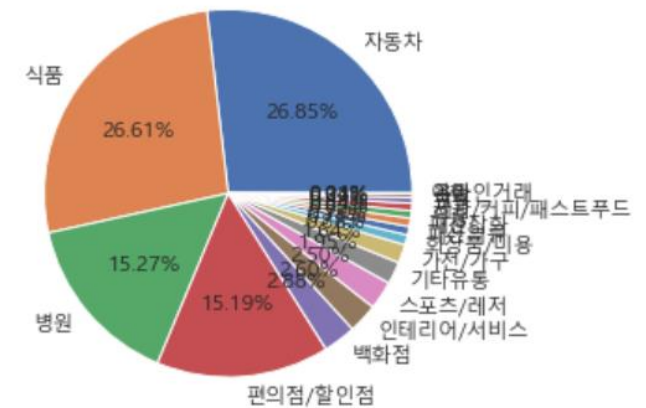
20대 남자의 소비 카테고리 순위



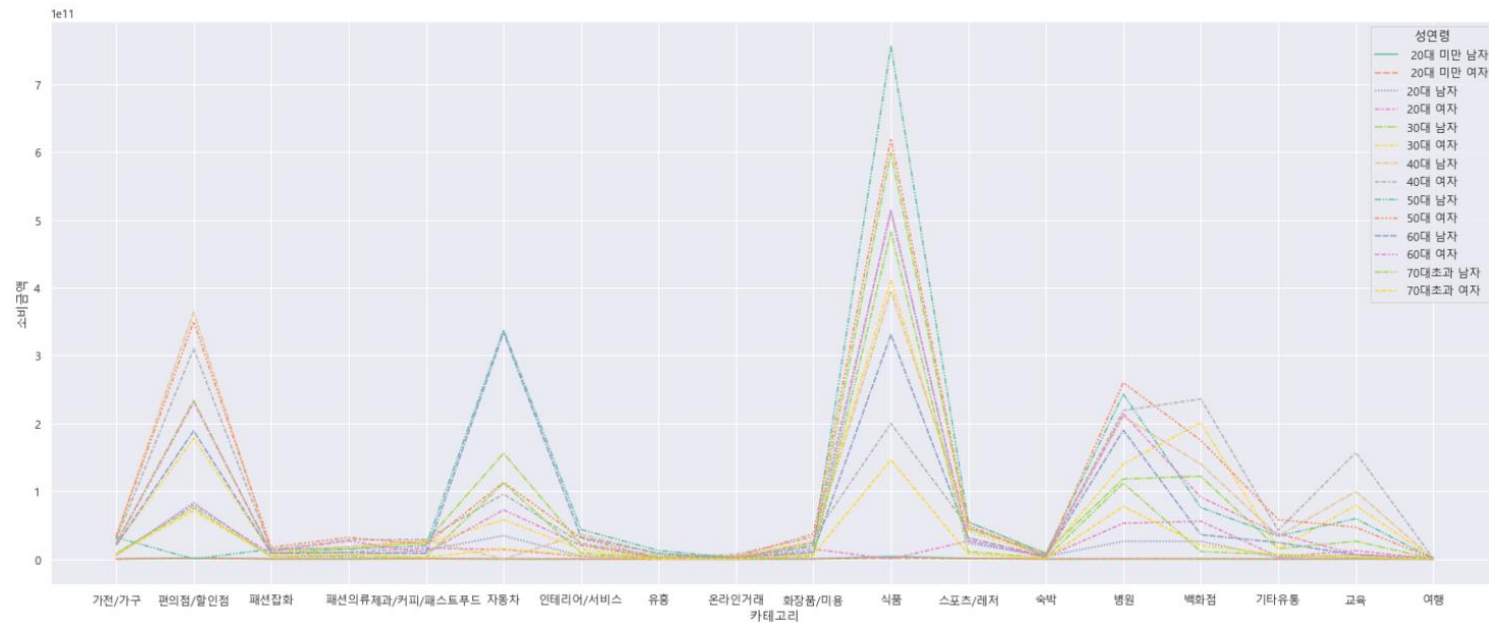
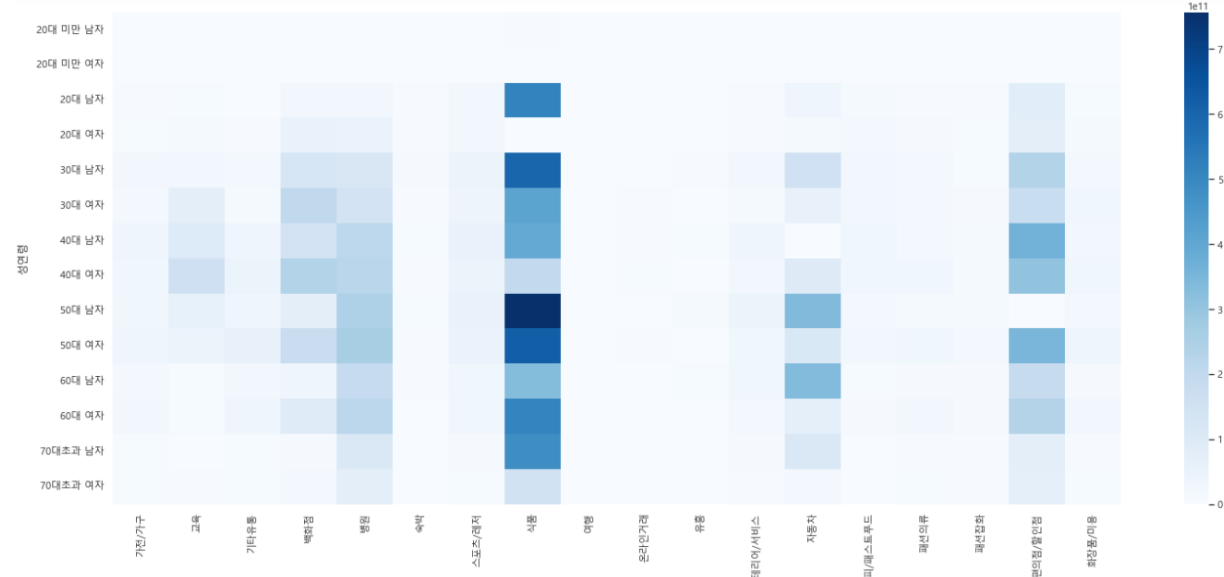
40대 남자의 소비 카테고리 순위



