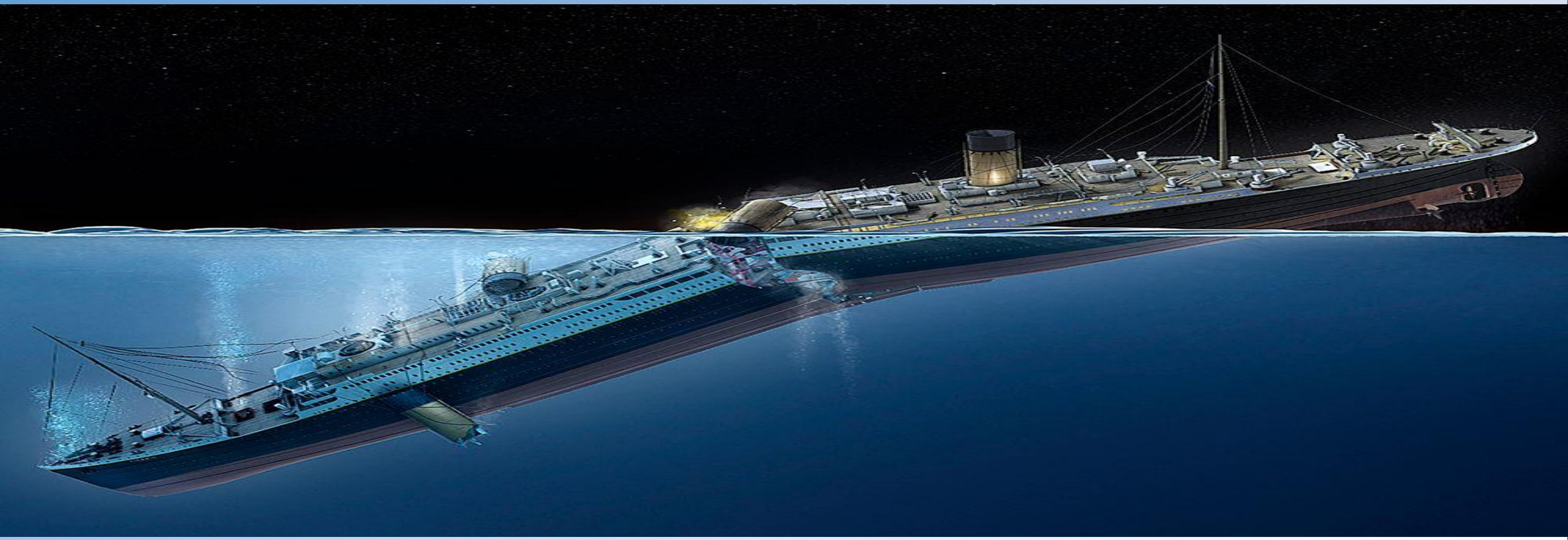
Kaggle Titanic Project



2021.07.05(월) ~ 07.09(금)

팀원
이두환 이정환
정혜민 전성민

한국 IT 교육원

Ⅳ. 목차

Chapter 1

대회 개요

1.1 대회 개요

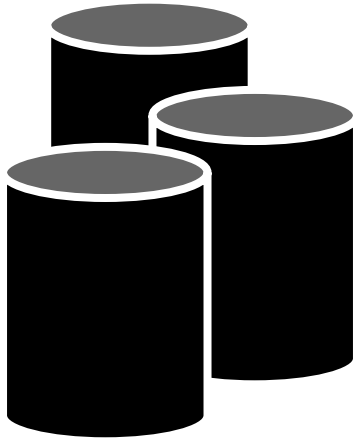
현재 Kaggle에서 진행하고 있는 데이터 분석 대회로써 타이타닉에 탑승했던 승객들의 생존 여부를 예측하는 것이 대회에 주제입니다.(검정 집합에 포함된 각 변수에 대해 0 또는 1 값을 예측해야 합니다)

구분	세부내용
대회 참여 목적	LightGBM, CatBoost, XGBoost 성능 비교 테스트 (분류 모형)
대회 참여 기간	2021.7.5 ~ 2021.7.9
대회 참여 인원	4명(이두환, 이정환, 전성민, 정혜민)
참여 인원 역할	이정환(PPT,Feature Engineering,LightGBM 모델 성능 평가) 이두환(XGBoost 모델 성능 평가) , 전성민(CatBoost 모델 성능 평가) 정혜민(Tableau & Plotly활용한 시각화)
성적	F1 스코어 / (1671/52673)
소스코드	EDA: https://public.tableau.com/app/profile/junghyemin/viz/titanic_0707_story/1_2 노트북: https://www.kaggle.com/ajalnine/titanic-keras-vs-lightgbm-vs-catboost-vs-xgboost

1.2.1 대회 데이터 소개

이번 대회에서의 전체 데이터 크기는 90.9KB 입니다.

