JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 28 No. 3, pp. 17-24, March 2023
https://doi.org/10.9708/jksci.2023.28.03.017

A Comparative Analysis of the Pre-Processing in the Kaggle Titanic Competition

Tai-Sung Hur*, Suyoung Bang*

*Professor, Dept. of Computer Science, Inha Technical College, Incheon, Korea
*Student, Dept. of Computer Science, Inha Technical College, Incheon, Korea

[Abstract]

Based on the problem of 'Tatanic – Machine Learning from Disaster', a representative competition of Kaggle that presents challenges related to data science and solves them, we want to see how data preprocessing and model construction affect prediction accuracy and score. We compare and analyze the features by selecting seven top-ranked solutions with high scores, except when using redundant models or ensemble techniques. It was confirmed that most of the pretreatment has unique and differentiated characteristics, and although the pretreatment process was almost the same, there were differences in scores depending on the type of model. The comparative analysis study in this paper is expected to help participants in the kaggle competition and data science beginners by understanding the characteristics and analysis flow of the preprocessing methods of the top score participants.

▸ **Key words**: Data Science, Machine Learning, Deep Learning, Algorithm, Kaggle

[요 약]

데이터 과학과 관련한 과제를 제시하고 참가자가 이를 해결하는 캐글(Kaggle)의 대표적인 대회인 'Tatanic – Machine Learning from Disaster' 문제를 기반으로 데이터 전처리 방식과 모델 구축이 예측 정확도와 점수에 어떤 영향을 미치는지 확인하고자 한다. 중복된 모델을 사용하였거나 앙상블 기법을 사용한 경우를 제외하고 높은 점수를 획득하여 상위 순위에 위치한 7건의 해결 방식을 선정하여 특징들을 비교 분석한다. 전처리를 진행하는 데 있어 대부분 고유하고 차별적인 특징을 가진 것을 확인하였으며, 거의 동일할 정도의 전처리 과정을 거쳤으나 모델의 종류에 따라 점수 차이가 존재하기도 하였다. 본 논문의 비교 분석 연구는 상위 점수 참가자의 전처리 방식의 특징과 분석 흐름을 이해함으로써 캐글 대회 참가자들과 데이터 과학 입문자들에게 많은 도움이 될 것으로 생각한다.

▸ **주제어**: Data Science, Machine Learning, Deep Learning, Algorithm, Kaggle

• First Author: Tai-Sung Hur, Corresponding Author: Suyoung Bang
*Tai-Sung Hur (tshur@inhatc.ac.kr) Dept. of Computer Science, Inha Technical College
*Suyoung Bang (swimmys2@naver.com), Dept. of Computer Science, Inha Technical College
• Received: 2023. 01. 20, Revised: 2023. 02. 28, Accepted: 2023. 02. 28.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

캐글(Kaggle)은 2010년 설립된 예측모델 및 분석 대회 플랫폼으로 기업 및 단체에서 데이터와 해결과제를 등록하면 데이터 과학자들이 이를 해결하는 모델을 개발하고 경쟁하게 된다[1]. 캐글의 대회 유형에는 일반적인 경쟁 유형, 기타 경쟁 유형이 있다. 일반적인 경쟁 유형에는 'Featured', 'Research', 'Getting Started', 'Playground' 4가지가 있다. 이 중 'Getting Started'는 가장 접근하기 쉬운 대회 유형으로 데이터 과학에 입문한 신규 이용자들을 위한 대회이다[2]. 'Getting Started' 대회는 지속적으로 진행되며, 문제 해결에 따른 보상은 주어지지 않는다. 이 중 가장 대표적인 대회가 'Titanic - Machine Learning from Disaster'이다.

'Titanic - Machine Learning from Disaster'는 타이타닉호 침몰 사건을 주제로 본 대회 참가자는 타이타닉호에 탑승한 승객의 이름, 나이 성별, 사회경제적 계층 등의 데이터를 활용하여 '어떤 부류의 사람들이 생존할 가능성이 더 높았느냐'라는 질문에 답하는 예측 모델을 구축하게 된다. 2022년 8월 10일 기준 14,398팀이 참여했으며, 이 대회의 평가 척도는 참가자가 정확하게 예측한 승객의 비율(정확도)이다.

