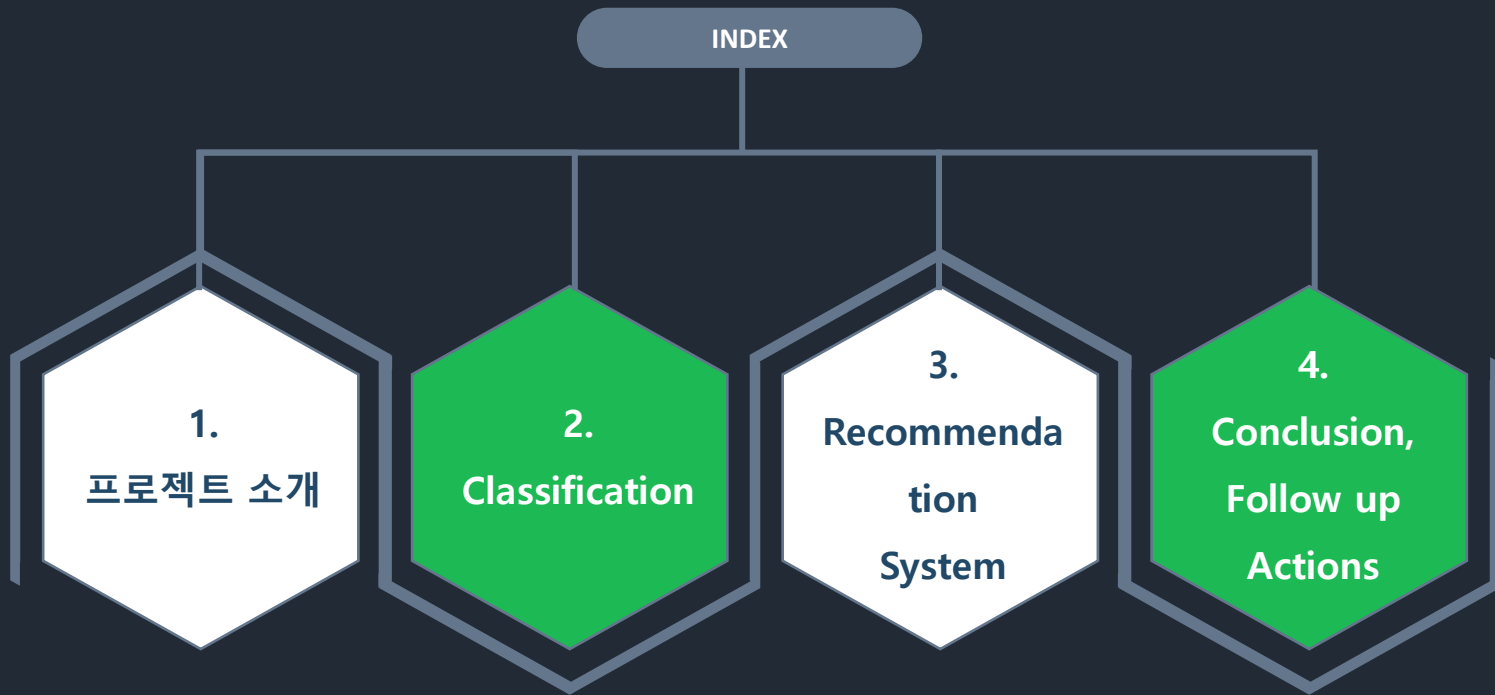


[KU-BIG '21 Spring Project]

# Spotify Music Track Preference Classification and Recommendation System

# INDEX



- 1) 주제 선정 배경 및 목표
- 2) 진행 과정

- 1) 데이터 셋 소개
- 2) 실험 모델 소개 및 결과
- 3) 결론

- 1) 데이터 셋 소개
- 2) 추천시스템 알고리즘 및 모델 종류
- 3) Surprise 패키지 소개
- 4) 실험 모델 소개 및 결과
- 5) 결론

- 1) Conclusion
- 2) Follow up Actions

Listen on

# 1. 프로젝트 소개

# 1. 프로젝트 소개

## 1) 주제 선정 배경 및 목표

Spotify



# 1. 프로젝트 소개

## 1) 주제 선정 배경 및 목표

### 19 예측 애널리틱스 팀 프로젝트 과제

주제 선정 배경

Play

언어적 표현 yes/no

음악을 즐김 : 감정, 기분, 느낌의 움직임을 즐김

음악의 선호도 판별

장르, 아티스트 영향

음악의 선호를 Linguistic 한 방법이 아닌  
수치적 속성들을 통해 판별할 수 있을까?

Follow Up Action

음악의 수치적 속성으로도 개인의 음악의 선호를 충분히 판별할 수 있음을 확인

Instance를 개개인에 맞추어 변경한 뒤 모델을 학습  
-> 나만의 음악 선호 판별 모델을 만들 수 있음

아티스트, 장르 등의 음원 Background 정보 없이 수치적 속성으로 만든 모델  
-> 음악에 대한 편견과 선입견 없이 즐길 수 있음

나아가 음악 추천 시스템의 핵심 모델로도 발전시킬 수 있음

# 1. 프로젝트 소개

## 1) 주제 선정 배경 및 목표

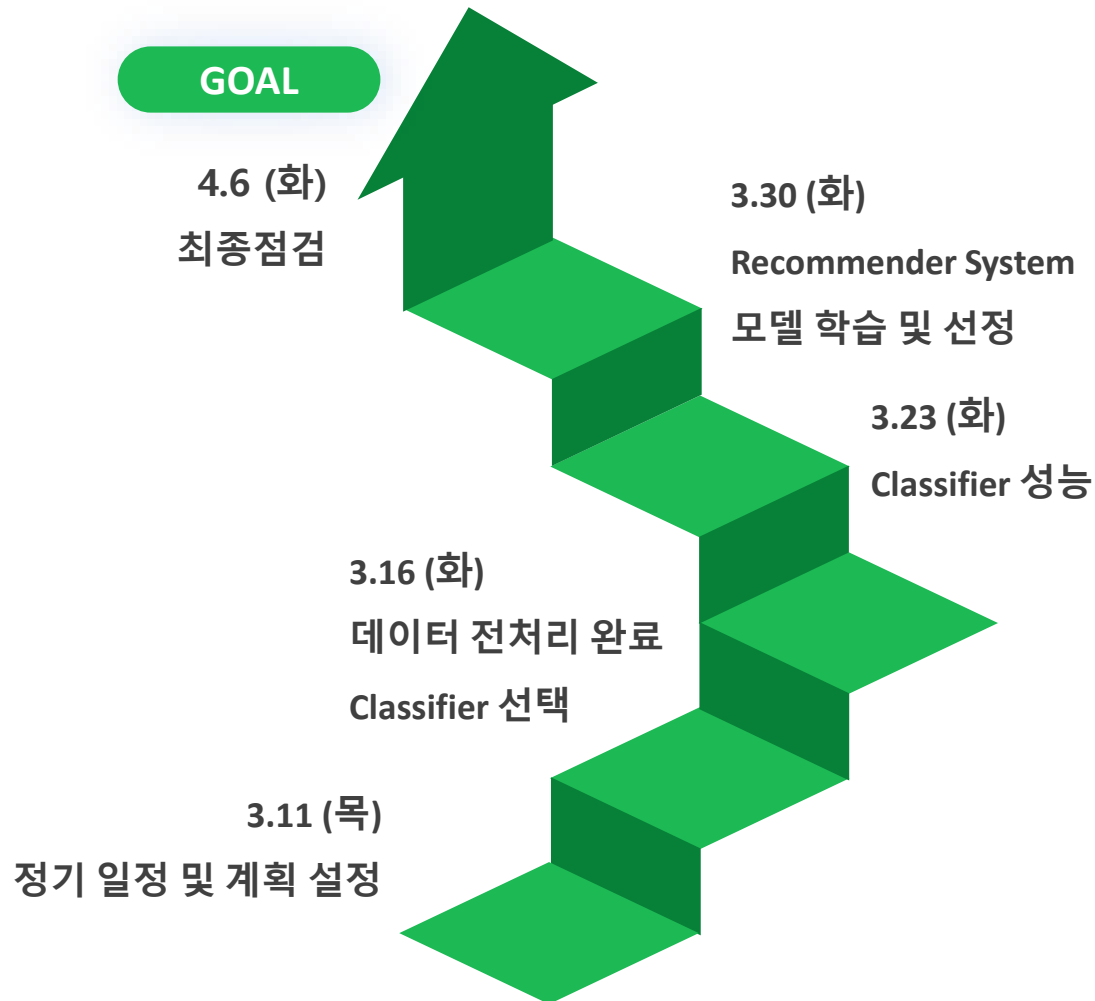
- 다양한 분류 모델 및 머신 러닝 주요 개념 학습, 실습 경험
- 추천 시스템 이론 학습, 실습 경험
- Python 스킬 향상