(Train data 행 891, 열 12), (Test data 행 418, 열 11)



데이터 이름	데이터 사이즈
Train	56.76KB
Test	27.96KB
Sample Submission	3.18KB

전체 데이터 크기: 90.9KB

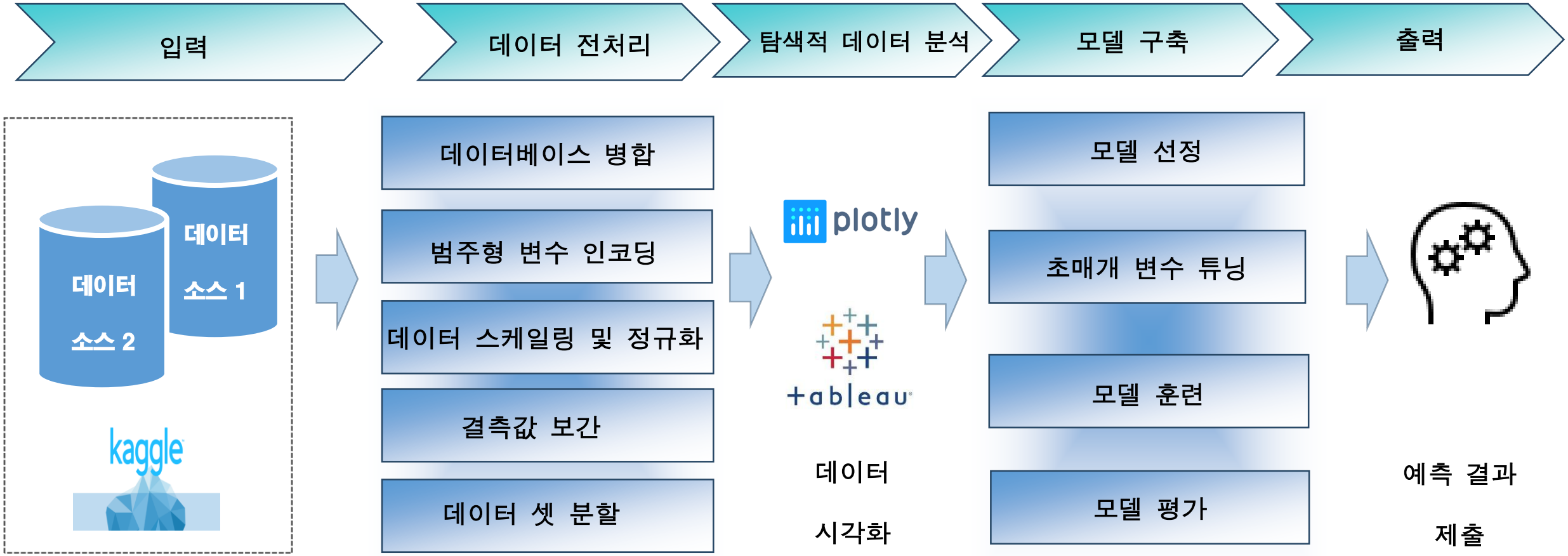
1.2.2 대회 데이터 소개

종속변수 Survival, 독립변수는 총 10개입니다.(object 4개, float[64]2개, integer[64] 4개)

Variable	Data Type	Definition	Key
Survival	int64	Survival	0 = No, 1 = Yes
Pclass	int64	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
Sex	objeck	Sex	
Age	float64	Age in years	
Sibsp	int64	# of siblings / spouses aboard the Titanic	
Parch	int64	# of parents / children aboard the Titanic	
Ticket	objeck	Ticket number	
Fare	float64	Passenger fare	
Cabin	object	Cabin number	
Embarked	object	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

1.3 머신러닝 워크플로우

데이터는 Kaggle 에서 다운 받아서, Kaggle Notebook으로 작업하도록 설계하였습니다.