본 연구에서는 타이타닉 대회의 상위 점수 참가자들이 작성한 커널을 토대로 데이터 전처리 방식과 머신러닝 모델 구축 방법을 비교하여 생존자 예측 정확도에 어떤 영향을 미치는지 확인하고자 한다. 대표 샘플을 기반으로 공개적으로 볼 수 있는 제출 점수인 public leaderboard(공개 리더보드)의 public score(공개점수)를 기준으로 하였다. 참가자가 본인의 전처리 및 모델 구축 과정을 공유한 25개의 커널을 일차적으로 수집하였다. 단, 상위 0.7% 이상의 커널들을 조사해 본 결과 작성된 코드 내에 알고리즘이 충분히 제시되지 않았거나, 제시된 알고리즘으로는 해석이 불가능한 경우는 제외하였다. 본 논문에서는 단일 모형을 사용한 경우만 비교하기 때문에 여러 모형을 복합적으로 결합하는 앙상블 기법을 사용하였거나 중복되는 머신러닝 모델을 가진 경우 또한 비교가 어려워 제외하였다. 결과적으로 1차 수집한 25팀 중 전처리와 머신러닝 모델의 기법을 달리한 7건의 커널을 선택하여 각 커널들의 특징을 비교분석하였다. 사용된 알고리즘은 랜덤포레스트(RandomForest), 의사결정나무(Decision Tree), MLP(Multi-Layer Perceptron), Light GBM, GradientBoosting, CatBoost이다. 커널들의 등수분포는 2022년 8월 6일 기준 195등부터 750등까지이며, public score(공개점수)의 분포는 0.81818점부터 0.79186점까지로 약 상위 5.6% 안에 드는 예측 결과를 담고 있다.

Table 1. Selected Kernel

index	Title	Algorithm	Public Score (rank/percent)
a.	Titanic KNN 2.0	KNN	0.81818 (195/1.36%)
b.	80.861% with RF+Mean encoding+BayesianOptimization	RandomForest	0.80861 (334/2.31%)
c.	notebookd6fe1ff5cb	Decision Tree	0.80622 (340/2.36%)
d.		MLP	0.80143 (469/3.25%)
e.	Titanic competition	LightGBM	0.79904 (524/3.63%)
f	titanic_1_	Gradient Boosting	0.79186 (703/4.88%)
g.	Prophet Titanic	CatBoost	0.79186 (730/5.07%)

본 논문의 구성은 다음과 같다. 제 2장에서는 타이타닉 대회에서 제공하는 데이터셋의 기본 feature에 대해 설명하고 제 3장에서는 선정한 커널별로 진행하는 데이터 시각화, feature engineering과 같은 전체적인 데이터 전처리 과정에 대해 알아본다. 제 4장에서는 7개의 노트북에서 사용한 각각 다른 알고리즘들에 대해 알아보고, 각 커널의 전처리 과정을 바탕으로 알고리즘의 생존자 예측 결과를 비교 분석한다. 마지막으로 결론을 제 5장에서 제시한다.

II. Preliminaries

캐글 타이타닉 대회(Titanic - Machine Learning from Disaster)에서는 이름, 나이, 성별, 사회경제적 등급 등과 같은 승객 정보를 포함하는 train(훈련)데이터셋과 test(시험) 데이터셋을 제공한다. train 데이터셋은 탑승한 승객에 대한 세부 정보와 생존 여부를 포함하여 총 891건의 승객 정보를 제시한다. test 데이터셋은 train 데이터셋에서 생존 여부를 제외한 418건의 데이터를 제시한다. train 데이터셋을 훈련한 모델에서 찾은 패턴을 test 데이터셋에 적용하여 승객 418명의 생존여부를 예측한다.

본 대회에서 제공하는 train 데이터는 Table 2에서 보는 바와 같이 12개의 feature로 이루어져있다. 'PassengerId'는 타이타닉호에 탑승한 승객들의 고유 번호를 의미한다. 'Survived'는 침몰 사고 이후 승객들의 생존 여부로, 0은 사망을 1은 생존을 나타낸다. 이는 train 데이터에만 존재하고 test 데이터에는 존재하지 않는다. 'Pclass'는 승객들의 티켓 등급이다. 1은 1등석, 2는 2등석, 3은 3등석을 나타내

며 이는 곧 사회, 경제적 지위(socioeconomic status, SES)를 의미한다. 'Name', 'Sex', 'Age'는 각각 승객의 이름, 성별, 나이이다. 'SibSp'는 Sibling과 Spouse의 합성어로 승객이 타이타닉호에 동반한 형제와 배우자의 수를 의미한다. 'Parch'는 Parents와 Children의 합성어로 승객이 타이타닉호에 동반한 부모와 자녀의 수를 의미한다. 다만, 부모가 아닌 사람과 함께 여행한 일부 아이들은 parch의 값이 0이 된다. 'Ticket'은 티켓의 고유 번호, 'Fare'은 티켓의 요금, 'Cabin'은 승객이 머물 객실의 번호이다. 'Embarked'는 승객이 타이타닉호에 승선한 항구를 나타내며 승객들은 Cherbourg의 'C', Queenstown의 'Q', Southampton의 'S' 총 3곳의 항구 중 한 곳에서 탑승한다.