# 1. 프로젝트 소개

## 1) 주제 선정 배경 및 목표

정기 세션: 화요일 밤 10시 30분



Listen on

## 2. Classification



## 2. Classification

### 1) 데이터 셋 소개

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	acousticness	danceability	duration_ms	energy	instrumentalness	key	liveness	loudness	mode	speechiness	tempo	time_signature	valence	target	song_title
2	0.0198	0.571	199640	0.87	6.60E-06	8	0.0564	-3.877	0	0.0641	172.716	4	0.815	0	Bye Bye B
3	0.3	0.55	266293	0.563	0	2	0.122	-7.019	1	0.0313	166.004	4	0.462	0	This I Pron
4	0.0479	0.647	191280	0.87	0	0	0.0485	-4.702	0	0.0798	165.103	4	0.906	0	It's Gonna
5	0.44	0.468	243493	0.535	0	10	0.101	-8.264	1	0.0436	168.001	4	0.298	0	God Must
6	0.0743	0.754	199787	0.946	0.000391	8	0.401	-2.149	0	0.0403	112.042	4	0.865	0	I Want You
7	0.0412	0.841	175520	0.866	0	11	0.212	-6.342	1	0.118	119.941	4	0.875	0	Pop - Radi
8	0.00799	0.685	209560	0.872	0	7	0.323	-6.939	1	0.0321	110.046	4	0.806	0	Tearin' up I
9	0.127	0.698	200105	0.882	0	4	0.15	-3.078	0	0.0863	106.083	4	0.462	0	Ciao Adios
10	0.905	0.42	257693	0.0802	0.857	7	0.135	-21.795	1	0.04	122.041	3	0.0513	0	Gary's The
11	0.0534	0.605	199904	0.873	0.837	1	0.314	-5.938	0	0.0344	126.026	4	0.47	0	Levels - R&
12	0.396	0.635	329907	0.53	0	8	0.118	-8.1	1	0.0328	116.631	4	0.204	0	On Bender
13	0.074	0.642	348893	0.421	0	5	0.0676	-9.32	0	0.0237	74.849	3	0.453	0	End Of Th
14	0.105	0.563	235853	0.487	0	2	0.0884	-7.775	1	0.0234	142.53	3	0.248	0	I'll Make L
15	0.468	0.578	201280	0.375	0	0	0.156	-12.951	1	0.0272	89.605	4	0.561	0	Water Run
16	0.499	0.571	301587	0.447	0	7	0.125	-8.757	1	0.0338	75.019	4	0.162	0	A Song Fo
17	0.16	0.773	211907	0.801	0	0	0.111	-5.562	1	0.0281	98.205	4	0.8	0	As Long as
18	0.066	0.798	232027	0.862	1.65E-06	11	0.168	-4.95	0	0.0336	100.031	4	0.893	0	Quit Playin

<https://developer.spotify.com/documentation/web-api/reference/tracks/get-several-tracks/>

<https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/>

## 2. Classification

### 1) 데이터 셋 소개

Acousticness	float	음원이 악기 소리로 이루어져 있는지
Danceability	float	춤추기에 얼마나 적절한지 (템포, 리듬의 일정성, 비트의 강도)
Duration_ms	int	음원의 길이
Energy	float	빠르고, 강하고, 시끄러운 정도
Instrumentalness	float	가사가 아닌 음가가 있는 소리
Key	int	음원의 조성
Liveness	float	공연 현장의 녹음으로 이루어져있는 정도

