



WIDE & DEEP LEARNING 기반

소비자 구매 패턴 분석을 통한 초개인화 추천 시스템 구현

2팀 ENJOY 박진 박현기 이다운 한규형 허선영



RECOMMEND

INDEX

1. 분석 배경 및 목적

배경
목적

2. 데이터 준비

데이터 정의
데이터 전처리

3. 모델링

모델 설명
적용 기법
성능 평가
웹서버 구현

4. 결론

시사점
활용방안
한계점

5. 참고 문헌



분석 배경 및 목적

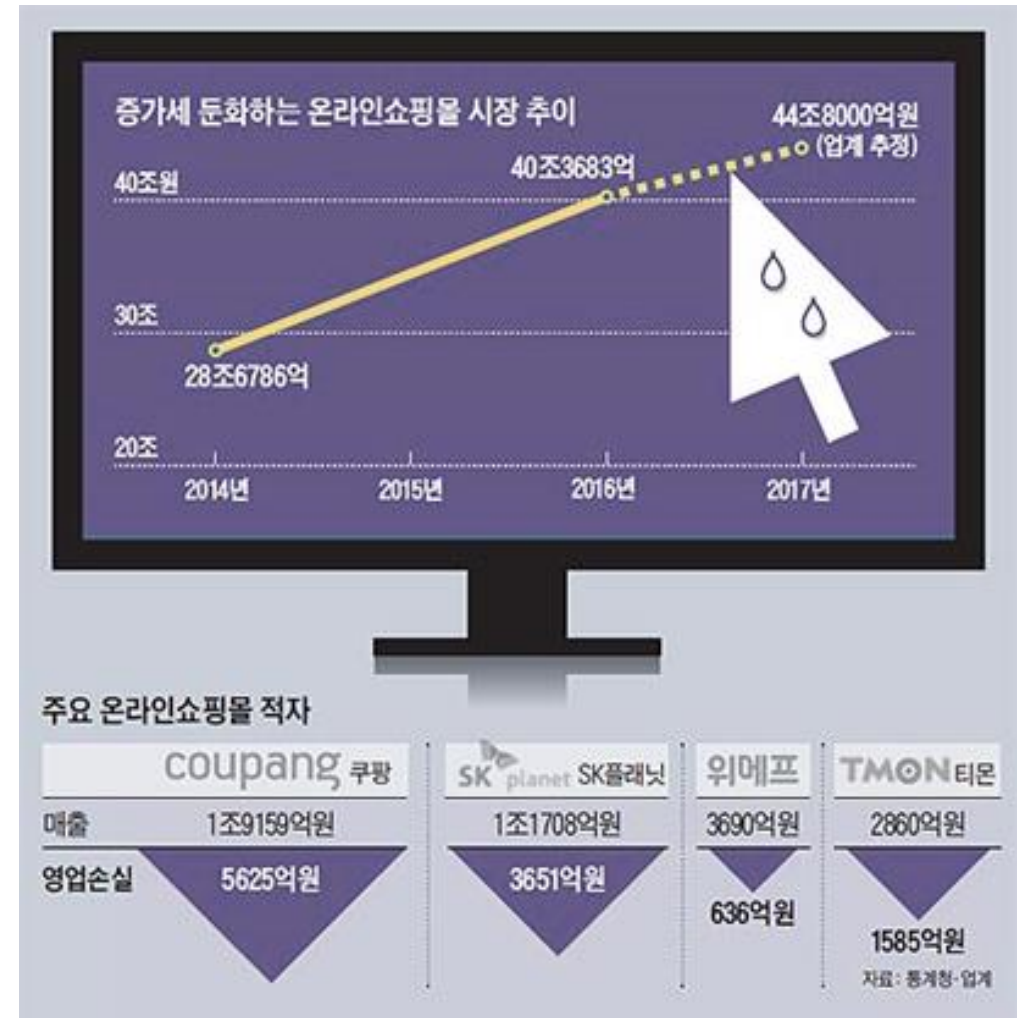
분석 배경 및 목적

온라인 쇼핑몰의 성장과 경쟁 심화

2022년 190조 '온라인쇼핑' 무한경쟁...너도나도 꿈꾸는 한국판 아마존

[NYT터닝포인트]연평균 20% 폭발적 성장, 롯데·신세계·현대百貨 등 온라인 투자 강화
쿠팡 3조, 11번가 5천억 등 '신흥' 온라인쇼핑 기업도 투자 맞불

(서울=뉴스1) 류정민 기자 | 2019-01-02 06:00 송고



기존 추천 시스템의 한계와 초개인화 마케팅의 등장

‘검색의 시대’ 지고 ‘추천의 시대’ 뜬다

2017/10/26 by 임백준

공유하기



세상을 잇(IT)는 이야기

‘검색의 시대’ 지고 ‘추천의 시대’ 뜬다

(손지은, 김성범, 김현중, 조성준, 2015)

추천시스템은 과거 구매행동을 통해 사용자가 향후 구매할 것이라 예상되는 제품을 자동으로 검색하여 제공하는 시스템이다. 이러한 추천시스템은 여러 전자상거래 업체에서 도입하고 있으며, 사용자의 편의성 및 수익에 긍정적인 영향을 미치고 있다. 하지만 사용자가 어떠한 기준을 가지고 제품을 평가하는지, 어떠한 요소가 구매 의사 결정에 영향을 미치는지는 반영할 수 없다는 한계가 있다. 이에 본 연구에서는 사용자가 직접 작성한 구매후기를 통해, 사용자 별 제품 평가요소를 활용할 수 있는 추천 모형 알고리즘을 개발하였다. 토픽 모델링을 활용하여 사용자들의 구매후기를 분석하였으며, 이러한 후기의 특성이 반영된 커널과 평가 점수가 반영된 커널 등을 함께 활용하여 다중 커널 학습 기반의 추천 모형을 개발하였다. 또한, 이러한 모형을 BestBuy 사례에 적용하여 검증하였다. 검증 결과, 기존 협업적 필터링 알고리즘보다 다중 커널 학습에 의한 추천 모형의 정확도가 우수하였고, 구매후기의 유사성을 반영하였기에, 사용자가 어떠한 요소를 평가하는지를 확인할 수 있었다. 또한, 기존 협업적 필터링 알고리즘보다 다양한 제품에 대한 추천이 가능함을 확인할 수 있었다. 본 연구는 토픽 모델링과 커널 학습 기반을 사용한 융합적인 추천모형으로서, 온라인 추천시스템의 새로운 방법을 제안한다.