Table 2. Titanic Dataset

Value	Definition
PassengerId	Passenger's unique number
Survived	Survival
Pclass	Ticket class
Name	Passenger's Name
Sex	Passenger's Sex
Age	Passenger's Age
SibSp	Siblings and Spouses
Parch	Parents and Children
Ticket	Ticket Number
Fare	Ticket Price
Cabin	Cabin number
Embarked	Port Information

타이타닉 train, test 데이터는 각각 3개의 feature에 결측치가 존재한다. 결측치를 처리하는 방법은 분석을 진행하는 사람에 따라 다르므로, 제 3장에서 각 커널들이 결측치를 포함한 데이터를 어떻게 전처리하는지 확인한다.

Table 3. Missing values of train and test data

Type	Missing value	Count (Percent)
Train Data	Age	177 (19.87%)
	Embarked	2 (0.22%)
	Cabin	687 (77.10%)
Test Data	Age	86 (20.57%)
	Embarked	1 (0.24%)
	Cabin	327 (78.23%)

III. The Proposed Scheme

1. Feature Engineering

a. 'Titanic KNN 2.0'[3]/ KNN(K-Nearest Neighbor)

a-1. Age 결측치 대체

Table 4. Mapping Name title of Titanic KNN 2.0

Mapping Title	Name Title
Mr	Mr, Major, Col, Sir, Don, Jonkheer, Capt, Dona
Mrs	Mrs, Lady, Countess
Miss	Miss, Mlle, Mme, Ms
Dr	Dr
Rev	Rev

정확하게 나이 결측치를 채우기 위하여 먼저 승객의 이름에서 승객의 수식어(title)을 가져온다. 17개의 수식어를 총 다섯 개의 title로 연결시킨 후 분류한다. 각 title별 나이의 중앙값을 구해 나이가 결측치인 승객의 title에 맞는 나이값을 채운다. 'title' feature는 나이의 결측치를 채우기 위해 만든 것이지 모델링 과정에 사용할 feature가 아니므로 삭제한다.

a-2. Family_Size

동승한 부모와 자녀의 수를 나타내는 'Parch'와 동승한 형제자매와 배우자 수를 나타내는 'SibSp'를 합쳐 총 가족 수를 나타내는 feature인 'Family_Size'를 새로 만든다.

a-2-1. Family_Survival

승객들의 데이터를 확인해보면 같은 성을 가지고, 같은 요금을 지불한 사람들이 가족이라고 유추할 수 있다. 대다수의 가족들이 함께 죽거나 모두 살았다는 점에서 생존한 가족 자체의 생존에 대한 여부를 하나의 feature로 사용한다. Family_Survival의 초기값은 0.5으로 시작한다. 가족이 모두 사망한 경우 즉, 가족 구성원 모두의 'Survived'가 0이면 Family_Survival은 0.0이 되며, 한 명이라도 생존한 경우 1.0이 된다. 가족이 아닌 동승자가 있을 수도 있다는 점을 고려해 같은 성이 아니더라도 같은 티켓 번호를 가지고 같은 요금을 지불하는 사람들은 함께 여행을 온 사람들이며, 가족과 마찬가지로 서로 도우면서 동시에 생존하거나 사망하였을 것이라고 예상한다. 따라서 위에서 거쳤던 과정을 동일하게 진행한다. Family_Survival의 정보를 가지고 있는 승객의 수는 총 546명이다.

a-3. FareBin & AgeBin

Fare 데이터의 중앙값으로 결측치를 대체한 후 데이터를 동일한 개수로 5개의 구간으로 나눈다. 요금을 작은 수에서 큰 수로 구간화한 것이므로 순서형 데이터이기 때문에 라벨 인코딩을 진행한다. Age 또한 데이터를 동일한 개수로 4개의 구간으로 나눈 다음 라벨 인코딩을 진행한다. 이후 Fare과 Age feature는 삭제한다.

a-4. Sex

성별의 남성(Male)은 0으로, 여성(Female)은 1로 인코딩한다.

b. '80.861% with RF+Mean encoding+BayesianOpti

mization'[4]/ 랜덤포레스트(RandomForest)

b-1. EDA(Exploratory Data Analysis, 탐색적 데이터 분석)

b-1-1. Name

승객의 이름에서 title만 따로 가져온 후 각각의 수식어들을 각각 고유의 값을 갖도록 유지한다. Table 5과 같이 이름의 길이를 5개의 구간으로 나눠 이름의 길이에 따른 평균 생존 여부를 확인해 보았을 때 이름의 길이가 길수록 생존 가능성이 높아진다는 것을 확인하였다.

