

고려대학교  
빅데이터 연구회

# KU-BIG

---

맥주 개인화 추천시스템

박혜원   정의석   최문규   김혜연   조은  
별



# 목 차

I

주제 선정

II

데이터 탐색 및 전처리

III

모델 적용

IV

분석 결과

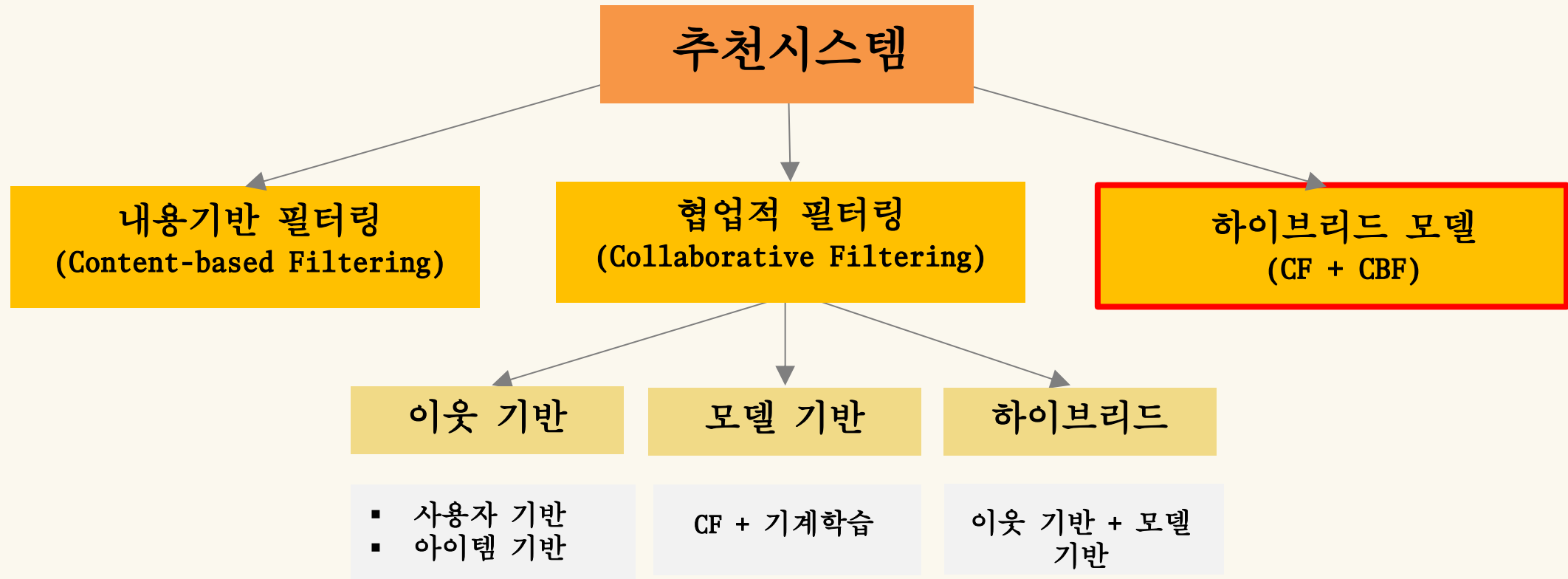
V

결론

## I. 주제 선정



# I . 주제 선정



## I. 주제 선정

‘맥알못’을 위한 (CBF)



오직 ‘너’를 위한 (CF)



맥주 추천시스템



## II. 데이터 탐색

Beer data

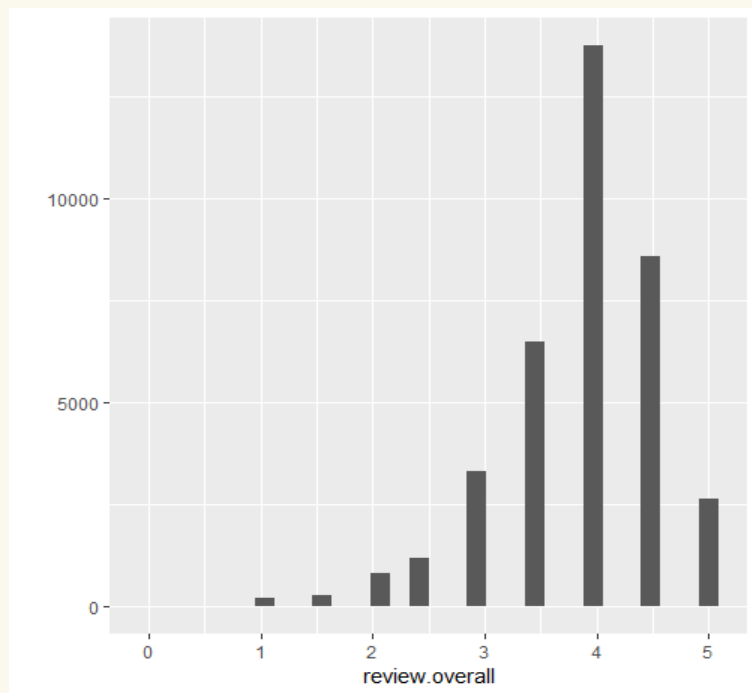
▶ 37175 obs (Kaggle)

변수명	형태(예시)	설명
beer.name	Caldera Pale Ale	맥주명
beer.breweryId	1075	양조장
beer.style	American Pale Ale (APA)	종류
beer.ABV	5.5	도수
user.profileName	vacax	사용자명

변수명	형태(예시)	설명
review.overall	2.5	총 평점
review.appearance	3.5	외관 평점
review.aroma	3	향 평점
review.palate	3.5	목넘김 평점
review.taste	2.5	맛 평점
review.text	I really like cans for beers...	텍스트
review.timeStruct	{'min': 1, 'hour': 1, 'mday': 5, ~'yday': 125}	업로드 시각

