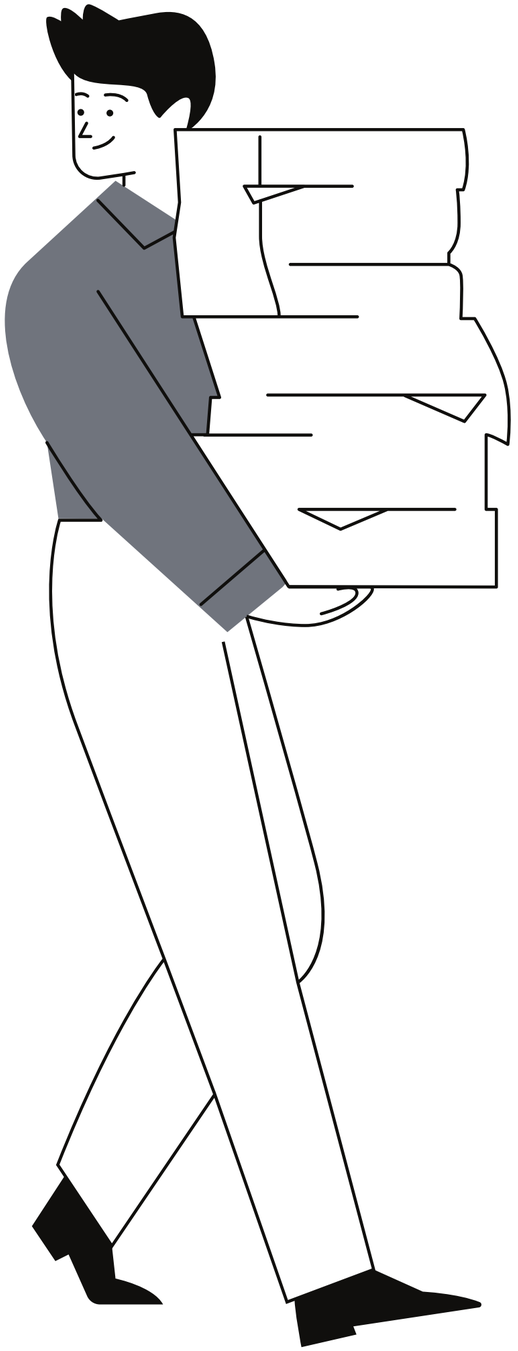
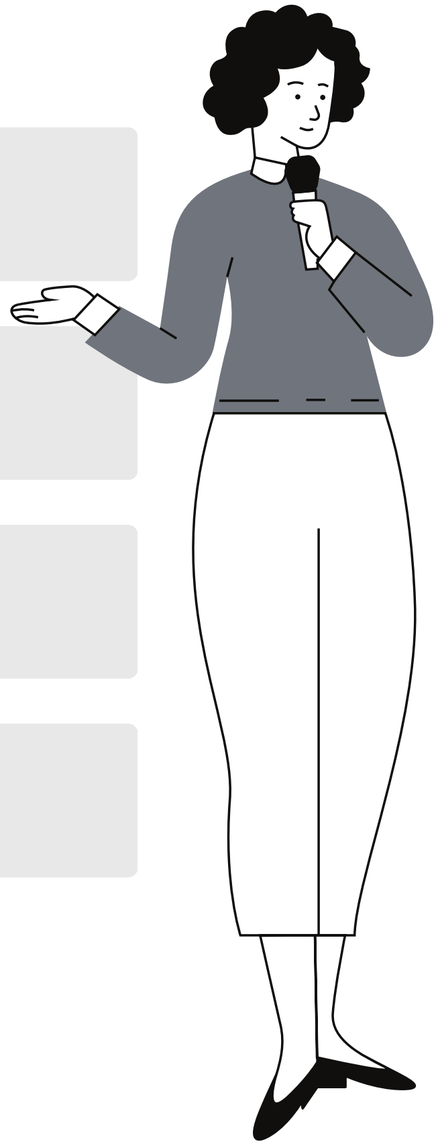
ESC 23-1 파이널프로젝트