추천시스템은 과거 구매행동을 통해 사용자가 향후 구매할 것이라 예상되는 제품을 자동으로 검색하여 제공하는 시스템

홈쇼핑, 고객 맞춤형 큐레이션 서비스 강화...머신러닝 적용



맞춤형 추천 인공지능 서비스를 내 앱에, 아마존 퍼스널라이즈

[이커머스 시장의 데이터드라이븐 마케팅 전략] 추천 알고리즘,
내 취향을 어떻게 그렇게 잘 알아?

[AI] "고객님, 이 물건 어때요?"...AI가 필요한 상품만 '콕콕'

"구매패턴 분석해 관심 높은 상품 추천해요"...홈쇼핑, 고객 맞춤형 큐레이션 서비스 강화

분석 배경 및 목적

목적



소비자 구매 패턴을
분석하여
구매 예정 상품군
추천



소비자의 편의 증진
및
만족도 향상



초개인화 추천을
통한 재구매 유도



매출 확대를 통한
쇼핑몰의 이윤 증대



데이터 준비

데이터 준비
데이터 정의



L.POINT 제공

총 3,988,587개 데이터

데이터 준비 데이터 정의



이용 데이터 SET	데이터 기간	사용 변수		출처
상품 구매 내역 (Product.csv)	2018.4.1. ~ 2018.9.30.	CLNT_ID	방문자에 랜덤으로 부여된 고유 ID	롯데 L.point
		SESS_ID	접속 후 세션이 시작될 때 부여된 고유 ID	
		PD_C	구매한 상품의 코드(최소단위)	
		HITS_SEQ	세션 내 활동 횟수	
회원 정보 (Custom.csv)		CLNT_ID	방문자에 랜덤으로 부여된 고유 ID	
		PD_C	구매한 상품의 코드(최소단위)	
		CLAC1_NM	상품의 대분류	
상품 분류 (Master.csv)		CLAC3_NM	상품의 소분류	
세션 내 활동 내역 (Session.csv)		CLNT_ID	방문자에 랜덤으로 부여된 고유 ID	
		SESS_ID	접속 후 세션이 시작될 때 부여된 고유 ID	
		SESS_SEQ	세션이 시작될 때 부여된 일련번호	

데이터 준비 테이블 정제

PRODUCT

CLNT_ID	SESS_ID	HITS_SEQ	PD_C
4139680	7605037	12	642112
4140076	10189797	13	570603
4142395	6158159	85	179538
4144914	7935714	12	554336
4144917	6406509	78	190306

SESSION

CLNT_ID	SESS_ID	SESS_SEQ
5873599	8641867	9
5873599	6616320	21
5873599	5886172	40
5873884	1050889	15
5874461	10298270	5

CUSTOM

CLNT_ID	CLNT_GENDER	CLNT_AGE
4830726	F	30
4830874	F	40
4830975	F	30
4831275	F	30
4825325	F	30

MASTER

PD_C	CLAC1_NM	CLAC3_NM
64382	스포츠패션	남성스포츠티셔츠
62282	속옷/양말/하웨어	여성일반양말
61729	유아동의류	영유아티셔츠/타
61537	패션잡화	남성부츠
58820	남성의류	남성남방셔츠



Product 테이블의 HITS_SEQ 가 1인 행은 이상치로 판단 후 제거



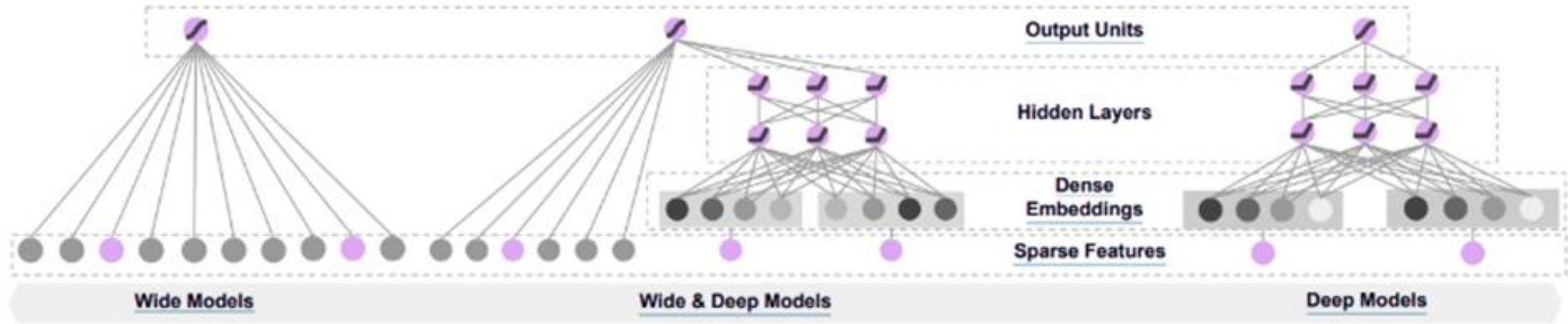
Product 테이블 기준으로 병합하여 구매내역 테이블 생성

데이터 준비 테이블 정제

CLNT_ID	SESS_ID	CLAC3_NM	SESS_SEQ	CLNT_GENDER	CLNT_AGE
451	2074820	샴푸, 림글로즈/틴트, 칫솔, 트리트먼트/팩, 아이브로우, 치약, 액상세탁세제, 섬유유연제/향기...	689	F	40.0
451	4327783	여성샴푸, 섬유유연제/향기지속제	615	F	40.0
451	4404129	여성스니커즈, 남성슬리폰	608	F	40.0
451	4464232	여성티셔츠/탑, 여성스커트	604	F	40.0
451	5050230	림글로즈/틴트, 페이스클렌저, 스킨/토너, 바디워시, 헤어에센스, 여성샴푸, 여성바지, 여성원피...	578	F	40.0

- ✓ 병합 과정에서 발생한 성별, 나이 변수의 결측치는 분석에서 제외
- ✓ 한 세션에서 2개 품목 이상인 거래
- ✓ 10번 이상의 세션이 발생한 고객 중 최근 10개의 세션
- ✓ 유저 별 SESS_SEQ 가장 높은 행 Test 나머지 9개의 행 Train으로 구분

Wide & Deep learning



Deep

임베딩을 통한 피드-포워드 신경망을 통해 특성 학습
새로운 변수 조합을 탐구하며 일반화에 강점

Wide

동시에 구매되는 품목을 학습하고 과거 이력에서 이용 가능한 상관관계를 뽑아내는 작업
암기와 해석에 강점

Wide
& Deep

입력값이 희소한 데이터를 임베딩을 통한 피드-포워드 신경망과
변수 변환을 통한 선형 모델을 함께 훈련

데이터 준비 Doc2Vec

여성티셔츠/탑,여성원피스

유아용욕조,분말세탁세제

여성재킷,남성런닝/트레이닝화,여성바지,여성티셔츠/탑

넵비,프라이팬

1	-0.005998	0.012753	-0.033853	-0.029541	-0.006895	0.041990	0.043170	0.019838	-0.022557
2	0.001308	-0.000922	0.000228	0.001078	0.000735	-0.000263	0.001063	-0.000943	-0.001087
3	-0.001810	0.000855	0.000017	-0.000029	-0.000377	-0.001479	0.001441	-0.000418	0.000131
4	0.003560	0.017498	0.017770	0.020282	0.026915	0.037074	0.003533	-0.007653	0.018898