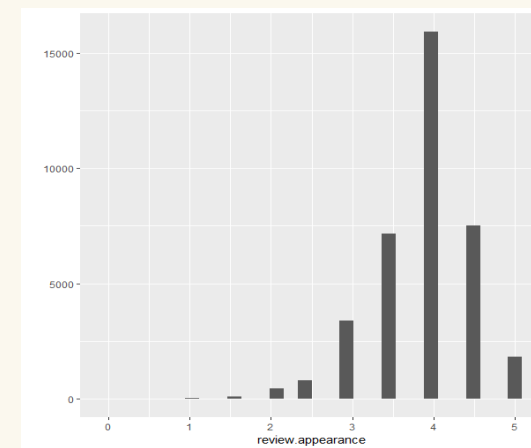
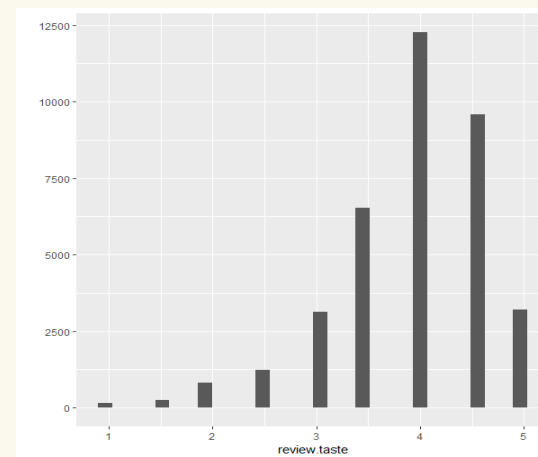
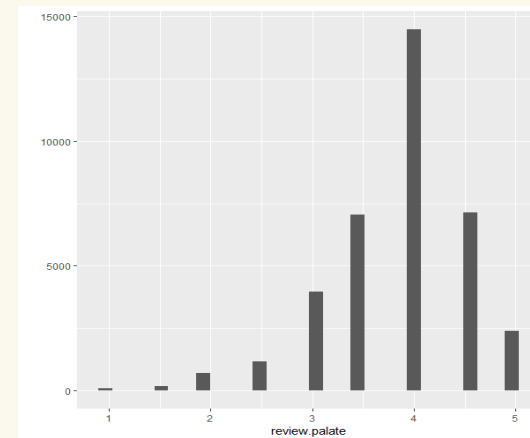
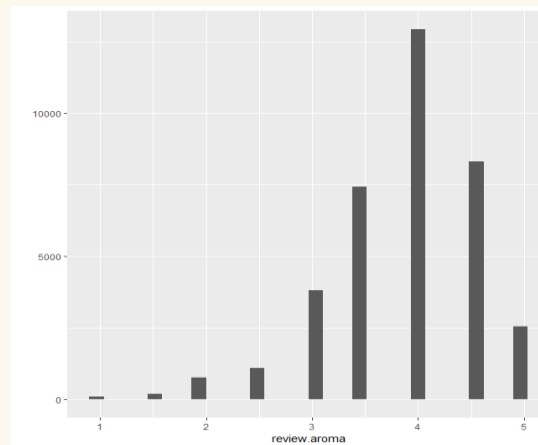
## II. 데이터 탐색

### 시각화



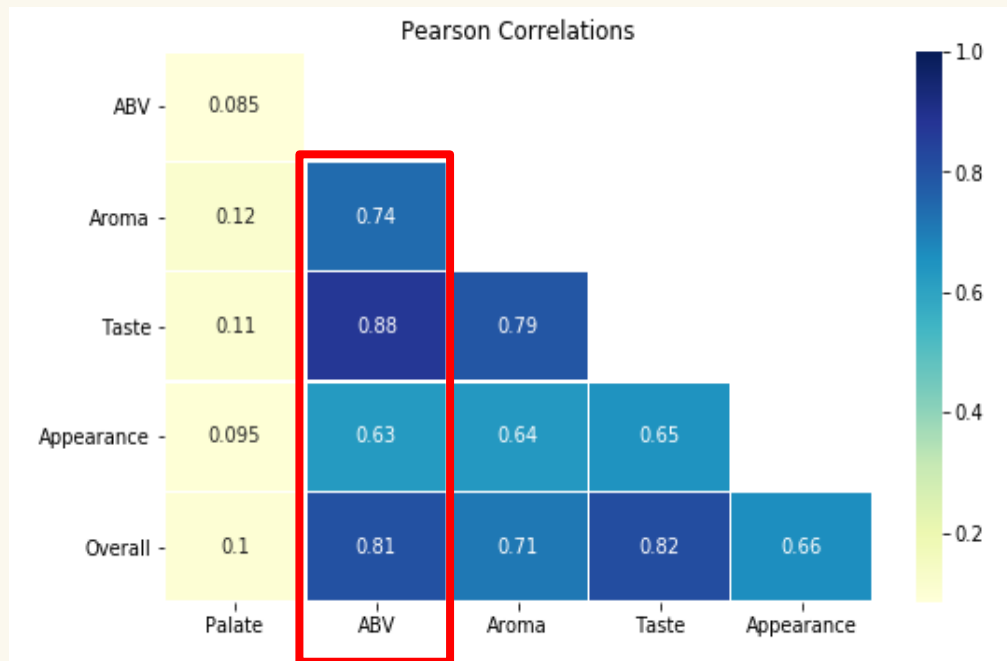
▶ review 분포

(overall/aroma/palate/taste/appearance)