1.4 대회 참여 목적

1. EDA를 태블로 또는 Plotly와 같은 동적 시각화로 작성

2. 각 모델 하이퍼 파라미터 튜닝

3. 최신 알고리즘 소개 및 테스트를 통해 적합한 모델 산출

4. 모델 최종 선정

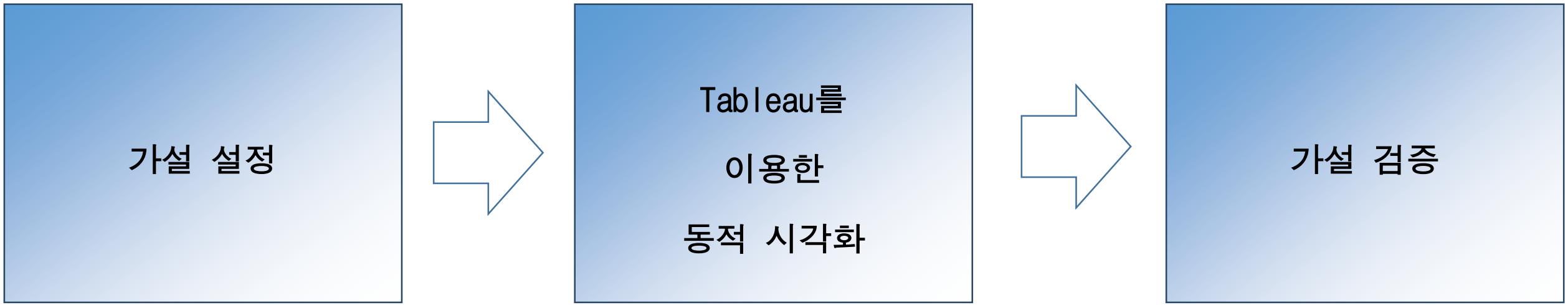
Chapter 2

Tableau & Plotly활용한 탐색적 자료분석

(EDA)

2.1 Exploratory Data Analysis(탐색적 자료 분석)

원 데이터(Raw data)를 가지고 유연하게 데이터를 탐색하고, 데이터의 특징과 구조로부터 얻은 정보를 바탕으로 통계 모형을 만드는 분석방법입니다.



2.2 가설 설정

설명:

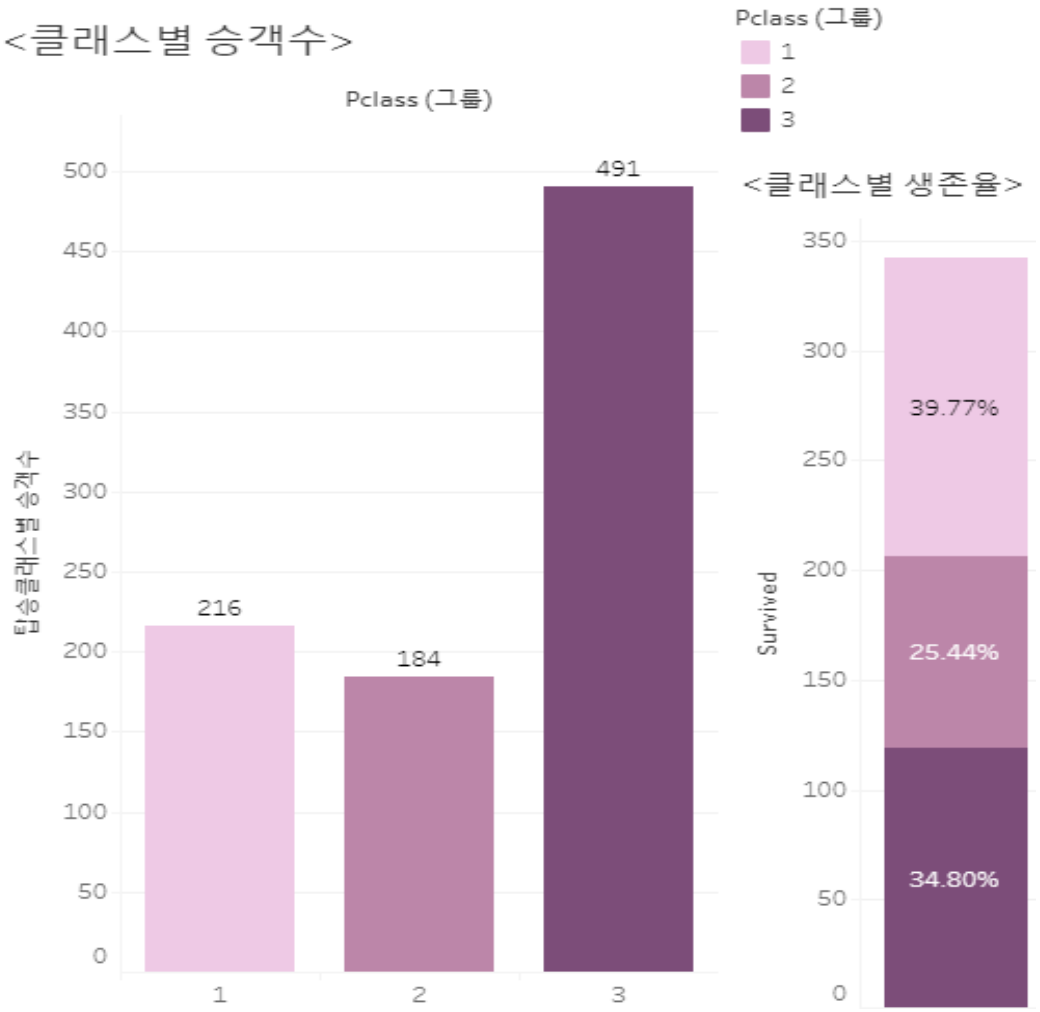
- **부자들이 더 많이 생존했을 것이다.(Pclass, Fare)**
- **여자들이 더 많이 생존했을 것이다.(Sex)**
- **사회적 약자(어린이, 노인)들이 더 많이 생존했을 것이다.(Age)**