- Deep 모델의 상품 임베딩
- Wide 모델에서 해당 상품에 대한 특징을 살리기 위해 사용

Wide & Deep learning – wide component

[선형모델]

CLAC1_NM_남성의류	CLAC1_NM_냉동식품	CLAC1_NM_냉장/세탁가전
0	0	0
1	0	0
1	0	0
0	0	0
0	0	0

교차곱변환

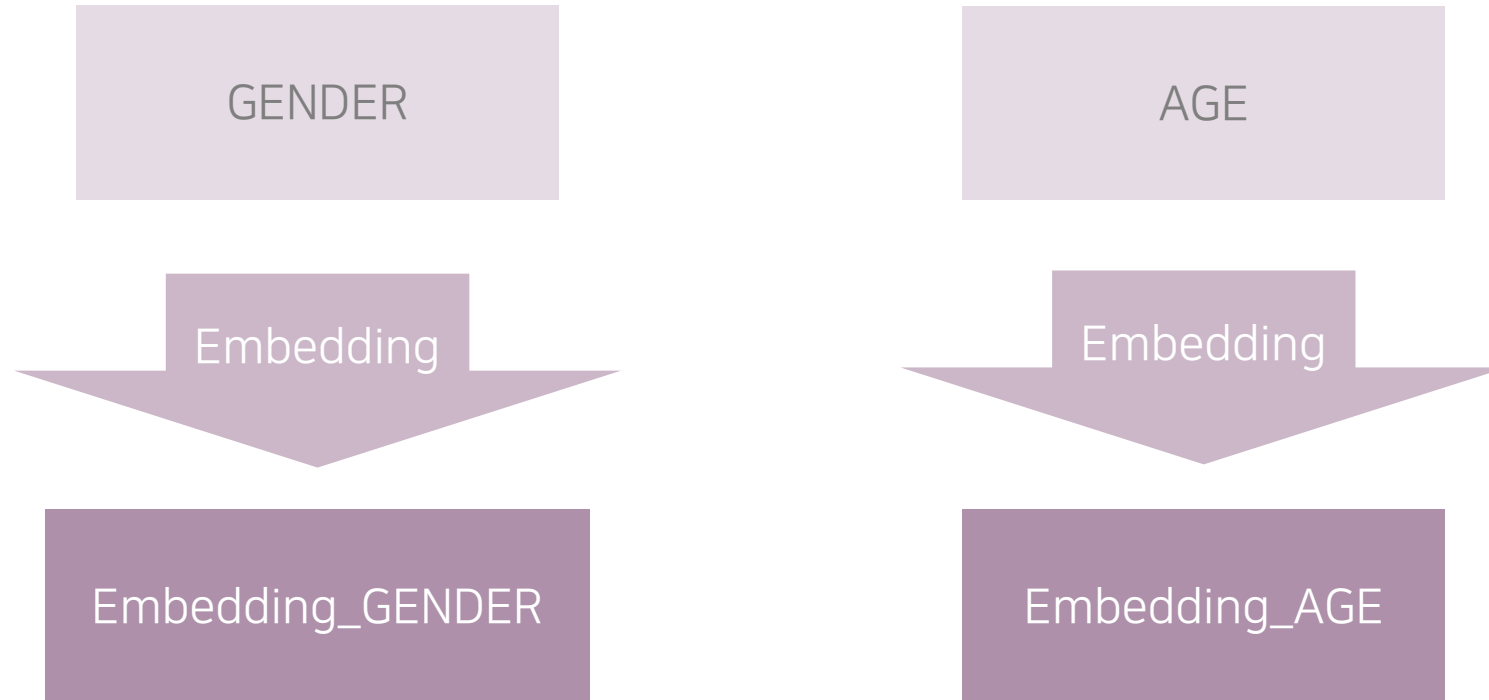
[비선형모델]

CLAC1_NM_구기/필드스포츠허브 +CLAC1_NM_아웃도어/레저_30	CLAC1_NM_구기/필드스포츠허브 +CLAC1_NM_여성의류_00	CLAC1_NM_구기/필드스포츠허브 +CLAC1_NM_여성의류_01	CLAC1_NM_구기/필드스포츠허브 +CLAC1_NM_여성의류_02
0	1	0	0
0	1	0	0
0	0	0	1
0	0	0	0
0	1	0	0

- ✓ 대분류를 이진변수화하고 교차곱 변환을 하여 일반화 선형 모델에 비선형성을 더해줌
- ✓ 과적합을 막아주고 변수간의 상호작용을 학습하기 때문에 효과적이고 해석이 용이
- ✓ 훈련 데이터에 나타나지 않으면 일반화가 어려움