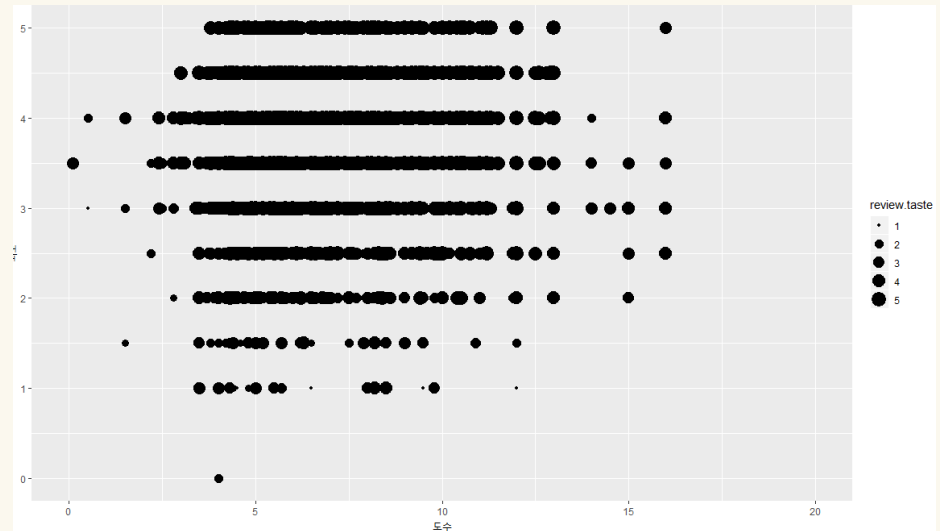


## II. 데이터 탐색

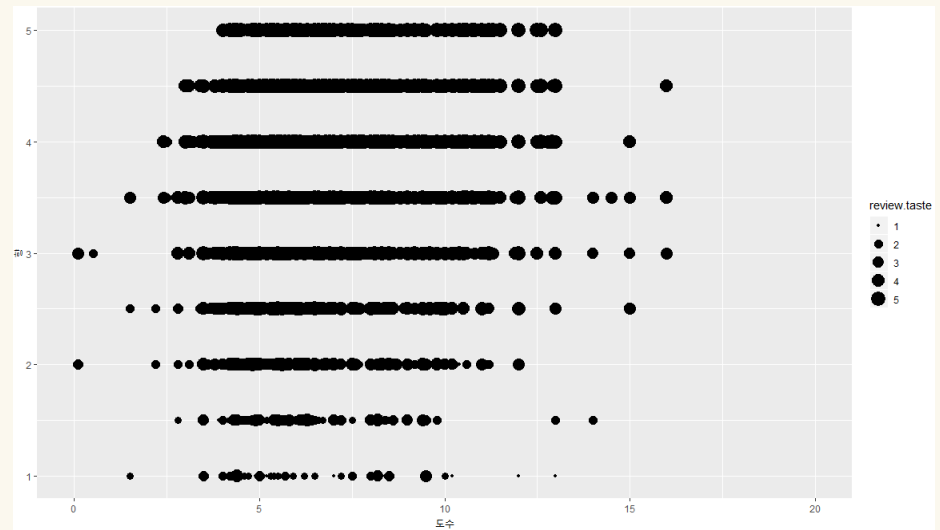
### 시각화



▶ ABV와 review간의 상관관계



▶ ABV, 외관, 맛 간의 상관관계(x: ABV, y: 외관, size: 맛)



▶ ABV, 향, 맛 간의 상관관계(x: ABV, y: 향, size: 맛)



## II. 데이터 전처리

CBF

▶ 각 변수 0~1값으로 변환

변수	변수 정보
beer.name	Factor (1688개)
beer.ABV	Numeric
Review.overall	Numeric (0~5)
review.appearance	Numeric (0~5)
review.aroma	Numeric (0~5)
review.palate	Numeric (0~5)
review.taste	Numeric (0~5)
review.text	Factor
beer.breweryId	Integer
beer.style	Factor (95종류)

CF

▶ Data set 변환

변수	변수 정보
user.profileName	Factor (7439명)
beer.name	Factor (1688개)
Review.overall	Numeric (0~5)

## II . 데이터 전처리 - CBF, CF

### 1) 결측값들 제거

```
na.index <- which(is.na(rate$rating))

rate <- rate[-na.index, ]
data <- data[-na.index, ]
```

### 2) 한 user가 같은 맥주에 여러 개의 평점을 내린 duplicates 제거

```
u <- spread(rate, beer, rating)

error <- c(14353, 42237, 42221, 41951, 41843, 41894, 41884, 41866, 43185,
          42521, 42941, 42677, 42648, 42900, 42557, 42489, 42673, 42449, 42525,
          41353, 41373, 41408, 41245, 8969, 7992, 8037, 8003, 7962, 8080, 8039, 8040,
          1449, 1463, 1280, 785, 867, 5040, 5362, 45283, 45309, 45308, 2460, 2454,
          9489, 34921, 34927, 34867, 34773, 40198, 38483, 12418, 14740, 2095, 2093,
          46718, 2202, 379, 177, 45942, 45954, 14227, 12056, 12058, 40071, 40069,
          40074, 39935, 40017, 39863)

rate <- rate[-error, ]
data <- data[-error, ]
```

## II . 데이터 전처리 - CBF

### 1) ABV, review 변수 점수화

- ABV 데이터는 변수간 상관관계 분석 결과에 따라 0~1 범위로 정규화
- Review 데이터 역시 0~1범위로 정규화

### 2) Review text 변수

- TF-IDF기법을 통해 중요 키워드 5개를 선별  
→ one-hot-encoding (0과 1값으로 변경)

Key1	Key2	Key3	Key4	Key5
lactic	banana	tend	visibl	under
teh	beig	averag	except	red
somr	oo	defect	compact	dust
camomil	compact	honest	hay	pil
cauldron	horsebrass	sandlewood	myslf	avalanch

▶ 키워드 변수 예시

- 유사도 계산 시, 선택한 beer 키워드 5개가 다른 beer 키워드와 겹치는지 계산

## II . 데이터 전처리 - CBF

### 3) Beer style 변수

- 1688종류 - 12종류로 분류

Pale Lager and Pilsner

Brown Ale

Porter

Wheat Beer

Dark Lager

Pale Ale

Stout

Wild &amp; Sour Ale

German Bock

India Pale Ale

Belgian-Style Ale

Specialty Beer

▶ webstaurantstore 사이트의 Different Styles of Beer 분류를 참고

```
style_df = df['beer.style']
onehot_style_df = pd.get_dummies(style_df)
onehot_style_df.head()
```

	Belgian-Style Ale	Brown Ale	Dark Lager	German Bock	India Pale Ale	Pale Ale	Pale Lager and Pilsner	Porter	Specialty Beer	Stout	Wheat Beer	Wild & Sour Ale
0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	0	1	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0

## III. 모델 적용 - CBF

### Cosine 유사도 기반 모델

1 widgets 함수로 beer 선택

```
import ipywidgets as widgets

user_beer=widgets.Dropdown(
    options=sorted(list(df["beer.name"])),
    description='Please choose a beer:',
    disabled=False,
)
user_beer
```