2.3.1 가설 검증 1-1

설명 :

승객수는 Third Class 가 491명으로 가장 많았지만 생존율은 First Class가 39.77%로 가장 높았습니다.

결과 적으로 부자들이 더 많이 생존했다는 것을 보여줍니다.

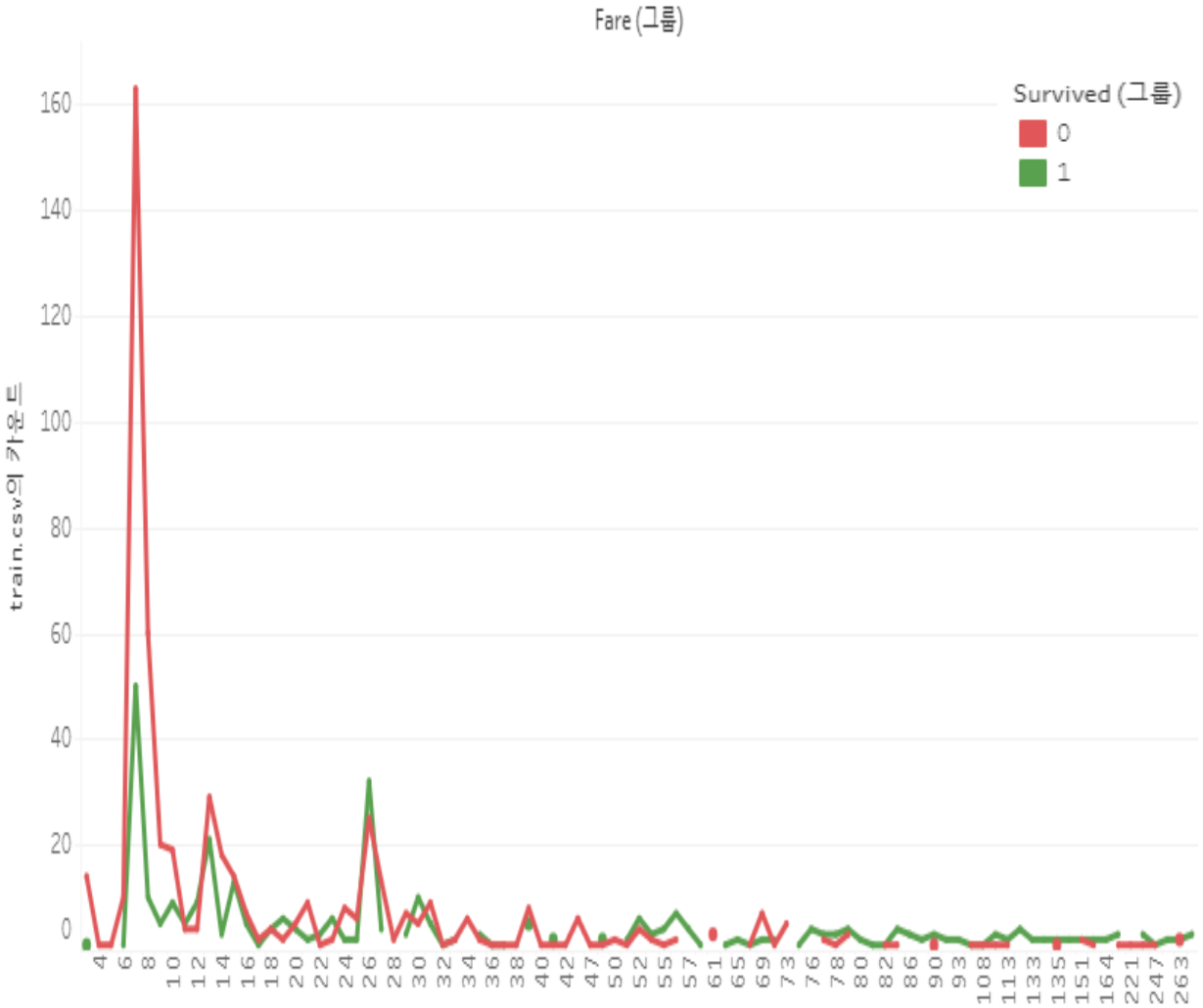


활용 도표: 막대그래프

2.3.2 가설 검증 1-2

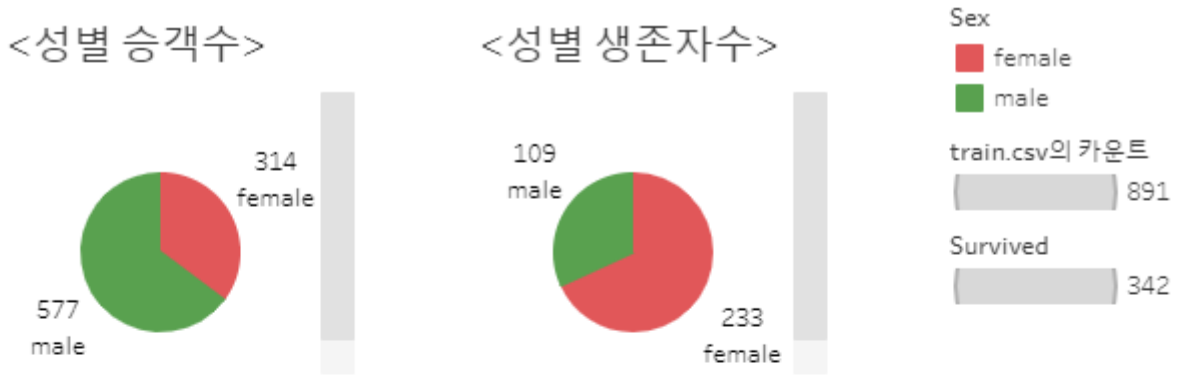
설명 :

Fare(티켓값) 데이터 활용해서 분석한 결과 상대적으로 비싼 티켓을 구매한 탑승객이 더 많이 살아남았다는 것을 확인할 수 있습니다.