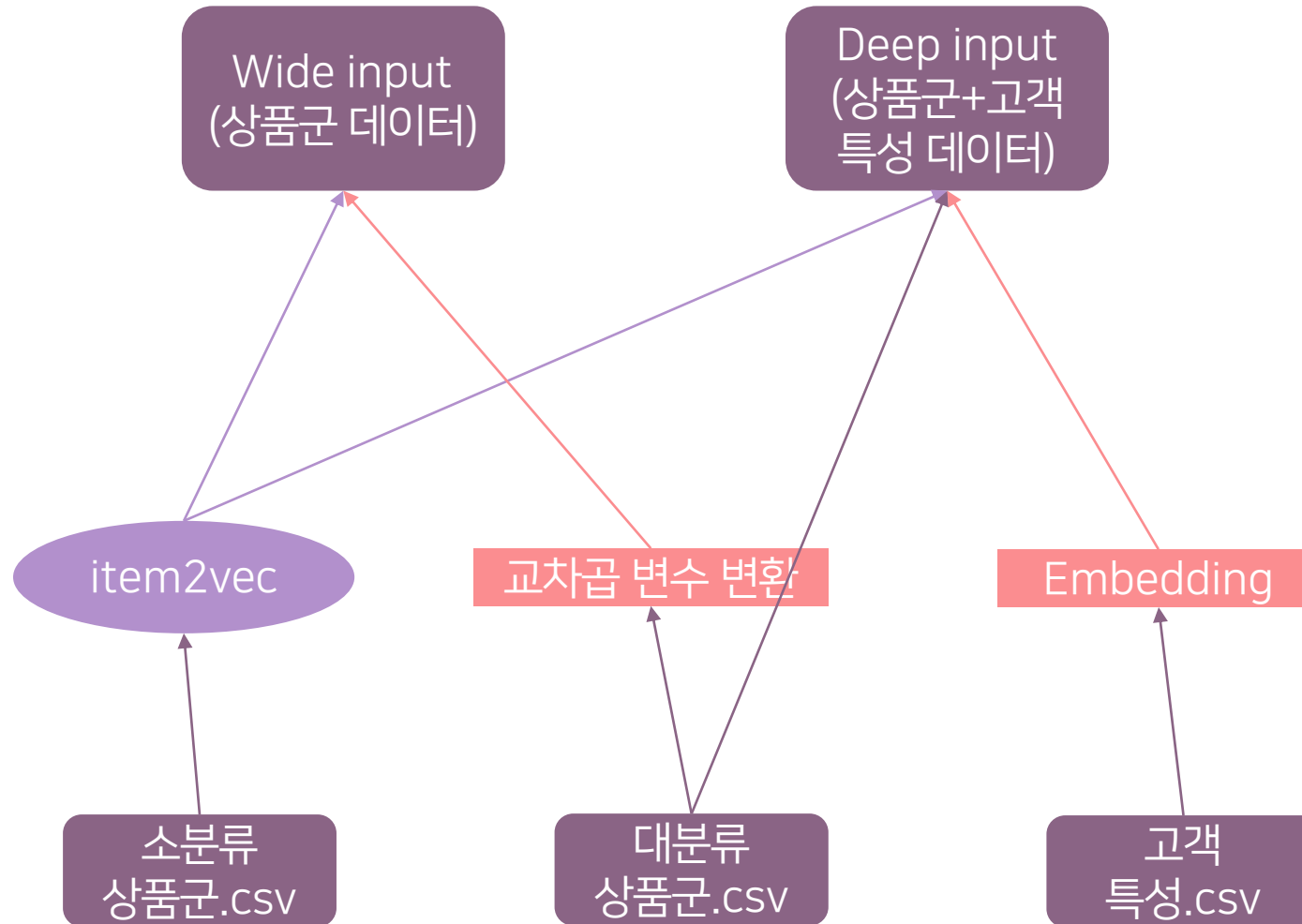


Wide & Deep learning – deep component



- 유저의 정보를 임베딩 하여 특성으로 사용
- 임베딩을 통해 학습되지않은 변수 조합에 대한 효과적인 일반화
- 고차원 임베딩은 관계없는 아이템을 추천을 하기 때문에 저차원 임베딩 사용

Wide & Deep learning - input





모델링

Wide model + deep model

Wide

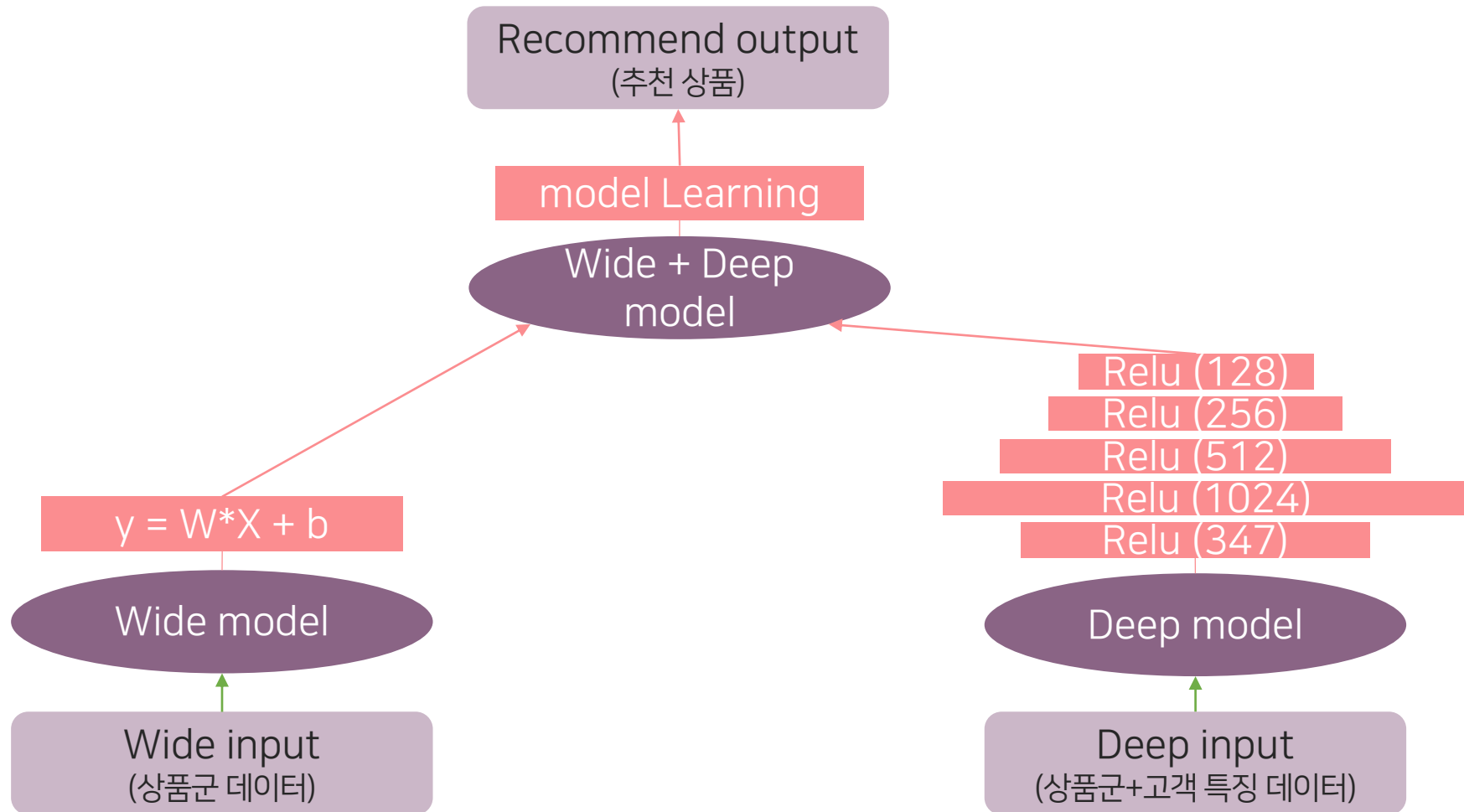
```
1 A = tf.get_variable("A", shape=[wide_size, output_size])
2 b = tf.Variable(tf.random_normal([output_size]))
3
4 wide = tf.add(tf.matmul(Wide, A), b)
```