AI 방법론을 통한 항공편 지연예측

정연섭 김지원 임세희 전상후



목차



1. 데이터 소개

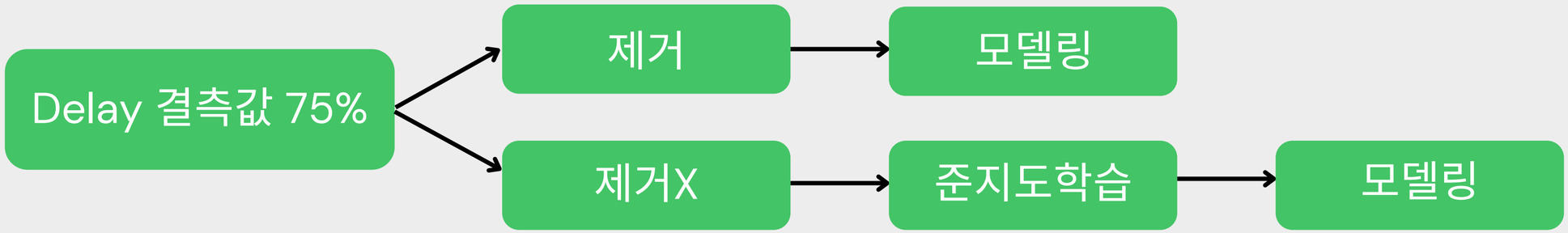
train.csv, test.csv 두 개의 데이터 내 항공편 운항 관련 정보(출발 시간, 도착 시간, 항공편 취소 여부, 경유 여부 등)를 통해 항공편 지연 확률을 구하고 지연 여부를 예측

Train/Test data

1
-1,000,000개 데이터
-그 중 train data의 target column인 delay 결측치 74.5%

2
-Estimated_Departure_Time(EDT), Estimated_Arrival_Time(EAT), Origin_State,Destination_State, Airline,Carrier_Code(IATA), Carrier_ID(DOT)에 결측치가 약 10%