Table 5. Survival rate by Length of Name

Name Length	Survival Rate
(11.999, 19.0]	0.220588
(19.0, 23.0]	0.301282
(23.0, 27.0]	0.319797
(27.0, 32.0]	0.442424
(32.0, 82.0]	0.674556

b-1-2. Sex

타이타닉호에 탑승한 승객 중 여성 승객은 0.352413, 남성 승객은 0.647587로 남성 승객의 수가 더 많은 것을 알 수 있었으며 성별에 따른 생존율을 확인한 결과, 여성 승객의 평균 생존율은 0.742038, 남성 승객의 평균 생존율은 0.188908이었다.

b-1-3. Age

Age에 결측치가 있는 데이터들의 평균 생존율이 약 0.293785였으며, 결측치가 없는 데이터들과 비교 했을 때 약 10%가 낮은 생존율을 보였다. 결측치 처리를 하기 전, 이런 특성을 설명할 수 있도록 Age_null flag feature를 만들어 포함시킨다.

b-1-4. Fare티켓의 요금에 따라 승객의 티켓 등급이 달라질 것이기 때문에 Fare과 Pclass 데이터의 분포와 양상이 유사할 것이며 이는 곧 생존 여부와도 분명한 관계가 있다. 실제로, Fare을 네 구간으로 나눈 후 Fare별 Pclass 별 탑승객 분포인 Table 6를 확인해 봤을 때 요금이 39.688 이상 512.329 미만이면 1등급에 해당하는 승객이 146명으로 가장 많으며 요금이 10.5 미만일 때 3등급에 해당하는 승객이 총 327명으로 3등급 승객이 가장 많이 분포해 있는 것을 볼 수 있었다.

Table 6. Distribution of Pclass according to Fare

Fare/Pclass	1	2	3
(-0.001, 7.854]	6	6	167
(7.854, 10.5]	0	24	160
(10.5, 21.679]	0	80	92
(21.679, 39.688]	64	64	52
(39.688, 512.329]	146	10	20

b-1-7. Cabin

Cabin은 결측치가 약 700개로 많은 편이지만 그 안에서 추출할 수 있는 정보들이 있다. 먼저, Cabin Letter은 각 객실의 첫 번째 문자를 가져온 것이다. Cabin Letter의 종류에는 A, B, C, D, E, F, G, T, n이 있다. 두 번째는 Cabin Number이다. Cabin Letter 뒤에 붙는 Cabin Number을 따로 떼어내 동일한 개수로 3개의 구간으로 나눈 후 이에 따른 평균 생존율을 확인했을 때 1.999부터 28.667까지 첫 번째 구간은 0.716418, 이후부터 65.667까지 두 번째 구간은 0.651515, 마지막으로 148.0까지의 세 번째 구간은 0.641791이었다. Cabin Number에 따른 생존율이 높은 수치를 나타내므로 결측치가 많아도 유지하도록 결정한다.

b-1-8. Embarked

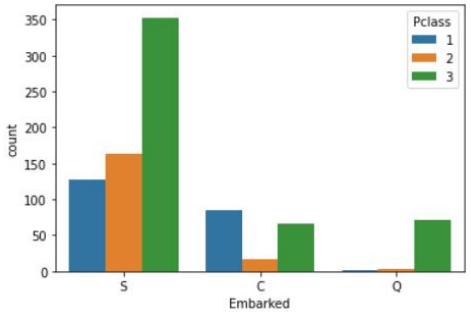

Fig. 1. Pclass distribution according to Embarked

'C(Cherbourg)'에서 탑승한 사람들이 다른 출항지보다 생존율이 20% 정도 높았다. Fig. 1에서 Pclass 별 Embarked의 분포를 확인해보았을 때 'C'에서 탑승한 상류층 승객들 때문일 것이라고 예상한다.

b-2. feature engineering

b-2-1. Name_Title&Name_Len

승객의 이름에서 수식어(title)을 추출해 'Name_Title'로, 이름의 총 길이를 추출해 'Name_Len'이라는 이름의 feature로 저장한다.

b-2-2. Age_Null_Flag

승객 이름의 수식어와 Pclass 별 나이의 평균을 구해 나이의 결측치를 채운다.

b-2-3. Fam_Size

부모와 자녀의 수를 나타내는 Parch와 형제와 배우자의 수를 나타내는 SipSp를 결합해 Fam_Size로 결합한 후, Fam_Size가 0이면 'solo', 3보다 작거나 같으면 'Nuclear', 그 이상은 'big'이라는 세 가지 범주로 나눠 그룹화를 진행한다.

b-2-4. Ticket_Lett&Ticket_Len

첫 글자와, TIcket의 총 길이를 분리하여 새로운 'Ticket_Lett'와 'Ticket_Len' feature를 만들어준다.

b-2-5. Cabin_Letter&Cabin_num

Cabin은 예를 들어 'A123'처럼 알파벳+숫자의 형태로 이루어져 있다. Ticket_Lett와 동일한 방법으로 알파벳과 숫자를 분리하여 Cabin_Letter, Cabin_num을 만들어준다. 그 숫자 값들을 동일한 개수로 3개의 구간으로 나눈다. 이때, NaN 결측치 값들은 첫 알파벳을 떼어내 추출하면 숫자가 아닌 'an'이라는 글자가 추출되는데 이 값은 다시 NaN으로 대체한다.

b-2-6. Embarked

Embarked의 결측치는 train 데이터셋에서 가장 많이 나타나는 값인 'S'로 대체한다.

c. 'notebookd6fe1ff5cb'[5]/ 의사결정나무(Decision Tree).