## 2. Classification

### 1) 데이터 셋 소개

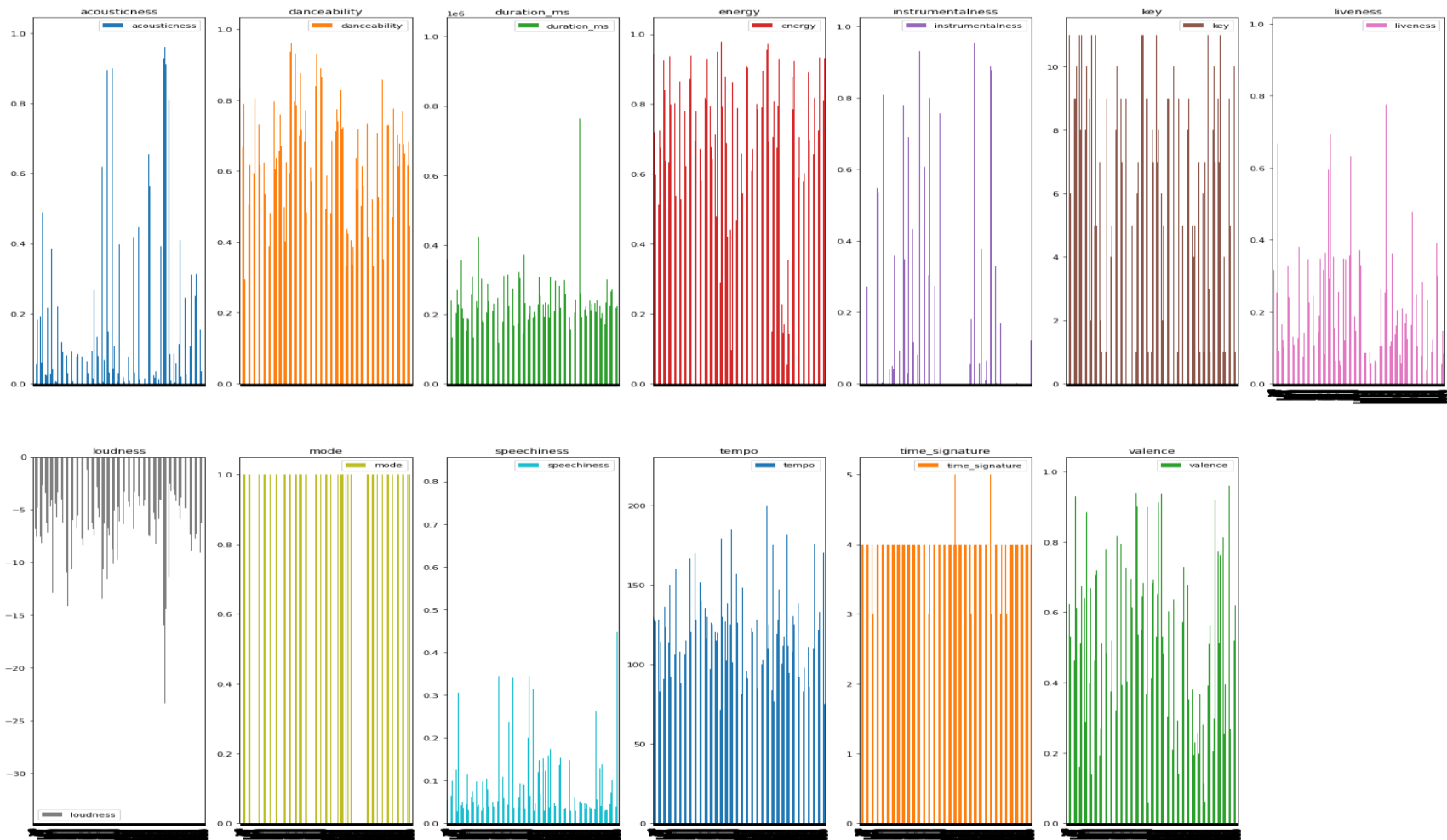
<b>loudness</b>	<b>float</b>	음원의 크기 (단위 :dB)
<b>mode</b>	<b>int</b>	장조 =1 / 단조 = 0
<b>speechiness</b>	<b>float</b>	가사의 비율 / $0.66 < :$ 내레이션 / $(0.33, 0.66)$ : 배경 음과 가사가 같이 공존 / $0.33 > :$ 가사가 없음
<b>tempo</b>	<b>float</b>	음원의 빠르기 (BPM)
<b>time_signature</b>	<b>int</b>	how many beats are in each bar / 4beat, 8beat, 16beat 등
<b>valence</b>	<b>float</b>	음원의 분위기 / 긍정적일수록 높음 / 어두울수록 낮음

\*<https://developer.spotify.com/documentation/web-api/reference/tracks/get-several-tracks/>

\*<https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/>

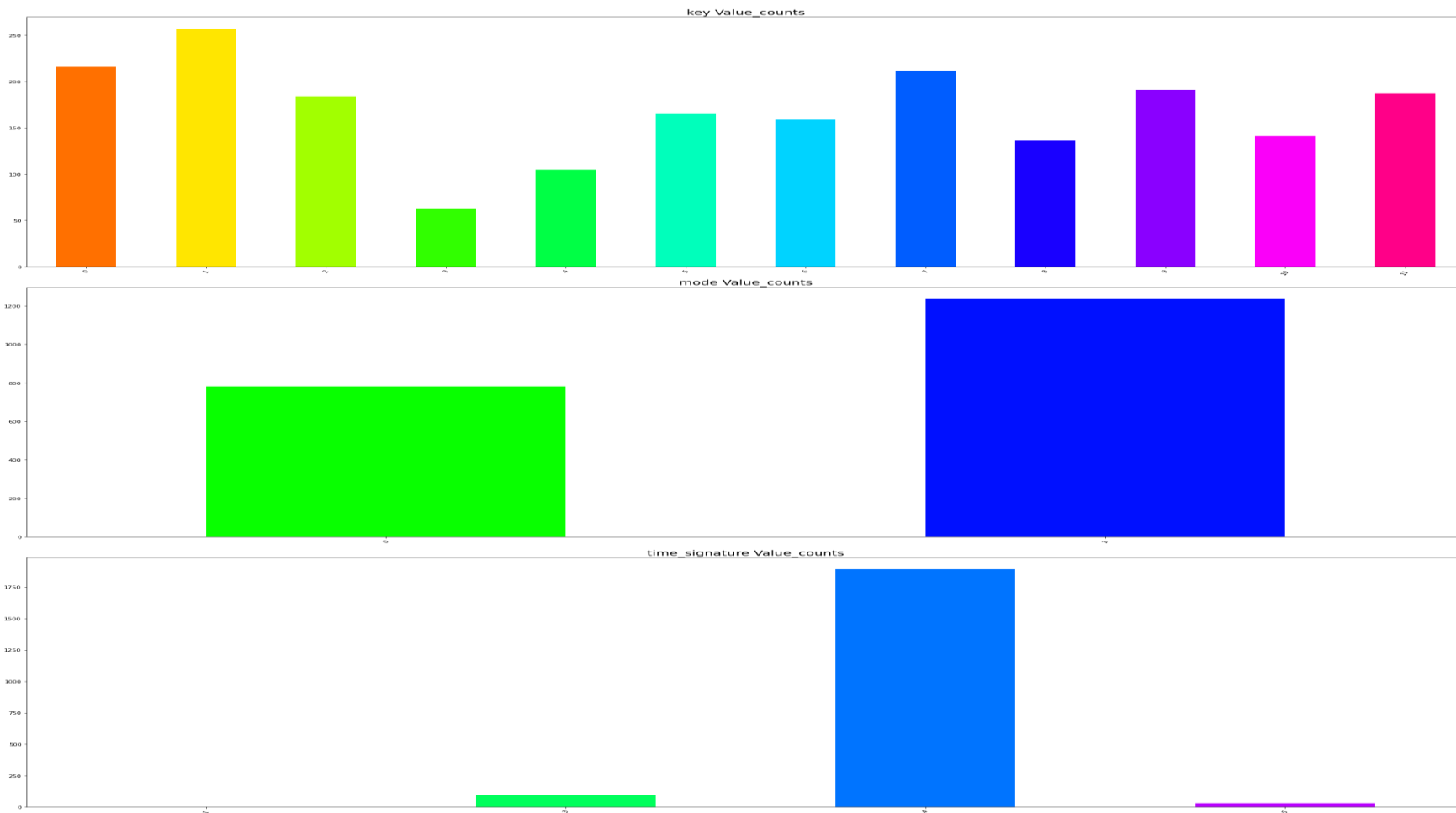
## 2. Classification

### 1) 데이터 셋 소개



## 2. Classification

### 1) 데이터 셋 소개



## 2. Classification

### 1) 데이터 셋 소개

VIF (Variance Inflation Factors, 분산팽창요인)

