AI 양재 허브 AI 웹개발 과정 5조

Inssa Food

김홍래 박찬 조성우

이런 경우, 한 번씩 있으시죠?

'영국남자'가 이 자리에 없었다면?

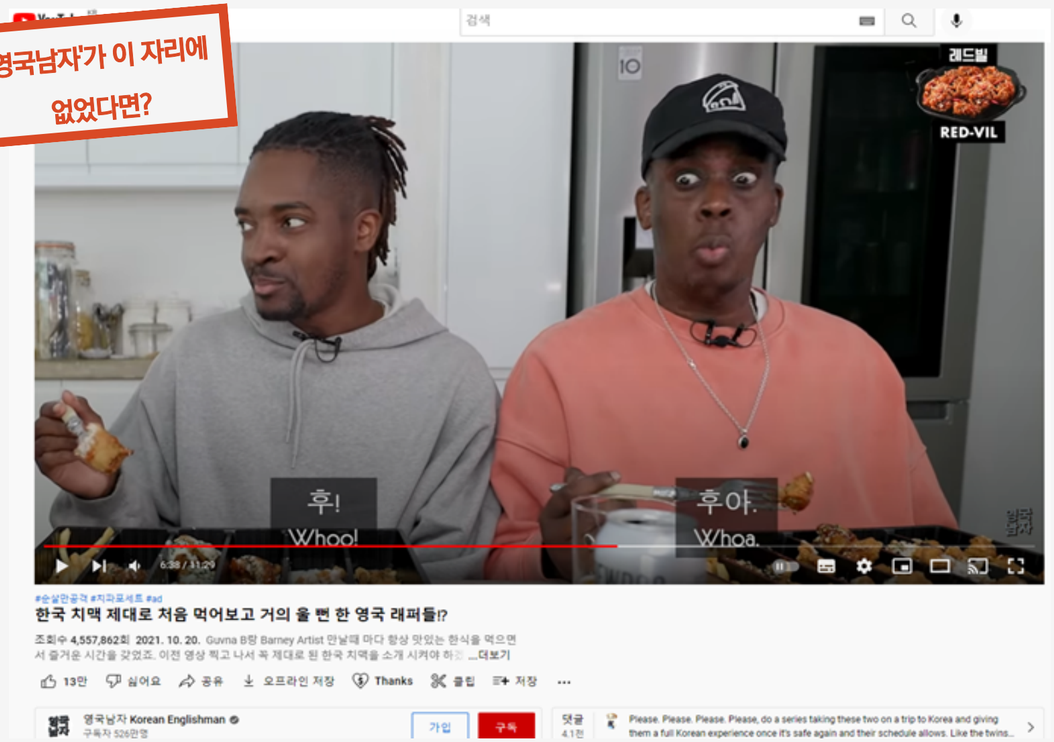


사진 갈무리 : 유튜브 〈 영국남자 〉, '군대 뉴스'