60대 남자의 소비 카테고리 순위

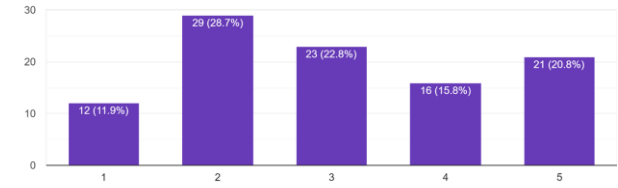


04 단기적금(1)소비패턴 분석 - 데이터 분석 및 시각화 (전체 성별,연령)

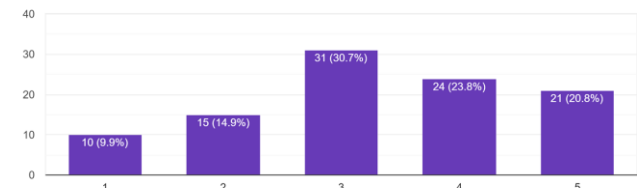


04 단기적금(2) - 설문조사

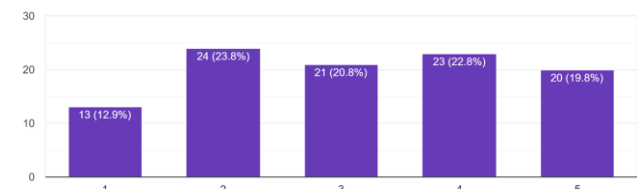
제과 / 커피 / 패스트푸드
응답 101개



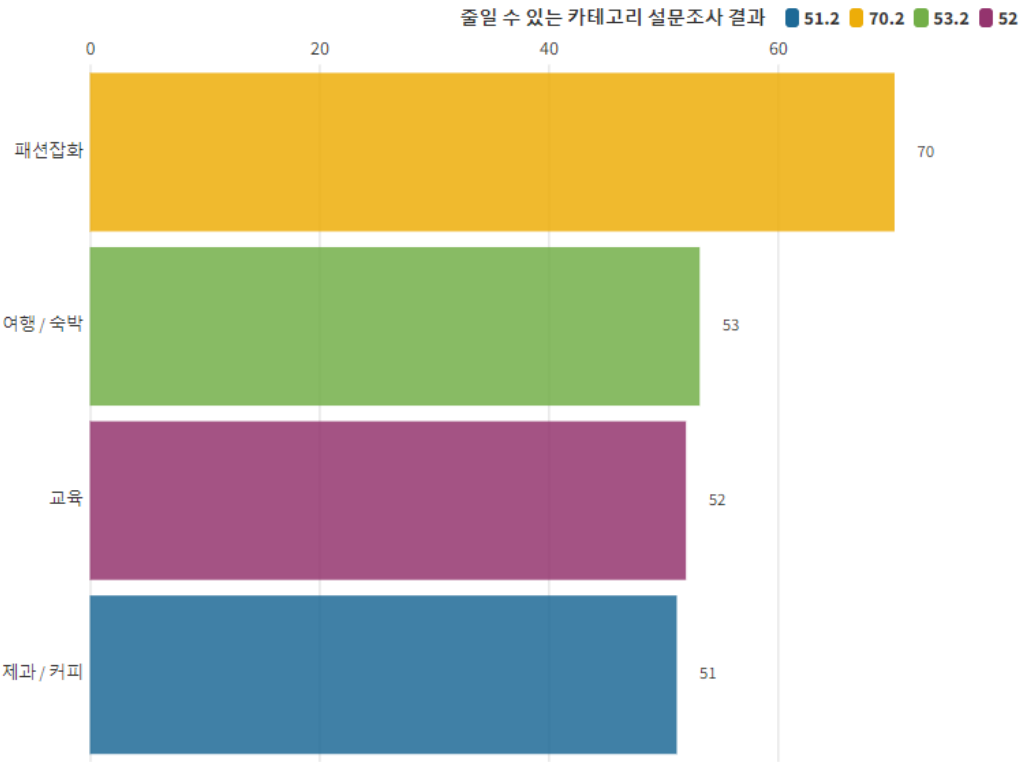
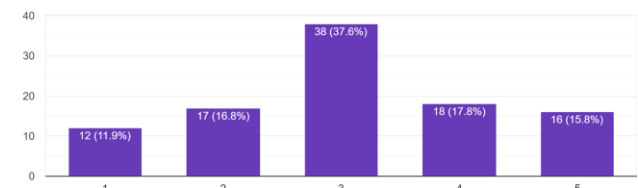
패션잡화
응답 101개



