

# Kaggle

최필주

- 2010년 설립된 머신러닝 경진대회 플랫폼
  - 기업 연계 주최 경진대회를 통해 문제와 데이터 제공
  - 개인/팀이 모여 높은 점수를 내기 위해 경쟁하는 구조
  - 기업: 우승자의 코드와 분석 기법 활용
  - 개인: 데이터 다룰 기회와 입상 시 상금 획득

## ○ 유명했던 대회들

### ■ Netflix Prize

- 문제: 사용자의 과거 영화 평점 데이터 → 새 영화 평점 예측
- 상금 \$1 M

### ■ 페이스북 V 체크인 예측 경진대회


- 문제: 페이스북 사용자가 체크인하는 장소 예측
- 입상 시 페이스북 채용 기회 제공

## ◎ 5가지 경진대회 유형

- Featured: 일반적인 경진대회, 상금와 캐글 포인트 부여
- Getting started: 머신러닝 입문자를 위한 예제 기반 학습용 경진대회, 상금 X
- Research: 연구 목적, 적은 상금
- Playground: 캐글 주최 경진대회
- Recruitment: 채용 목적, 상금 대신 채용 면접권 부여

## ◎ 2019년 초 Titanic competition 수행

- <https://www.kaggle.com/c/2019-1st-ml-month-with-kakr/overview>

 InClass Prediction Competition

### 2019 1st ML month with KaKR

캐글 코리아와 함께하는 1st ML 대회 - 타이타닉 생존자를 예측하라!

353 teams · 6 months ago

[Overview](#) [Data](#) [Kernels](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [My Submissions](#) [Late Submission](#)

#### Overview

<a href="#">Description</a>	<h4>Introduction</h4> <p>본 대회는 구글 코리아가 후원하고, 캐글 코리아(비영리 페이스북 온라인 커뮤니티)가 진행하는 데이터 사이언스 대회입니다. Academic 목적이며, 대한민국 누구나 참여하실 수 있습니다.</p> <h4>Competition background</h4> <p>RMS 타이타닉의 침몰은 역사상 가장 악명 높은 참사 중 하나입니다. 1912년 4월 15일, 첫 항해 도중 타이타닉은 빙산과 충돌한 후 침몰하였고, 이로 인해 2224명의 승객과 승무원 중 1502명이 사망했습니다. 이 비극은 국제 사회에 큰 충격을 주었고, 선박 안전 규정을 개선하는 계기가 되었습니다.</p> <p>많은 사망자가 생긴 이유 중 하나는 승객과 승무원을 위한 구명정이 충분하지 않았기 때문입니다. 침몰에서 살아남는 데는 여러 요소가 있었겠지만, 여성, 어린이 및 상류층과 같은 특정 그룹의 사람들이 생존 가능성이 더 컸습니다.</p> <p>이 컴퍼티션에서 우리는 어떤 부류의 사람들이 생존할 가능성이 높았던 지에 대해 분석을 하고, 이를 기반으로 하여 머신러닝 모델을 만든 뒤 승선한 사람들의 생존유무를 예측합니다.</p>
<a href="#">Evaluation</a>	
<a href="#">Prize</a>	
<a href="#">Timeline</a>	

## ○ 데이터 이해

- 데이터에 대한 기초적인 통계와 시각화, 변수간 관계 확인

## ○ 평가 척도 이해

- 대회 문제 의도 파악, 패널티 확인

## ○ 교차 검증 기법 선정

- 일정 비율로 훈련/검증 데이터로 분리
- 훈련 데이터로 모델 학습하고 검증 데이터로 평가 → 다수 반복

## ◎ 피처 엔지니어링(\*)

- 학습에 사용할 데이터 준비: 스케일링, 이상값 제거, 결측값 대체, 범주형 데이터 변환, 변수 선정, 파생 변수 생성 등

## ◎ 모델 튜닝

- 교차 검증 점수를 기반으로 모델의 최적 파라미터 파악

## ◎ 앙상블

- 다수의 모델을 조합하여 사용

## ○ Baseline 모델

- 최소한의 성능을 보이는 기본 머신러닝 파이프라인
- Baseline 모델을 구축해야 되는 이유
  - 올바르게 동작하는 기본 초석
  - 성능 비교의 기준점