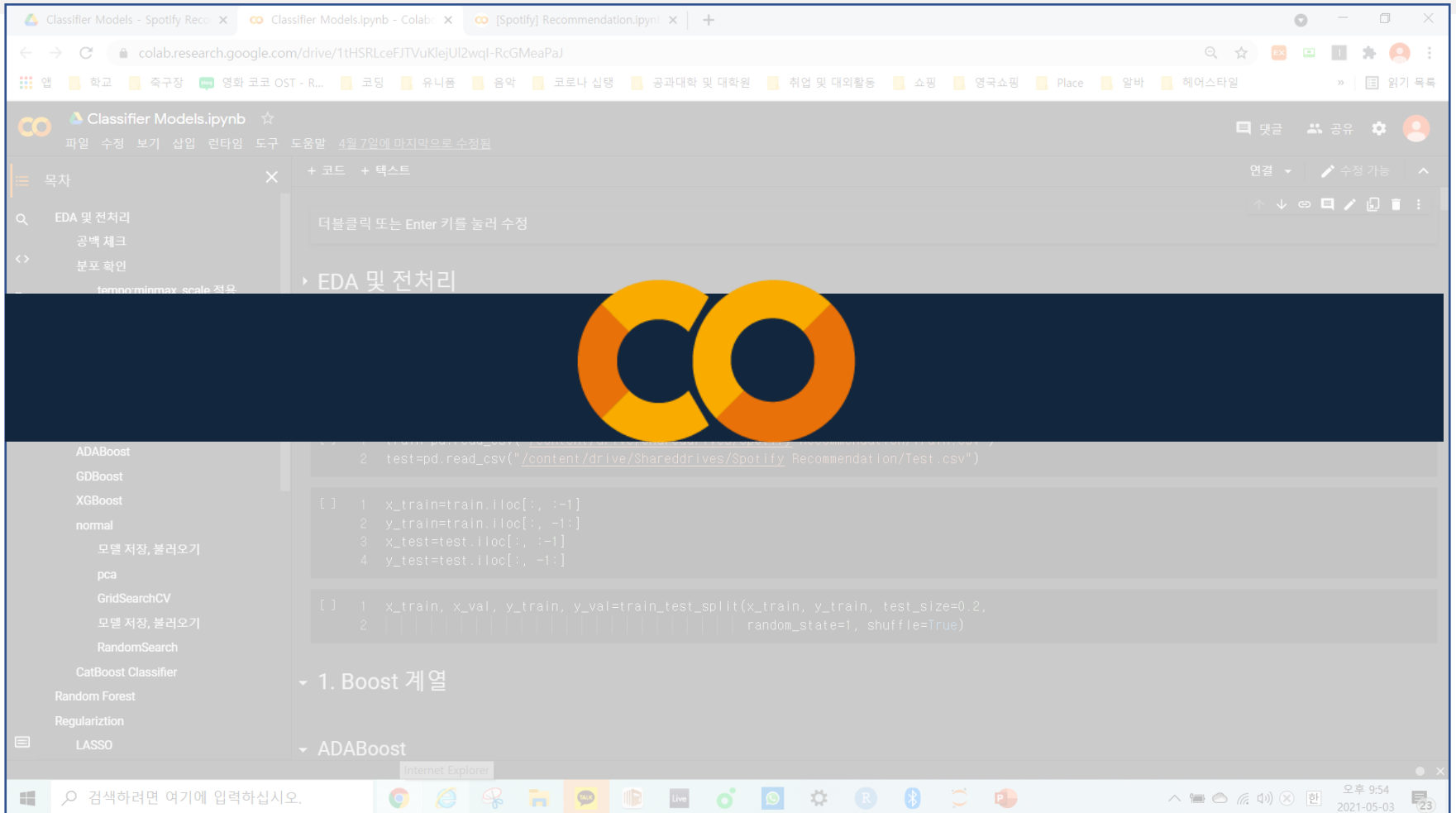
$\geq 10$  : 다중공선성 존재

$$VIF_i = \frac{1}{1 - R_i^2}$$

	VIF Factor	features
0	1.546169	instrumentalness
1	2.047553	speechiness
2	2.281422	acousticness
3	2.383877	mode
4	2.432860	liveness
5	2.965915	key
6	4.025178	valence
7	7.366587	loudness
8	8.267952	duration_ms

# 2. Classification

## 2) 코드 소개



The screenshot displays a Google Colab notebook environment. The browser tabs at the top include 'Classifier Models - Spotify Reco...', 'Classifier Models.ipynb - Colab', and '[Spotify] Recommendation.ipynb'. The address bar shows the Colab URL. The notebook interface features a left sidebar with a file explorer showing 'EDA 및 전처리', '공백 체크', and '분포 확인'. The main area contains a code editor with the following Python code:

```
1 train=pd.read_csv("/content/drive/SharedDrives/Spotify Recommendation/Test.csv")
2 test=pd.read_csv("/content/drive/SharedDrives/Spotify Recommendation/Test.csv")

[ ] 1 x_train=train.iloc[:, :-1]
     2 y_train=train.iloc[:, -1:]
     3 x_test=test.iloc[:, :-1]
     4 y_test=test.iloc[:, -1:]

[ ] 1 x_train, x_val, y_train, y_val=train_test_split(x_train, y_train, test_size=0.2,
     2 random_state=1, shuffle=True)
```

Below the code editor, a table of contents is visible, listing sections such as '1. Boost 계열' and 'ADABOOST'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 9:54 on 2021-05-03.

## 2. Classification

### 3) 실험 모델 결과

#### <Bagging>

모델명	F1 score
RandomForest (Randomized CV)	72.98%
RandomForest (Grid Search CV)	71.09%
Random Forest	72.03%

#### <Boost>

모델명	F1 score
XGBoost (Grid Search CV)	75.00%
CatBoost	100.00%
ADABOOST (Randomized CV)	71.69%
ADABOOST (Grid Search CV)	70.75%
GDBOOST (Randomized CV)	70.81%
GDBOOST (Grid Search CV)	70.24%

#### <Voting & Regularization>

모델명	F1 score
Voting( XGBoost + KNN + SVM +MLP)	72.12%
Voting(XGBoost + XGBoost)	71.84%
Voting(ADABOOST + GDBOOST)	72.30%
Lasso	63.41%
Ridge	63.81%
Elastic Net	62.86%



## 2. Classification

### 4) 결론

- Ensemble, Deep Learning, Regularization Model 중 소수의 Tabular Data에 가장 좋은 성능을 보인 모델은 Ensemble !
- Ensemble : Bagging(ex. Random Forest), Boost (ex. XG Boost, Ada Boost, CAT Boost, GD Boost) Voting 중 Boost 계열의 성능이 Best
- 한 사용자의 음악 트랙에 대한 선호 유무를 분류해주는 것 또한 추천 시스템 모델이라고 볼 수 있음

Listen on

## 3. Recommendation

# 3. Recommendation

## 1) 데이터셋 소개

### <Track Features>

: acousticness, danceability, duration\_ms, energy, instrumentality, key, liveness, loudness, mode, speechless, time\_signature, tempo, valence, song\_title, artist

Spotify\_Data .xlsx ☆ 📎 ☁

파일 수정 보기 삽입 서식 데이터 도구 도움말 어제 오후 4:33에 마지막으로 수정했습니다.