여행 / 숙박 / 스포츠
응답 101개



교육
응답 101개



04

데이터 분석 - (3) 물품 추천

- 데이터 수집
- 데이터 분석 및 시각화

물품 추천

-네이버 데이터랩 클릭량 통계 기반으로
성별, 연령별 관심 카테고리를 분석하여,
1~6위 카테고리 각각 6개의 물품을 추천

-10대 여성~ 60대 남성
2020~2021 1년 기준

자료출처 :
네이버 데이터랩

1. 네이버 데이터랩 DATA

날짜	패션의류	패션잡화	화장품/미용	디지털/가	가구/인테리어	출산/육아	스포츠/레저	생활/건강	여가/생활편의
2020-12-26	38	57	91	92	98	52	25	90	30
2020-12-27	40	60	93	92	100	52	25	94	29
2020-12-28	35	54	90	86	92	52	21	93	31
2020-12-29	35	52	86	85	88	50	22	90	28
2020-12-30	35	53	86	85	90	50	23	90	32
2020-12-31	29	45	74	73	78	46	19	77	37
2021-01-01	29	43	71	72	80	44	20	78	25
2021-01-02	32	49	93	76	86	51	22	82	21
2021-01-03	34	52	90	76	87	52	22	82	19
2021-01-04	30	44	75	66	72	53	20	78	20
2021-01-05	28	42	75	66	70	51	17	75	18
2021-01-06	29	43	71	67	69	58	20	72	20
2021-01-07	29	44	72	69	69	100	21	72	20
2021-01-08	30	48	75	71	71	68	22	71	18
2021-01-09	34	54	82	78	86	59	26	77	19
2021-01-10	36	58	85	81	95	54	26	80	19
⋮									
2021-12-22	27	40	52	54	50	28	21	71	81
2021-12-23	28	42	54	57	51	27	22	71	90
2021-12-24	24	39	48	54	48	27	18	64	100
2021-12-25	30	49	53	68	55	37	24	77	91
2021-12-26	40	61	70	74	63	33	34	91	83

04 데이터 분석 - 물품추천 - 데이터 분석 및 시각화

10대 여성의 Best 카테고리 - 생활/건강

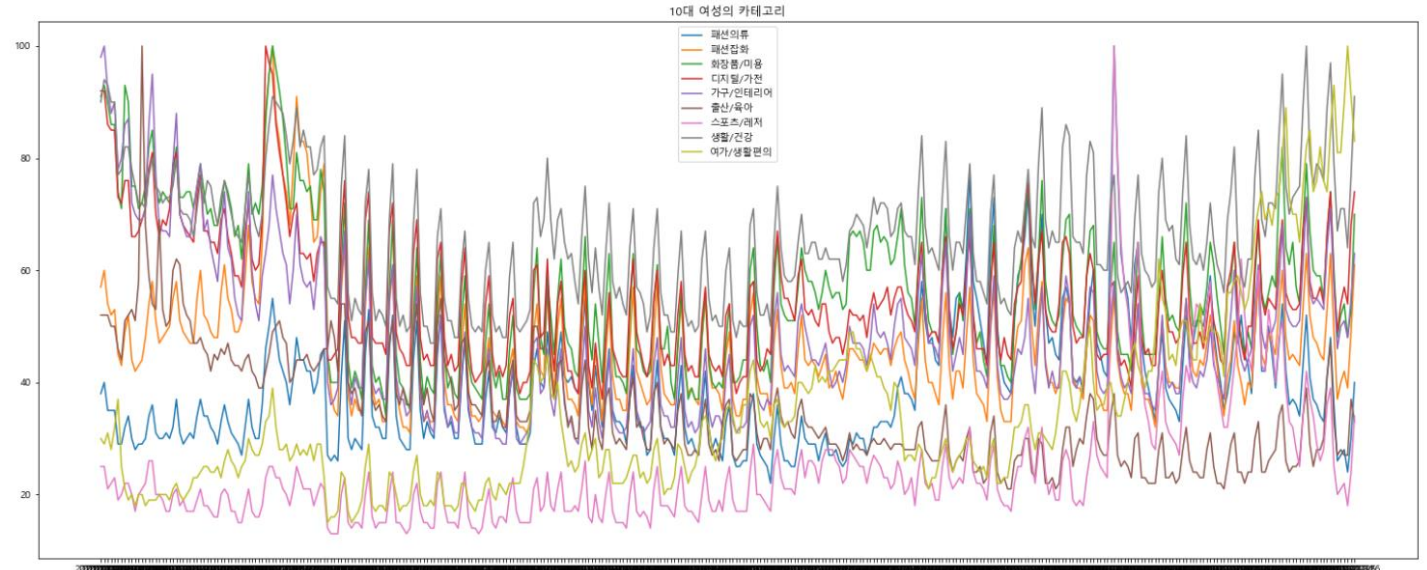
```
df_F01 = pd.DataFrame(data_F01)
df_F01.head()
```