EDA 과정에서는 데이터의 통계량과 결측치, 데이터의 타입 및 전체적인 정보를 확인한다. feature engineering 과정은 첫 번째 커널과 동일하게 진행한다.

d. 'temp_1'[6]/ MLP(Multi Layer Perceptron)

d-1. 결측치 대체

데이터 상 결측치가 존재하는 'Embarked', 'Fare', 'Age'에 각각 최빈값, 중앙값, 중앙값으로 결측치를 대체한다.

d-2. Title

승객의 이름에서 수식어(title)을 추출한다. 추출한 18개의 수식어들을 큰 4가지의 범주로 묶어 새로운 feature를 만든다. 'Mr'은 0, 'Miss'는 1, 'Mrs'는 2, 그리고 'Master', 'Dr', 'Rev' 등 나머지 소수의 수식어들은 3으로 대응시킨다.

d-3. Sex

성별의 남성(Male)은 0으로, 여성(Female)은 1로 대체한다.

d-4. Embarked

'S' 항구는 0, 'C' 항구는 1, 'Q' 항구는 2로 대체한다.

d-5. Age & Fare

Age는 0세 초과 12세 이하, 12세 초과 20세 이하, 20세 초과 40세 이하, 40세 초과 120세 이하 총 4개의 범위로 구간화 한다. Fare은 17달러 이하, 17달러 초과 30달러 이하, 30달러 초과 100달러 이하, 100달러 초과 600달러 이하 총 4개의 범위로 구간화 하고 각각을 0, 1, 2, 3으로 Label Encoding[7]한다.

d-6. Cabin

알파벳+숫자의 형태로 이루어진 Cabin의 첫 글자 알파벳을 숫자와 분리 후 실수값와 대응시킨다. 'A'는 0, 'B'는 0.4, 'C'는 0.8, 'D'는 1.2, 'E'는 1.6, 'F'는 2, 'G'는 2.4, 'T'는 2.8로 대응시킨다. Cabin의 결측치는 앞서 실수로 대응시킨 값들의 중앙값으로 대체한다.

d-7. Family_Size

부모와 자녀의 수를 나타내는 Parch와 형제와 배우자의 수를 나타내는 SipSp를 결합해 새로운 feature로 생성한다.

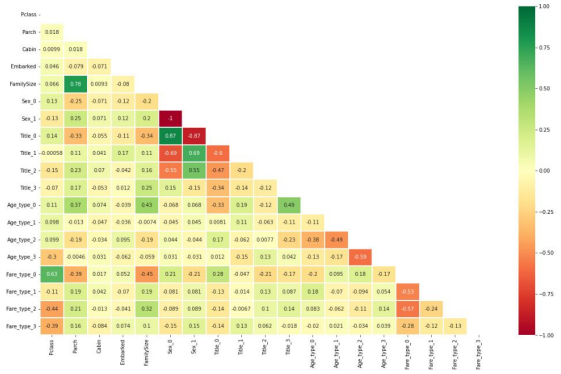

Fig. 2. correlation coefficient between features

d-8. Fig. 2와 같이 최종 데이터셋의 feature들 간 상관계수를 확인한다.

e. 'Titanic competition'[8]/ LightGBM

e-1. 결측치 대체

다섯 번째 커널은 train 데이터와 test 데이터를 결합한 상태에서 진행한다. 결합한 데이터의 결측치의 구체적인 값은 Table 7와 같다.

Table 7. Value and percentage of missing values for combined train and test data

Value	NaNs	Percent	Type
Fare	1	0.08	float64
Embarked	2	0.15	object
Age	263	20.09	float64
Survived	418	31.93	float64
Cabin	1014	77.46	object

LightBGM 모델 생성 후 존재하는 전체 데이터를 활용해 해당 feature의 결측치 값을 예측하여 대체하는 방법을 선택했다. 결측치가 존재하는 다섯 가지 feature 중 'Cabin'은 총 데이터 1309건 중 1014건이 결측치로, 결측치의 값이 전체의 70% 이상이기 때문에 예측값을 넣어 결측치를 대체하는 방법에는 적합하지 않다고 판단해 제외한다. 'Embarked', 'Fare', 'Age', 'Survived' 네 개의 feature의 결측치를 예측하고 대체한다. 'Age', 'Survived', 'Cabin' 세 개의 feature은 결측치가 10%이상이기 때문에

모델을 생성하고 예측하는 데이터로서는 사용하지 않는다.