2. 전처리



나머지 결측치 각 10%

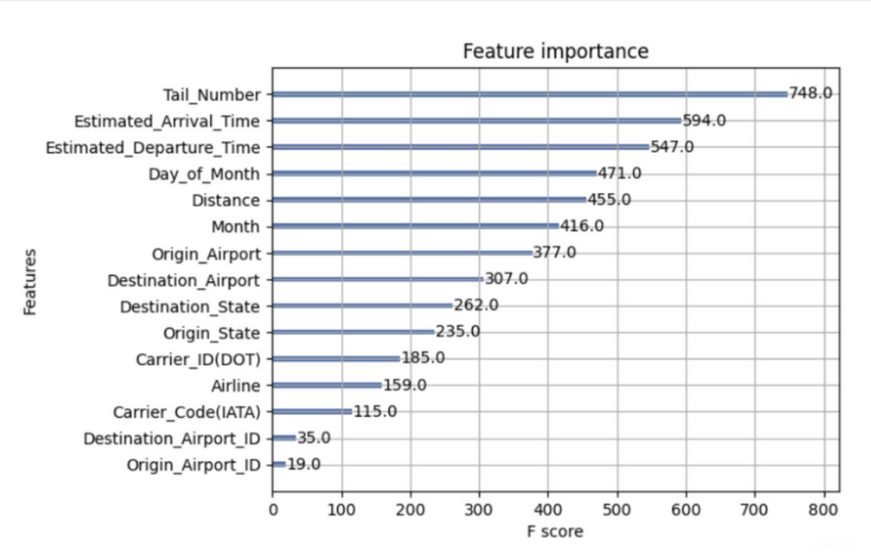
1 feature importance

2 카이제곱 분석

3 비행시간 처리

2. 전처리

1. feature importance



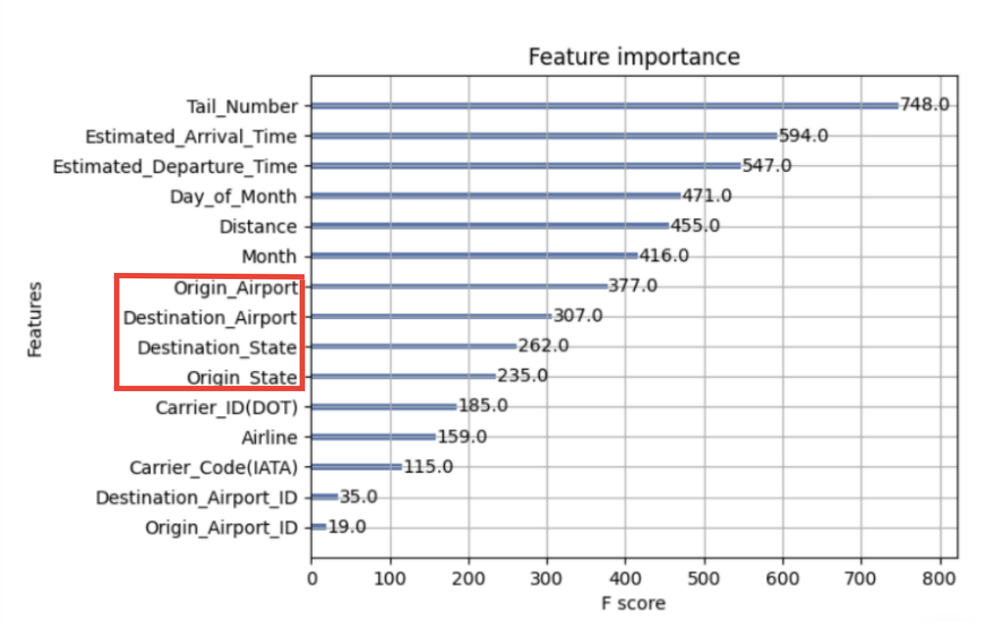
- 중요도를 알아보기 위해 우선 결측값이 있는 모든 행들을 제거하고, XGBoost 모델을 적합 후 중요도 분석

- 중요도가 200이하인 변수는 제거

(Carrier_ID,Airline,Carrier_Code,Destination_Airport_ID,Origin_Airport_ID)

2. 전처리

2. 카이제곱 분석



- 여러개의 Airport이 하나의 state에 존재할 수 있으며, 연착여부는 사실상 State보다는 Airport에 영향을 더 많이 받을 것이다(활주로 크기 등). 따라서 Airport이 State를 대표할 수 있으면 State도 제거할 수 있을 것이다.

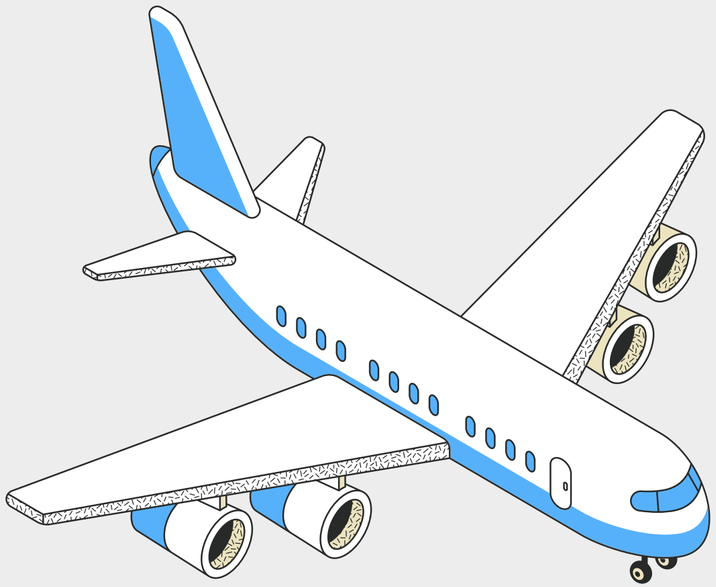
- 상관분석을 통해 알아보고자 하였으나 이는 연속형 데이터만 사용가능하기에 카이제곱 분석을 사용하였다. 분석한 결과, 유의수준 5%하에 귀무가설를 기각하기에 두 변수는 서로 종속하다.