해외여행을 가서 맛있는 음식을 먹었는데,
그게 무슨 음식인지 모르겠다

한국에 돌아와서도 그 맛을 잊지 못하고,
소중한 기억으로 간직하고 싶다

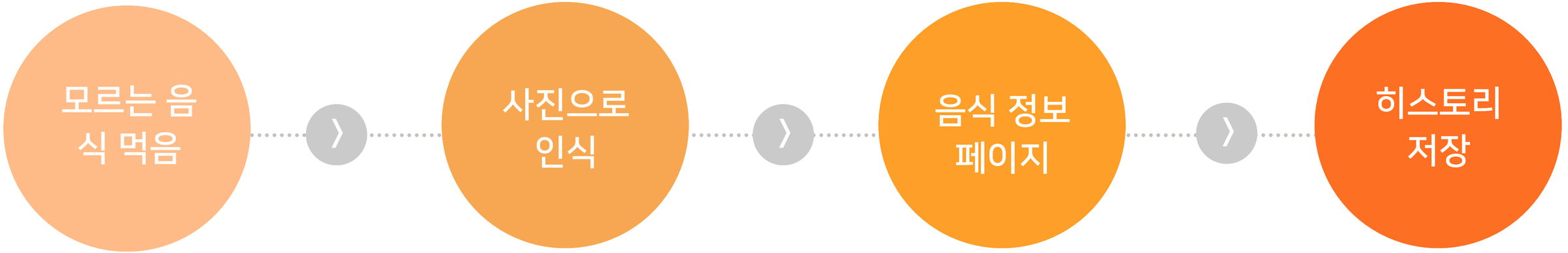
Inssa Food

한국음식, 제대로 알고, 먹고, 저장하자!

