대구디지털산업진흥원(DIP)에서 인턴으로 근무할 당시, 제가 진행했던 프로젝트가 기억에 남습니다.

 # 프로젝트 배경
- 단체급식업체(구내식당, 학교 급식 등)가 식수인원을 20명씩만 틀려도, 대형업체의 경우 1만인분이 넘게 음식이 버려진다는 뉴스를 접함
- 사내 구내식당에 물어보니, 코로나로 인해 식수인원 예측에 어려움을 겪고 있다고 호소.

 # 프로젝트 목표
- 급식업체의 식수인원 예측 오차를 최소화하는 것
- 대형 급식업체는 보안상의 문제로 데이터를 공유해줄 수 없다하여, 일단 사내 구내식당업체의 동의를 받아 사내 구내식당의 식수인원 예측 오차를 최소화하는 것을 목표

 # 당시 고려한 식수인원에 영향을 주는 매개변수들
1. 메뉴
2. 작일 코로나확진자 수
3. 날씨
4. 월 & 요일 (Month & Weekday)
5. 건물 출입 예상인원 및 자택근무 예상인원
6. 건물 출입인원들의 연령대

 # 직면했던 문제점
1. 필요했던 매개변수 중, 5번과 6번은 민감한 정보라 접근권한이 없었음.
   => 해당 건물에, 여러회사가 함께 상주중이었기에, 협력을 구하기 힘들었습니다.

2. 데이터의 양이 16개월분 밖에 되지않음. 거기다 여러영업장이 아닌 한 영업장의 데이터 밖에 구할 수 없었으므로 데이터가 극히 부족.
   => 데이터 증식에 대한 필요를 느꼈으나, 당시엔 이미지데이터만 학습데이터로써 증식이 가능하다 생각하였고, 이러한 rawdata를 임의로 증식했다간 예측모델성능이 저하될 것이 자명하다 생각했기에 해결치 못했습니다.
   => 다시 돌아보니, 딥러닝모델중에 GAN이라는 모델구조를 커스터마이징하여 데이터 증식을 시도 해봄직 하다 생각합니다.

3. 어디까지 동일한 메뉴로 묶어야 하느냐와 이를 자동화를 어떻게 할 것 인가에 대한 딜레마 (* 가장어려웠던 문제점)
   (예시 1) [반달단무지, 단무지, 단무지무침]은 사실상 같은메뉴임
   (예시 2) [고등어조림, 고등어구이]는 조리법이 다를뿐 원재료는 같음
   (예시 3) [돈육메추리알장조림, 꽈리고추메추리알조림] 은 소비자 입장에서 엄연히 다른메뉴
   (메인메뉴를 메추리알로 보는지, 돈육으로 보는지에 대한 관점문제이므로 다른메뉴로 취급해야함)
   (예시 4) [산나물무침, 취나물무침, 비름나물무침, 나물겉절이, 돌나물무침등]은 나물의 종류에 따라 크게 식수인원이 변동되지 않으므로 같은 나물무침으로 봐야함.

 # 접근방법 및 해결과정 (* 가장어려웠던 문제점인 3번에 대해서 자세히 기술하겠습니다.)
1. 성능을 최대한 끌어낼 수 있는 방법을 프로젝트 기한 내에 찾지 못할때를 대비하여, 최소한 이 프로그램이 작동이라도 할 수 있게, brute-force 한 방법이라도 먼저 만들어야겠다 생각했습니다. 그리하여 가장 간단한 매칭방법이라 생각하는 글자 일치율과 글자 일치갯수, 단어포함관계등을 통하여 메뉴를 매핑하는 법을 일단 고안했습니다.
2. 메인메뉴가 메뉴 중에서 가장 강력한 영향력을 가질 것이므로, 메인메뉴 판별기를 만들고, 메인메뉴로 선정된 음식에 가중치를 주는 방법을 고려하고 시도했습니다.
3. 메뉴의 이름을보고 원 식자재들과 조리방법으로 분리, 이를 매개변수로 삼으면 어떠한 식당이라도 동일한 방법으로 적용하여 자동화가 가능할 것이라 생각하고 시도하였습니다.
예시) [꽈리고추멸치볶음] 은 꽈리고추/멸치 + 볶음(조리방법) 으로 분리가 가능합니다.
=> (하지만, 탕수육, 탕평채와 같은 메뉴이름 자체에 조리방법이 포함되지않은, 식자재명도 포함되지않는 메뉴들이 있어 해결방법을 고민했습니다.)

4. 이를, 예전 텍스트압축프로그램을 만들때 사용했던 참조데이터 격인 meta데이터를 만들어, 관리 및 매핑을 하면 자동화가 가능할것이라 생각하고 시도하였습니다.)
5. 하지만, 이러한 특수한이름의 음식명이 너무많아 메타데이터를 일일이 만들기엔 , 프로젝트 기한내에 모든 단체급식업체의 자동화는 불가능하다 판단하고, 해당 구내식당에만 커스터마이징하여 한정적으로 적용하였습니다. (1번의 방법론 + 4번의 방법론을 적용하였습니다. 2번은 적용하였으나 예측모델의 성능저하가 발견되어 제거 하였습니다.)

 # 실제결과

- 깃허브 주소 : https://github.com/Sun26-Avrin/streamlit01-restaurant_headcount
- 웹앱 주소 : https://share.streamlit.io/sun26-avrin/streamlit01-restaurant_headcount/DureBiang_Streamlit.py
- 훈련용데이터로 교차검증을 했을 시, 평균 오차인원은 10명내외였습니다. (훈련용데이터가 너무 적어, 과적합이라 생각합니다.)

 # 아쉬웠던 점 및 여전한 고민

- 남은시간 최대한 사용자가 지속적으로 사용하는데 불편함이 없도록, 자동화와 모듈화에 신경을 썼으나, 타협점을 찾지 못한 것이 사용자인 '영양사에게 모델추가학습권한을 줘도 되는가' 였습니다. 이를 인터페이스로 만들어 제공하자니, 실수로 들어가는 데이터가 발생시 어떻게 모델복원을 시켜드려야하는지도 고민해야했고, 컴퓨터 복원기능처럼 타임라인을 보여드리고 그에 사용한 데이터가 무엇인지 보여드리자니 너무 웹앱자체가 깔끔하지 않고 무거워진다 생각하여 해당기능을 넣지 않았습니다. 이는 아직도 고민중인 고민거리입니다.

- 초기 목표이자 궁극적 목표였던 모든 급식업체에 적용되는 자동화된 성능좋은 예측모델을 만드는 것에는 실패하였지만, 관리자와 사용자입장에서 지속적으로 사용할 수 있게끔 하려 모듈들을 구축하며 어떤 구조로 유지보수를 할것인가에 대해 고민했던 경험은 정말 좋은 경험이었습니다.