- 따라서 Destination_State, Origin_State도 제거하여 마지막엔 총 8개의 변수만 남게된다.

(Tail_Number, EAT, EDT, Day_of_Month, Distance, Month, Origin_Airport, Destination_Airport)

2. 전처리

3. 비행시간 처리



- Estimated_Departure_Time (EDT), Estimated_Arrival_Time (EAT)는 약10%의 결측치를 가지고 있으며, 제거하려 하였으나 제출 파일 행 수를 맞추기 위해 최빈값으로 대체

- EDT>EAT인 경우도 존재하였으며, 그 이유는 현지시간 기준으로 측정되었기에 시차로 인한 오류가 발생했다. 이를 해결하기 위해서는 외부 데이터 시차데이터를 도입해야 한다.

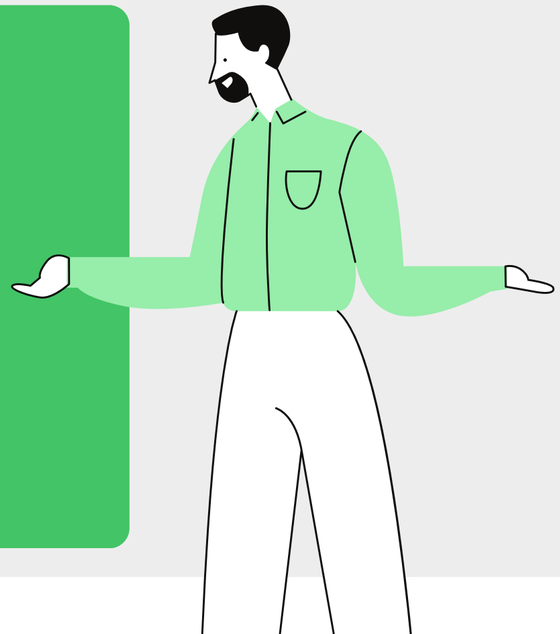
3.모델링-1

1. Delay에서 결측치를 가진 모든 행을 제거

2. Random forest classifier 모델 적용

Random forest classifier이란?

- Classification Decision Tree
  단점: overfitting, Missing Data처리 어려움...
- Bagging : Ensemble method
  여러 개의 Training data, 여러 개의 모델, 여러 개의 예측값 중 최빈값 선택
- Random forest: Bagging -> 여러 개의 Decision Tree 생성
  장점: High Accuracy, overfitting 방지