100%




\$ % .0 .00 123




Arial



10



**B** *I* S **A**


   


  


  


 









A1

# 3. Recommendation

## 1) 데이터셋 소개

### <Ratings>

: 긍정적(1), 모르는 노래(0), 부정적(-1)

: 총 29명의 users

	users	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	users	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
2	도윤_He	1	0	1	1	0	1	1	0	1	0	1	1	1	0	1	1	0	0	1	0	1	0	1
3	도윤_M	1	0	1	0	0	0	0	-1	0	1	0	0	1	0	0	1	1	0	0	0	1	0	1
4	도윤_M	-1	-1	-1	-1	-1	-1	-1	1	1	1	-1	1	1	1	-1	1	-1	-1	-1	-1	1	-1	-1
5	도윤_Jl	1	-1	1	-1	1	-1	1	1	-1	1	1	1	1	1	-1	1	1	0	1	1	1	1	0
6	도윤_Ma	1	1	0	1	1	0	1	1	1	1	1	1	1	1	1	-1	-1	1	1	1	1	0	0
7	도윤_Ml	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	도윤_Ge	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1
9	도윤_Tif	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1
10	형록1	-1	-1	-1	-1	-1	-1	0	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
11	형록2	0	0	1	1	0	1	0	-1	-1	1	0	0	0	0	0	1	0	1	1	1	0	1	1
12	형록3	-1	-1	-1	-1	-1	-1	-1	-1	0	1	1	-1	-1	-1	1	0	-1	1	0	-1	1	1	1
13	형록4	1	-1	-1	-1	-1	1	1	1	1	1	1	-1	-1	1	-1	1	0	-1	1	1	1	1	-1
14	창현_승	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	-1	0	-1	-1	-1	0
15	창현	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	-1	0	-1	-1	-1	0
16	창현_세	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
17	다현	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-1	1	0	1	1	1	1	0	-1
18	광민	0	0	0	1	0	1	-1	1	0	1	-1	1	1	-1	0	0	1	-1	1	0	1	0	1
19	정인	1	-1	0	0	0	-1	1	1	1	1	-1	0	0	-1	1	1	0	1	-1	0	0	0	0

# 3. Recommendation

## 1) 데이터셋 소개

〈Users〉 : gender(M/W), age, nationality

	A	B	C	D
1	users	gender	age	nationality
2	도윤	m	27	korea
3	도윤_Herng	m	26	singapore
4	도윤_Melanie	w	23	uk
5	도윤_Jinju	w	22	uk
6	도윤_Maya	w	22	uk
7	도윤_MI	m	26	singapore
8	도윤_Geraldine	w	26	singapore
9	도윤_Tiffany	w	23	honkong
10	형록1	m	23	korea
11	형록2	m	24	hongkong
12	형록3	w	25	korea
13	형록4	w	22	singapore
14	창현_승런	m	27	korea
15	창현	m	27	uk
16	창현_세영	m	27	korea
17	다연	w	23	korea
18	다연2_광민	m	20	singapore
19	다연3_정인	w	24	singapore
20	나윤1	w	22	korea
21	나윤2	w	22	korea
22	나윤3	w	22	singapore
23	나윤4	w	22	korea
24	효진1	w	22	korea
25	효진2	w	23	uk

## 3. Recommendation

### 1) 데이터셋 소개

#### 〈Main 데이터〉

	users	tracks	score
3631	효진1	No New Friends - SFTB Remix_DJ Khaled	-1.0
888	도윤_MI	Skepta Interlude_Drake	0.0
1720	형록3	Best I Ever Had_Drake	-1.0
1061	도윤_Geraldine	Lollipop_Lil Wayne	0.0
2266	창현	Don't Let Me Down - Zomboy Remix_The Chainsmo...	1.0
...	...	...	...
4378	윤지현	I Want You Back - Radio Edit_NSYNC	0.0
3979	효진3	Live Your Life - feat. Rihanna_T.I.	1.0
3793	효진2	No New Friends - SFTB Remix_DJ Khaled	0.0
1934	형록4	SexyBack_Justin Timberlake	1.0
2884	정인	Popular Song_MIKA	1.0

# 3. Recommendation

## 2) 추천시스템 알고리즘 및 모델 종류

### 1. 협업 필터링 (Collaborative Filtering)

- 구매, 소비한 제품에 대한 각 사용자의 평가 이용
- User-Based CF, Item-Based CF
- 대부분의 온라인 콘텐츠 제공 서비스 기업에서 활용 (ex. 아마존, 넷플릭스)

### 2. 내용 기반 필터링 (Content-Based Filtering)

- 제품의 내용을 분석 (ex. 뉴스, 블로그 포스트 / 텍스트 정보)

## 3. Recommendation

---

### 2) 추천시스템 알고리즘 및 모델 종류

#### 3. 지식 기반 필터링 (Knowledge-Based Filtering)

- CF, CBF에서 확인할 수 없는 'Why'에 대한 전문 지식을 활용

#### 4. 딥러닝 (Deep Learning)

- 사용자, 아이템의 특징 값 활용 후 예상 선호도 출력

#### 5. 하이브리드 (Hybrid)



# 3. Recommendation

## 2) 추천시스템 알고리즘 및 모델 종류

### <Memory Based>

**KNN**

**Slope One (Item CF)**

**Co-Clustering(Item&User CF)**

### <Model Based>

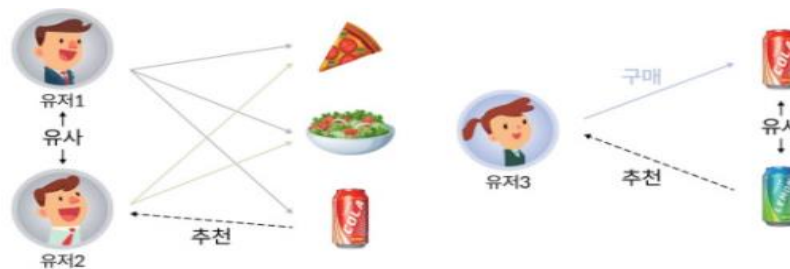
**Matrix Factorization**

**Neural Network**

### 3. Recommendation

#### 2) 추천시스템 알고리즘 및 모델 종류 : Memory Based

##### <Collaborative Filtering원리>



	사용자 기반 협업 필터링	아이템 기반 협업 필터링
특징	사용자 간의 유사도 계산하여 추천	아이템 간의 유사도를 측정하여 추천
장점	아이템 정보 없이 추천 가능, 알고리즘 구현이 간단함	아이템 정보 없이 추천 가능, 신규 사용자에게 추천이 가능함
단점	사용자, 아이템 증가에 따라 연산 급증, 신규 가입자는 유사도 부여가 어려움	사용자, 아이템 증가에 따라 연산 급증, 초기 데이터 양이 적으면 정확도 낮음

	러브라이브	아이돌마스터	어벤져스	아이언맨	워썬더	(유사도)
User A	5	-	1	2	-	-
User B	5	5	1	3	-	4.2
User C	4	5	2	-	3	4.5
User D	5	4 ②	1	1	2	4.1 ①
User E	1	2	5	5	4	2.3

### 3. Recommendation

#### 2) 추천시스템 알고리즘 및 모델 종류 : Memory Based

Slope One : 간단하면서도, 정확도가 높은 것이 특징인 Item-based CF algorithm

A simple yet accurate collaborative filtering algorithm.

