



국민대학교
전자정보통신대학
컴퓨터공학부

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	Flex Ads
팀 명	둥글둥글
문서 제목	중간보고서

Version	1.9
Date	2019-APR-17

팀원	민 지수 (조장)
	유 지원
	윤 지영
	이 성재
	황 수진
지도교수	이 경용 교수

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17


CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인 I 수강 학생 중 프로젝트 "Flex Ads"를 수행하는 팀 "둥글둥글"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "둥글둥글"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


Filename	중간보고서-Flex Ads.doc
원안작성자	민지수, 유지원, 윤지영, 이성재, 황수진
수정작업자	민지수, 유지원, 윤지영, 이성재, 황수진

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2019-04-05	황수진	1.0	최초 작성	1, 2.1 작성
2019-04-08	민지수	1.1	최초 작성	2.2, 3.1 작성
2019-04-08	윤지영	1.2	최초작성	향후 추진 계획 작성
2019-04-09	이성재	1.3	내용 추가	2.1, 2.2 추가 작성
2019-04-10	유지원	1.4	내용 추가	2.2 추천알고리즘 구현내용 작성
2019-04-11	황수진	1.5	내용 추가	1.5 내용 추가 2.1 내용 추가
2019-04-11	민지수	1.6	내용 수정	2.2 수행내용 사진 추가, 3, 4 내용 수정
2019-04-13	이성재	1.7	내용 수정	2.2 수행내용 사진 추가
2019-04-15	윤지영	1.8	검토	전체 내용 검토
2019-04-17	황수진	1.9	작성 완료	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

목 차

1	프로젝트 목표	4
2	수행 내용 및 중간결과	5
2.1	계획서 상의 연구내용	5
2.2	수행내용	7
3	수정된 연구내용 및 추진 방향	23
3.1	수정사항	23
4	향후 추진계획	25
4.1	향후 계획의 세부 내용	25
5	고충 및 건의사항	26

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

1 프로젝트 목표

소비자가 상품을 구매하는 데에 있어서, 광고는 중요한 요소이다. 광고는 소비자에게 상품을 인지시키고 구매로 이어질 수 있도록 유도하여 상품 판매를 증가시키는 역할을 한다. 그렇기 때문에 상품 생산자와 판매자는 소비자의 이목을 끌 수 있는 창의적인 광고 제작에 힘쓴다.


현재 온라인 마켓인 쿠팡, 11 번가, 아마존에서는 소비자가 클릭하거나 구매한 상품을 기반으로 상품 추천 광고 시스템을 제공하고 있다. 이는 온라인 마켓의 모든 방문자를 대상으로 하는 광고가 아니라 소비자 개개인의 구매 성향을 파악하여 광고를 제공하는 방식이다. 따라서 평범한 무작위의 광고보다 소비자가 흥미를 가지게 되고 구매까지 이어질 수 있는 가능성이 높기 때문에 뛰어난 광고 효과를 보인다.

하지만 현재 오프라인 마켓인 홈플러스와 이마트에서의 광고는 전단지를 이용한 고전적인 방식이 이어지고 있다. 이는 소비자의 눈길을 끌지 못하여 광고 효과가 낮다. 이를 보완하기 위해 광고 영상을 디스플레이에 몇 개의 특정 영상을 반복하여 재생하는 광고를 도입하였다. 하지만 이 또한 소비자와 특별한 상호작용을 하지 않기 때문에 소비자의 흥미를 끌기는 쉽지 않다.

따라서 본 프로젝트는 온라인 마켓의 개인 맞춤형 광고 시스템을 회원제 오프라인 마켓에 적용해 소비자인 회원이 마켓에 방문하였을 때 구매 기록을 기반으로 회원이 눈길을 사로잡을 만한 제품 광고를 제공하는 것을 목표로 한다.

세부 목표

- Nvidia Jetson TX1 / TX2 를 이용하여 실시간으로 얼굴을 인식한다.
- 인식된 얼굴의 이미지만을 Crop 하여 AWS S3 의 Bucket 에 저장한다.
- AWS Rekognition 에 의해 해당 얼굴에 대한 회원 ID 정보를 반환한다.
- 회원의 구매 내역을 AWS RDS 에 저장하여 추천 알고리즘은 지속적으로 작동하게 한다.
- 추천 알고리즘의 결과를 AWS DynamoDB 에 저장한다.
- 반환된 회원 정보는 AWS DynamoDB 와 연동되어 적절한 광고 결과를 낸다.
- 광고 결과는 AWS API Gateway 를 통해 디스플레이에 송출한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

2 수행 내용 및 중간결과

2.1 계획서 상의 연구내용

연구 내용은 크게 얼굴 인식 기능과 구매내역 기반의 광고 추천 기능, 두 가지로 나누어 작성되었다. 프로젝트 수행 계획서의 해당 연구 내용은 다음과 같다.


얼굴 인식 기능의 경우, 마켓에 등록된 회원이 입장하여 카메라에 포착되는 순간, NVIDIA Jetson TX1 / TX2 모듈에서 작동하는 딥러닝 모델에 의해 얼굴이 인식된다. 인식된 회원의 얼굴 이미지는 사각형 형태로 Crop 되어 AWS 의 S3 에 전송된다. 얼굴을 인식하여 Cropping 하는 과정에서 딥러닝 모델이 사용되는데, 해당 모델은 Fddb 의 얼굴 데이터셋을 24,000 회 학습한 것이다. OpenCV 의 Haar Cascade 와 같은 알고리즘을 이용하여 딥러닝 이외의 방법으로 얼굴 인식 시스템을 구현할 방법을 고려해 보았다. 하지만 Jetson 에서 TensorRT 를 이용해 딥러닝 모델을 Cuda 기반으로 빠르고 정확하게 작동시킬 수 있도록 셋팅 되어있기 때문에 효율성 측면에서 적절하지 않다고 판단하였고, FaceNet-120 모델을 사용하는 것으로 결정하였다.