	날짜	패션의류	패션잡화	화장품/미용	디지털/가전	가구/인테리어	출산/육아	스포츠/레저	생활/건강	여가/생활편의
0	2020-12-26	38	57	91	92	98	52	25	90	30
1	2020-12-27	40	60	93	92	100	52	25	94	29
2	2020-12-28	35	54	90	86	92	52	21	93	31
3	2020-12-29	35	52	86	85	88	50	22	90	28
4	2020-12-30	35	53	86	85	90	50	23	90	32

10대 여성 선호 카테고리

생활/건강 : 66.12841530054645
화장품/미용 : 56.59289617486339
디지털/가전 : 54.96174863387978
가구/인테리어 : 47.94808743169399
패션잡화 : 45.94808743169399
패션의류 : 39.005464480874316
출산/육아 : 34.904371584699454
여가/생활편의 : 34.53005464480874
스포츠/레저 : 24.472677595628415

10대 여성 Worst 카테고리 : 스포츠/레저



생활 건강 - 런닝머신 추천 !

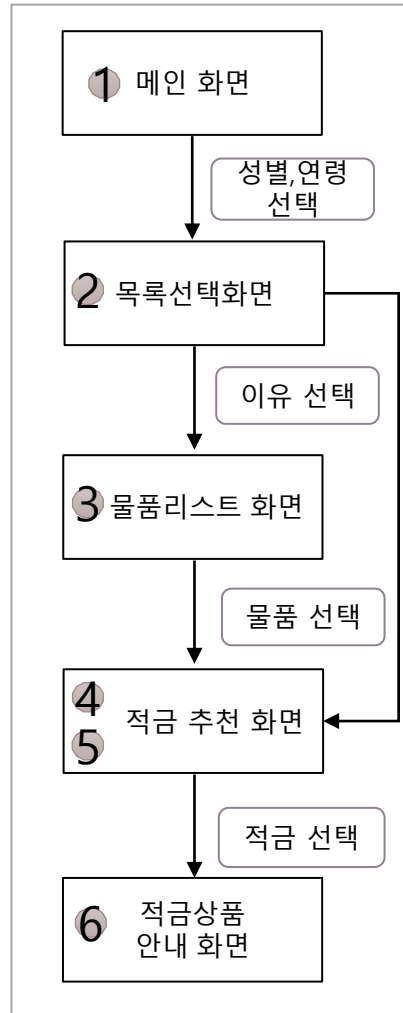
⋮

패션의류 - 트레이닝복 추천 !