Deep

```
1 W1 = tf.get_variable("W1", shape=[347, 512], initializer=He)
2 b1 = tf.Variable(tf.random_normal([512]))
3 L1 = tf.nn.relu(tf.matmul(Deep, W1) + b1)
4
5 W2 = tf.get_variable("W2", shape=[512, 256], initializer=He)
6 b2 = tf.Variable(tf.random_normal([256]))
7 L2 = tf.nn.relu(tf.matmul(L1, W2) + b2)
8
9
10 W3 = tf.get_variable("W3", shape=[256, 128], initializer=He)
11 b3 = tf.Variable(tf.random_normal([128]))
12 L3 = tf.nn.relu(tf.matmul(L2, W3) + b3)
13
14 W4 = tf.get_variable("W4", shape=[128, 64], initializer=He)
15 b4 = tf.Variable(tf.random_normal([64]))
16 L4 = tf.nn.relu(tf.matmul(L3, W4) + b4)
17
18 W5 = tf.get_variable("W5", shape=[64, output_size], initializer=He)
19 b5 = tf.Variable(tf.random_normal([output_size]))
20
21 deep = tf.add(tf.matmul(L4, W5), b5)
```

1 hypothesis = wide + deep

Modeling Training 과정



Learning

```
1 # parameters
2 keep_prob = 0.5
3 learning_rate = 1e-4
4 training_epochs = 10
5 batch_size = 256
```

```
1 with tf.Session() as sess:
2     sess.run(tf.global_variables_initializer())
3     Model_train()
```

```
[epoch: 0][Cost: 0.39455][2019-11-18 03:40:41]
[epoch: 1][Cost: 0.04384][2019-11-18 03:40:52]
[epoch: 2][Cost: 0.02528][2019-11-18 03:41:02]
[epoch: 3][Cost: 0.01979][2019-11-18 03:41:12]
[epoch: 4][Cost: 0.01731][2019-11-18 03:41:21]
[epoch: 5][Cost: 0.01594][2019-11-18 03:41:31]
[epoch: 6][Cost: 0.01506][2019-11-18 03:41:41]
[epoch: 7][Cost: 0.01443][2019-11-18 03:41:51]
[epoch: 8][Cost: 0.01394][2019-11-18 03:42:02]
[epoch: 9][Cost: 0.01353][2019-11-18 03:42:13]
Learning Finished!
```

모델 정확도



```
1 with tf.Session() as sess:
2     sess.run(tf.global_variables_initializer())
3     YY = tf.equal(Y, 1)
4     XX = tf.greater(tf.sigmoid(hypothesis), 0.5)
5     correct = tf.equal(XX, YY)
6     # score = sess.run(correct, feed_dict={Wide: Wide_data_test,
7     #                                     Deep_un_embed: Deep_data_test[Deep_col[:-2]],
8     #                                     Deep_embed: Deep_data_test[Deep_col[-2:]],
9     #                                     Deep: Deep_data_test,
10    #                                     Y: test_label})
11    accuracy = tf.reduce_mean(tf.cast(correct, 'float'))
12
13    feed_dict_test = {Wide: Wide_data_test,
14                      Deep_un_embed: Deep_data_test[Deep_col[:-2]],
15                      Deep_embed: Deep_data_test[Deep_col[-2:]],
16                      Y: test_label}
17
18    print("Accuracy: {:>.5f}".format(sess.run(accuracy, feed_dict=feed_dict_test)))
```

Accuracy: 0.50226

ID 13516 의 실제 구매/추천 상품

```
1 train_docs[105]
```

(['남성정장바지', '기타패션잡화', '유아용품티슈', '남성티셔츠', '남성정장셔츠'], 13516)

```
1 list(score.iloc[105].nlargest(10).index)
```

```
['냉동고',
 '상푸/린스세트',
 '남성양말선물세트',
 '머플러',
 '남성골프베스트',
 '롤플레이잉완구',
 '유아동스포츠스웨트셔츠/후드/집업',
 '여성스포츠전신/원피스',
 '크림/밤/오일',
 '유아동침구매트']
```

Model 개선

- ✓ 추천 상품 확인: 실제 구매한 상품과 추천 상품 비교
 - 정확도보다 구매한 상품과 추천된 상품 간의 유사성이 더 중요
- ✓ 파라미터 변경: 추천된 상품을 구매할 확률 개선
 - Deep model의 Relu layer 추가
 - 훈련 횟수
 - learning_rate
 - output 기준 score



모델링 개선 Model Learning

Learning

```
1 # parameters
2 keep_prob = 0.7
3 learning_rate = 1e-7
4 training_epochs = 501
5 batch_size = 256
```

```
1 with tf.Session() as sess:
2     sess.run(tf.global_variables_initializer())
3     Model_train()
```

```
[epoch: 0] [Cost: 1.23259] [2019-11-20 10:20:37]
[epoch: 100] [Cost: 0.52140] [2019-11-20 10:54:51]
[epoch: 200] [Cost: 0.20757] [2019-11-20 11:29:57]
[epoch: 300] [Cost: 0.08334] [2019-11-20 12:05:47]
[epoch: 400] [Cost: 0.03852] [2019-11-20 12:41:05]
[epoch: 500] [Cost: 0.02341] [2019-11-20 13:17:14]
Learning Finished!
```



모델 정확도

```
1 with tf.Session() as sess:
2     sess.run(tf.global_variables_initializer())
3     YY = tf.equal(Y, 1)
4     XX = tf.greater(tf.sigmoid(hypothesis), 0.7)
5     correct = tf.equal(XX, YY)
6     # score = sess.run(correct, feed_dict={Wide: Wide_data_test,
7     #                                     Deep_un_embed: Deep_data_test[Deep_col[:-2]],
8     #                                     Deep_embed: Deep_data_test[Deep_col[-2:]],
9     #                                     Deep: Deep_data_test,
10    #                                     Y: test_label})
11     accuracy = tf.reduce_mean(tf.cast(correct, 'float'))
12
13     feed_dict_test = {Wide: Wide_data_test,
14                       Deep_un_embed: Deep_data_test[Deep_col[:-2]],
15                       Deep_embed: Deep_data_test[Deep_col[-2:]],
16                       Y: test_label}
17
18     print("Accuracy: {:.5f}".format(sess.run(accuracy, feed_dict=feed_dict_test)))
```

Accuracy: 0.65597



ID 13516 의 실제 구매/추천 상품

```
1 train_docs[105]
```

(['남성정장바지', '기타패션잡화', '유아용품티슈', '남성티셔츠', '남성정장셔츠'], 13516)

```
1 list(score.iloc[105].nlargest(10).index)
```

['영유아티셔츠/탑',
'남성남방셔츠',
'면봉/화장솜',
'남성정장셔츠',
'여성가운',
'필통',
'오메가3/기타추출오일',
'기타냉동간편식',
'성인이불/아불커버',
'오리고기']