AWS S3 의 특정 Bucket 에 얼굴 이미지가 저장되는 순간, AWS Lambda Function 과 연동된 트리거에 의해 이미지를 분석하기 위한 AWS Rekognition 으로 해당 이미지가 전달된다. 전달된 이미지는 이미 학습된 Rekognition 모델에 의해 분석되며, 얼굴에 맞는 회원 정보를 Lambda Function 으로 반환된다. 회원 정보는 회원 ID 를 포함하여 다양한 정보가 존재하지만, 네트워크 오버헤드를 최소화하기 위해 회원 ID 만을 전송하도록 제한한다.

얼굴 인식 기능과 함께 구현될 구매내역 기반의 광고 추천 기능을 살펴보면 다음과 같다.

고객의 구매 내역 전체를 AWS RDS 에 저장한 상태에서, AWS EC2 서버를 이용해 각 회원이 가장 필요로 하는 상품 정보를 분석한다. 그리고 이를 기반으로 고객에게 보여줄 광고 정보를 DynamoDB 에 저장한다. AWS Lambda 의 호출에 따라 DynamoDB 로부터 특정 회원에 해당하는 광고 ID 가 반환되며, 해당 광고는 AWS S3 로부터 불러 Web 에 나타난다.

우리는 광고 추천 알고리즘의 구현을 위해 Instacart 에서 제공하는 약 300 만 개의 구매 내역 데이터를 사용한다. 이 데이터를 바탕으로 우리는 고객이 실제로 구매할 것 같은 제품 즉, 관심을 가질 제품을 추천하게 된다. 맞춤형 광고 추천을 위한 알고리즘으로 Collaborative Filtering 과 boosting 을 사용한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

그 외의 적용 가능한 알고리즘은 지속적으로 탐색한다. Collaborative Filtering 과 boosting 을 통해 각각 유사 고객의 정보와 구매 여부를 판단하여 추천할 광고를 결정한다.

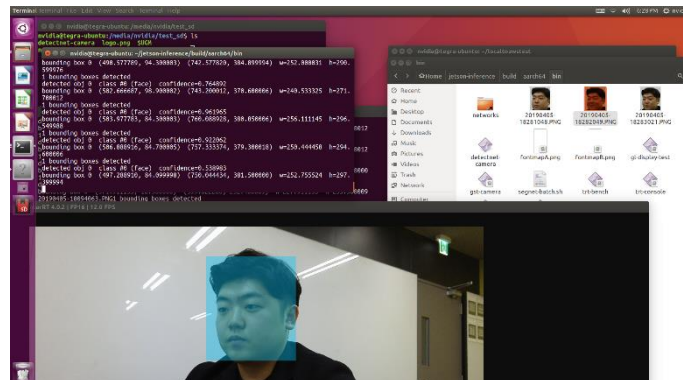
광고 추천 알고리즘의 결과로 반환되는 회원 ID 는 광고 송출을 위한 AWS Lambda Function 으로 전송된다. 해당 ID 는 고객 맞춤형 광고 추천 개발 과정에서 맞춤형 광고가 저장된 AWS DynamoDB 를 통해서 송출에 필요한 광고 정보를 찾아내는 Key 가 된다. 송출하고자 하는 광고의 이미지 데이터는 AWS S3 저장소에 저장되어 있으며, Lambda Function 에 의해 AWS API Gateway 를 통해 송출된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서	
	프로젝트 명	Flex Ads
	팀 명	둥글둥글
	Confidential Restricted	Version 1.9
		2019-APR-17

2.2 수행내용

- 얼굴 인식 기능을 구현하기 위한 수행 내용


Jetson TX1, TX2 모듈에 Jetpack 패키지를 이용하여 TensorRT 를 설치하고, 이를 기반으로 구현된 jetson-inference 의 코드를 활용하여 얼굴 Detect 기능을 구현하였다. AWS SDK 를 직접 해당 코드에 구현하는 데 문제가 있어, Detect 된 얼굴 이미지를 Local 에 저장하고, 해당 이벤트에 대해 인식하여 AWS S3 로 전송하는 Python 코드를 구현하였다. 해당 코드에는 작성되어 있지 않은 Detect 된 얼굴의 Bounding Box 에 따라, Image Cropping 과 Save 작업을 loadImage 코드에 함수 형태로 구현하였으며, detectnet-camera 코드에서 호출하여 사용하였다. Image File Save 과정에서 Disk Write 가 발생하여 Frame Drop 을 확인하였으며, 이에 대한 해결방안으로 SD Card 에서의 프로그램 실행을 구현하였다.



[그림 1] 실시간 얼굴 인식 및 이미지의 Local File System 저장 화면



[그림 2] 실시간 다중 얼굴 인식

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17


- Face Cropping 범위의 변경 및 크기 지정

Jetson 에서 FaceNet-120 을 작동하여 얼굴을 인식한 후 얼굴 영역만을 Bounding Box 처리한다. 인식된 얼굴 이미지를 Rekognition 으로 보내 누구인지 판별하는 작업이 진행되어야 한다. Rekognition 에서는 '적절한 얼굴 이미지 input 은 가려진 부분이 없고 화면을 꽉 채우지 않은 상태의 얼굴이어야 하고, 이미지에 사람의 머리와 어깨가 포함되어야 한다. 또한 얼굴 경계 상자로 자른 상태는 아니어야 한다.'는 제약조건이 존재한다. 따라서 인식된 얼굴을 좌측, 우측은 width * 0.6 만큼, 상단은 height * 0.3, 하단을 height * 0.6 만큼 확장하여 Cropping 하기로 하였다.

회원이 특정 구간을 지나갈 때 적절하게 얼굴을 인식할 수 있도록 Cropping 하기 위한 얼굴 영역의 픽셀 사이즈에 제약 조건을 두었다. 해당 조건은 두 가지를 고려하여 결정된 것으로, 하나는 Rekognition 시스템의 입력 이미지에서 얼굴의 최소 크기가 50x50 픽셀 이상이어야 한다는 점이며, 다른 하나는 너무 큰 얼굴 이미지가 저장될 경우 네트워크나 File I/O 의 오버헤드가 발생하여 인식 속도가 느려진다는 점이다. 이러한 점을 고려하여 얼굴이 인식된 영역의 크기가 120px 이상, 200px 이하일 경우에만 이미지를 저장하도록 구현하였다.

