데분기 최종 보고서

도전! 데이터 분석 Project (370조)

EDA, numpy 및 pandas 활용 분석 및 결측치 처리, 그리고 시각화

(조장) 2020312223 김나연
2022314484 김서윤
2022312233 남다현

1. 데이터 분석 목적

승객들이 Transported 되는 데 영향을 미치는 변수들을 관찰하고자 합니다. 주어진 train 데이터셋을 사용해 'Transported' 열의 값을 예측하는 모델을 개발하고자 하였습니다. 이를 test 데이터셋에 적용하여 항해 중에 다른 차원으로 이동될 가능성이 있는 승객들을 식별할 수 있습니다. 결론적으로, 항해 중 다른 차원으로 이동될 가능성이 높은 승객들을 특정하여 재난 상황에서 재난 약자가 될 가능성이 높은 집단을 파악하고, 해당 승객들에 대한 안전 및 보안 절차를 개선할 수 있습니다. 미래의 우주여행에서 유사한 상황이 발생할 경우, 위험을 예측하고 그에 따른 정책 및 계획을 수립하여 대비할 수 있기를 기대합니다.

2. 데이터 분석 과정

2.1. 사용한 데이터셋

https://www.kaggle.com/competitions/spaceship-titanic -> test.csv, train.csv

이 데이터셋은 2919년을 배경으로, 우주선 타이타닉이 새로운 행성으로 이민 중이던 중 시공간 이상과 충돌하여 거의 절반의 승객이 다른 차원으로 이동되었다는 상황을 다룹니다.

train.csv: 이 파일은 약 2/3인 대략 8,700명의 승객에 대한 개인 기록을 포함하며, 훈련 데이터로 사용됩니다.
test.csv: 이 파일은 나머지 약 1/3, 대략 4,300명의 승객에 대한 개인 기록을 포함하며, 테스트 데이터로 사용됩니다.
PassengerId: 각 승객에 대한 고유한 식별자입니다. 각 ID는 gggg_pp 형식을 취하며, 여기서 gggg는 승객이 함께 여행 중인 그룹을 나타내고 pp는 해당 그룹 내 승객의 번호입니다. 그룹 내 사람들은 주로 가족 구성원이지만 항상 그렇지는 않습니다.
HomePlanet: 승객이 출발한 행성으로, 일반적으로 그들의 영구적인 거주 행성입니다.
CryoSleep: 항해 동안 부적절한 행동으로 인해 객실 안으로 이동이 제한된 승객을 나타내는 지표입니다.
Cabin: 승객이 머무는 객실 번호입니다. deck/num/side 형식을 취하며, 여기서 side는 Port를 나타내는 P 또는 Starboard를 나타내는 S로 표현됩니다.
Destination: 승객이 하차할 행성입니다.
Age: 승객의 나이입니다.
VIP: 승객이 항해 중에 특별한 VIP 서비스를 지급했는지 여부를 나타냅니다.
RoomService, FoodCourt, ShoppingMall, Spa, VRDeck: 승객이 Spaceship Titanic의 여러 고급 편의 시설에서 청구한 금액입니다.

Name: 승객의 이름과 성입니다.
Transported: 승객이 다른 차원으로 이동되었는지를 나타냅니다. 이것은 종속변수로, 예측하려는 대상인 열입니다.

2.2. 결측치 처리 및 데이터 분석

- Home Planet

가장 먼저 'Home Planet' 열을 기준으로 데이터프레임을 그룹화하고, 각 그룹의 인덱스를 행성 이름 별로 매핑하여 최대 인덱스가 저장된 딕셔너리 max_indexes를 생성했습니다. 그다음, 각 그룹의 인원수를 계산하여 group_sizes 시리즈에 저장했습니다.

또한, 생존율을 계산하고 정렬하였습니다. 각 그룹에서 'Transported' 열의 평균값을 계산하여 생존율을 나타내는 homeplanet_missing 시리즈를 생성하고 생존율에 따라 정렬하였습니다. 그리고 'Home Planet' 열의 결측치를 최빈값으로 채웠습니다. 'Home Planet' 열의 데이터는 모두 행성의 이름으로, 평균, 최빈값, 중앙값 등의 대푯값 중에서 자료가 숫자로 주어지지 않았을 때도 산출할 수 있는 최빈값을 활용하였습니다. fillna() 메서드를 사용하여 결측치를 채우며, 결과는 데이터프레임에 반영됩니다.