This is a straightforward implementation of the SlopeOne algorithm [lemire2007a].

The prediction  $\hat{r}_{ui}$  is set as:

$$\hat{r}_{ui} = \mu_u + \frac{1}{|R_i(u)|} \sum_{j \in R_i(u)} \text{dev}(i, j),$$

where  $R_i(u)$  is the set of relevant items, i.e. the set of items  $j$  rated by  $u$  that also have at least one common user with  $i$ .  $\text{dev}(i, j)$  is defined as the average difference between the ratings of  $i$  and those of  $j$ :

$$\text{dev}(i, j) = \frac{1}{|U_{ij}|} \sum_{u \in U_{ij}} r_{ui} - r_{uj}$$

### 3. Recommendation

#### 2) 추천시스템 알고리즘 및 모델 종류 : Memory Based

Co-Clustering : 사용자와 아이템을 동시에 클러스터링하는 기법을 적용한 CF algorithm

The prediction  $\hat{r}_{ui}$  is set as:

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i}),$$

where  $\overline{C_{ui}}$  is the average rating of co-cluster  $C_{ui}$ ,  $\overline{C_u}$  is the average rating of  $u$ 's cluster, and  $\overline{C_i}$  is the average rating of  $i$ 's cluster. If the user is unknown, the prediction is  $\hat{r}_{ui} = \mu_i$ . If the item is unknown, the prediction is  $\hat{r}_{ui} = \mu_u$ . If both the user and the item are unknown, the prediction is  $\hat{r}_{ui} = \mu$ .

$C_u, C_i, C_{ui}$ : cluster of users, items, co – clusters

$$\hat{r}_{ui} = \mu_i \text{ (user unknown)}$$

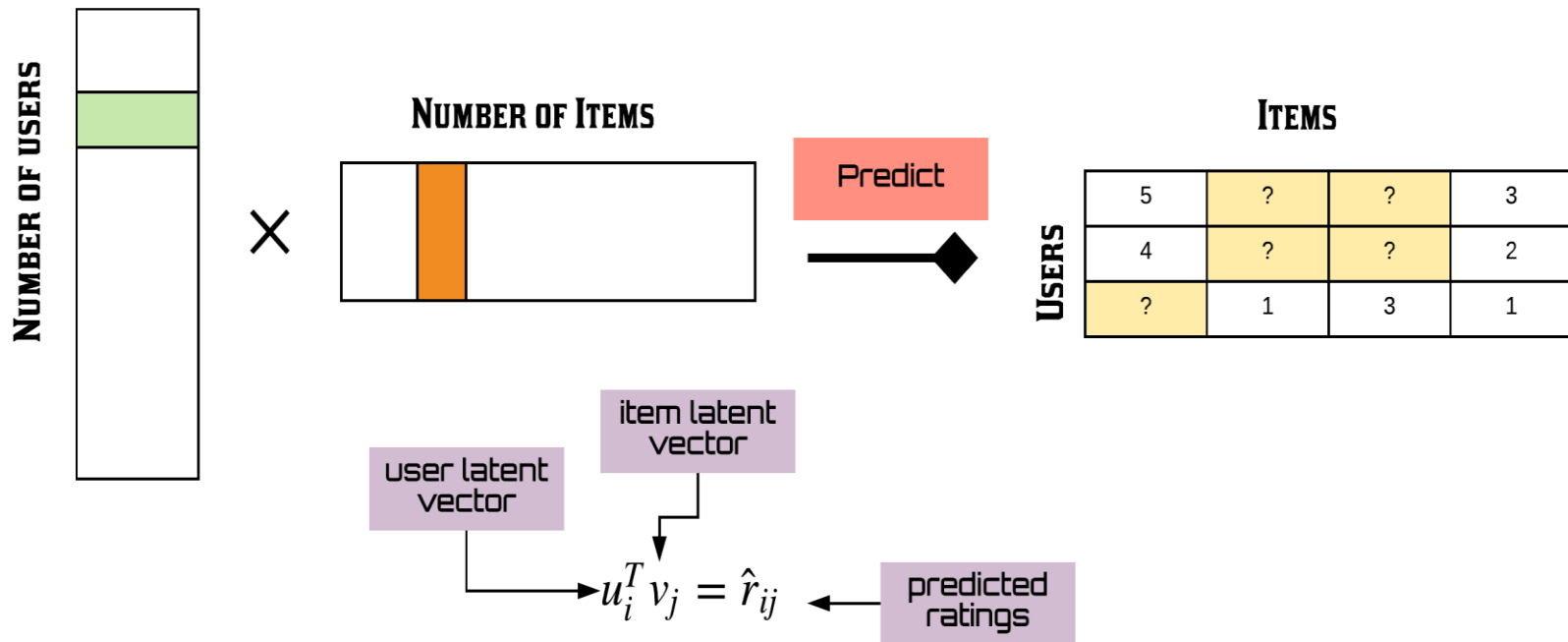
$$\hat{r}_{ui} = \mu_u \text{ (item unknown)}$$