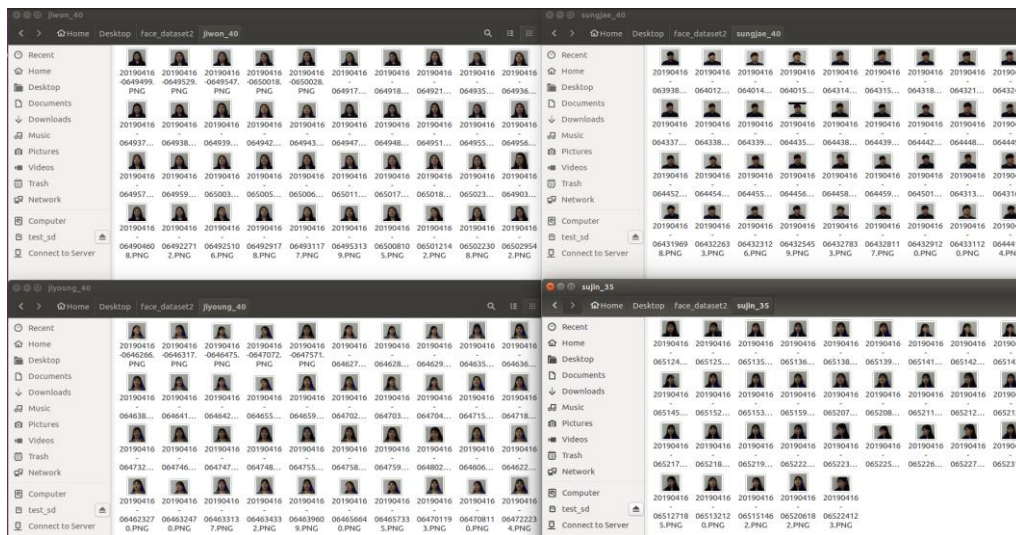
- Image Save to Local 주기의 지정

얼굴이 Detect 된 이미지를 로컬에 저장한 후 AWS S3 에 업로드 한다. 로컬에 저장할 시에 Detect 될 때마다 모두 저장된다면 이는 한정적인 메모리 자원을 불필요하게 많이 사용하는 것이다. 그렇기 때문에 이미지를 저장하는 적절한 주기가 필요하다. 우리는 이 주기를 1 초로 지정하였다.

	중간보고서		
	국민대학교	프로젝트 명	
	컴퓨터공학부	Flex Ads	
	캡스톤 디자인 I	동글동글	
		Confidential Restricted	Version 1.9
			2019-APR-17

● Face Collection 생성의 필요

AWS Rekognition Face Collection 을 이용한 얼굴 이미지 사전 학습이 필요하며, 이를 수행하기 위해 학습 얼굴 이미지 데이터셋을 Jetson 을 이용하여 생성해야 한다. 정면 이미지, 얼굴을 왼쪽으로 돌린 이미지(요 45 도 이내), 얼굴을 오른쪽으로 돌린 이미지(요 45 도 이내), 얼굴을 아래로 기울인 이미지(피치 30 도 이내), 얼굴을 위로 젖힌 이미지(피치 45 도 이내)를 인덱싱하라는 권장 사항이 있다. 이 권장사항에 따라서 테스트를 위해 캡스톤 팀원의 얼굴로 데이터셋을 생성하였다. [그림 3]



[그림 3] 팀원의 얼굴 이미지 데이터 셋


● 추천 알고리즘 구현을 위한 데이터베이스 관련 수행 내용

설계한 대로 두 데이터베이스를 사용했다. 먼저, 고객의 구매기록을 RDS 에 저장하였다. 구매기록에 대한 정보와 제품에 관련된 정보는 서로 다른 테이블에 저장되어 있으며 아래 그림과 같다. RDS 에 저장된 데이터와 그 내용을 살펴보면 다음과 같다. 아래 이미지는 터미널에서 sql 을 사용하여 출력한 테이블의 레코드이다.

우리는 aisle 과 department 에 대한 정보를 가지고 있다. [그림 4] aisle 은 제품의 소분류를 의미하며 department 는 제품의 대분류를 의미한다. 우리가 가진 구매기록에는 총 134 개의 소분류와 21 개의 대분류가 존재한다. 예를 들어 'chocolate sandwich cookies'는 소분류 'cookies cakes'에 속하고, 대분류 'snacks'에 속한다.

aisle_id	aisle	department_id	department
1	prepared soups salads	1	frozen
2	specialty cheeses	2	other
3	energy granola bars	3	bakery
4	instant foods	4	produce
5	marinades meat preparation	5	alcohol
6	other	6	international
7	packaged meat	7	beverages
8	bakery desserts	8	pets
9	pasta sauce	9	dry goods pasta
10	kitchen supplies	10	bulk

[그림 4] aisle 과 department

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

제품은 총 49688 개 존재하며, DynamoDB 에 아래와 같이 테이블로 제품명과 제품이 속한 분류와 함께 저장되어있다. [그림 5]

product_id	product_name	aisle_id	department_id
1	Chocolate Sandwich Cookies	61	19
2	All-Seasons Salt	104	13
3	Robust Golden Unsweetened Oolong	94	7
4	Smart Ones Classic Favorites Min	38	1
5	Green Chile Anytime Sauce	5	13
6	Dry Nose Oil	11	11
7	Pure Coconut Water With Orange	98	7
8	Cut Russet Potatoes Steam N' Mas	116	1
9	Light Strawberry Blueberry Yogur	120	16
10	Sparkling Orange Juice & Prickly	115	7

[그림 5] product 정보

약 342 만개의 주문기록을 가지고 있다. 주문기록에는 몇 가지 정보가 존재한다. 어떤 고객이 주문을 하였는지 (user_id), 해당 고객이 몇 번째 구매를 한 것인지 (order_number), 무슨 요일 (order_dow), 몇 시 (order_hour_of_day)에 구매를 했는지, 이전 구매로부터 며칠 이후에 구매한 것인지 (days_since_prior_order)에 대한 정보가 있다. [그림 6]