## ○ 재현성

- 랜덤값을 사용하는 경우 seed 값을 고정하여 재현 가능하도록 설정



# 실습1. Titanic 생존자 맞추기

## ○ 문제 정의

- Titanic 호 탑승객의 특징(동승자, 성별, 티켓 클래스 등)에 따른 생존 결과 예측하기
- 범주형 회귀 분석에 해당

## ○ 데이터 읽어오기

- rain.csv - 예측 모델을 만들기 위해 사용하는 학습셋
- test.csv - 예측 모델을 이용하여 예측할 탑승객 정보가 담긴 테스트셋
- sampleSubmission.csv - 제출시 사용할 수 있는 csv 파일

## ○ 데이터 설명

- 종속변수 (train 데이터 셋에만 포함되어 있음)
  - Survival: 생존유무, target 값. (0 = 사망, 1 = 생존)
- 독립변수
  - Pclass: 티켓 클래스. (1 = 1st, 2 = 2nd, 3 = 3rd)
  - Sex: 성별
  - Age: 나이(세)
  - Sibsp (Siblings and spouse): 함께 탑승한 형제자매, 배우자 수 총합
  - Parch (Parents and children): 함께 탑승한 부모, 자녀 수 총합
  - Ticket: 티켓 넘버
  - Fare: 탑승 요금
  - Cabin: 객실 넘버
  - Embarked: 탑승 항구
- 주의점: 결측값(NaN) 존재

## ○ 데이터 정보 확인

- 데이터의 양 활용: `shape`
- 각 열별 데이터 정보 확인: `info()`
  - 각 열별 데이터의 개수가 동일하지 않음: 결측값 존재
- 결측값 확인: `isnull().sum()` 활용
  - Train의 결측값이 있는 열: Age, Cabin, Embarked
  - Test의 결측값이 있는 열: Age, Fare, Cabin

## ○ 데이터 정보 확인

- Bar chart 활용
  - Survived와 dead 각각의 feature별 수 확인
  - 각 feature별 생존자 수와 비율 확인
- 열 간 correlation 확인
  - `df.corr()` 사용

## ○ Feature engineering

- 숫자가 아닌 데이터 → 숫자로 맵핑
  - `df[col] = df[col].map(mapping)`
- 숫자 → 범주형으로 변환하기
  - 예: 나이 → 나이 대(10대, 20대 또는 아동, 청년, 노년 등)를 구분

## ◎ Feature engineering

### ■ 결측값 채우기

- fillna 함수 사용
- 일괄 값으로 채우기: min, mean, max, median 등 사용
- 연관되어 있는 열 정보를 기준으로 값 채우기
  - `df[열1].fillna(df.groupby(열2)[열1].transform('median'), inplace = True)`
  - 열1: 결측값을 채울 열
  - 열2: 관련이 있는 열
  - `df.groupby(열2)`: 열2를 기준으로 그룹을 나누기
  - `df.groupby(열2)[열1].transform('median')`: 열2를 기준으로 그룹을 나눈 후 열1 선택하여 중앙값으로 채우기



## ○ 모델링

### ■ 사용하는 모델

- KNeighborsClassifier(n\_neighbors = k)
  - 가까운 k명의 이웃들의 생존 결과의 다수결 결과 사용
- DecisionTreeClassifier()
  - Decision tree 사용
- RandomForestClassifier(n\_estimators=k)
  - k 개의 작은 decision tree 사용
- GaussianNB()
  - Naïve Bayes 활용
- SVC(gamma = 'auto')
  - Support vector를 기준으로 생존 결과 결정