e-2. Age Category

탑승자의 나이를 5로 나누어 각각의 값을 Age Category로 생성한다. 예를 들어 첫 번째 탑승자의 나이는 22세이고 5로 나누어 4라는 값을 얻어 그 값 자체를 하나의 범주로 사용한다.

e-3. Ticket

'문자+숫자'의 형태로 이루어져 있는 Ticket feature를 문자와 숫자로 분할한다. 문자부분에서 생기는 결측치는 'without'으로 채운다.

f. 'titanic_1_'[9]/ GradientBoosting

f-1. 결측치가 많은 데이터 삭제

결측치가 데이터의 많은 부분을 차지하고 있는 'Cabin', 'Name', 'Ticket' feature를 삭제한다.

f-2. Age, Fare, Embarked 결측치 대체

Pclass와 Sex(성별)에 따른 나이를 구한 후, 그 값의 중앙값으로 결측치를 대체한다. 'Fare' 또한 'Age'와 동일하게 Pclass와 성별에 따른 요금값의 중앙값으로 결측치를 대체한다. 'Embarked'는 결측치가 존재하는 행을 삭제함으로서 결측치를 처리한다.

f-3. Sex

성별의 남성(Male)은 0으로, 여성(Female)은 1로 대체한다.

f-4. Embarked

'S' 항구는 0, 'C' 항구는 1, 'Q' 항구는 2로 대체한다.

기본 feature 이외에 더 이상의 feature을 추가적으로 생성하지 않고 진행한다.

g. 'Prophet Titanic'[10]/ CatBoost

g-1. 결측치 대체

'Age'는 전체 나이의 중앙값으로 결측치를 대체한다. 'Embarked'와 'Cabin'은 'U'로 결측치를 대체한다. 'Fare'은 0.0으로 결측치를 대체한다.

g-2. Sex

성별의 남성(Male)은 0으로, 여성(Female)은 1로 대체한다.

g-3. Embarked

'S' 항구는 0, 'C' 항구는 1, 'Q' 항구는 2로 대체한다.

1-4. Cabin

A는 0, B는 1, C는 2, D는 3, E는 4, F는 5, G는 6, T는 7, U는 8로 각각 대체한다.

추가적인 feature 생성은 하지 않고 진행한다.

2. Delete Feature

7개의 커널은 공통적으로 생존자를 예측하는 모델을 만들 때 도움이 되지 않아 사용하지 않을 feature들을 제거하여 최종 데이터셋을 결정한다.

Table 8. Features to delete

Index	Delete Feature
a.	Name, PassangerId, SibSp, Parch, Ticket, Cabin, Embarked, Fare, Age
b.	Name, PassangerId, SibSp, Parch, Ticket, Cabin
c.	Name, PassangerId, SibSp, Parch, Ticket, Cabin, Embarked, Fare, Age
d.	Age, Name, Fare, Ticket, PassengerId, SibSp
e.	Age, CabinLevel
f.	Cabin, Name, Ticket, Age, Fare, PassengerId, SipSp
g.	Name, PassengerId, Ticket

IV. Applied algorithm

1. Algorithm

앞서 소개된 캐글 커널은 총 7개로 각기 다른 단일 알고리즘을 사용한 커널들을 선정했기 때문에 알고리즘 또한 7가지로 다음과 같다.

1.1 KNN(K-Nearest Neighbor)

예측하려는 데이터 X가 주어졌을 때 기존 데이터 중 속성이 비슷한 K개의 이웃을 먼저 찾는 방식이다. 데이터 X를 둘러싼 K개의 가장 가까운 이웃을 찾고, 이웃 데이터가 가장 많이 속해 있는 목표 클래스를 예측값으로 결정하는 알고리즘이다. K의 값에 따라 KNN 모델이 예측하는 클래스가 달라지므로 적정한 K값을 설정해야 한다[11].

1.2 RandomForest

의사결정나무 모형을 다수 만들어 더 정확한 예측을 하는 것이 목적이다. 부트스트랩 표본을 다수 생성하고 의사결정나무 모형을 적용하여 그 결과를 종합하는 앙상블 방법(ensemble methods)으로, 무작위성이 더해지기에 의사결정나무의 수가 증가할수록 예측 오차가 줄어든다. 높은 예측력과 train data와 test data로 나누어 모형 타당화를 시도할 필요가 없다는 점이 장점으로 꼽힌다[12].

1.3 Decision Tree

트리(tree) 구조를 사용하고 각 분기점(node)에는 분석 대상의 속성들이 위치하는 방식이다. 각 분기점에서 최적의 속성을 선택할 때, 해당 속성을 기준으로 분류한 값들이 구분되는 정도를 측정한다. 다른 종류의 값들이 섞여있는 정도가 낮을수록 분류가 잘 된 것이다[13].

1.4 MLP(Multi Layer Perceptron)

입력층(input Layer)과 출력층(output Layer)으로만 이루어져 있는 퍼셉트론의 문제점을 보완한 인공신경망으로 입력층과 출력층 사이에 은닉층(hiddn Layer)를 가지고 있는 신경망이다[14].