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
1	112108	train	4	4	10	9
2	202279	prior	3	5	9	8
3	205970	prior	16	5	17	12
4	178520	prior	36	1	9	7
5	156122	prior	42	6	16	9
6	22352	prior	4	1	12	30
7	142903	prior	11	2	14	30
8	3107	prior	5	4	6	17
9	139016	prior	14	0	19	5
10	135442	prior	4	6	8	8

[그림 6] 주문 기록

또한 각각의 구매에 어떤 제품이 구매되었는지 또한 알 수 있다. 아래 데이터는 구매에 대해서 어떤 제품들이 구매되었는지를 나타낸다. reordered 는 이전에 이 제품을 고객이 구매했는지 알려준다. [그림 7]


order_id	product_id	add_to_cart_order	reordered
2	33120	1	1
2	28985	2	1
2	9327	3	0
2	45918	4	1
2	30035	5	0
2	17794	6	1
2	40141	7	1
2	1819	8	1
2	43668	9	0
3	33754	1	1
3	24838	2	1
3	17704	3	1
3	21903	4	1

[그림 7] 주문기록과 해당 주문에 구매한 제품

한편, 고객에게 어떤 광고를 추천할지에 대한 정보는 DynamoDB 에 저장되어있다. 현재 추천 알고리즘을 통해 고객에게 추천될 광고의 정보를 DynamoDB 에 저장하지 않은 상태이기 때문에 데이터베이스에 고객에게 추천할 광고 데이터가 존재하지 않는다. 따라서 임의로 고객 ID(user_id)와 추천할 광고에 나타날 제품 정보인 제품 id 와(product_id) 제품명(product_name)을 기록해 테스트 중이다. [그림 8]

<input type="checkbox"/>	user_id ⓘ ▲	update_ver ▼	product_id ▼	product_name▼
<input type="checkbox"/>	23	1	25	apple
<input type="checkbox"/>	124	1	24	banana

[그림 8] DynamoDB 에 저장된 고객 ID 와 광고할 제품 정보

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

● 추천 알고리즘 구현 내용

먼저, 가지고 있는 데이터의 분석을 진행하였다. 꾸준히 구매하던 제품을 사지 않다가 다른 대체품을 사는 행위를 포착하는 등 데이터의 특징을 살펴보았다. 그리고 추천알고리즘 구현을 위해 boosting 에 관한 학습을 했다. 이 알고리즘이 데이터 형태에 맞는 입출력에 맞도록 세부적인 알고리즘 설계 과정을 고안하였다. 현재는 사용할 feature 를 계속 생성하고 있다. 요일별 또는 상품별 재구매율, 특정 단어(fresh, organic 등)가 들어간 제품의 재구매율 등 7~8 여개의 feature 를 만들어 알고리즘에 적용하였다. [그림 9]

현재까지 만든 feature 의 내용은 다음과 같다.(1~6)

1. 고객별로 자주 구매하는 제품
2. 장바구니에 가장 먼저 담긴 제품
3. 고객이 최초로 해당 제품을 구매한 시점에서, 현재까지 그 제품을 구매한 비율
4. 제품이 많이 팔리는 요일, 시간, 대분류, 소분류
5. 많이 팔린 제품
6. 특정 단어가 들어간 제품

user_id	organic_prob	user_hour_prob	user_dow_prob	user_aisle_prob
131433	0.022727	0.006803	0.058824	0.003311
122809	0.037037	0.033333	0.032258	0.003333
44202	0.032258	0.027778	0.033333	0.083333
38813	0.026316	0.012346	0.031250	0.005988
55598	0.020408	0.047619	0.005814	0.010526
127103	0.058824	0.024390	0.024390	0.014925

[그림 9] 기존 데이터에서 추천 알고리즘에 사용할 feature 가 만들어진 모습

feature 를 생성함과 동시에 boosting model 에 적용하여 evaluation 하는 것을 진행 중이다.

train set 로부터 feature 을 추출하였으며[그림 10], 각 데이터의 실제 값을 가지고 있다. 여기서 실제값은 재구매 정보(재구매가 된 제품인지 아닌지 1 또는 0 으로 표기)[그림 11] 를 말한다. 만들어진 모델을 통해 test set 의 구매기록에서 어떤 제품이 재구매 될 것인지 예측하게 된다.

add_to_cart_order	order_number	order_dow	order_hour_of_day	user_dep_prob	user_aisle_prob	user_dow_prob
8	7	4	12	0.007752	0.022727	0.022222
5	3	3	10	0.015625	0.007407	0.090909
1	7	0	17	0.004619	0.100000	0.026316
6	5	2	13	0.004619	0.002375	0.026316
9	3	2	11	0.011364	0.035714	0.250000
3	6	0	11	0.003610	0.011494	0.250000
4	7	1	22	0.015385	0.019231	0.052632
1	16	0	16	0.010309	0.023256	0.166667

[그림 10] train set 에서 사용한 feature(training data)