물품 가격에 따른
고금리 적금 상품 추천으로
이어짐

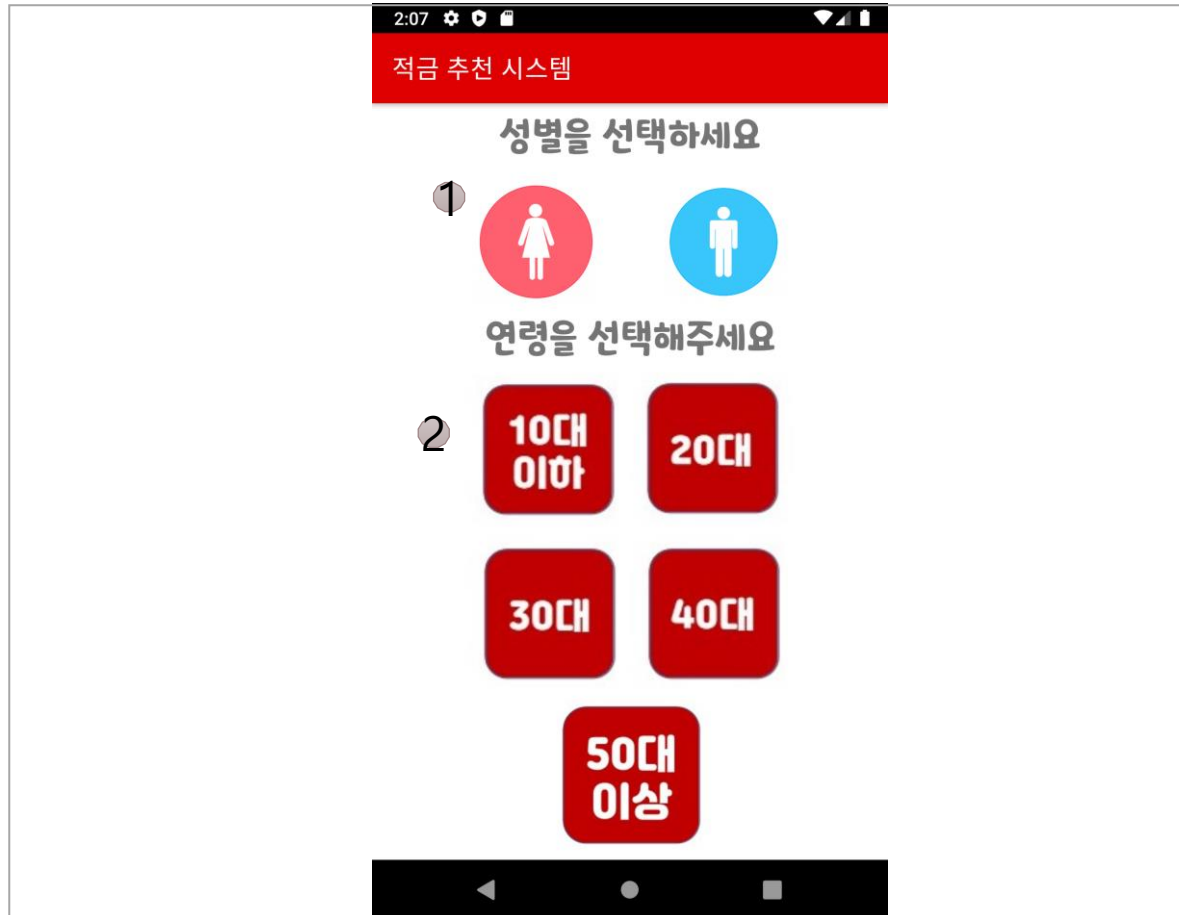
04 앱 구현

전체 흐름도



자신의 연령대를 선택합니다.

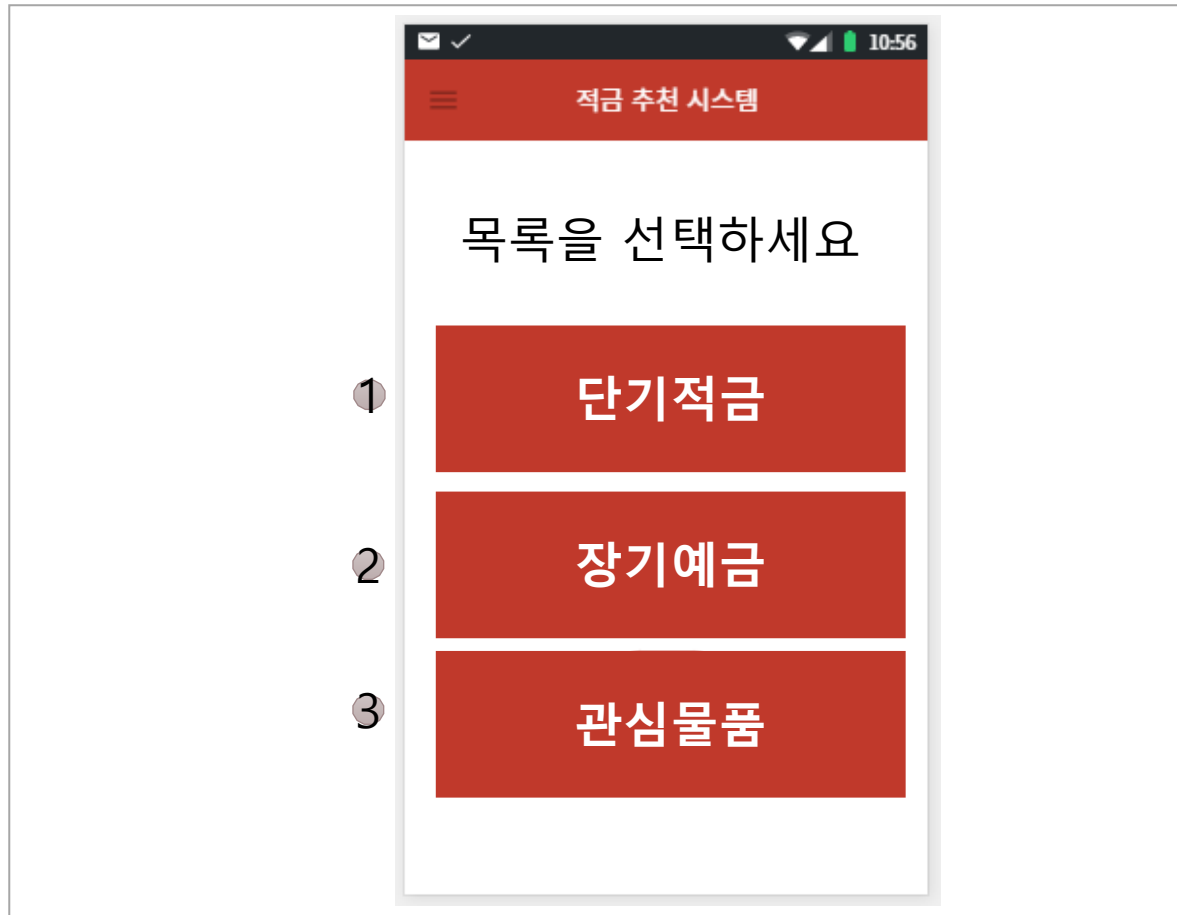
UI	성별, 연령대 선택 화면	소스 코드	GenderActivity
----	---------------	-------	----------------