Ingredient Insight, 인싸푸드

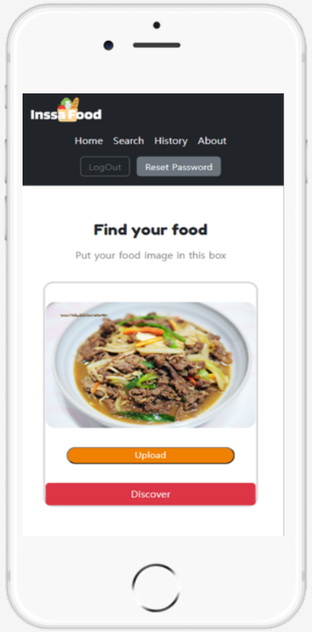
InssaFood

사용자 경험



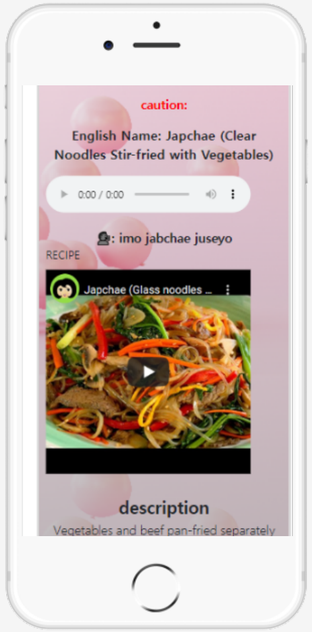
핵심 기능

음식 사진 분류기



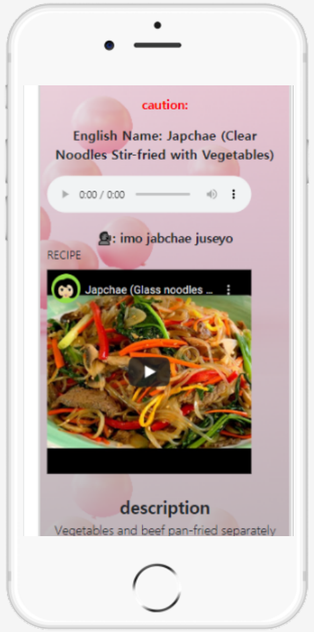
사용자가 사진을 집어넣음 → 어떤 음식인지 분류

음식 정보 제공



이름, 발음, 설명, 레시피, 재료, 맵기 정보 제공

히스토리 저장



먹은 음식 사진과 소감, 그 음식의 정보를 정리

Inssa Food

Demo

웹사이트 시연

InssaFood

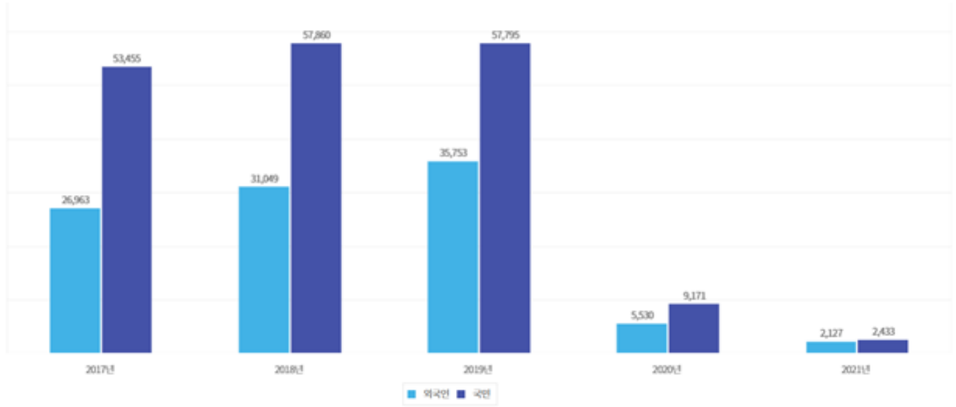
시장성 분석

Inssa Food

TARGET

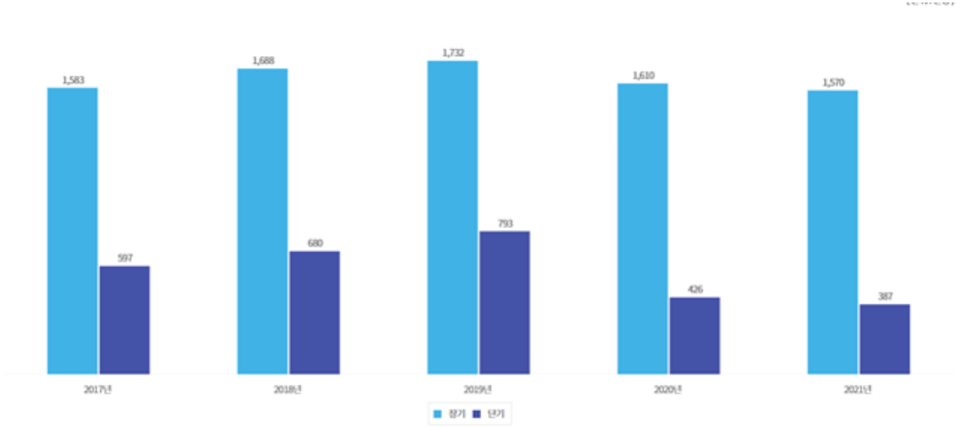
국내 방문 / 체류 외국인

자료 출처 : 법무부 출입국통계 https://www.moj.go.kr/moj/2411/subview.do



(단위:명)

구분	2017	2018	2019	2020	2021
외국인	26,962,672	31,048,752	35,752,704	5,530,350	2,127,176
국민	53,455,030	57,859,670	57,795,389	9,171,481	2,432,517



(단위:명)

구분	2017	2018	2019	2020	2021
장기 체류외국인	1,583,009	1,687,733	1,731,803	1,610,323	1,569,836
단기 체류외국인	597,399	679,874	792,853	425,752	386,945

연도별 출입국자 현황

코로나 19의 영향으로 2020년부터 출입국자가 급감.
그러나 2019년까지의 국내 방문객은 꾸준한 증가 추세.
코로나 19 유행이 끝나면 다시 외국인 방문객이 폭발적으로 증가할 것으로 보임.