활용 도표: 라인 차트

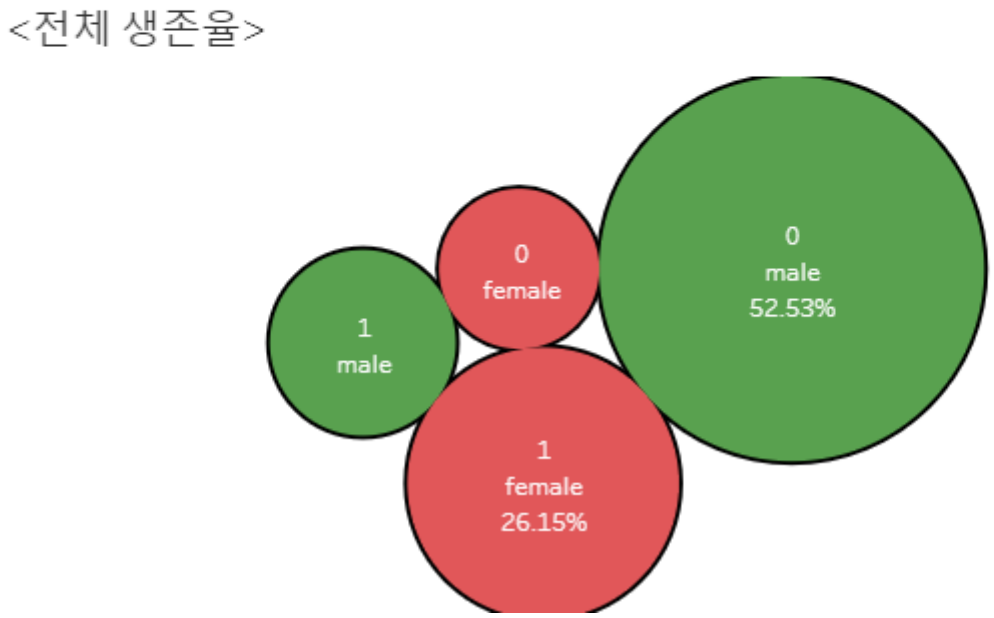
2.3.3 가설 검증 2



설명 :

승객수는 남자가 더 많았지만 생존자 수는

여자가 더 많은 것을 확인할 수 있습니다.



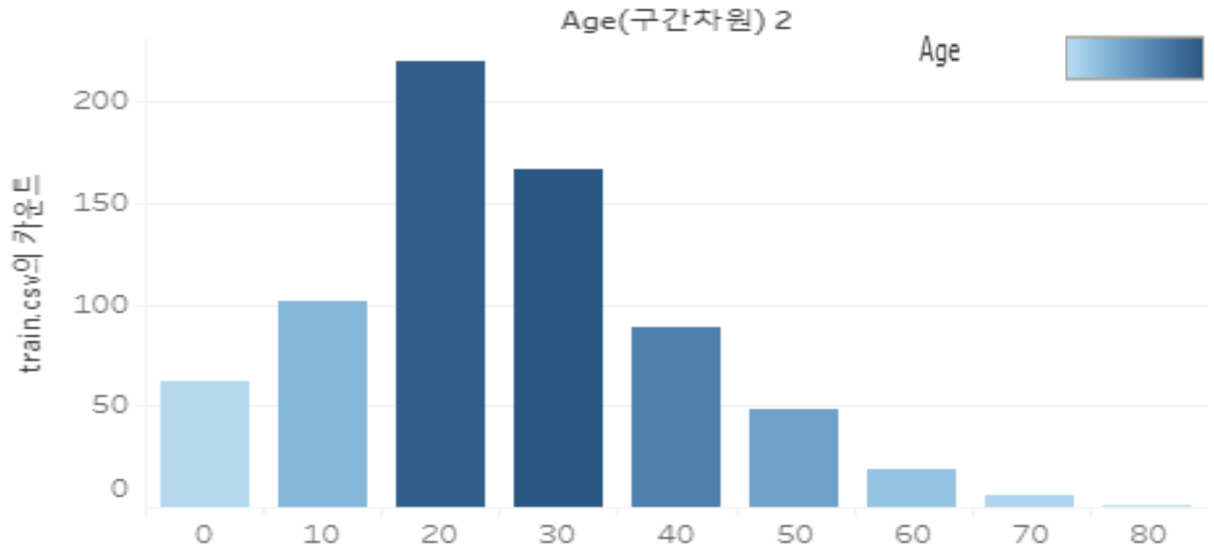