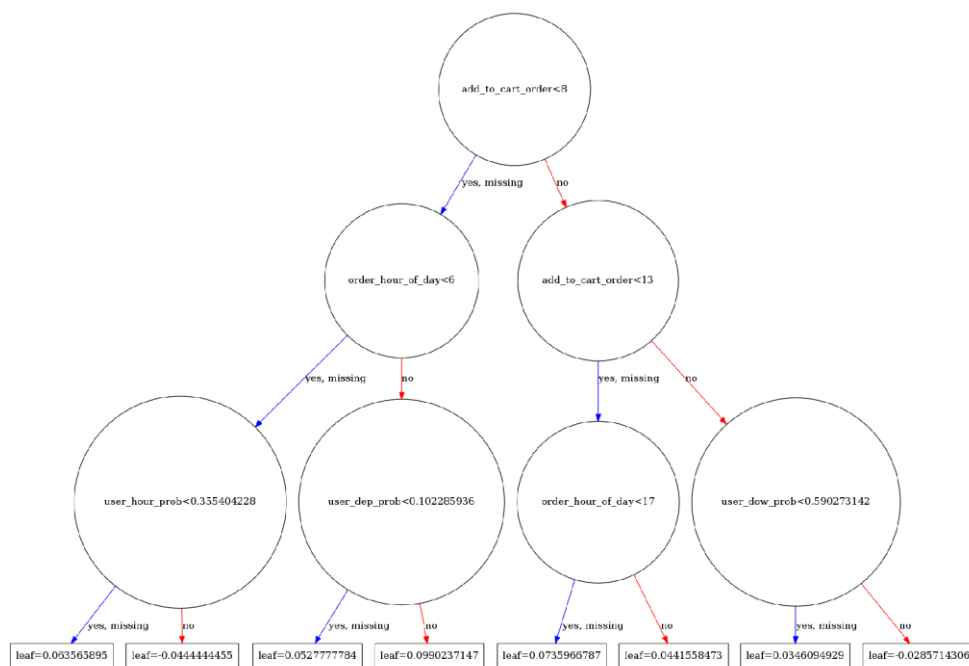
```
train_y
array([1, 0, 0, ..., 0, 0, 0])
```

[그림 11] train set 의 실제값


 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

boosting tree 는 [그림 12]와 같으며 만드는 과정에서 필요한 파라미터는 아래에 기술하였다. (1~8)
boosting tree 는 복잡하게 생성되어 간단하게 표현한 것이다.

1. eval_metric : 모델의 평가 함수를 정의한다.
2. eta : learning rate, 단계별 가중치를 줄여서 강한 모델을 제작하기 위해 사용한다.
3. subsample : 각 트리마다 관측 데이터 샘플링 비율이다.
4. min_child_weight : overfitting 을 컨트롤, child 의 최소 가중치의 합이다.
5. colsampe_bytree : feature 샘플링 비율이다.
6. max_depth : 트리의 최대 깊이를 정의한다.
7. silent : 동작 메시지를 출력하려면 0, 출력을 원하지 않으면 1 로 설정한다.
8. seed : 난수 생성 seed 이다.



[그림 12] boosting tree

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17


model 에 test set 으로부터 train set 과 마찬가지로 만들어낸 feature[그림 13]를 넣으면, 제품이 해당 구매에 실제로 구매될 것인지 예측 가능하다. 예측값은 xgboost 로부터 생성된 boosting tree 의 결과로 만들어지는데, leaf node 의 값이 일정 기준(threshold)을 넘게 되면 구매가 되는 것이고 그렇지 않다면 구매되지 않는 것이다. 우리는 구매될 것이라고 예측된 제품을 table 형태로 만들었다[그림 14]. 이 중 랜덤으로 1 개의 광고를 추천한다.

	add_to_cart_order	order_number	order_dow	order_hour_of_day	dep_prob	aisle_prob	dow_prob	hour_prob	organic_prob
0	2	1	1	13	0.406667	0.173481	0.181818	0.114796	0.602740
1	7	2	0	9	0.359589	0.080221	0.342205	0.172515	0.366366
2	1	1	1	13	0.320611	0.118263	0.182432	0.159817	0.499014
3	3	4	5	13	0.281734	0.196429	0.172414	0.234568	0.658103
4	3	2	0	9	0.509915	0.107692	0.479365	0.080292	0.847240

[그림 13] test set 에서 사용할 feature(test data)

	user_id	product_id
0	3	1819 21903 17668 14992 47766 16797 9387 43961 ...
1	4	25146 21573 25623
2	11	17794 13176 41290 44632 27451 4799 42736 22950...
3	12	13176 14992 31506 42736 11520 44422 22959
4	15	196 14715 43721 37710 27839 11266 12427 10441 ...

[그림 14] 유저에게 추천할 상품 목록

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

- 광고 송출을 위한 수행 내용

추천 알고리즘으로부터 나온 결과를 DynamoDB 에 저장하지 않아, 고객에게 송출할 광고를 임의로 저장하여 테스트하였다. DynamoDB 에는 고객에게 송출할 광고 정보(product_id)가 [그림 15]과같이 {user_id: value, product_id: value} 형태로 임시로 저장되어 있다.

user_id ⓘ	update_ver	product_id	product_name
23	1	25	apple
124	1	24	banana

[그림 15] DynamoDB 에 저장된 데이터


AWS API Gateway 는 얼굴인식 결과로부터 받은 user_id 를 입력으로 받아 Lambda 를 trigger 한다. 이 때, 얼굴인식이 아직 구현되지 않았으므로 임의의 user_id 를 [그림 16]처럼 json 형식으로 입력으로 받았다. 그리고 Lambda 는 DynamoDB 에 저장된 고객-광고 데이터를 탐색한다. 그리고 Lambda 는 이 고객에 해당하는 데이터를 DB 에서 읽어온다[그림 17].

```
1 {
2   "user_id": 23
3 }
```

[그림 16] Lambda event

▼ Execution results	Status: Succeeded
Response:	
{	
"update_ver": 1,	
"user_id": 23,	
"product_name": "apple",	
"product_id": 25	
}	