Please choose a beer: Founders Porter ▼

2 x = 선택한 beer에 맞는 user profile 생성  
y = 선택한 beer를 제외한 item profile 생성

```
import sklearn.metrics.pairwise
print("Beer selected is " + user_beer.value)
user_profile=df.loc[df['beer.name']==str(user_beer.value)]
```

```
Key1=user_profile['Key1'].tolist()
Key2=user_profile['Key2'].tolist()
Key3=user_profile['Key3'].tolist()
Key4=user_profile['Key4'].tolist()
Key5=user_profile['Key5'].tolist()
```

```
key_list=[Key1[0],Key2[0],Key3[0],Key4[0],Key5[0]]
```

```
user_profile['Key1'] = np.where(user_profile.Key1.isin(Key1),1,0)
user_profile['Key2'] = np.where(user_profile.Key2.isin(Key2),1,0)
user_profile['Key3'] = np.where(user_profile.Key3.isin(Key3),1,0)
user_profile['Key4'] = np.where(user_profile.Key4.isin(Key4),1,0)
user_profile['Key5'] = np.where(user_profile.Key5.isin(Key5),1,0)
```

```
item_profiles=item_profiles.loc[~(data['beer.name']==str(user_beer.value))]
item_profiles['Key1'] = np.where(item_profiles.Key1.isin(key_list),1,0)
item_profiles['Key2'] = np.where(item_profiles.Key2.isin(key_list),1,0)
item_profiles['Key3'] = np.where(item_profiles.Key3.isin(key_list),1,0)
item_profiles['Key4'] = np.where(item_profiles.Key4.isin(key_list),1,0)
item_profiles['Key5'] = np.where(item_profiles.Key5.isin(key_list),1,0)
```

```
item_profiles
x=user_profile.drop(['beer.beerId','beer.brewerId','beer.name'],axis=1)
y=item_profiles.drop(['beer.beerId','beer.brewerId','beer.name'],axis=1)
```

3 x,y 간의 cosine유사도 기반 모델 생성

	Unnamed: 0.1	beer.ABV	beer.beerId	beer.brewerId	beer.name	review.appearance	re
5627	5627	0.50	38552	765	Trafalgar Smoked Oatmeal Stout	0.3	0.1
6157	6157	1.10	3003	395	Bearded Pat's Barleywine	0.4	0.4
10766	10766	0.45	17655	14	Honey-Ginger Ale	0.4	0.4
5507	5507	0.50	41315	765	Trafalgar Oak-Aged Rye	0.3	0.1
5700	5700	0.55	32710	765	Strawberry Anti-Social	0.3	0.1

## IV. 분석 결과- CBF

### CBF 추천 예시

Recommended Beers for Debau Cherry Bitter Ale:

Trafalgar Smoked Oatmeal Stout  
Bearded Pat's Barleywine  
Honey-Ginger Ale  
Trafalgar Oak-Aged Rye  
Strawberry Anti-Social



#### • 선택 맥주

진한 **밀향**과 **카라멜향**이 베이스, 건포도나 포도와 같은 **과일향**이 진하고 매우 **달**

#### • 추천 맥주

- 1 : **오트밀** 베이스에 **체리향**이 감미된 맥주
- 2 : **건포도**, **카라멜**, 바닐라 향이 진한 도수 높은 맥주
- 3 : **오렌지**, 생강, 꿀의 향과 맛이 강한 **단맛**의 맥주
- 4 : **오렌지** 향과 **카라멜향**, **밀향**이 섞인 **과일향**이 강한 맥주
- 5 : 풍선껌 같이 단 **과일향**(딸기향)과 **카라멜향**이 섞인 매우 **단 맛**의 맥주

카라멜, 과일 향이 가미된, 향이 진하고 단 맛의 맥주를 추천

## Ⅲ. 모델 적용 - CF

### Recommenderlab 패키지

- R에서 추천 시스템 알고리즘을 구축할 때 주로 쓰이는 패키지
- 기존 데이터 셋에서 가져온 데이터 프레임 형식에서 추천을 위해서는 “**realRatingsMatrix**” 형태로 변환 필요
- ex) method=“IBCF” (아이템 기반 협업 필터링 모델) / method=“UBCF” (사용자 기반 협업 필터링 모델)

### Cold Start Problem 발생

- train set에 있는 정보는 추천 모델이 잘 구축되지만,  
test set에 있는 정보로는 구축 x
- train set과 test set에 뽑히는 맥주 종류 수가 다르기 때문

\* Cold Start

: 새로운 사용자나 새로운 아이템이 등장했을 경우, 데이터 부족으로 인해 추천되지 않는 문제점

	Item				
U S E R					
		Rating			

## II . 데이터 전처리 - CF

### 1) 3개 이상 Review를 남긴 사용자 추출

```
user_review<-table(rate$user)
user_review_table<-user_review[user_review>=3]
index.vector<-rep(F,length(rate$user))
rate$user<-as.character(rate$user)
index<-which(rate$user %in% names(user_review_table))
index.vector[index]<-T
rate_user<-rate[index.vector,]
rate_user$user<-as.factor(rate_user$user)
```

- Cold Start Problem 해결책
  - review를 1,2개만 남긴 user로 CF를 학습시키면 특정 사용자에게 추천되지 않는 문제점 발생



## Ⅲ. 모델 적용 - CF

### Code 참고

```
install.packages("recommenderlab")
library(recommenderlab)
setwd("C:/Users/user/Desktop/실습파일")

# 기본 데이터 셋에서 user, beer, rating 변수만 불러와 rate 데이터셋 생성
beer<-read.csv("beer.csv",header=T)
rate <- beer[,c(15, 5, 9)]
colnames(rate) <- c("user", "beer", "rating")

# train과 test 데이터셋 생성 후 추천 모델에 맞는 realRatingsMatrix 형태로 변환
tr_ind <- sample(1:7439, 5579, replace=F)
train <- rate[tr_ind, ]
test <- rate[-tr_ind, ]
rate_tr<-as(train,"realRatingMatrix")
rate_te<-as(test,"realRatingMatrix")

# 협업 필터링 모델 생성
ibcf_model<-Recommender(data=rate_tr,method="IBCF")
ubcf_model<-Recommender(data=rate_tr,method="UBCF")

# 추천 리스트 반환해 보기
i_pred_tr<-predict(ibcf_model, newdata = rate_tr, n = 5)
u_pred_tr<-predict(ubcf_model, newdata = rate_tr, n = 5)
i_pred_te<-predict(ibcf_model, newdata = rate_te, n = 10) # 데이터셋과 cold start 문제점을 보여줌
```

```
> i_pred_tr<-predict(ibcf_model, newdata = rate_tr, n = 5)
> u_pred_tr<-predict(ubcf_model, newdata = rate_tr, n = 5)
> i_pred_te<-predict(ibcf_model, newdata = rate_te, n = 10) # 데이터셋과 cold start 문제점을 보여줌
Error in object@predict(object@model, newdata, n = n, data = data, type = type, :
  number of items in newdata does not match model.
```

## Ⅲ. 모델 적용 - CF

### reshape 패키지

- 데이터 프레임의 format을 변형시킬 때 쓰는 패키지
- melt() : wide format data를 long-format으로 변환
- cast() : long format data를 wide-format으로 변환