그다음 코드는 데이터프레임에서 여전히 남아있는 결측치의 개수를 확인하는 코드입니다. 이 코드를 활용하여 결측치가 무사히 잘 채워졌음을 확인하고, 결측치를 최빈값으로 채운 후의 생존율을 다시 확인해 보았습니다. 결측치 처리가 생존율에 미치는 영향을 살펴보았지만, 큰 변동이 없음을 확인한 후 바 차트로 시각화하여 사라진 승객의 비율을 시각적으로 확인할 수 있도록 하였습니다.

- VIP

VIP 여부에 영향을 미치는 변수로 'RoomService, FoodCourt, ShoppingMall, Spa, VRDeck'을 고려하였습니다. VIP 여부에 따라 이러한 서비스 이용에 대한 소비 금액이 차이가 있을 것으로 가정해 보았습니다. 즉, VIP 고객은 이러한 서비스를 이용하는 데 더 많은 돈을 지출할 것으로 예상하였습니다. 따라서 먼저 이 5가지 서비스와 관련된 결측치를 채우기 위해 VIP가 아닌 고객과 VIP 고객 각각에 대해 이 서비스를 이용한 평균 소비 금액을 대입하였습니다. 이후에는 VIP 열의 결측치를 채우는 데에 집중했습니다. 총체적 서비스 이용 금액의 평균 이상을 기록하는 승객만 VIP 열의 값을 true로, 나머지는 false로 처리하였습니다. 그 후 VIP의 여부에 따른 실종률을 비교해 봤더니 VIP에 속한 사람들의 실종률이 0.36, 속하지 않은 사람들의 실종률이 0.5로 나타나 VIP들의 실종률이 더 낮음을 확인할 수 있었습니다.

- CryoSleep

먼저, 'CryoSleep' 열의 데이터를 처리하고 해당 열에서 True와 False 값을 분석했습니다. 'CryoSleep' 열의 값을 NumPy 배열로 변환하고, True와 False 값을 가진 행을 찾아 해당 인덱스를 기록했습니다. 이렇게 얻은 정보로 'CryoSleep' 열에서 True와 False 값을 가진 사람들의 수를 파악하였습니다. 이로써 감금 여부가 생존율에 얼마나 영향을 미치는지 확인하였습니다.

두 번째로, 결측치 처리를 위한 코드를 작성하였습니다. 이 과정에서 VIP인 사람들은 감금되지 않으리라고 가정하였으며, 이를 기반으로 'CryoSleep' 열의 결측치를 'VIP' 열을 기준으로 채우려고 하였습니다. 'VIP' 열을 기준으로 데이터를 그룹화하고, 각 그룹에서

'CryoSleep' 열의 평균을 계산하여 그 값을 'CryoSleep' 값에 따라 오름차순으로 정렬한 데이터프레임을 생성하였습니다. 'CryoSleep' 열의 결측치인 행을 확인하고 해당 행의 'CryoSleep' 값을 결정하기 위해 'VIP' 열을 검사하며, VIP가 아닌 경우에만 'CryoSleep' 열의 값을 True로 설정했습니다. 이후 진행한 분석 결과에서 'CryoSleep' 열의 값이 결측치를 채우기 전과 큰 변동을 나타내지 않는다는 것을 확인하여 VIP와 CryoSleep간의 관계에 관련된 가정에 힘을 실어주었습니다.

마지막으로 'VIP' 열의 값에 따라 'CryoSleep' 열의 분포를 시각화하는 작업을 수행했습니다. 두 개의 원그래프를 나란히 그려 VIP인 사람의 경우 90.2%의 확률로 감금이 되지 않았고, VIP가 아닌 사람의 경우 38.3%의 확률로 감금이 되어, VIP가 아닌 사람이 VIP인 사람보다 약 4배의 확률로 더 감금되었음을 시각적으로 확인할 수 있었습니다.