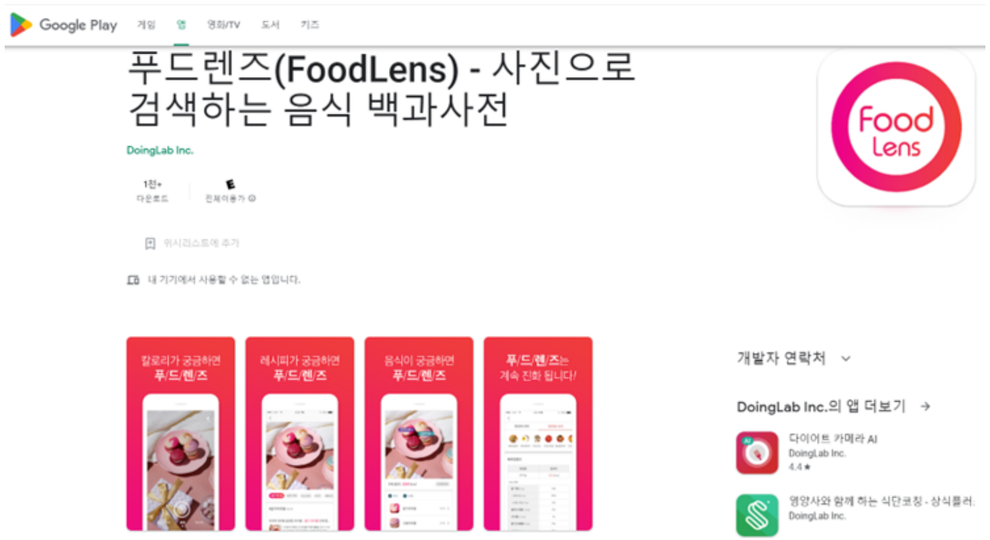
국내 체류 외국인

코로나 19에도 불구하고 국내 체류 외국인 수는 안정적으로 유지되고 있음.
이들 중 많은 이용자를 유입시키면 빠르게 성장하고, 이 집단과 연계된 국내 방문 외국인 고객 확보에도 유리할 것으로 생각됨

유사 서비스

푸드렌즈

Foodlens



유사한 기술, 다른 타겟

인공지능 기술로 음식 이미지 분류 후 영양정보 제공
다이어트에 특화된 기능, 현재는 B2B 기업으로 피벗한 것으로 보임.
기존 푸드렌즈 어플은 플레이스토어 기준 1000회 단위 다운로드 수 기록.

인싸푸드는 웹 기반 → 더 많은 사용자 유입 기대
외국인 대상 음식 분류, 정보 제공, 추억 저장 기능으로 차별화된 타겟층

기술 스택

InssaFood

TECH STACK

FrontEnd

HTML, CSS, Javascript(ECMA 6)

React

Bootstrap, Material UI

BackEnd

서버 : NodeJS (ExpressJS), Flask

DB : MongoDB

모델 : Tensorflow (python)

개발환경

IDE : Visual Studio Code

Tensorflow 모델 학습 : Colab,

Jupyter Notebook

배포환경

Naver Cloud Platform

[MICRO] 1vCPU, 1GB Mem [g1]

협업

Version control : Github

To Do List : Github Project

Messenger : Discord

회의록 : Notion

OS

개발환경 : Windows 10, Windows 11

배포환경 : Linux (Ubuntu 18.04)

Inssa Food

Model - MobileNet

데이터셋 : 〈한국 이미지(음식)〉 , 2018, AI 허브

한국 음식 150종(종별 약 1천 장)의 데이터를 구축한 이미지 데이터 제공

https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=79

이 중 10개의 클래스 활용, 각 클래스 당 1000장의 이미지 학습

```
# x = tf.keras.layers.experimental.preprocessing.Resizing(224, 224)(inputs)
x = data_augmentation(inputs)
x = layers.experimental.preprocessing.Rescaling(1./225)(x)
# x = tf.keras.applications.vgg19.preprocess_input(inputs)
x = base_model(x, training = False)
x = Flatten()(x)                # Fully Connected에 온전하게 학습을 위해 펼쳐준다
# x = Dense(2048, activation='relu')(x)
# x = Dropout(0.5)(x)
x = Dense(64, activation='relu')(x)
x = Dropout(0.5)(x)
outputs = Dense(10, activation = 'softmax')(x)  # Softmax 함수로 10개 분류하는 분류기
model_res = tf.keras.Model(inputs, outputs) # model_res 란 이름의 인풋과 아웃풋이 정해진 모델 생성
```

```
model_res.summary()
```

```
Model: "model_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_14 (InputLayer)       [(None, 224, 224, 3)]     0

 sequential_4 (Sequential)   (None, 224, 224, 3)       0

 rescaling_6 (Rescaling)     (None, 224, 224, 3)       0

 mobilenet_1.00_224 (Functio  (None, 7, 7, 1024)       3228864
 nal)

 flatten_6 (Flatten)         (None, 50176)             0

 dense_13 (Dense)            (None, 64)                3211328

 dropout_7 (Dropout)         (None, 64)                0

 dense_14 (Dense)            (None, 10)                650
```