Description	
1	성별 선택 시 비선택된 이미지는 흐리게 변경
2	연령을 선택하면,
+	뒤로가기 터치 시 직전화면
DB관련	
Table	컬럼
	CRUD

자신의 연령대를 선택합니다.

UI	목록 선택 화면	소스 코드	Activity
----	----------	-------	----------

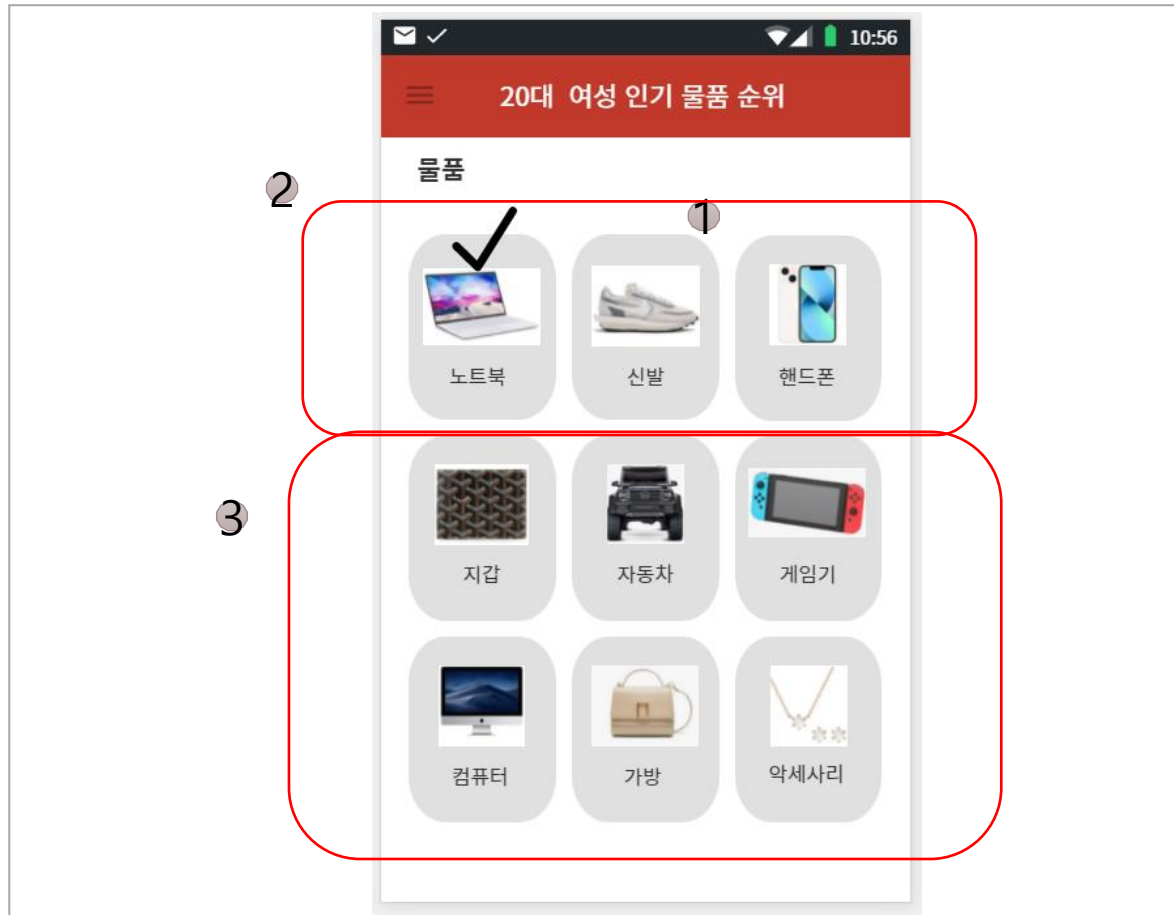


Description	
1	단기적금 버튼 클릭 시 적금리스트 화면(Installment2Activity)으로 이동
2	장기적금 버튼 클릭 시 적금리스트 화면(Installment2Activity)으로 이동
3	관심물품 버튼 클릭 시 사고싶은 관심물품화면으로 이동 (ChooseGoodsActivity)
+	뒤로가기 터치 시 직전화면

DB관련		
Table	컬럼	CRUD

성별, 연령별 소비 패턴 분석 결과, 가장 인기있는 물품 9개를 추천해줍니다.