#### ▶ 코딩 결과

<shipdata>

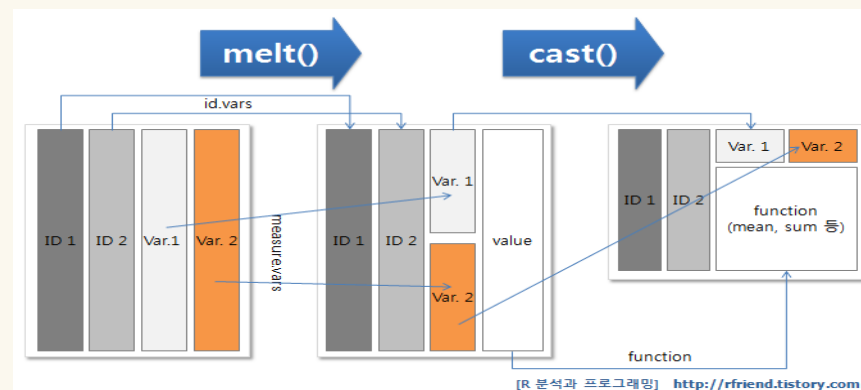
	type	year	period	service	incidents
1	A	60	60	127	0
2	A	60	75	63	0
3	A	65	60	1095	3
4	A	65	75	1095	4
5	A	70	60	1512	6
6	A	70	75	3353	18
7	A	75	60	0	0
8	A	75	75	2244	11
9	B	60	60	44882	39
10	B	60	75	17176	29

<melt.ships>

	type	year	variable	value
1	A	60	period	60
2	A	60	period	75
3	A	65	period	60
4	A	65	period	75
5	A	70	period	60
6	A	70	period	75
7	A	75	period	60
8	A	75	period	75
9	B	60	period	60
10	B	60	period	75
11	A	60	service	127
12	A	60	service	63
13	A	65	service	1095
14	A	65	service	1095
15	A	70	service	1512

<cast.ships>

	type	year	period	service	incidents
1	A	60	135	190	0
2	A	65	135	2190	7
3	A	70	135	4865	24
4	A	75	135	2244	11
5	B	60	135	62058	68



### Ⅲ. 모델 적용 - CF

#### Beer data에 적용

- 기존 long 형태 > realRatingMatrix형태에 맞게 변형 - reshape2 패키지의 acast() 함수 활용
- cast() 함수
- dcast()는 데이터프레임을 결과로 반환 / acast()는 벡터, 행렬, 배열을 결과로 반환

#### ▶ 코딩 결과

<기존 데이터셋>

	user	beer	rating
1	stcules	Sausa We	1.5
2	stcules	Red Moor	3
3	stcules	Black Hor	3
4	stcules	Sausa Pils	3
5	johnmiche	Cauldron	4
6	oline73	Caldera G	3
7	Reidrover	Caldera G	3.5
8	alpinebry	Caldera G	3
9	augustgar	Caldera G	4.5
10	MadelnOr	Caldera G	5
11	rawthar	Caldera G	4
12	Halcyonda	Caldera G	4
13	RedDiam	Caldera C	2
14	beerguy10	Caldera C	4



<변환 데이터셋>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1		Pooya Poi	The Wind	A.M. Ale	10 Blocks	10 Year Cl	12 Year Ar	13th Ann	16 Year Ar	1871 Laga	1906 Rese	1984 Gold	1er Anniv	2 Brew	2009 Arq	24 K	2e Anniv	2X Choc	3rd Anniv	4e Anniv	5 Day IPA	60 Schill	7 Threads	7GA IPA
2	0110x011	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
3	05Harley	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
4	0tt0	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
5	100floods	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
6	1099	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
7	110x18	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
8	12publoy	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
9	12vUnion	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
10	1759Gid	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
11	184601	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
12	1993Heel	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
13	1Adam12	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
14	1fast28	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
15	1morebee	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
16	1p3261	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
17	1santore	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
18	1thinmint	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
19	1whiskey	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
20	2378GCGT	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
21	2ndstage	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
22	2roosters	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
23	2xHops	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

8

- 기존 데이터 셋에서 user를 왼쪽 기준변수, beer를 열 변수로, 그리고 그 벡터 사이에 review가 들어간 u 데이터셋을 생성
- 사용자가 평가를 내린 맥주 이외 값에는 NA값이 들어가고, 평가한 항목에만 실제 평가 점수가 들어감

## Ⅲ. 모델 적용 - CF

### Beer data 출력값 분석

#### [1] 사용자 기반 협업 필터링 (UBCF)

u_pred	S4 (recommenderlab::topNList)	S4 object of class topNList
items	list [1860]	List of length 1860
0110x011	integer [5]	572 575 638 112 615
1099	integer [5]	1 2 3 4 5
12puebloyankee	integer [5]	572 37 228 1022 1175
1759Girl	integer [5]	572 617 1412 1611 37
184601	integer [0]	
1fastz28	integer [5]	575 1190 130 188 638
1santore	integer [5]	576 38 35 37 572
321jeff	integer [5]	617 638 636 590 634

```
> head(as(u_pred, "list"), 3)
$`0110x011`
[1] "Founders Breakfast Stout"      "Founders CBS Imperial Stout"
[3] "Founders Red's Rye PA"        "B.O.R.I.S. The Crusher Oatmeal-Imperial Stout"
[5] "Founders Imperial Stout"
```

#### ▶ 출력값 분석

- 이전과 달리 test 데이터 셋으로도 추천 알고리즘이 돌아가는 데 성공!
- 하지만, 특정 user에 대해 추천이 이루어지지 않는 문제점 여전히 발생..