활용 도표: 파이차트

2.3.4 가설 검증 3

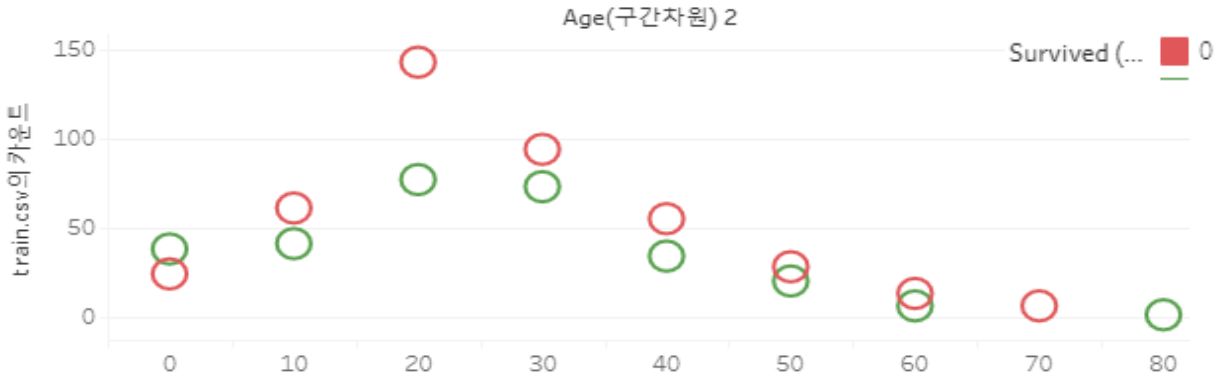
설명 :

승객수는 20대-30대가 제일 많았지만 상대적으로 영유아 및 노인들이 많이 생존한 것을 확인할 수 있습니다.