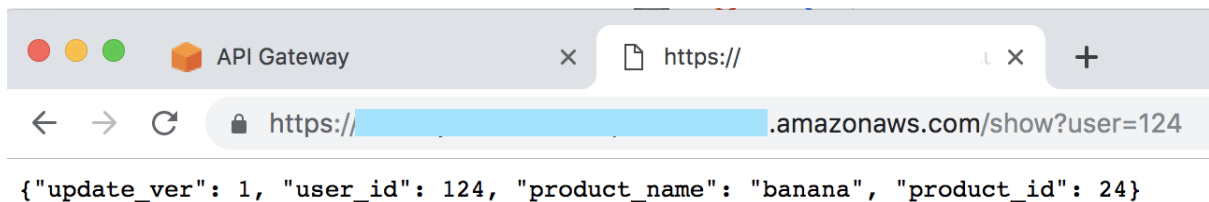
[그림 17] Lambda 가 DynamoDB 에서 읽어 들인 값

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

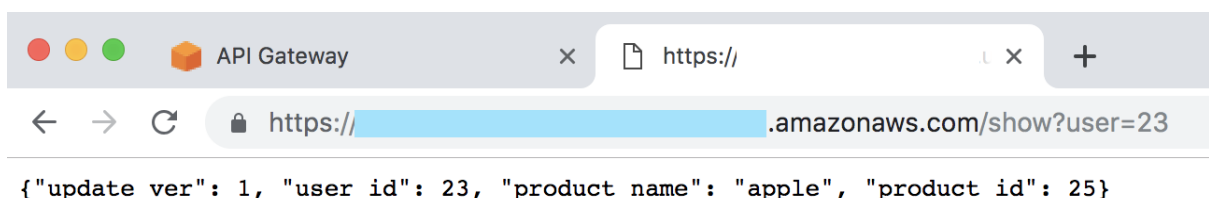
그리고 읽어온 정보는 AWS API Gateway 에서 제공하는 URL 을 통해 출력된다. 이를 수행하기 위하여 아래 세 가지 작업을 진행하였다.

1. [그림 18]와 같이 작동하는 Lambda Function 을 GET method 로 추가한다.
2. GET method 의 Integration Request 에서 Mapping Templates 를 추가한다. 이는 Lambda 의 입력을 json 형식으로 만들어 주기 위함이다. 그리고 입력을 URL 을 통해 주기 위하여 Method Request 에서, user 라는 parameter 를 인식하도록 URL Query String Parameters 에 추가했다. (이때, user 는 user_id 를 의미하며 다른 변수들과 혼동하지 않도록 user 이라고 지정하였다.)
3. user_id 를 URL 을 통해 입력으로 받고 이를 호출하기 위하여 API 를 Deploy 하였다. Deploy 를 통해 생성되는 Stage 에서 Invoke URL 을 제공한다. 우리는 이 URL 을 통해 광고를 송출하는 것이다.

[그림 18]와 [그림 19]는 광고가 송출될 URL 을 통해 송출될 광고의 정보를 출력한 모습이다. 우리는 GET method 를 이용하여 API 를 호출하기 때문에 URL 마지막에 ?user=[인식된 고객의 id]를 추가하여 Web 에 URL 을 띄운다. [그림 D]는 user_id=124 에 해당하는 고객에게 추천할 광고 정보를 송출한 것이며 [그림 19]는 user=23 에 해당하는 고객에게 추천할 광고 정보를 나타낸 것이다.



[그림 18]




[그림 19]

Flex Ads 팀의 계획서상 진도는 다음과 같이 명시되어 있다.

항목	세부내용	12 월	1 월	2 월	3 월	4 월	5 월	비고
요구사항 분석	주제 결정 및 시나리오 작성							
	시나리오에 따른 역할 분담							
	SRS 작성							
	학습한 결과에 따른 시나리오 수정							
관련분야 연구	얼굴 인식 알고리즘 학습 및 코드 분석							(1)
	추천 알고리즘(머신러닝) 학습 - User based algorithm - Product based algorithm							(2)
	머신러닝과 딥러닝 학습							(3)
	AWS 클라우드 서비스 사용 공부 - S3, EC2, RDS, DynamoDB, Lambda, Rekognition							(4)
	추천 알고리즘 설계 및 개발 환경 구축							(5)
설계	NVIDIA Jetson TX1 / TX2 보드와 AWS 를 이용한 얼굴인식 시스템 설계							(6)
	Serverless 를 이용한 얼굴 인식 시스템과 추천 시스템 연동							(7)
	얼굴 인식 시스템 구현							(8)
구현	NVIDIA Jetson TX1 / TX2 보드, Rekognition, S3 연동 구현							(9)
	추천 알고리즘 구현							(10)
	얼굴 인식과 추천 시스템 연동 작업							(11)
	데모를 위한 전체 프로세스 시각화							
	추천 알고리즘 디버깅 및 테스트							
테스트	전체 프로세스 및 시스템 테스트							

주제 결정 및 시나리오 작성이 완료되어 전체적인 요구사항 분석과 전반적인 관련 분야에 대한 연구를 마친 상태이다. 이하 표의 비고란 붙임 번호에 따라 계획서상의 일정과 현재 수행하고 있는 진도 현황을 비교한다. 4 월까지 완료한 수행 목록과 계획서의 진도를 비교한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

(1) 얼굴 인식 알고리즘 학습 및 코드 분석

Jetson TX1, TX2 의 보드에서 얼굴 인식은 C++ 기반의 CUDA 코드로 이루어져 있다. 해당 코드는 Jetson Tutorial 코드인 Jetson-Inference 에 작성되어 있으며, 이를 기반으로 얼굴 인식 알고리즘과 코드를 분석할 수 있었다. 얼굴 인식은 Jetson 의 TensorRT 기반의 DetectNet 코드로 구현이 되어 있었으며 FaceNet-120 모델이 사용되고, GPU 병렬 처리를 통해 빠른 속도의 인식 및 시각화가 가능하다는 점을 확인할 수 있었다.

(2) 추천 알고리즘(머신러닝) 학습

추천 알고리즘으로 boosting 과 collaborative filtering 을 사용하는 것을 계획하고 이 두 알고리즘을 학습하였다. 우리가 사용할 boosting 알고리즘은 Gradient Boosting 을 기반으로 구현된 것이다. 이것은 고객이 제품을 구매할지에 대한 여부를 예측하게 된다. 또한 Collaborative filtering 은 user 로부터 얻는 관심사에 따라 user 의 기호도를 예측한다. 추천 시스템에서 주로 사용되는데 해당 item 에 대해서 user 의 기호를 예측하게 된다.

(3) 머신러닝과 딥러닝 학습

HKUST 김성훈 교수의 머신러닝과 딥러닝 BASIC 강의를 수강하였다. 이를 통해 머신러닝에서의 Hypothesis, Cost 등 주요 용어의 개념과 쓰임에 대해 배웠고 얼굴 인식 딥러닝 모델에 대한 기초를 학습하였다. 또한 강의 수강 후 Python 으로 실습을 진행하였다.