UI	물품 리스트 화면	소스 코드	ChooseGoodsActivity
----	-----------	-------	---------------------



Description		
1	물품 선택 시, 적금 추천 화면으로 이동. 선택한 물품의 가격 정보와 물품명 정보 전송.	
2	이미지뷰 3개	
3	리사이클뷰 6개	
+	뒤로가기 터치 시 직전화면	
DB관련		
Table	컬럼	CRUD
item	*	select

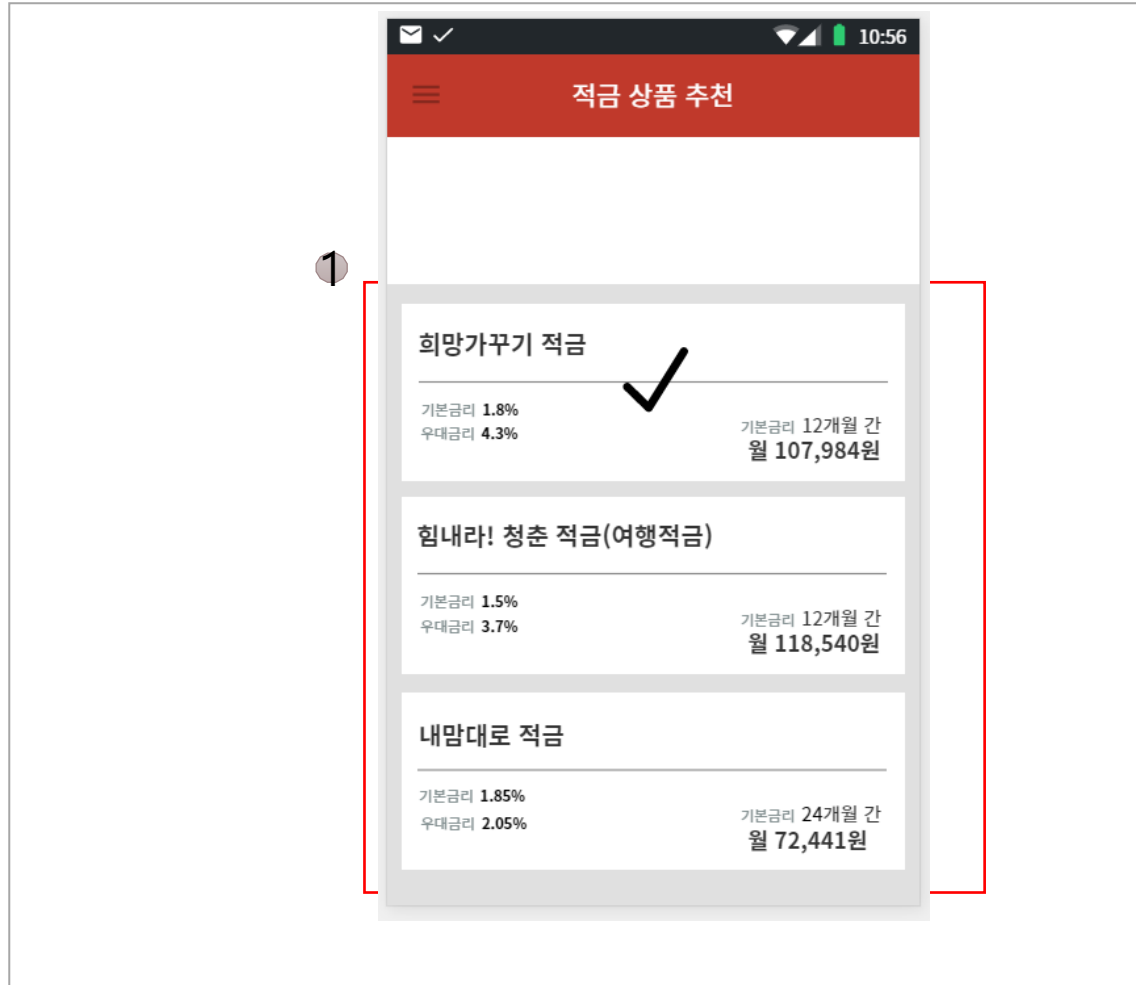
선택한 물품을 기반으로 적금 상품을 추천해줍니다.

UI	적금 추천 화면	소스 코드	InstallmentActivity
----	----------	-------	---------------------

Description		
1	물품 리스트에서 선택한 물품 가격, 이름, 추천된 적금 상품 중 최고 금리 표시.	
2	추천 적금 상품 리스트 표시. 선택 시 적금 상품 안내 화면으로 이동.	
3	기본 금리를 기준으로 월 납입금액 출력 (단리 기준, 일반 과세15.4%)	
+	뒤로가기 터치 시 직전화면	
+	적금 정보 크롤링으로 가져와서 DB에 저장하기 (기본금리, 가입기간)	
DB관련		
Table	컬럼	CRUD
item	*	select
installment	*	select

선택한 물품을 기반으로 적금 상품을 추천해줍니다.