<연령대별 승객수>



<연령대별 생존자수>



활용 도표: 히스토그램 차트

Chapter 3

각 모델에 대한 하이퍼 파라미터 튜닝

3.1 LightGBM 하이퍼 파라미터 튜닝

- n_esitmators(500) : 반복 수행하는 트리 개수(크면 성능이 올라가나 너무 크면 과적합)
- learning_rate(0.1):
- Boosting:
  - DART: tree dropout 적용(수행시간 긴편)

  [트리]
  - max_depth(7) : 깊이, Level-wise 방식보단 leaf-wise 방식이 상대적으로 깊음
  - num_leaves(31) : 하나의 트리가 가질 수 있는 최대 리프 개수

  [샘플링]
  - sub_sample(0.9) : 행 샘플링
  - Colsample_bytree(0.9) : 칼럼에 대한 샘플링
  - n_jobs(5): 병렬 스레드 수

  [규제]
  - lambda_l1(reg_lambda), lambda_l2(reg_lambda)

3.2 CatBoost 하이퍼 파라미터 튜닝

- LightGBM 같은 접근법으로 튜닝
- Iterations(500) : 반복 수행하는 트리 개수(크면 성능이 올라가나 너무 크면 과적합)
- learning_rate(0.09):
- Boosting:
  - DART: tree dropout 적용(수행시간 긴편)

  [트리]

  - depth(9) : 깊이, Level-wise 방식보단 leaf-wise 방식이 상대적으로 깊음

  [샘플링]

  - Sub_sample(0.9) : 행 샘플링
  - Colsample_bytree(0.9) : 칼럼에 대한 샘플링

  [규제]

  - lambda_l1(reg_lambda), lambda_l2(reg_lambda)

3.3 XGBoost 하이퍼 파라미터 튜닝

- n_esitmators(num_iterations)(500) : 반복 수행하는 트리 개수(크면 성능이 올라가나 너무 크면 과적합)
- learning_rate(0.01):
- Boosting:
  - DART: tree dropout 적용(수행시간 긴편)

  [트리]

  - max_depth(9) : 깊이, Level-wise 방식보단 leaf-wise 방식이 상대적으로 깊음
  - num_leaves(31) : 하나의 트리가 가질 수 있는 최대 리프 개수

  [샘플링]

  - sub_sample(0.9) : 행 샘플링
  - Colsample_bytree(0.9) : 칼럼에 대한 샘플링
  - n_jobs(5): 병렬 스레드 수

  [규제]

  - lambda_l1(reg_lambda), lambda_l2(reg_lambda)

Chapter 4

모델 소개 및 비교

(최신 알고리즘)

4.1 LightGBM

- LightGBM은 XGBoost와 함께 인기 있는 부스팅 계열 모델입니다.
- 장점: 학습과 예측 속도가 XGBoost에 비해 빠름, 카테고리형 피처 자동 변환

- 트리 분할: 리프 중심(Leaf-wist) 트리 분할 방식 (비대칭적인 규칙 트리)
  - 장점 : 예측 오류 손실 최소화
  - 단점 : 오버피팅에 약함

4.2 CatBoost

- 대칭 트리(CatBoost와 다른 부스팅 알고리즘의 주요 차이점)

예측 시간을 줄이는 데 도움

지연 시간이 짧은 환경에서 매우 중요

- 기존의 부스팅 모델 : 일괄적으로 모든 훈련 데이터를 대상으로 잔차 계산

Catboost : 일부만 가지고 잔차 계산 -> 모델 생성 -> 모델로 예측한 값을 남은 데이터의 잔차로 사용

- 현재 데이터의 타깃 값을 사용하지 않고, 이전 데이터들의 타깃 값만을 사용 -> Data Leakage 없음

- 파라미터 최적화 : 파라미터 튜닝에 크게 신경 쓰지 않아도 됨.

- 단점 : 데이터 대부분이 수치형 변수인 경우, LightGBM 보다 학습 속도가 느림.