3.모델링-2

1. 데이터에 대해 standardscaler사용

2. 하이퍼파라미터 튜닝과 함께 XGBClassifier 모델 사용

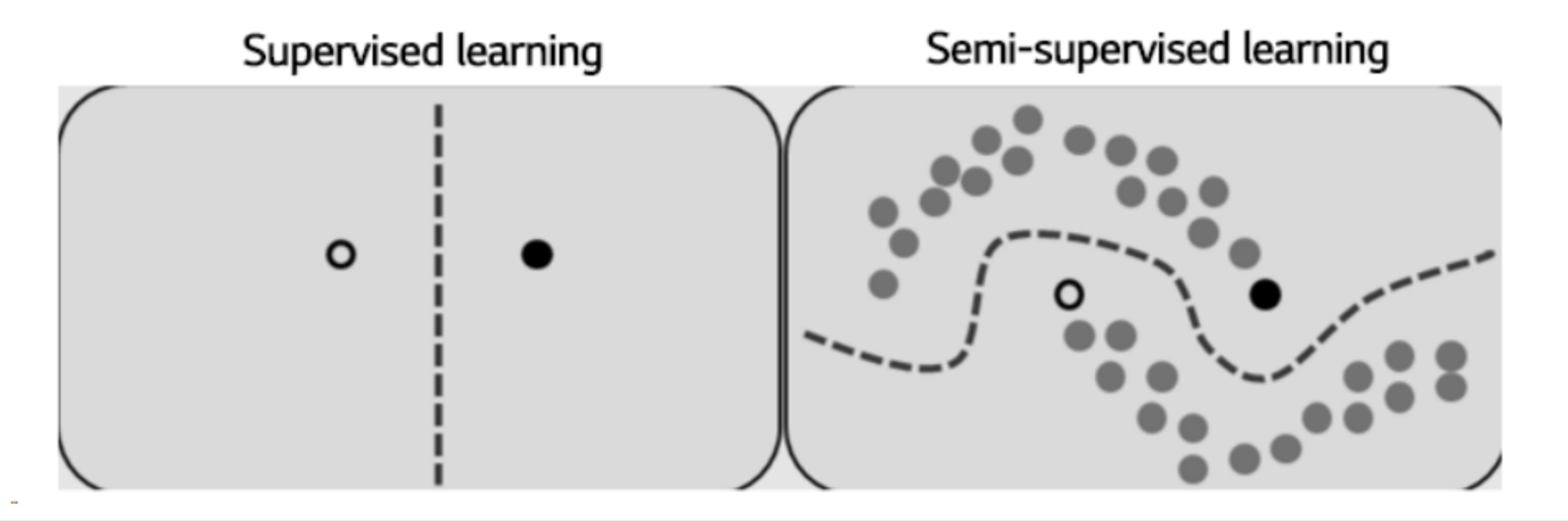
XGB classifier이란?

- 앙상블 알고리즘 중 하나
  장점: (1) 뛰어난 예측 성능
  (2) GBM 대비 빠른 수행 시간
  (3) 과적합 규제(Overfitting Regularization)
  (4) Tree pruning(트리 가지치기) : 긍정 이득이 없는 분할을 가지치기해서 분할 수를 줄임
  (5) 자체 내장된 교차 검증
  (6) 결손값 자체 처리

4. 준지도 학습

준지도 학습(Semi-supervised learning):

- 지도학습과 비지도학습 그 사이: 적은 양의 labeled data와 많은 양의 unlabeled data를 가질 시 사용
- 적은 양의 labeled data에 지도학습을 사용하고, unlabeled data에 비지도학습을 사용한다.



4. 준지도 학습

준지도 학습 중 Self Training Classifier를 사용

1. 레이블이 달린 데이터를 가지고 학습

2. 이 모델을 가지고 레이블이 달리지 않은 데이터를 예측

3. 레이블이 있는 기존 데이터와 학습 후 예측한 데이터를 결합해
모델링에 사용. (모델링은 앞의 두 모델 사용)

* 지연 여부의 결측행을 제거하지 않고 채워서 모델링 한 것이 포인트!
(하지만 예측의 정확도를 보장하지는 못함)

결과값

최신순 점수순

	제목	제출 일시	public점수 private점수	제출선택	
854586	SelfTrained_submission.csv edit	2023-06-01 12:04:33	2.79517487 1.8111087041	☐	Self training classifier Random forest classifier
854480	optimized_submission_origin.csv edit	2023-06-01 00:17:26	1.1435043848 0.8024744954	☐	
854428	optimized_submission.csv edit	2023-05-31 21:58:30	1.1536386189 0.8094384286	☐	XGB classifier
854415	baseline_submission_origin.csv edit	2023-05-31 21:20:19	0.9878956801 0.7339283627	☐	
854405	baseline_submission.csv edit	2023-05-31 21:07:58	1.1601582817 0.8205821332	☑	Random forest classifier

감사합니다!

정연섭 김지원 임세희 전상후