Model 추천 상품 비교



개선 전 추천 상품

```
1 train_docs[105]
```

(['남성정장바지', '기타패션잡화', '유아용품티슈', '남성티셔츠', '남성정장셔츠'], 13516)

```
1 list(score.iloc[105].nlargest(10).index)
```

['냉동고',
'샴푸/린스세트',
'남성양말선물세트',
'머플러',
'남성골프베스트',
'롤플레이밍완구',
'유아동스포츠헌트셔츠/후드/집업',
'여성스포츠헌트/원피스',
'크림/밤/오일',
'유아동침구매트']

개선 후 추천 상품

```
1 train_docs[105]
```

(['남성정장바지', '기타패션잡화', '유아용품티슈', '남성티셔츠', '남성정장셔츠'], 13516)

```
1 list(score.iloc[105].nlargest(10).index)
```

['영유아티셔츠/탑',
'남성남방셔츠',
'면봉/화장솜',
'남성정장셔츠',
'여성가운',
'필통',
'오메가3/기타추출오일',
'기타냉동간편식',
'성인이불/이불커버',
'오리고기']



모델링



웹 페이지 구현



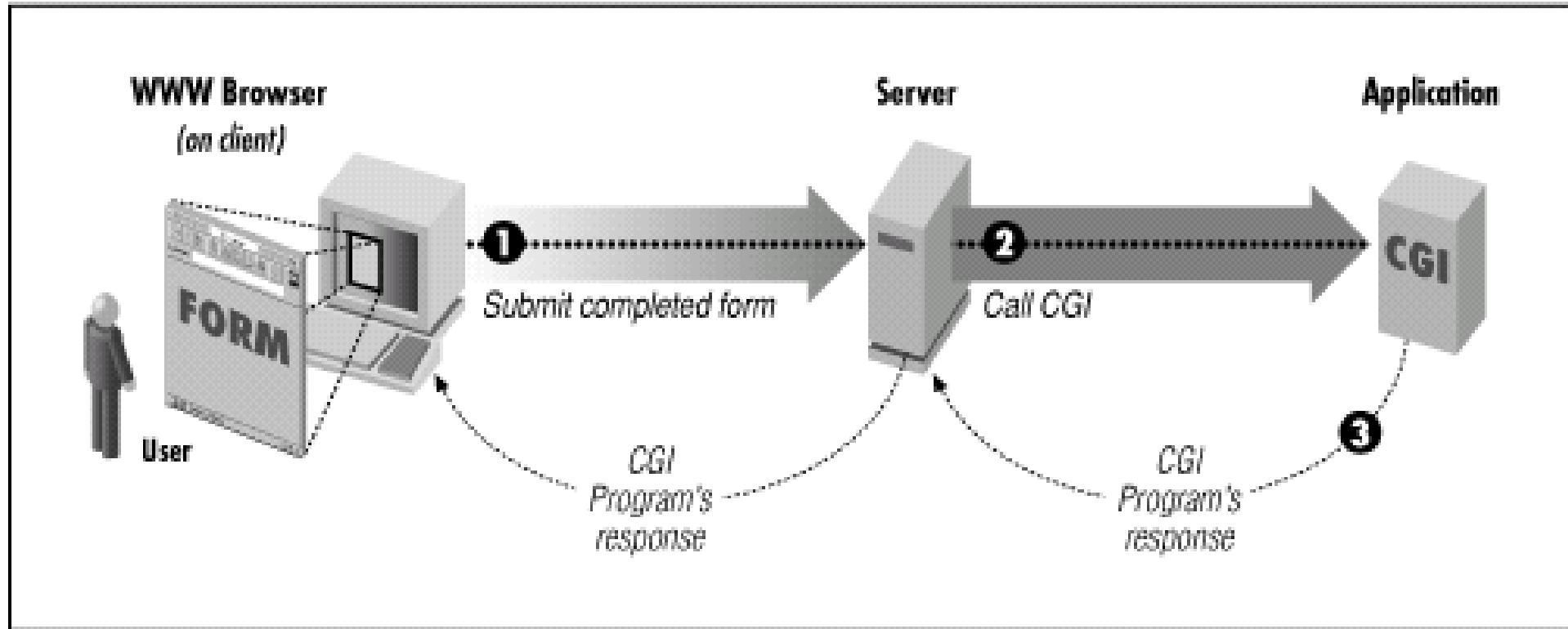
Common Gateway Interface

공통 게이트웨이 인터페이스의 약어로, 웹 서버 상에서 사용자 프로그램을 동작시키기 위한 조합을 의미함

이전에 서버 프로그램에서 다른 프로그램을 불러내고, 그 처리 결과를 클라이언트에 송신하는 방법이 제안되었음. 이를 실현하기 위한 서버 프로그램과 외부 프로그램과의 연계법을 정한 것이 CGI.

모델링 - 웹서버 구현

'CGI'란



쉽게 말해 CGI는 웹 서버를 운영하는 사람이 사용자들로부터 특정 정보를 얻어 자신의 프로그램에 사용하려고 할 때 필요한 인터페이스

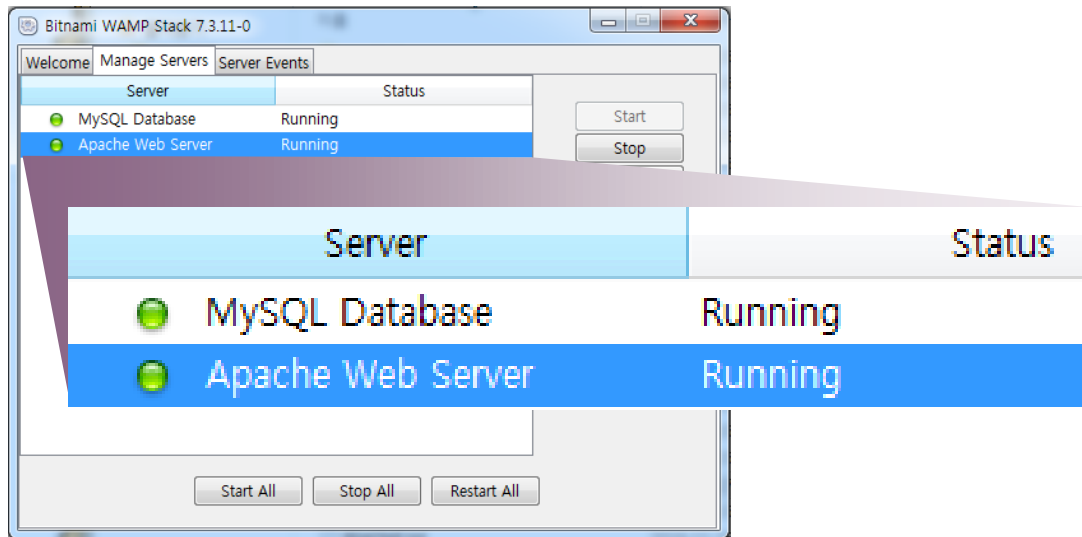
모델링 - 웹페이지 구현

비트나미 패키지 설치



[bitnami WAMP Stack]