4.3 XGBoost

- 장점 : 병렬 연산을 지원하여 계산 속도를 향상. GPU를 사용할 수 있는 옵션을 제공

과적합(Overfitting) 을 제어하기 위해 두 가지 방법

1. 모델의 복잡도를 제어 - 하이퍼 파라미터(max_depth, min_child_weight, gamma) 사용

2. Random Value를 추가 - subsample, colsample_bytree 파라미터 사용

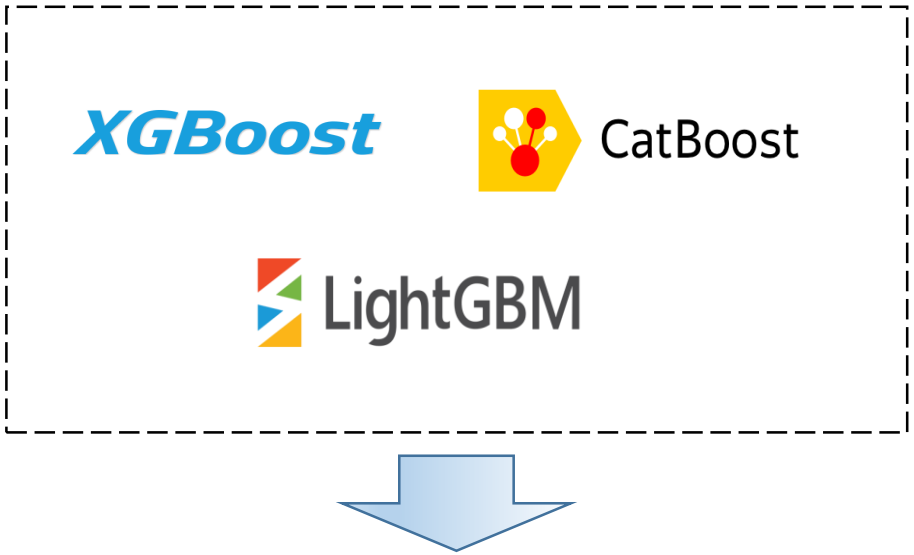
단점

- 학습 데이터에 예민하게 반응

- 튜닝을 할 때 손 봐야할 파라미터가 너무 많음

4.4 최종 모델 선정 / 모델 성능 평가

하이퍼 파라미터 튜닝, 그리드 서치, 변수 선택을 통해 F1-Score 가장 좋은 성능을 보인 XGBoost모델을 최종 모델로 선정하였습니다.



모델	변수	리더보드
XGBoost	Ticket	0.82296
CatBoost	Ticket	0.78947
LightGBM	Ticket	0.77511

Chapter 5

결론

5.1 시사점

1. 태블로를 통해 EDA를 Plotly와 같은 동적 시각화 방식으로 대시보드를 구성하였습니다.

2. 각 모델에 대한 하이퍼 파라미터 튜닝을 통해 모델을 측정하여 정밀한 모델을 구현하였습니다.

3. Kaggle 대회 참여를 통해 최신 알고리즘에 대한 이해도 향상과 데이터 분석에 대한 흐름을 전반적으로 경험할 수 있었습니다.

한계점

1. 시간 부족으로 인해서 다양한 전처리 방법 및 기존 알고리즘들을 최신 알고리즘과 테스트하지 못한점

2. 불순도 기반 Feature Importance의 한계점을 고려해야 합니다. train 데이터 셋으로부터 얻은 통계량으로 계산된 중요도 이기 때문에, test 데이터 셋에서는 이 변수 중요도가 어떻게 변하는지 알 수 없기 때문입니다.

5.2 참고 자료

- 태블로 대시보드
  [https://public.tableau.com/app/profile/junghyemin/viz/titanic_0707_story/1_2]
- 대시보드 참고 홈페이지
  [https://public.tableau.com/app/profile/shyamala.g/viz/TitanicDashboards/SurvivalontheTitanicbyAgeGenderClassFare]

- 케글 노트북
  [https://www.kaggle.com/ajalnine/titanic-keras-vs-lightgbm-vs-catboost-vs-xgboost]
- XGBoost 공식 홈페이지
  [https://xgboost.readthedocs.io/en/latest]
- CatBoost 참고 홈페이지
  [https://hanishrohit.medium.com/whats-so-special-about-catboost-335d64d754ae]
- LightGBM 공식 홈페이지
  [https://lightgbm.readthedocs.io/en/latest]
- LightGBM PPT 유튜브 영상
  [https://www.youtube.com/watch?v=1dSfWpFnvP0&t=532s]

감사합니다