모델 : MobileNet 기준으로 전이학습

Tensorflow.keras에서 MobileNet을 No Top 옵션으로 가져와,

64개의 노드를 가진 Dense Layer 1층과 Dropout Layer 1층을 컴파일.

Loss Function : sparse-cross-entropy

Optimizer : Adam

Inssa Food

Model - MobileNet

```
# 모델 컴파일 진행 - 아까와 달리 categorical_crossentropy 사용 > label이 숫자형 데이터이므로
model_res.compile(optimizer = tf.keras.optimizers.Adam(learning_rate= 0.000001),
                  loss = 'sparse_categorical_crossentropy',
                  metrics=['accuracy'])

# early stopping 설정
early = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3)

# 모델 fitting
model_res.fit(train_dataset,
              epochs = 30,
              validation_data=validation_dataset,
              batch_size= 32,
              callbacks=[early],
              verbose=1)
```

```
Epoch 1/30
213/213 [==============================] - 74s 171ms/step - loss: 0.3138 - accuracy: 0.8871 - val_loss: 0.4271 - val_accuracy: 0.8858
Epoch 2/30
213/213 [==============================] - 33s 152ms/step - loss: 0.3152 - accuracy: 0.8875 - val_loss: 0.4277 - val_accuracy: 0.8817
Epoch 3/30
213/213 [==============================] - 33s 152ms/step - loss: 0.3160 - accuracy: 0.8903 - val_loss: 0.4285 - val_accuracy: 0.8833
Epoch 4/30
213/213 [==============================] - 34s 153ms/step - loss: 0.3108 - accuracy: 0.8844 - val_loss: 0.4279 - val_accuracy: 0.8825
<keras.callbacks.History at 0x7fc06be4a090>
```

성능 : train 88.44%, validation 88.25%

```
[ ]  model_res.save('foodie_mobilenet_88_25.h5')
```

.h5 형식으로 저장 후 server 내 Flask Server 폴더에 저장.

이후 Flask 내에서 model.load() 한 후 model.predict() 실행.