$$\hat{r}_{ui} = \mu \text{ (user\&item unknown)}$$

### 3. Recommendation

#### 2) 추천시스템 알고리즘 및 모델 종류 : Model Based

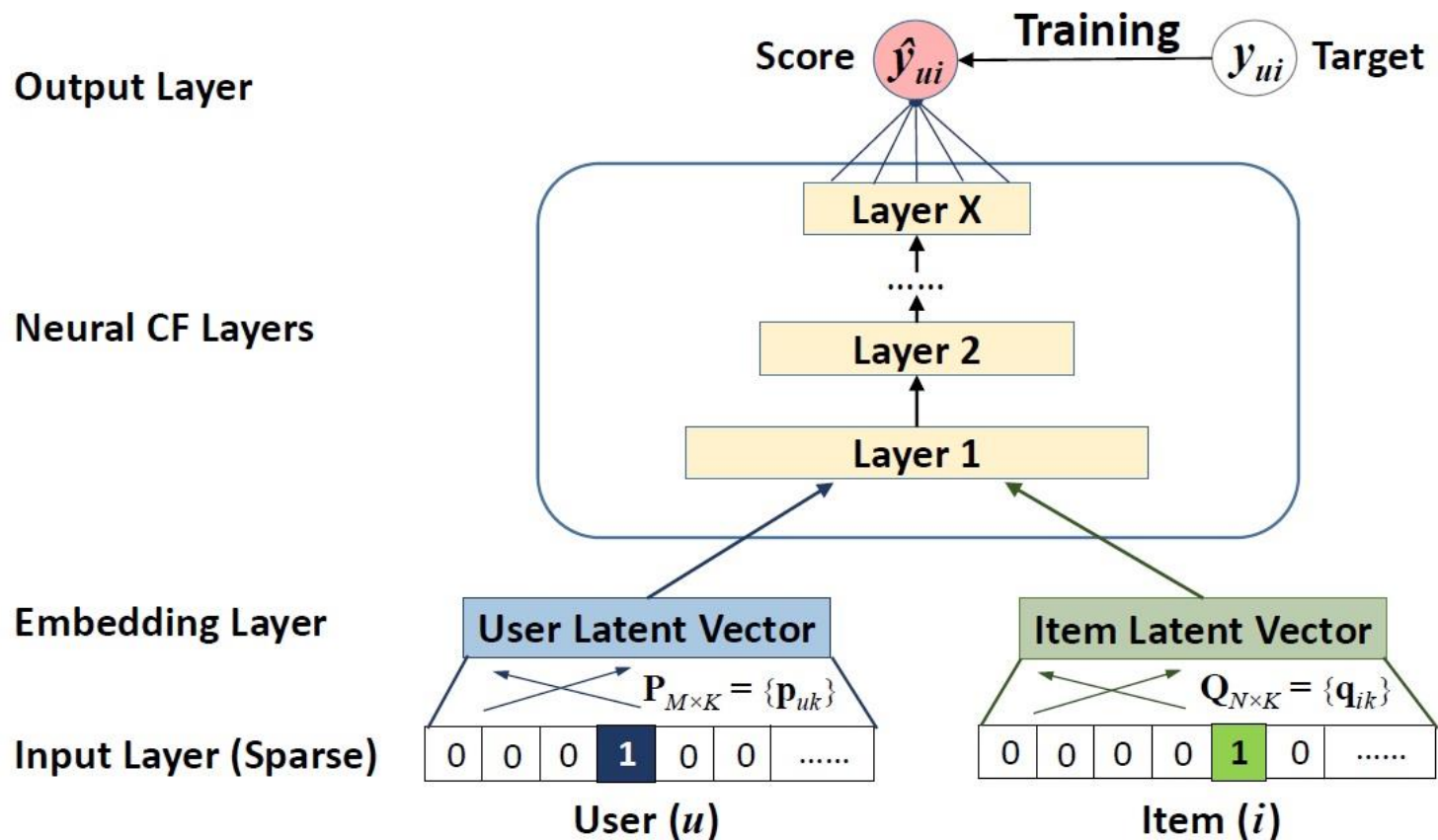
##### <Matrix Factorization 원리>



### 3. Recommendation

#### 2) 추천시스템 알고리즘 및 모델 종류 : Model Based

##### <Neural Network 원리>



# 3. Recommendation

## 2) 추천시스템 알고리즘 및 모델 종류 : Hybrid Recommender System

CF (Collaborative Filtering)+MF (Matrix Factorization)

: 각각의 단점을 보완하고 장점을 부각시키기 위해 weight를 달리 하여 합쳐 사용

```
def recommender0(recomm_list, mf):
    recommendations = np.array([mf.get_one_prediction(user, movie) for (user, movie) in recomm_list])
    return recommendations

def recommender1(recomm_list, neighbor_size=0):
    recommendations = np.array([CF_knn_bias(user, movie, neighbor_size) for (user, movie) in recomm_list])
    return recommendations


recomm_list = np.array(ratings_test.iloc[:, [0, 1]])
predictions0 = recommender0(recomm_list, mf)
RMSE2(ratings_test.iloc[:, 2], predictions0)
predictions1 = recommender1(recomm_list, 37)
RMSE2(ratings_test.iloc[:, 2], predictions1)

for i in np.arange(0, 1, 0.01):
    weight = [i, 1.0 - i]
    predictions = predictions0 * weight[0] + predictions1 * weight[1]
    print("Weights - %.2f : %.2f ; RMSE = %.7f" % (weight[0],
    weight[1], RMSE2(ratings_test.iloc[:, 2], predictions)))
```

# 3. Recommendation

## 3) Surprise 패키지 소개

: 파이썬 기반의 추천 시스템 구축을 위한 전용 패키지  
(SVD, SVD++, SLOPE ONE, K-NN, K-NN Baseline etc)

 **Surprise**  
stable

USER GUIDE

Getting Started

Basic usage

Use a custom dataset

Use cross-validation iterators

Tune algorithm parameters with GridSearchCV

Command line usage

Using prediction algorithms

How to build your own prediction algorithm

Notation standards, References

FAQ

API REFERENCE

prediction\_algorithms package

The model\_selection package

similarities module

accuracy module

dataset module

[Docs](#) » [Getting Started](#)

[Edit on GitHub](#)

## Getting Started

### Basic usage

#### Automatic cross-validation

[Surprise](#) has a set of built-in [algorithms](#) and [datasets](#) for you to play with. In its simplest form, it only takes a few lines of code to run a cross-validation procedure:

From file `examples/basic_usage.py`

```
from surprise import SVD
from surprise import Dataset
from surprise.model_selection import cross_validate

# Load the movieLens-100k dataset (download it if needed),
data = Dataset.load_builtin('ml-100k')

# We'll use the famous SVD algorithm.
algo = SVD()

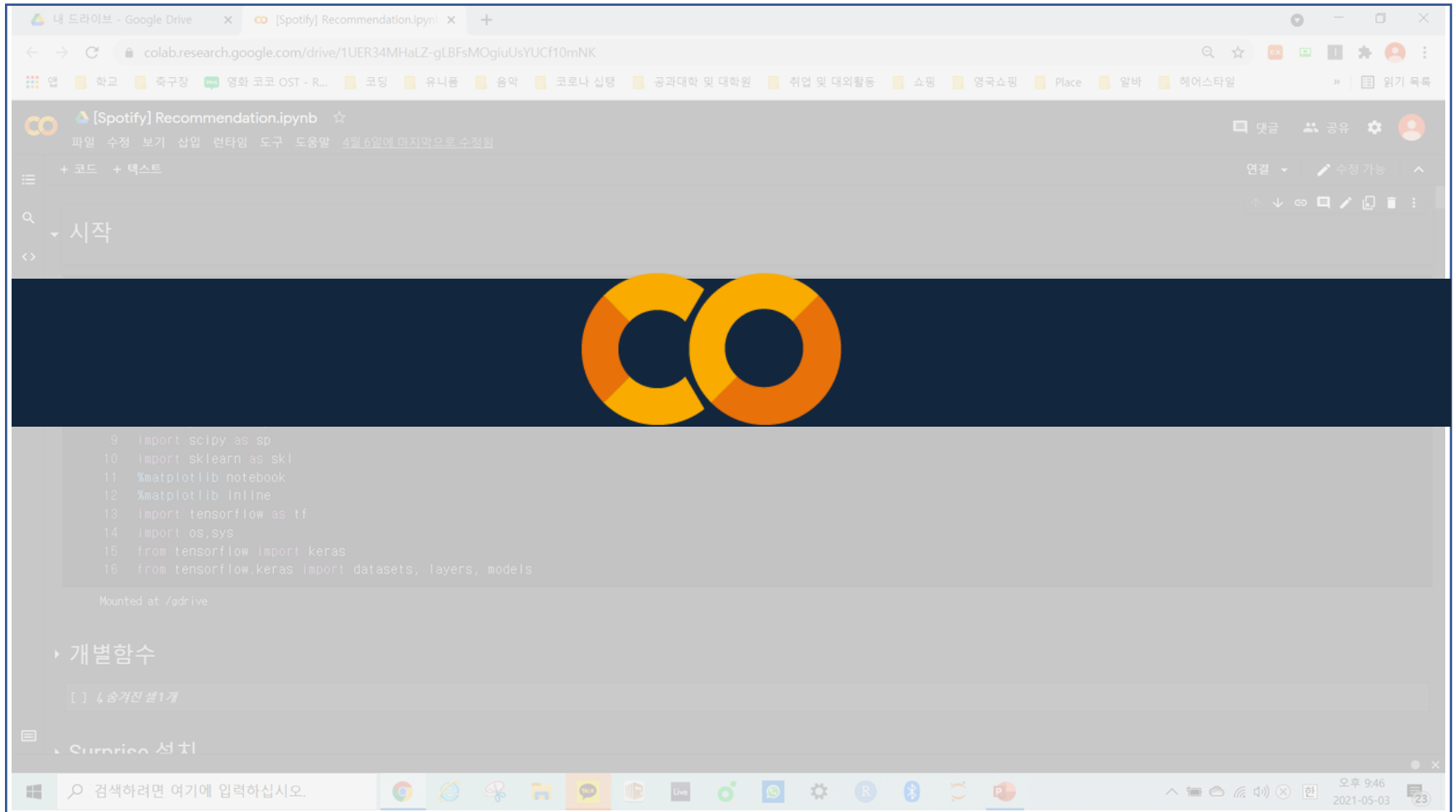
# Run 5-fold cross-validation and print results
cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
```

The result should be as follows (actual values may vary due to randomization):



# 3. Recommendation

## 4) 코드 소개



The screenshot shows a Google Colab notebook interface. The browser tabs at the top include '내 드라이브 - Google Drive' and '[Spotify] Recommendation.ipynb'. The address bar shows the URL 'colab.research.google.com/drive/1UER34MHalZ-glBFsMOgiuUsYUCf10mNK'. The notebook title is '[Spotify] Recommendation.ipynb' with a star icon. Below the title, there are tabs for '파일', '수정', '보기', '삽입', '런타임', '도구', and '도움말'. The main code area contains the following Python code:

```
9 import scipy as sp
10 import sklearn as skl
11 %matplotlib notebook
12 %matplotlib inline
13 import tensorflow as tf
14 import os,sys
15 from tensorflow import keras
16 from tensorflow.keras import datasets, layers, models
```

Below the code, it says 'Mounted at /gdrive'. The left sidebar shows a file explorer with a search bar and a list of files. The bottom status bar shows the time '오후 9:46' and the date '2021-05-03'.

### 3. Recommendation

#### 5) 실험 모델 결과

##### Memory Based

<KNN>

모델명	Base	RMSE
Basic	User	0.7712
	Item	0.7216
Means	User	0.7568
	Item	0.7254
Baseline	User	0.7545
	Item	0.7317
Z Score	User	0.7571
	Item	0.7339

<CF variation>

모델명	RMSE
Slope One	0.7591
Co Clustering	0.7479

##### Model Based

Matrix  
<Factorization>

모델명	RMSE
MF	0.7232

<Neural Net>

모델명	RMSE
Basic	0.7604
Layers	0.726
Layers, Nationality	0.725

## 3. Recommendation

### 5) 실험 모델 결과

RMSE: 0.7233

<윤지현 (a.k.a. 윤영룡) 님을 위한 추천곡>

Lucky\_\_Britney Spears => 89.42 %의 확률로 좋아하실 거예요

Get Lucky\_\_Daft Punk => 79.67 %의 확률로 좋아하실 거예요

So Sick\_\_Ne-Yo => 71.57 %의 확률로 좋아하실 거예요

Billie Jean\_\_Michael Jackson => 69.15 %의 확률로 좋아하실 거예요

When Will My Life Begin - From "Tangled"/Soundtrack Version\_\_Mandy Moore => 57.84 %의 확률로 좋아하실 거예요

<도윤 님을 위한 추천곡>

If I Were a Boy\_\_Beyoncé => 100 %의 확률로 좋아하실 거예요

Shape of You (Major Lazer Remix) [feat. Nyla & Kranium]\_\_Ed Sheeran => 100 %의 확률로 좋아하실 거예요

The Monster\_\_Eminem => 100 %의 확률로 좋아하실 거예요

Counting Stars\_\_OneRepublic => 100 %의 확률로 좋아하실 거예요

Live Your Life - feat. Rihanna\_\_T.I. => 82.24 %의 확률로 좋아하실 거예요

<도윤 님을 위한 추천곡>

Counting Stars\_\_OneRepublic => 65.27 %의 확률로 좋아하실 거예요

Shape of You (Major Lazer Remix) [feat. Nyla & Kranium]\_\_Ed Sheeran => 50.03 %의 확률로 좋아하실 거예요

The Monster\_\_Eminem => 50.01 %의 확률로 좋아하실 거예요

I Wish\_\_Stevie Wonder => 48.4 %의 확률로 좋아하실 거예요

Sexy Love\_\_Ne-Yo => 45.0 %의 확률로 좋아하실 거예요

## 4. Conclusion & Follow up Actions

## 4. Conclusion & Follow up Actions

---

- 각 음악 트랙에 대한 개인 선호 유무를 분류해주는 분류기를 생성할 수 있고 이를 개인 맞춤 음악 추천 시스템으로 칭함.
- 추천시스템 모델의 경우, 사용자의 수와 음악 트랙의 수가 좀 더 많았더라면 더 좋은 성능을 얻었을 듯함.
- 절대적으로 성능이 좋은 모델은 없으며 데이터의 특성에 따라 모델 선정 필요
- 사용자 / 음악 트랙의 특징 추가 활용 방안 필요  
(ex. 사용자의 성별, 음악 트랙의 수치적 특성들)

## 4. Conclusion & Follow up Actions

### 멤버소개

- **조장:**

김도윤(산업경영공학부 14)

- **구성원:**

기다연 (통계학과 19)

김창현 (경영대학원)

임효진 (통계학과 19)

이나윤 (통계학과 19)



A large crowd of people is visible at a concert at night. The stage is illuminated with bright green lights, and the crowd is mostly dark, with some people holding up phones or cameras. The overall atmosphere is that of a large-scale music event.

# Thank you

**[KU-BIG '21 Spring Project]**

**Spotify Music Track  
Preference Classification and Recommendation System**