(4)AWS 클라우드 서비스 사용 공부

- S3

Amazon Simple Storage Service (S3)는 확장성과 데이터 가용성 및 보안과 성능을 제공하는 객체 스토리지 서비스이다. 사용하기 쉬운 관리 기능을 제공하므로 비즈니스, 조직 및 요구사항에 따라 데이터를 조직화하고 세부적인 액세스 제어를 구성할 수 있다.

- EC2

Amazon Elastic Compute Cloud(EC2)는 안전하고 크기 조절이 가능한 컴퓨팅 파워를 클라우드에서 제공하는 Web 서비스이다. Amazon EC2 의 간단한 Web 서비스 인터페이스를 통해 간편하게 필요한 용량을 얻고 구성할 수 있다. 또한 개발자가 장애에 대한 복원력이 뛰어나고 일반적인 오류 상황에 영향을 받지 않는 애플리케이션을 구축할 수 있도록 도구를 제공한다.



- RDS

Amazon Relational Database Service(RDS)를 사용하여 클라우드에서 관계형 데이터베이스를 간편하게 설정, 운영 및 확장한다. 시간 소모적인 관리 작업을 자동화하면서 크기 조정 가능한 용량을 이용할 수 있다. Amazon RDS 는 여러 데이터베이스 인스턴스 유형(메모리, 성능 또는 I/O 최적화)으로 제공된다. AWS Database Migration Service 를 사용하여 기존 데이터베이스를 Amazon RDS 로 손쉽게 마이그레이션 또는 복제할 수 있다.

- DynamoDB

Amazon DynamoDB 는 어떤 규모에서도 10 밀리초 미만의 성능을 제공하는 키-값 및 문서 데이터베이스이다. DynamoDB 는 완전관리형의 다중 리전, 다중 마스터 데이터베이스로서, 인터넷 규모 애플리케이션을 위한 보안 기능, 백업 및 복원, in memory caching 을 제공한다. DynamoDB 는 하루에 10 조 개 이상의 요청을 처리한다.

- Lambda


AWS Lambda 는 이벤트에 대한 응답으로 코드를 실행하고 자동으로 기본 컴퓨팅 리소스를 관리하는 서버리스 컴퓨팅 서비스이다. AWS Lambda 를 사용하여 사용자 지정 로직을 통해 다른 AWS 서비스를 확장하거나, AWS 규모, 성능 및 보안으로 작동하는 자체 백엔드 서비스를 만들 수 있다. AWS Lambda 는 Amazon API Gateway 를 통한 HTTP 요청, Amazon S3 Bucket 에 있는 객체에 대한 변경 사항, Amazon DynamoDB 의 테이블 업데이트 또는 AWS Step Functions 의 상태 전환과 같은 다양한 이벤트에 대한 응답으로 코드를 자동 실행한다.

- Rekognition

이미지와 비디오 분석을 애플리케이션에 쉽게 추가할 수 있도록 합니다. Rekognition API 에 이미지나 비디오를 제공하면 서비스에서 객체, 사람, 텍스트, 장면 및 활동을 파악할 수 있습니다. 부적절한 콘텐츠를 감지할 수도 있습니다. Amazon Rekognition 도 매우 정확한 얼굴 분석과 얼굴 인식을 제공합니다. 사용자 확인, 카탈로그 작성, 인원 계산 및 공공 안전을 포함하여 다양한 사용 사례에서 얼굴을 탐지, 분석 및 비교할 수 있습니다.

(5) 추천 알고리즘 설계 및 개발 환경 구축

추천 알고리즘 설계는 후에 기재되어 있는 3.1 의 수정사항에 따라 boosting 모델을 이용해 광고 추천 알고리즘을 설계한다. 추천 알고리즘 개발은 AWS EC2 인스턴스 환경에서 이루어지고 있으며 현재 RDS 에 존재하는 구매기록 데이터를 EC2 인스턴스가 가져와 알고리즘을 실행한다. 그리고 알고리즘을 통해 나온 고객에게 추천될 광고 목록은 DynamoDB 에 저장된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

(6) Jetson TX1 / TX2 보드와 AWS 를 이용한 얼굴인식 시스템 설계

Jetson TX1 / TX2 의 on board camera 를 이용하였고, FaceNet-120 모델을 작동하여 카메라에 포착되는 순간 얼굴이 인식되면서 해당 영역에 Bounding Box 처리된다.

(7) Serverless 를 이용한 얼굴 인식 시스템과 추천 시스템 설계

얼굴 인식 시스템과 추천 시스템은 Lambda 를 통해 Serverless 로 연동된다. Lambda 는 얼굴 인식 결과로부터 나온 고객 ID 를 바탕으로 추천될 광고를 가져오게 되며, 이를 AWS API Gateway 를 통해 Web 으로 송출하게 된다.

(8) 얼굴 인식 시스템 구현

Bounding Box 처리된 부분을 3.1 의 수정사항에 따라 Cropping 하는 영역의 크기를 Rekognition 에서 제대로 동작할 수 있게 Bounding Box 보다 더 넓은 영역을 Cropping 하였다.

(9) Jetson TX1 / TX2 보드, Rekognition, S3 연동 구현

Jetson TX1 / TX2 보드와 AWS S3 연동은 inotify for python 을 이용하여 Local 의 특정 폴더에 이미지가 저장되면 AWS S3 의 지정된 Bucket 에 저장하는 것까지 완료하였다.

(10) 추천 알고리즘 구현

효율적인 추천 알고리즘 구현을 위해 feature 를 생성하는 과정에 있다. 정확도를 높이기 위해서 최대한 많은 feature 들을 만들어 evaluation 을 통해 유효한 feature 들을 분류하여 이용할 예정이다.

(11) 얼굴 인식과 추천 시스템 연동 작업

위의 (7)에 설계한 내용을 바탕으로 연동 작업이 이루어졌다. 현재 Lambda 는 추천할 광고를 DynamoDB 에서 가져와 Web 에 그 내용을 출력하도록 구현되어 있다. Lambda 가 해야 할 일의 약 30% 정도가 완성되었다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

3 수정된 연구내용 및 추진 방향

3.1 수정사항

- Cropped Face Image 의 AWS S3 전송 방법의 변경

AWS SDK 를 C++로 직접 구현하여 AWS S3 로 Cropping 된 사진을 전송하는 방식으로 진행하려 했으나, Img Local Save 와 inotify for python 을 이용하는 시나리오로 변경하였다.