1.5 LightGBM

GB(Gradient Boosting)을 기반으로 하는 프레임워크로, 빠른 학습 속도를 가지고 있으며 정밀도, 모델 안정성 및 컴퓨팅 효율성 측면에서 좋은 성능을 보이는 알고리즘이다[15].

1.6 GradientBoosting

약한 예측 모델들의 앙상블 형태로 예측 모델을 제공하는 알고리즘이다. 경사 하강법과 부스팅이 합쳐진 개념으로서, 여러 개의 모델이 순차적으로 학습을 진행하면서 이전 모델의 오차에 가중치를 부여하는 형태로 손실 함수를 최소화하면서 학습이 이루어진다[16].

1.7 CatBoost

GradientBoosted Decision tree를 기반으로 하며 훈련하는 동안 decision tree 세트가 연속적으로 구축되는 방식이다[17]. 범주형 feature에 대한 처리, 빠른 GPU 훈련, 대칭 트리 사용 등이 가능하다[18].

Table 9. Hyperparameters by algorithm

Index	Algorithm	Value
a.	KNN	algorithm='auto', leaf_size=26, metric='minkowski', metric_params=None, n_jobs=1, n_neighbors=6, p=2, weights='uniform'
b.	Random Forest	max_depth=8, n_estimators=905, min_samples_split=14, min_samples_leaf=1, max_features=0.3964, oob_score=True, random_state=42, n_jobs=-1
c.	Decision Tree	criterion='entropy', max_depth= 9, min_samples_split= 8, max_features= 'auto', splitter= 'random'
d.	MLP	BATCH_SIZE=16, optimizer=torch.optim.SGD(model.parameters(), lr=0.01), error=BCELoss(), EPOCHS=1000
e.	LightGBM	subsample=0.9, num_leaves= 750, n_jobs= -1, n_estimators= 2000, min_split_gain=0.01, min_child_samples=5, max_depth=5, learning_rate=0.01, lambda_l2=0.1, lambda_l1=0.1, feature_fraction=0.3, bagging_seed=16, bagging_fraction=0.7
f.	Gradient Boosting	min_samples_split=20, min_samples_leaf=60, max_depth=3, max_features=7
g.	Catboost	loss_function='Logloss', eval_metric='Accuracy', depth=4, l2_leaf_reg=1, iterations=150, learning_rate=0.1

2. Parameter setting according to algorithm

최적의 결과를 도출하기 위한 하이퍼파라미터 설정 과정을 진행한다. 각 알고리즘에서 설정한 하이퍼파라미터의 값은 Table 9와 같다.

최적의 하이퍼 파라미터를 찾기 위해 여러 가지 방법이 존재한다. a, b, g는 GridSearch를 통해 최적의 하이퍼 파라미터를 찾았으며 다섯 번째 커널인 e에서는 Randomized Search를 활용했다. c는 Bayesian Optimization을 통해 구했으며 나머지는 작성자가 직접 수동으로 조정하며 최적의 하이퍼파라미터를 선정했다.

3. Prediction result score by algorithm

앞선 데이터 전처리와 여러 방법을 통해 찾은 최적의 하이퍼파라미터에 따른 알고리즘별 타이타닉 대회 점수 결과는 Table 10과 Fig. 3에서 보는 바와 같다.

Table 10. Results of Titanic competition public scores by algorithms

Index	Title	Algorithm	public score
a.	Titanic KNN 2.0	KNN	0.81818
b.	80.861% with RF+Mean encoding+BayesianOptimization	Random Forest	0.80861
c.	notebookd6fe1ff5cb	Decision Tree	0.80622
d.	temp_1	MLP	0.80143
e.	Titanic competition	LightGBM	0.79904
f.	titanic_1_	GradientBoosting	0.79186
g.	Prophet Titanic	Catboost	0.79186

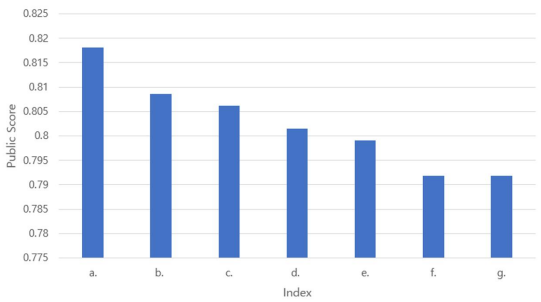

Fig. 3. Results of Titanic competition public scores by algorithms

V. Conclusions

본 논문에서는 1912년 발생한 타이타닉호 침몰 사건을 주제로 한 대회인 'Titanic - Machine Learning from Disaster'를 다룬다. 앙상블 기법, 중복된 모델을 제외하고 전처리와 머신러닝 모델의 기법을 달리한 상위 7건을 선택하여 각 커널들의 특징을 비교분석함으로써 캐글 타이타닉 대회에 참가하는 참가자들이 데이터 전처리의 특징과 분석 흐름을 이해하고 고득점을 취득할 수 있을 유용한 정보를 제공하는데 목적을 두었다.