MobileNet 선정 이유

VGG19 모델을 이용한 전이학습 모델로 90% 이상의 정확도를 달성했음

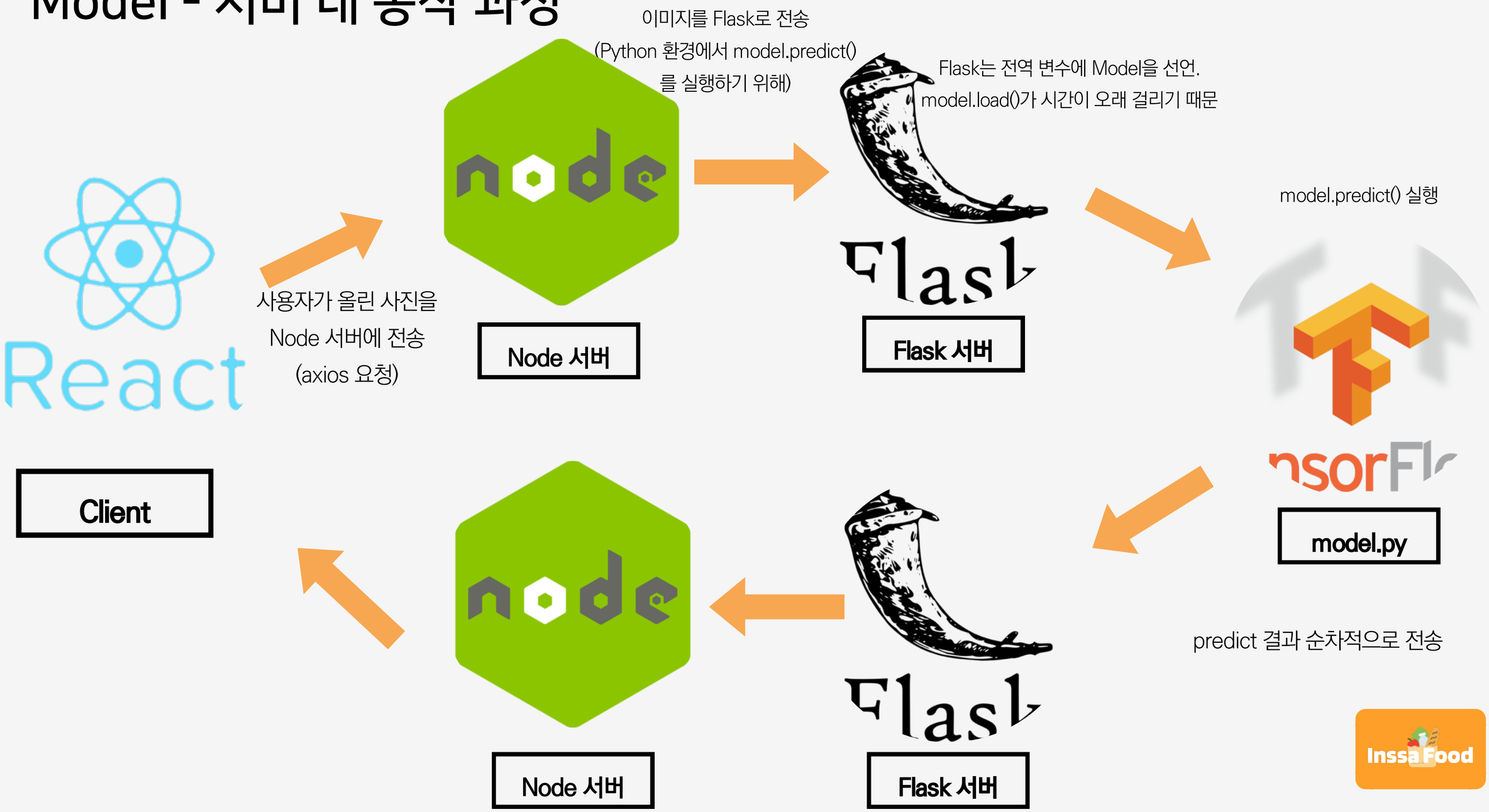
그러나, h5 형식으로 저장하니 850MB에 달하는 용량

배포 환경의 메모리가 1GB인 관계로 OOM 에러 지속적으로 발생함

따라서 모델을 경량화 해야 한다는 결정

Inssa Food

Model - 서버 내 동작 과정



개발 과정

InssaFood

PROCESS ROADMAP

01

기획

Data-Driven Decision
한국음식 데이터 발견
문제점 인식
인공지능을 활용한 해결방안 찾기
시장성 조사

02

설계

사용자 경험 중심 설계
개발환경, 협업 tool 결정
DataBase Schema 결정
작동 순서 결정

03

구현

기능 단위 구현
FE, BE 구분하지 않고
기능을 맡아서 개발

02

피드백

팀 밖의 피드백을 적용
아이디어 피벗
새로운 기능 추가

Inssa Food

어려움

01. Front-End

Layout 관련 이슈

Mobile 기준 Layout과 Web 기준 Layout 간 호환성 : 반응형 웹

해결 : Client의 대부분이 Mobile로 사진을 찍어 올릴 것이라고 판단,
모바일 보기에 최적화

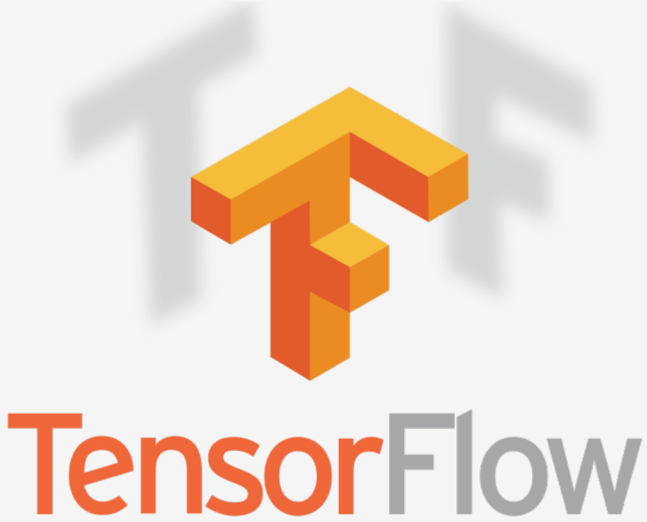
Bootstrap, Material UI 동시에 적용

해결 : Bootstrap 요소들은 HTML 태그 내에서 스타일을 부여

Inssa Food

어려움

02. TensorFlow



Local 환경

TensorFlow JS 사용 시도 : Python 모델을 변환 실패

Python 도입 결정, nodeJS의 'child-process' 중 spawn() 메서드 도입 시도

로컬 환경 구성 이슈

spawn() 메서드 실행 시 가상환경에서 실행하기 어렵다는 점 깨달음.
Flask 서버를 열어 Tensorflow 모델만 실행 결정

InssaFood