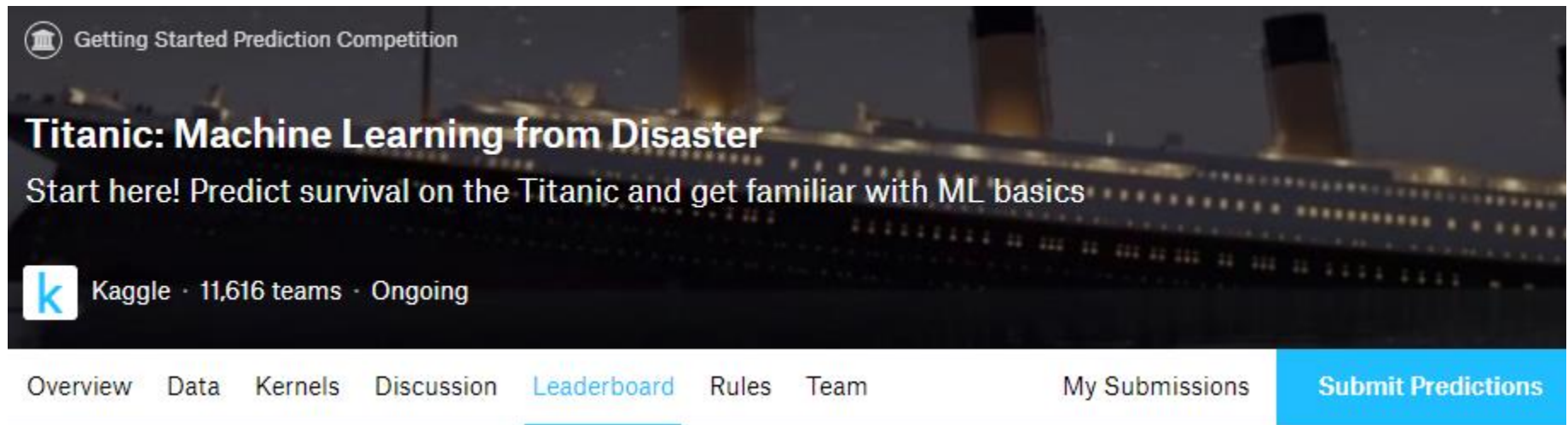
## ○ 모델링

### ■ 모델 적용 및 예측

- train용 데이터(train\_data)와 target 선택
- 모델 선택: `clf = 모델_함수(파라미터)`
- 데이터 적용: `clf.fit(train_data, target)`
- 예측: `clf.predict(test_data)`
- 모델의 평가: train용 데이터를 일부 나눠 학습과 검증용으로 활용

## ○ 제출용 파일 생성

- 하나 이상의 모델의 예측값 활용하여 제출용 파일 만들기
- PassengerId와 Survived(예측값)만 담아 파일 제출
- 파일의 제출: Submit Predictions 버튼을 눌러 제출




The screenshot shows the Kaggle competition page for 'Titanic: Machine Learning from Disaster'. The background is a dark image of the Titanic ship at night. The text on the page includes 'Getting Started Prediction Competition' with a small icon, the title 'Titanic: Machine Learning from Disaster', the subtitle 'Start here! Predict survival on the Titanic and get familiar with ML basics', the Kaggle logo, and the text 'Kaggle · 11,616 teams · Ongoing'. At the bottom, there is a navigation bar with links: 'Overview', 'Data', 'Kernels', 'Discussion', 'Leaderboard' (which is highlighted with a blue underline), 'Rules', 'Team', 'My Submissions', and a large blue button labeled 'Submit Predictions'.

Getting Started Prediction Competition

### Titanic: Machine Learning from Disaster

Start here! Predict survival on the Titanic and get familiar with ML basics

 Kaggle · 11,616 teams · Ongoing

Overview Data Kernels Discussion Leaderboard Rules Team My Submissions [Submit Predictions](#)

## ○ Feature engineering – Name

- Title 정보로 부터 성별, 결혼 여부 확인 가능
- Mr, Miss, Mrs 추출(그 외는 others로)
  - `str.extract('RE', expand = False)` 사용
  - `'([A-Za-z]+)\\.'`의 의미: 알파벳으로 시작해서 .으로 끝나는 단어
    - `[A-Za-z]`: 모든 알파벳
    - `+`: 1 or more
    - `.`: any character
    - `\\.`: comma(.)

## ◎ 교차 검증 (Cross validation)

- kFold 함수 사용
- 설정
  - `k_fold = KFold(n_splits=10, shuffle=True, random_state=0)`
  - `shuffle`: 쪼개기 전에 랜덤하게 섞기
  - `random_state`: random 수 생성 시 seed 값
- 교차검증
  - `cross_val_score(clf, train_data, target, cv=k_fold, n_jobs=1, scoring='accuracy')`
  - `scoring`: 반환할 값의 종류
    - 분류일 경우 `accuracy`
    - 회귀일 경우 `RMSE`(Root mean square error =  $\text{RSS}/\text{root}(\#)$ )

## ◎ 앙상블

- 5가지 모델의 예측 결과를 모두 활용