- Cabin

Cabin 열의 결측치의 경우, 199개가 존재합니다. 먼저 같은 그룹에 속한 사람들이 같은 곳에서 묵을 것이라는 가정하에 위 결측치를 채우고자 했습니다. passengerId 열의 결측치를 채우는 과정에서 생성한 maxIndex 열의 값이 2 이상, 즉 싱글이 아닐 때 그 사람이 속한 그룹이 묵고 있는 Cabin 값과 같은 값을 결측치에 채웠습니다. 이러한 과정을 바탕으로 100개의 결측치를 채우는 데 성공했으며, 남은 99개의 결측치는 Cabin 값의 형식을 활용하여 처리하였습니다. Cabin 열의 값은 deck(갑판 위치)/방 번호/side(배의 방향) 형식으로 이루어져 있으므로 문자열을 인덱싱하는 방식을 이용하여 deck 과 side의 값을 각각 추출하여 새로운 열을 생성하였습니다. 그 후 deck과 side 별 생존율을 확인하였는데, deck의 경우 B>C>G>A>F>D>E>T 순으로 생존율이 낮은 것을 확인할 수 있었으며 side의 경우 port에 탑승한 승객들의 생존율이 더 높은 것을 확인할 수 있었습니다. 마찬가지로 cabin과 side에 각각 99개의 결측치가 존재하고 있을 것입니다. cabin의 경우 99개의 값이 갑작스레 늘어나도 cabin 간의 실종률 순서에 영향을 주지 않을 F의 값으로 결측치를 채웠고, side의 경우 균형을 맞추기 위해 결측치에 실종률이 더 낮은 P를 할당했습니다. 위 사항들의 그래프를 그려서 확인해 보더라도 배의 좌측(port)에 위치한 사람들이 구조 상황에서 유리한 위치를 선점했음을 알 수 있었습니다.

- Age

Age 변수의 분포를 확인하기 위해 데이터를 10대, 20대, 30대, 40대, 50대, 60대, 70대, 80대 이상으로 구분하여 막대그래프로 시각화하였습니다. 확인 결과, Age 변수의 데이터들은 대부분 20~29 범위의 값을 가지고 있으므로 최빈값을 이용해 결측치를 채웠습니다.

- Destination

Destination의 결측치 처리 방식을 결정하기 위해 먼저 Destination과 다른 변수 간에 어떤 상관관계가 존재하는지 알아보고자 하였습니다.

첫 번째로 Destination과 Age의 관계를 알아보기 위해 Age에 따른 Destination의 분포를 시각화해 살펴보았습니다. 그 결과 다양한 나이에서 Destination의 분포가 모두 유사하게 나타남을 확인했습니다. 따라서 '나이에 따른 목적지 선호도는 존재하지 않는다'라는 결론을 내렸습니다.

두 번째로 Destination과 VIP 변수 간 유의미한 상관관계가 있는지 분석하기 위해 VIP가 TRUE인 경우와 FALSE인 경우의 Destination을 각각 막대그래프로 시각화했습니다. 그러나 VIP가 TRUE인 경우가 FALSE인 경우보다 현저히 적어 분포를 제대로 확인하기 어려웠습니다. 따라서 정확한 분석을 위해 막대그래프가 아닌 원그래프를 활용하여 분포를 시각화했고 VIP가 TRUE인 경우와 FALSE인 경우 모두 Destination의 분포가 유사함을 알 수 있었습니다. 그러므로 '두 변수 간 유의미한 상관관계가 없다'라는 결론을 내렸습니다.
이처럼 Destination과 다른 변수들 사이에 유의미한 상관관계가 없으므로 Destination의 최빈값을 활용하여 결측치를 처리해도 Destination의 분포에 큰 변화가 없을 것이라고 예상했습니다. 따라서, 최빈값으로 결측치를 처리하였습니다.

- PassengerId

gggg_pp의 형태를 가진 값들입니다. 우리는 '그룹에 속한 사람들의 수가 많을수록 실종률이 낮을 것이다.'라는 가정을 바탕으로 위 데이터를 분석하고자 했습니다. gggg가 같은 값을 지니고 있다면 같은 그룹에 속해 있다는 말이고, 같은 그룹 내에 몇 명이나 속해 있는지 확인하기 위해서는 pp가 가장 커지는 값을 관찰하면 될 것입니다.
위 아이디어에 착안하여 passengerId를 인덱싱하여 각각 group, index, maxIndex 열로 쪼갠 후, 그룹의 인원수가 생존율에 얼마나 큰 영향을 끼치는지 관찰했습니다. 8명>1명(single)>2명>7명>5명>3명>6명>4명 순으로 생존율이 높았고, 따라서 무조건 그룹에 속한 사람들이 많을수록 생존율이 높을 것이란 가정은 채택할 수 없음이 드러났습니다. 8명으로 속한 그룹의 생존율이 이례적으로 높은 결과는, 8명이 속한 그룹의 비율이 전체 인원수에서 상당히 소수의 비율을 차지하고 있으므로 발생했을 것이라 추론하였습니다. 따라서 무조건 많은 사람이 포함된 그룹이 생존율이 높다고 속단할 수는 없을 것입니다.