#### [2] 아이템 기반 협업 필터링 (IBCF)

i_pred	S4 (recommenderlab::topNList)	S4 object of class topNList
items	list [1860]	List of length 1860
0110x011	integer [5]	14 24 29 30 41
1099	integer [0]	
12puebloyankee	integer [0]	
1759Girl	integer [5]	346 460 506 530 684
184601	integer [0]	
1fastz28	integer [5]	70 1034 26 127 237
1santore	integer [0]	
321jeff	integer [5]	19 24 52 68 90

```
> head(as(i_pred, "list"), 3)
$`0110x011`
[1] "2009 Arquebus, Ch. Boswell Barrel Finish With Brett" "80 Weight Mild Amber Scotch Ale"
[3] "Able I.P.A." "Ackerman's Imperial Double Stout (Indiana Replicale 2011)"
[5] "Agent Provocateur"
```

```
$`1099`
character(0)
```

```
$`12puebloyankee`
character(0)
```

## Ⅲ. 모델 적용 - CF

### Beer data 출력값 분석

#### [1] 사용자 기반 협업 필터링(UBCF)

u_pred_1	S4 (recommenderlab::topNList)	S4 object of class topNList
items	list [1063]	List of length 1063
1759Girl	integer [5]	566 608 610 558 629
33nickadams	integer [5]	566 631 570 1394 1417
49degrees	integer [5]	610 566 608 631 629
4DAloveofSTOUT	integer [5]	566 631 570 1394 33
57md	integer [5]	610 1417 556 560 185
908carson	integer [5]	566 610 631 586 560
9InchNails	integer [5]	610 608 1409 128 620
AaronHomoya	integer [5]	570 610 35 225 629
abcsofbeer	integer [5]	610 566 631 586 33
abecall98	integer [5]	37 33 608 586 1588
Absumaster	integer [5]	610 1409 120 1150 185
adam42381	integer [5]	610 570 569 560 185
AdamBear	integer [5]	1162 569 1150 286 1296

```
> head(as(u_pred_1, "list"),20)
$`1759Girl`
[1] "Founders Breakfast Stout"
[2] "Founders Imperial Stout"
[3] "Founders KBS (Kentucky Breakfast Stout)"
[4] "Founders Black Rye"
[5] "Founders Porter"

$`33nickadams`
[1] "Founders Breakfast Stout"
[2] "Founders Red's Rye PA"
[3] "Founders Centennial IPA"
[4] "Stout's Double IPA (India Pale Ale)"
[5] "Sue"
```

#### ▶ 출력값 분석

- 이전과 달리 모든 user에 대해 추천이 이루어지는 점 확인 가능!!

#### [2] 아이템 기반 협업 필터링(IBCF)

i_pred_1	S4 (recommenderlab::topNList)	S4 object of class topNList
items	list [1063]	List of length 1063
1759Girl	integer [5]	33 343 404 427 487
33nickadams	integer [5]	54 66 286 364 379
49degrees	integer [5]	3 63 79 166 181
4DAloveofSTOUT	integer [5]	216 436 672 972 1108
57md	integer [5]	4 71 75 84 101
908carson	integer [5]	31 33 34 37 38
9InchNails	integer [5]	1 3 8 11 14
AaronHomoya	integer [5]	4 23 84 117 119
abcsofbeer	integer [5]	13 32 42 51 67
abecall98	integer [5]	68 263 348 530 622
Absumaster	integer [5]	288 490 1375 477 707
adam42381	integer [5]	242 313 498 525 717
AdamBear	integer [5]	54 364 453 1135 1169

```
> head(as(i_pred_1, "list"),20)
$`1759Girl`
[1] "Aecht Schlenkerla Eiche" "CBC E.S.B. \dank Nugs\" "Cross Buttock Ale" "Dark"
[5] "Emmerdale"

$`33nickadams`
[1] "Alt-errior Motive" "American Red Ale" "Breznak Schwarzbier"
[4] "Chilli-Out" "Clare's Right Hook - Lemon Zest"

$`49degrees`
[1] "1 A.M. Ale" "Amber Ale - German Style Altbier" "Anglers Ale"
[4] "Bernard Celebration Lager" "Biere De Miel"
```

## IV. 분석 결과- CF

### 정확도 평가-1

#### ▶ 코딩

```
scheme_cross<-evaluationScheme(r,method="cross-validation",train=0.8,
                               given=3,k=10,goodRating=3)
user_model_2<-Recommender(data = getData(scheme_cross, "train"),
                           method = "UBCF",
                           parameter = "COSINE")
item_model_2<-Recommender(data = getData(scheme_cross, "train"),
                           method = "IBCF",
                           parameter = "COSINE")
u_pred_2 <- predict(user_model_2, newdata = getData(scheme_cross, "known"),
                    n = 10, type = "ratings")
i_pred_2 <- predict(item_model_2,newdata = getData(scheme_cross, "known"),
                    n = 10, type = "ratings")
Accuracy <- rbind(
  calcPredictionAccuracy(x=u_pred_2,
                        data = getData(scheme_cross, "unknown"),byUser = FALSE),
  calcPredictionAccuracy(x=i_pred_2,
                        data = getData(scheme_cross, "unknown"),byUser = FALSE)
)
```

#### ▶ 결과값 해석

```
> rownames(Accuracy)<- c("UBCF", "IBCF")
> Accuracy
```

	RMSE	MSE	MAE
UBCF	0.7319416	0.5357385	0.5518433
IBCF	0.9807698	0.9619094	0.6866502

- UBCF 방법이 IBCF 방법보다 좀 더 정확하다고 할 수 있다!!

#### • evaluationScheme

- MSE로 추천 모델 성능 평가하는 recommenderlab 내장 함수

• train 비율 0.8, 반복 수 10인 cross-validation 방법

활용 evaluationScheme 방법으로 known 정보로 예측한 평점과

unknown한 정보로 얻은 평점과 비교하여 MSE, RMSE, MAE 계산

Mean squared error	MSE	=	$\frac{1}{n} \sum_{t=1}^n e_t^2$
Root mean squared error	RMSE	=	$\sqrt{\frac{1}{n} \sum_{t=1}^n e_t^2}$
Mean absolute error	MAE	=	$\frac{1}{n} \sum_{t=1}^n  e_t $