첫 번째 커널은 7개의 커널들 중 본 대회에 대한 체계적인 학습이 가능한 튜토리얼 방식을 띄며 가족 자체의 생존에 대한 여부인 'Family_Survival'과 과감한 feature 삭제가 특징적이다. 두 번째 커널에서는 결측치가 많은 'Cabin'에서 숫자만 따로 분리하여 구간을 나눈 후 평균 생존율을 확인해 생존과 Cabin과의 관계를 파악한다. 세 번째 커널은 데이터 전처리 과정에 있어서는 첫 번째 커널을 참고하여 별다른 특징이 없지만 오히려 동일한 전처리 과정을 진행하고 다른 알고리즘을 사용했을 때 public score에 어떤 차이가 존재하는지 확인할 수 있다. 네 번째 커널은 7건 중 유일하게 딥러닝 과정을 진행했다는 점, 다섯 번째 커널은 결측치 대체에 있어 예측 모델을 생성했다는 점이 특징적이다. 마지막 두 개는 추가 feature를 생성하지 않고 기본으로 주어진 feature들만을 가지고 진행하여 다른 커널들에 비해 간결하다. 각 커널들은 생존자 예측에 앞서 데이터 전처리 시 참고할 만한 몇 가지 특징을 지닌다.

본 논문에서 진행한 비교 분석 연구가 캐글 타이타닉 대회 참가자들과 데이터 사이언스 입문자들에게 많은 도움이 될 것으로 생각된다.

REFERENCES

[1] Kaggle, https://en.wikipedia.org/wiki/Kaggle

[2] How to Use Kaggle, https://www.kaggle.com/docs/competitions

[3] Titanic KNN 2.0, https://www.kaggle.com/code/nehalgordhan/titanic-knn-2-0/notebook?scriptVersionId=97830121

[4] 80.861% with RF+Mean encoding+BayesianOptimization, https://www.kaggle.com/code/xavier001/80-861-with-rf-mean-encoding-bayesianoptimization/notebook?scriptVersionId=78369436

[5] Notebookd6fe1ff5cb, https://www.kaggle.com/code/rossanneadams/notebookd6fe1ff5cb/notebook?scriptVersionId=97927669

[6] Temp_1, https://www.kaggle.com/code/quandang1210/temp-1/notebook?scriptVersionId=99981538

[7],[11] S. H. Oh, "*Python Deep Learning Machine Learning Introduction*", Information Publishing Group, pp.90, pp.167, 2021.

[8] Titanic competition, https://www.kaggle.com/code/artyomkolas/titanic-competition/notebook?scriptVersionId=48346776

[9] Titanic_1_, https://www.kaggle.com/code/akshayr009/titanic-1/notebook?scriptVersionId=82013284

[10] Prophet Titanic, https://www.kaggle.com/code/mirfanazam/prophet-titanic/notebook?scriptVersionId=98443252

[12] Yoo Jin Eun, "Random Forest: Data Mining Techniques as an Alternative to Decision Trees", Journal of Educational Evaluation, VoL 28, No. 2, pp. 427-448, June 2015.

[13] S. H. Oh, "*Python Machine Learning Pandas Data Analysis*", Information Publishing Group, pp.323-324, 2019.

[14] In Gook Chun, "*Deep Learning EXPRESS*", Life and Power Press, pp.183,190-191, 2021.

[15] Yan, J., Xu, Y., etc. "LightGBM: accelerated genomically designed crop breeding through ensemble learning", Genome biology, vol. 22, pp. 1-24, 2021, DOI:10.1186/s13059-021-02492-y

[16] Jihye Kim, Soo Jin Lee, "Darknet Traffic Detection and Classification Using Gradient Boosting Techniques", Journal of the Korea Institute of Information Security & Cryptology, 32(2), pp. 371-379, 2022.

[17] How training is performed, https://catboost.ai/en/docs/concepts/algorithm-main-stages

[18] CatBoost, https://en.wikipedia.org/wiki/Catboost

Authors

Tai-Sung Hur received the B.S degree in Dept. of Computer Science from Inha University in 1984, and M.S degree in Dept. of Computer engineering from Soongsil University in 1987, and Ph. D. degree in Dept. of Computer engineering from Inha University in 1992. Dr. Hur has over 35 years of computer education. He is currently a Professor in the Dept. of Computer Science, Inha Technical College. He is interested in Data Science, Big data, Database and Internet of Things.

Suyoung Bang received B.S. degree in 2023 from the Department of Computer Science Inha Technical College, Incheon Korea. Her research interests include Data Science, Big data and Data engineering.