Windows 환경에서 Apache, Mysql, Php를
한번에 설치할 수 있는 통합 솔루션



Apache의 htdoc파일에서 File 태그 추가

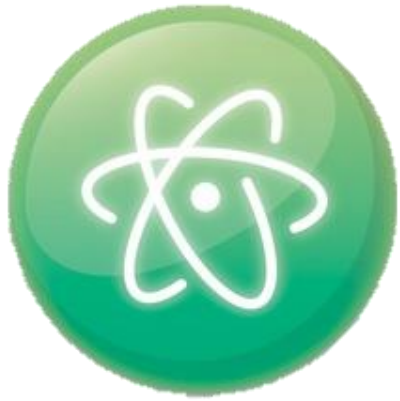
```
<Files "*.py">
```

```
options ExesCGI  
AddHandler cgi-script .py
```

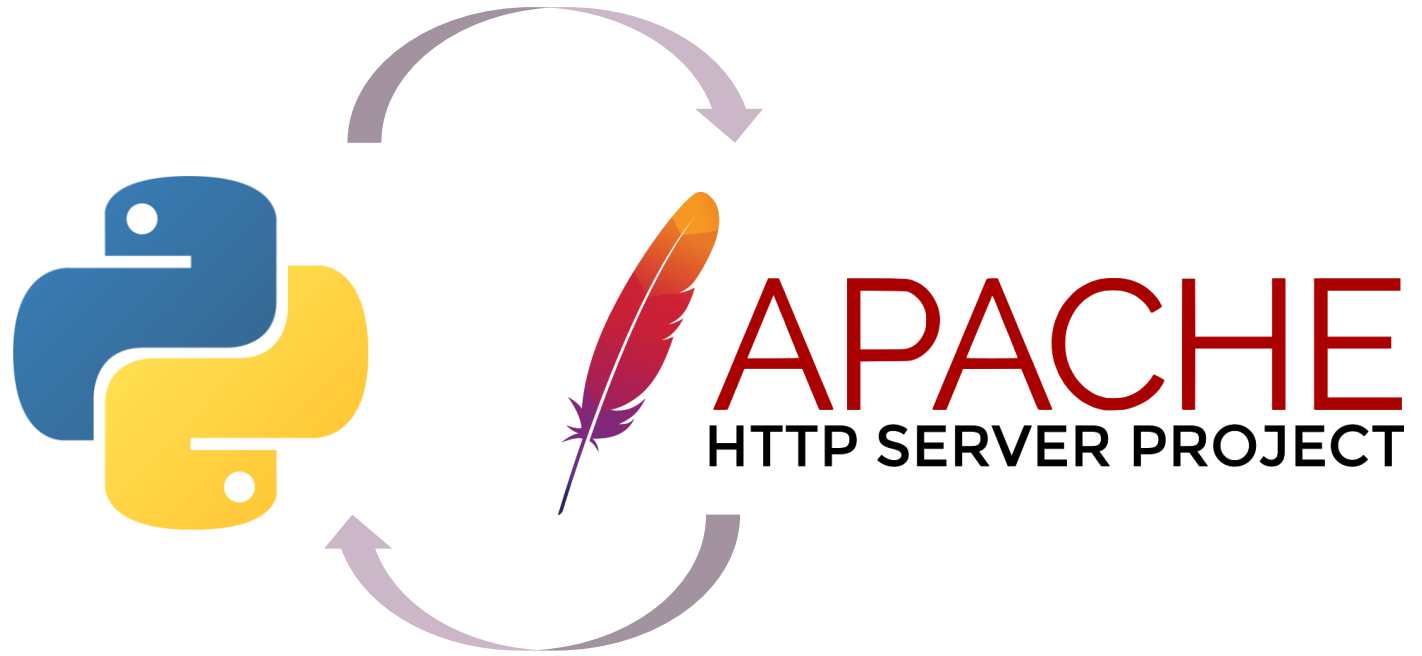
```
</Files>
```

∴ 확장자가 .py인 파일을 CGI방식으로 실행

파이썬 응용프로그램 활용 방법

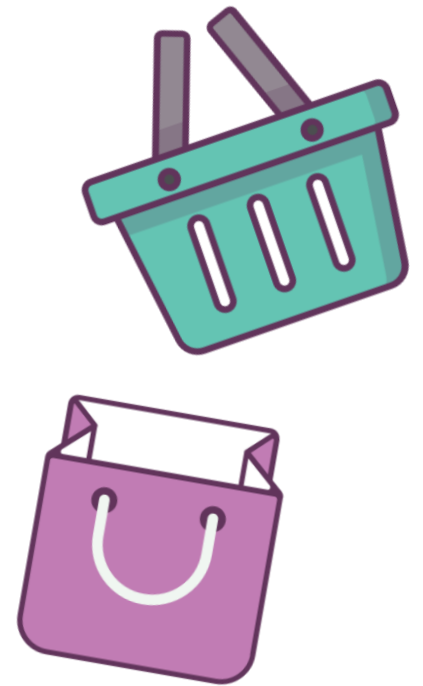
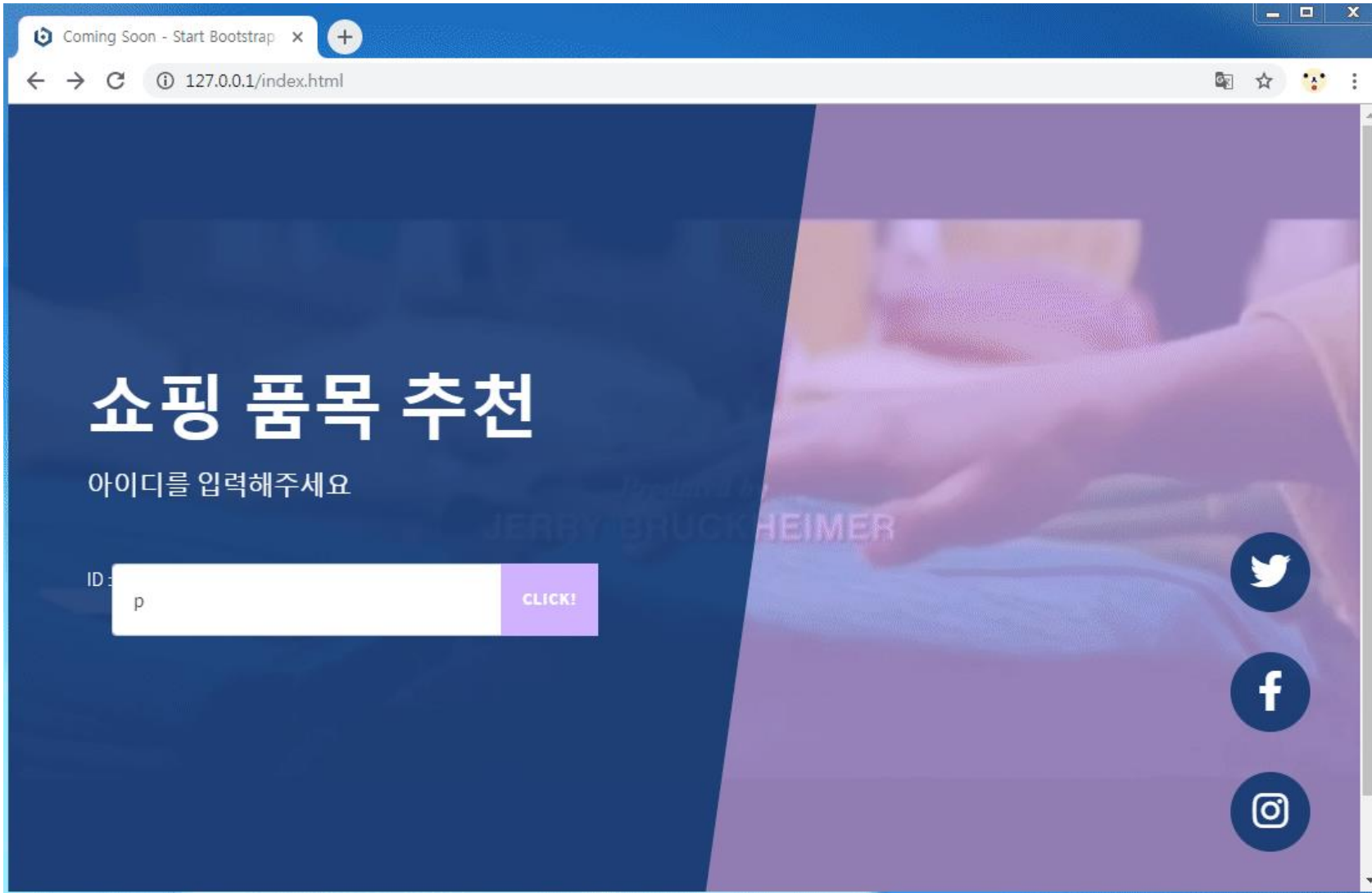


아톰 에디터를 통해
html, python코드를 편집



CGI를 통해 파이썬과 웹서버(아파치)를 연동시켜서
웹 브라우저에 파이썬 소스코드에 대한 결과가 눈에 보이게 함

모델링 - 웹페이지 구현 사용자 화면





결론

결론

시사점

기존 추천시스템과의 차별성

- ① 후기가 부족한 전자상거래 기업에도 적용 가능
- ② 개인의 과거 구매성향 및 패턴을 기반으로 한 추천시스템으로서 구매 만족도를 높일 수 있음
- ③ 초개인화 추천 시스템 구현

학술적 시사점

- ① 기존의 연구들은 대다수가 사용자 후기를 기반으로 하였으나, 본 연구는 구매이력을 기반으로 한 추천 시스템을 구축하였다는 점에서 의의를 지님

실무적 시사점

- ① 초개인화 추천 통한 개인의 소비 만족도 향상
- ② 초개인화 추천 통한 고객 증가
- ③ 수익성 제고를 위한 전략적 시스템으로써의 가치를 지님



향후 활용 방안

초개인화 추천 시스템을 통한 마케팅 전략

- 재로그인 시기, 재구매 등 구매 패턴을 기반으로 초개인화 마케팅 전략 수립 가능
- 개인 맞춤형 쿠폰 제공

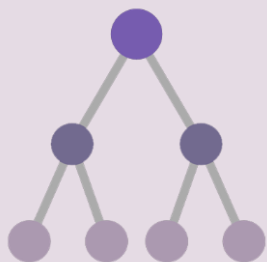
추천 시스템을 활용한 서비스 기획 방향성 제시