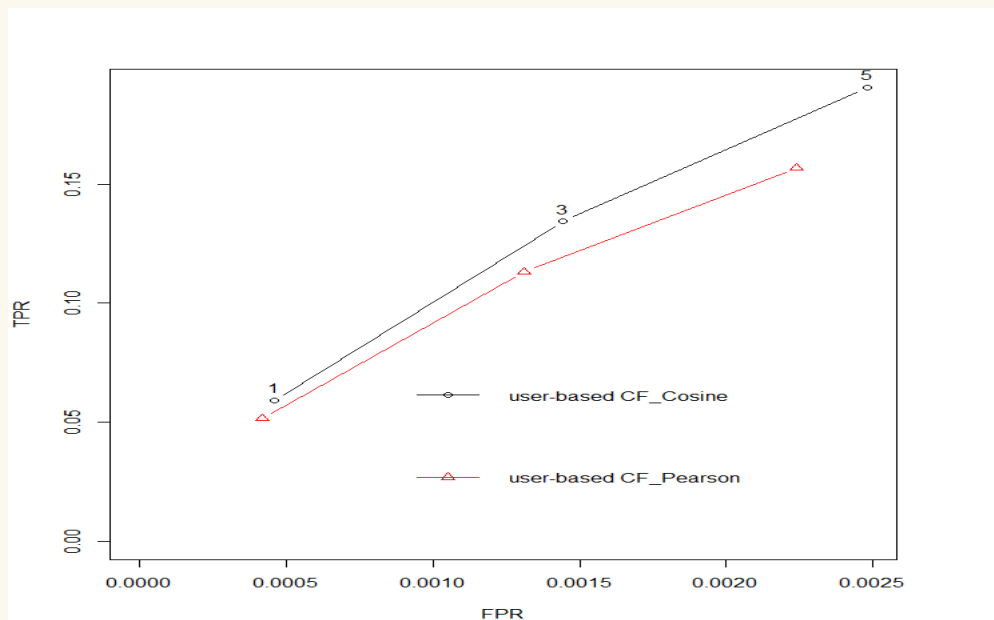
## IV. 분석 결과- CF

### 정확도 평가-2

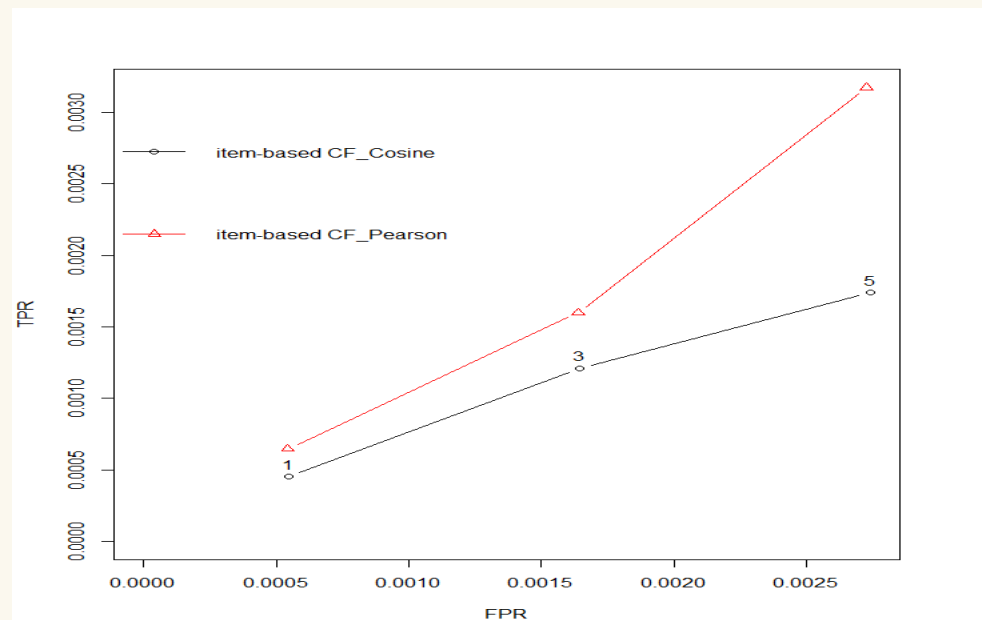
#### ▶ ROC CURVE

: 민감도를 나타내는 TPR과 특이도를 나타내는 FPR을 축으로 하는 곡선으로 왼쪽 위 꼭지점에 가까울수록 정확함을 의미

<UBCF>



<IBCF>



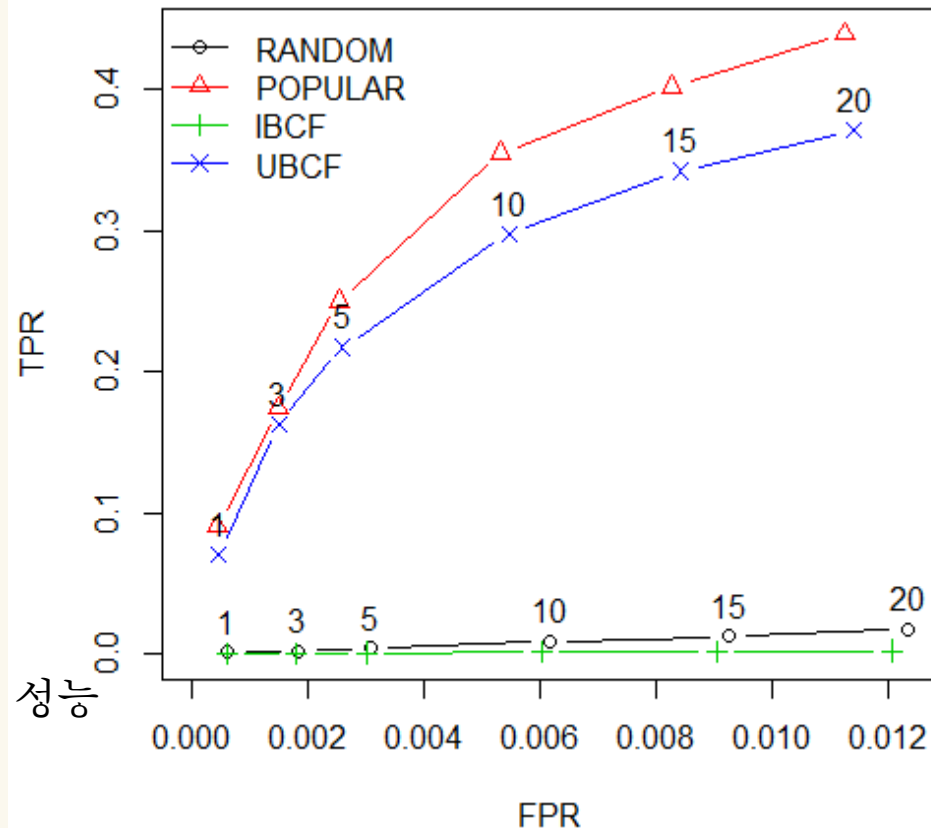
- UBCF는 코사인 방식이, IBCF는 피어슨 방식이 좀 더 성과가 좋을 수 있다!!!