어려움

03. Naver Cloud Platform

배포 이후

Version Control

NodeJS, Python, Ubuntu, Flask, Tensorflow의 버전을 모두 맞춰줘야 함

Out of Memory

무료인 micro 1GB 메모리 서버를 사용했기 때문에, 모델 Load 중 지속적으로 에러 발생 → 모델 경량화 결정

모델 학습

모델 경량화 (800MB → 50MB) 이후 Validation accuracy 88.25% 달성. 그러나, 실제로 model.predict() 를 했을 때 모든 이미지가 같은 클래스를 return하는 결과 발견 → 사용자의 image를 preprocessing 하는 과정에서 오류 예상

Inssa Food

발전 방향

InssaFood

발전 방향

1 메뉴판 OCR

치명적인 약점 : 음식을 주문하기 전에 사용자에게 미리 정보를 알려줘야 한다! 기획 단계에서 메뉴판 OCR 기능은 구상되었지만, 기술적 한계에 직면함

2 분류 클래스 추가

현재는 한국 음식 10종에 대해 이미지 분류기가 완성된 상태.
더 많은 음식이 추가되면 추가될수록 서비스의 퀄리티가 높아질 것으로 기대됨

3 정확도 향상

현재의 정확도는 Validation dataset 기준으로 88.25% 수준. 더 높은 정확도는 더 많은 사용자로 이어질 것으로 기대됨

4 외국음식 추가

한국 음식 뿐 아니라 외국 음식을 분류기에 추가해 서비스한다면, 해외에 나가는 한국인에게까지 고객층을 확장시킬 수 있을 것으로 기대됨

5 Social Login

외국인들이 주 타겟인 것에 비해, 외국인들이 주로 사용하는 소셜 로그인 구현이 부족함.
카카오 로그인은 국내 거주 외국인에게만 사용을 기대할 수 있다는 단점이 있음

6 History Export

기획 단계에서 History result 페이지를 작게 만들어서 인스타그램에 바로 업로드할 수 있게끔 구상했음. 자연스럽게 SNS를 통한 홍보 효과 기대.

Inssa Food

감사합니다

인싸푸드