- 장바구니에 상품을 담았을 때, 구매 직후, 로그인 시점에서 자동 추천 기능 제공
- 추후 챗봇 구현을 통해 개인 정보 기입 시 추천 상품을 제안할 수 있는 기능을 추가할 수도 있을 것

CRM에 필요한 기초 지표로 활용 가능

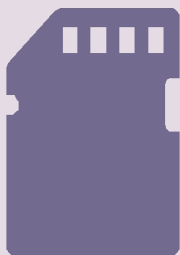
- 고객의 잠재적 Needs를 파악하여 전략적 제안을 할 수 있도록 하는데 참고 자료로 사용될 수 있음

데이터(Data)가 가진 한계점

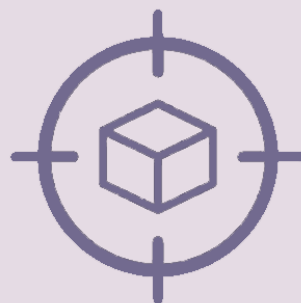


상품 분류의 어려움

검색이 실제 구매로 이어진
상품과 검색에서 그친 상품에
대한 명확한 분류가 어려움



상품명에 따른
매트릭스 구현 시
메모리 부족 문제 발생
소분류 추천으로 변경



메모리 부족 문제

분석 대상 감소로 인해
정확도 낮게 측정



만족도 파악의 어려움

데이터 자체에 평점이
존재하지 않아 상품에 대한
소비자의 만족도 파악 어려움

모델(Modeling)의 한계점

텐서플로우 모델 구현 시 낮은 Accuracy

- ✓ 너무 많은 양의 label -> 모든 label을 정확하게 맞추는 것이 어려움
- ✓ 추천시스템은 정확도로 판단하는 모델이 아니므로 크게 고려할 사항은 아니나, 설명력을 높이기 위해 Accuracy를 향상시켜야 할 필요성이 있음



참고 문헌

참고 문헌

- Wide & Deep Learning for Recommend Systems
(Google Inc., 2016.6.24.)
- Word / Document embedding : Word2Vec / Doc2Vec
(<https://lovit.github.io/>, 2018.3.26.)
- 파이썬을 이용한 머신러닝, 딥러닝 실전 개발 입문
(윤인성, 위키독스)
- 생활코딩
(<https://opentutorials.org/>)
- 출처 : 롯데멤버스, L.pay|L.POINT, 제6회 L.POINT Big Data Competition



RECOMMEND

THANK YOU