## IV. 분석 결과- CF

### 시스템 성능 평가

```
> s.split <- evaluationScheme(data = r, method = "split", train = 0.8, given = 5, k = 5)
>
> s.cross <- evaluationScheme(data = r, method = "cross-validation", train = 0.8,
+                             k = 10, goodRating = 0, given = 5)
> algorithm <- list(RANDOM = list(name = "RANDOM", param = NULL),
+                  POPULAR = list(name = "POPULAR", param = NULL),
+                  IBCF = list(name = "IBCF", param = NULL),
+                  UBCF = list(name = "UBCF", param = NULL))
>
```

- FPR = False Positive Rate (좋아하지 않는 아이템을 추천한 비율)
- TPR = True Positive Rate (좋아하는 아이템을 추천한 비율)
- ROC Cruve이며 그래프 밑의 넓이가 1일수록 좋은 모델
- 인기가 좋은 상품 순으로 추천하는 “POPULAR”가 제일 좋은 성능
- 랜덤하게 아이템 추천하는 방식보다도 좋지 못한 IBCF



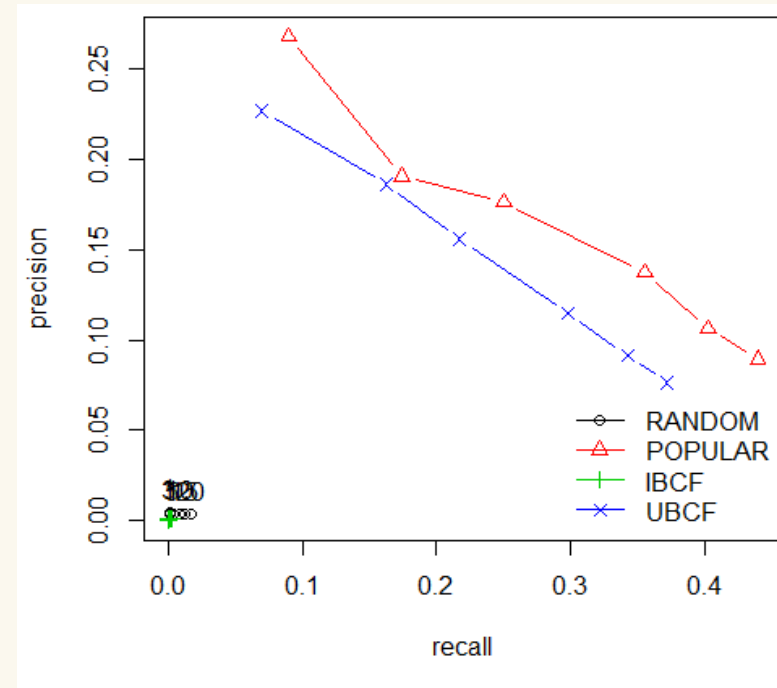
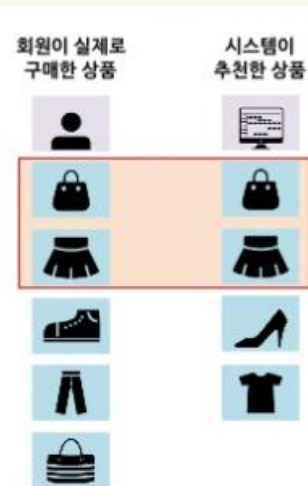


## IV. 분석 결과- CF

### 시스템 성능 평가

- Precision = 추천된 아이템 중 실제로 사용자가 좋아한 비중
- Recall = 사용자가 좋아한 영화 중 실제로 추천된 아이템의 비중
- $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$   $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
- Precision과 Recall은 역수 관계

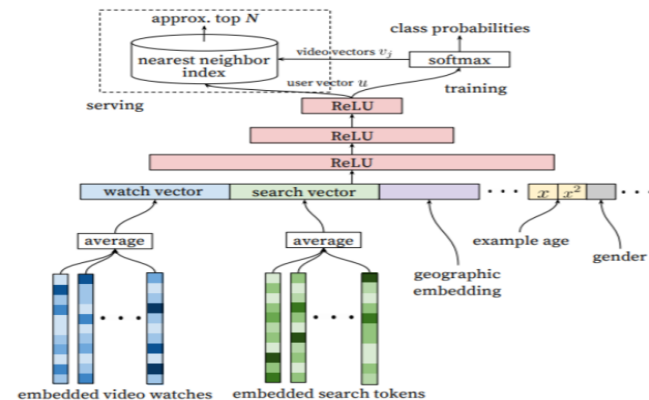
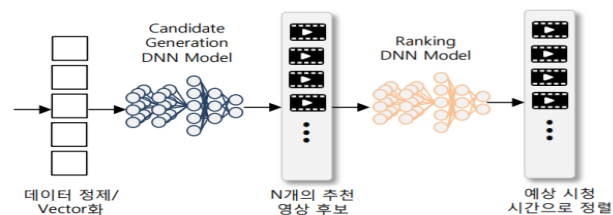
		예측	
		POSITIVE	NEGATIVE
실제	TRUE	True Positive	True Negative
	FLASE	False Positive	False Negative



## IV. 분석 결과- CF

### 개선 방안

- ❑ 사용자의 콘텐츠 구매(재생) 이력을 Vector화 하고 DNN/RNN으로 학습하고 이를 통하여 특정 사용자가 다음에 볼 콘텐츠를 확률순으로 출력하여 추천
  - ✓ 대규모 사용자/아이템에 대해 우수한 성능
  - ✓ 사용자 선호도에 크게 민감하지 않음
  - ✓ Hybrid(CF + CB + @) 시스템으로 확장이 용이
  - ✓ 대부분 Global top player는 MF에서 DL로 전환



K-nn, 딥러닝 등 더 심화된 알고리즘들 필요

요

## V. 기대효과&활용방안

편의점과의 연계



▶ 수입맥주 점유율 60%(2018)를 차지하는 편의점에서 활용된다면 시너지를 낼 것으로 예상

- 자체 데이터를 활용 → 맥주 큐레이션 서비스 제공
- 오픈 데이터를 활용 → 신규 맥주 소싱(Buying)에 필요한 시장 분석 모델로도 활용 가능

## V. 기대효과&활용방안

### 큐레이션 서비스와 연계



▶ 기존 맥주 큐레이션 서비스(ex, 벨루가 브루어리)와 제휴

- 국내 맥주 애호가들의 평가가 더해진다면 '국내판 Beer advocate' 과 같은 맥주 추천 서비스로 성장 가능할 것

A stylized illustration of a person from the chest up, wearing a grey suit jacket, a white shirt, and a dark tie. The person's head is partially visible at the top, with a red circle representing the mouth. A large, thick black speech bubble originates from the mouth and points towards the center of the image. Inside the speech bubble, the text "Do you have any question?" is written. The background is a solid light beige color.

Do you have any  
**question?**

Thank you  
for your attention.