- 추천 알고리즘

우리는 사용자의 과거 구매기록을 바탕으로 사용자의 방문 시점에서 이 사람이 구매할 제품을 예측한다. 초기 광고 추천알고리즘으로 Boosting 과 Collaborative Filtering 를 사용하는 Hybrid Approach 를 제안했다. 모든 유저와 제품에 대해서 구매 예측 여부를 두 알고리즘을 이용하여 판단하는 것으로 계획하였다[식 1].

$$f(u, p) = w_1 \cdot f_1(u, p) + w_2 \cdot f_2(u, p)$$


$f_1(u, p)$: boosting에서 user가 product를 구매할 것인지 예측한 결과

$f_2(u, p)$: Collaborative filtering에서 user가 product를 구매할 것인지 예측한 결과

$f(u, p)$: user이 product를 구매할 것인지 예측한 결과(광고 추천 여부)

[식 1] 초기에 계획한 추천 알고리즘

하지만, 구매 기록은 개별 고객의 취향을 반영한다는 점을 고려하여 고객 간의 유사도를 활용하는 Collaborative Filtering 은 알고리즘에서 사용하지 않는다. 따라서 광고 추천을 위한 모델은 boosting 으로 구현한다.


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

- API 를 이용한 Web 으로의 광고 송출

광고가 송출될 Web 의 URL 은 마련된 상태이다. 고객이 카메라를 통해 인식된 후 Local Computer 를 통해 이 URL 이 띄워져야 실제로 광고를 볼 수 있게 된다. 따라서 이를 수행하기 위해 AWS 의 SNS(Simple Notification Service)를 통하여 인식 정보를 받아내고 Local 과 네트워킹할 수 있도록 작업할 예정이다. 즉, Web 에 실시간으로 광고를 보여주기 위하여 하나의 기능이 추가된 것이다.

- 광고 이미지를 저장할 저장소

광고 이미지를 S3 에 저장하는 것을 계획하였다. 하지만 S3 저장소는 하드디스크 기반이기 때문에 값을 가지고 오는 데 시간이 걸릴 것으로 예상된다. 따라서 Amazon ElastiCache 를 통해 디스크를 왕복해야 하는 것을 없애고, 인 메모리 데이터 스토어를 한다. 데이터를 인 메모리에서 관리함으로써 빠른 작업을 수행할 수 있으며 평균 읽기 속도가 1 밀리초 미만임을 가정하였을 때 실시간으로 광고 송출에 적합하다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	둥글둥글	
	Confidential Restricted	Version 1.9	2019-APR-17

4 향후 추진계획

4.1 향후 계획의 세부 내용

- AWS Rekognition 의 Collection 을 이용한 얼굴 이미지 학습 및 분석

현재 AWS Rekognition 의 Face Collection 기능을 이용하여 Face Index 를 생성하고, 특정 얼굴 이미지가 입력되었을 때 분석 가능한 시스템을 개발 중이다. 이를 위하여 학습에 필요한 얼굴 이미지 데이터를 Jetson 보드를 이용하여 수집한 다음, Face Collection 으로 생성할 예정이다.

- AWS Lambda 를 이용한 Rekognition 시스템 연동


앞에서 분석이 완료된 얼굴 데이터는 AWS Lambda 를 통하여 관련 정보를 처리하게 된다. Lambda 는 분석된 결과를 데이터베이스와 연동하여 적절한 개인 맞춤형 광고 정보를 가져오는 역할을 하게 된다. 이러한 과정에서 Rekognition 에서 반환되는 값을 기반으로 다른 시스템과 연동하는 부분의 구현이 필요하게 되며, 앞의 수행 계획이 완료되는 대로 실행할 예정이다.

- 광고 추천 알고리즘의 성능 향상을 위한 feature 생성

높은 성능의 boosting 모델을 구축하기 위해서, training data 를 이용한 feature 를 가공해야 한다. feature 이 추가될 때마다 모델을 평가하여 최적의 모델을 찾는다. 우리는 고객이 어떤 제품을 구매할 것인지 예측하며 이를 f1-score 를 통해 평가한다.

- 추천할 제품에 해당하는 광고를 Web 을 통해 송출

광고 이미지를 AWS 의 Redis 에 저장하는 작업을 진행한다. 2~3 개의 임의의 광고 이미지를 넣어 테스트할 예정이다. 이 후 광고 이미지의 개수를 늘린다. 이 이미지를 Web 에 추천된 제품 이름과 함께 송출한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	Flex Ads	
	팀 명	동글동글	
	Confidential Restricted	Version 1.9	2019-APR-17

5 고충 및 건의사항

캡스톤 디자인 과목의 프로젝트를 진행하기 위해서 각 팀은 작업을 수행할 공간이 필요합니다. 하지만 자율주행 스튜디오 외에는 적당한 공간이 없고 자율주행 스튜디오 또한 수요가 많아 적절한 공간이 없습니다. 따로 해당 강의를 위해 사용할 수 있는 공간을 활용할 수 있도록 마련해 주셨으면 합니다. 2-3 팀이 모여서 강의실을 대여할 수 있도록 해주시면 감사하겠습니다.

또한 프로젝트에 진행에 있어 필요한 물품 중 과사무실에 구비가 되어있어 대여가 가능한 물품(예를 들면 HDMI, 무선 키보드-마우스 등)의 목록과 대여 가능 여부를 미리 공지해 주셨으면 합니다. 이를 통해 학생들이 구비되어 있는 물품을 개인적으로 구매하거나 Link 사업단을 통해 구매 요청하는 경우가 적어지고 필요한 물품을 바로 대여 후 사용할 수 있기 때문에 시간이 절약되어 더욱 프로젝트에 집중할 수 있을 것이라고 생각합니다.