UI	적금 추천 화면	소스 코드	Installment2Activity
----	----------	-------	----------------------

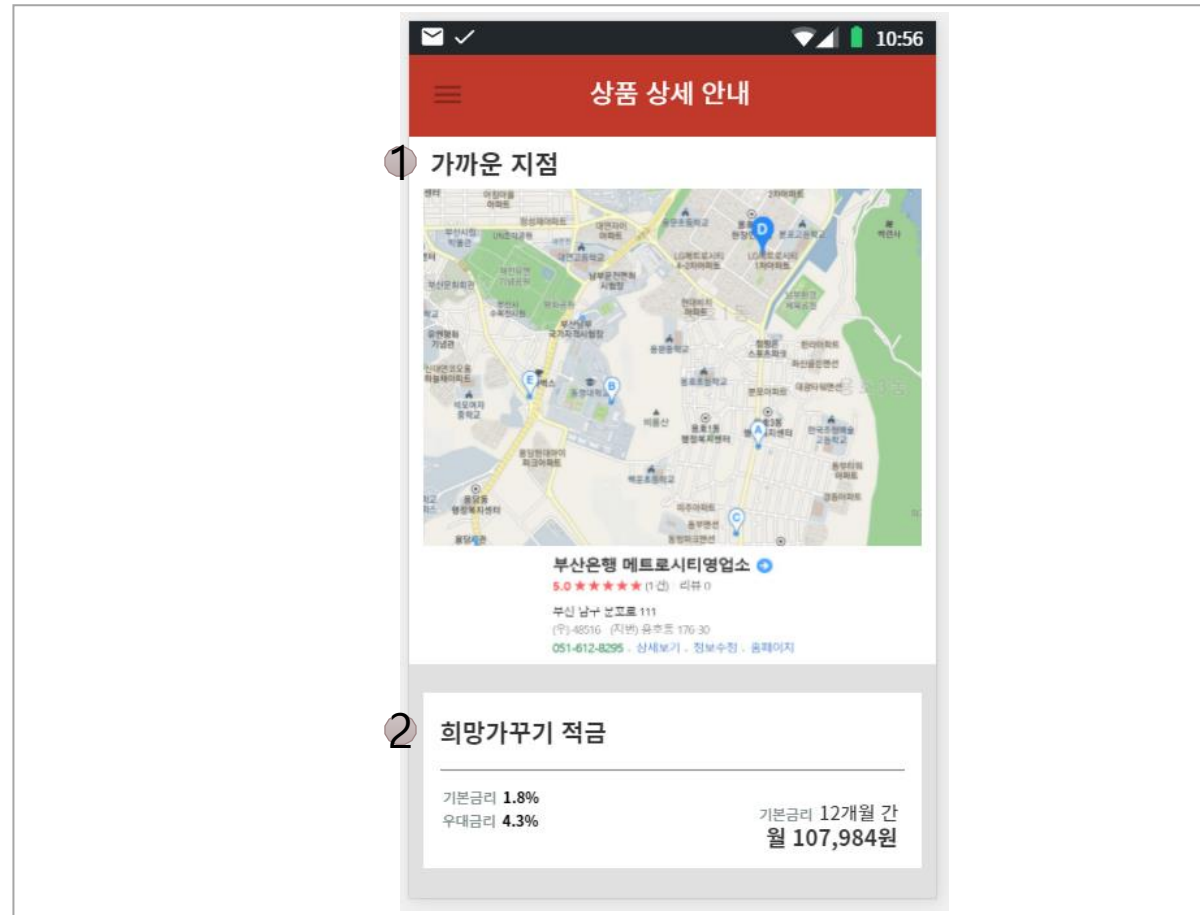


Description		
1	추천 적금 상품 리스트 표시. 선택 시 적금 상품 안내 화면으로 이동.	
2	기본 금리를 기준으로 월 납입금액 출력 (단리 기준, 일반 과세15.4%)	
3		
+	뒤로가기 터치 시 직전화면	
+	적금 정보 크롤링으로 가져와서 DB에 저장하기 (기본금리, 가입기간)	
DB관련		
Table	컬럼	CRUD
installment	*	select

적금 상품 상세 안내 페이지입니다.

Naver map을 이용하여 자신과 가장 가까운 지점과 적금 상세 내용 화면 출력

UI	적금 상품 안내 화면	소스 코드	MapDetailsActivity
----	-------------	-------	--------------------



Description	
1	현재 위치를 기준으로 가장 가까운 지점을 지도에 표시
2	선택한 적금 상품의 상세 정보 표시 (기본금리, 우대금리, 월 납입금액)
+	뒤로가기 터치 시 직전화면
+	레트로핏으로 맵 출력

DB관련		
Table	컬럼	CRUD
bankinfo	*	select
installment	*	select