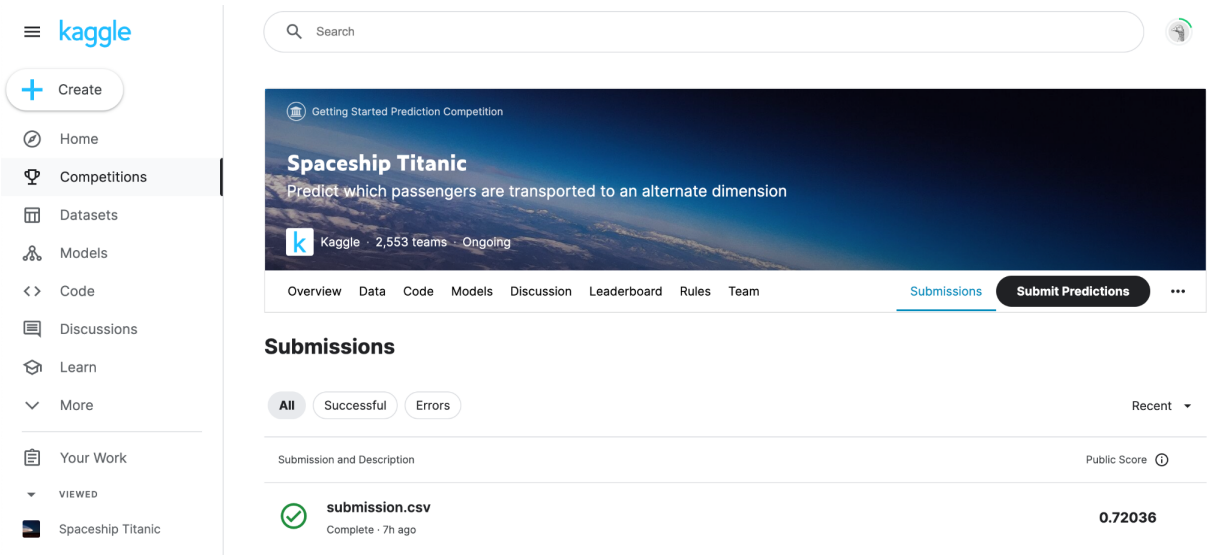
- Name

가장 먼저 'Name' 열의 값을 받아 이름과 성을 분리하는 split_name 함수를 정의하였습니다. 'Name'은 결측치를 채우는 방법을 취할 수 없으므로, 결측치가 있는 경우 두 부분 모두를 None으로 처리하였습니다. 영문으로 이름을 표기할 때는 '성, 이름'의 방식과 '이름 성'의 방식이 있는데, 'Name' 열의 값에서 이름과 성을 구분하는 기호가 없으므로, 이름과 성을 분리할 때는 후자의 방식을 택하여 공백을 기준으로 나눕니다. 이렇게 분리된 이름과 성을 새로운 'Given' 및 'Family' 열에 할당합니다.
이 코드를 통해 'Family' 열을 기준으로 데이터프레임을 그룹화하고, 각 가족 그룹의 크기를 'Family Size' 열에 추가했습니다. 그 후, 'IsSingle' 열을 생성하여 각 승객이 단일 승객 여부를 나타내는 불리언(Boolean) 값을 할당했습니다. 이후 'IsSingle' 열을 기준으로 데이터프레임을 그룹화하여 각 그룹(단일 승객 또는 다중 승객)의 생존율을 계산했습니다.
마지막으로, 데이터프레임에서 여전히 남아있는 결측치의 개수를 확인합니다. 이름은 결측치를 처리할 수 없었기 때문에 200개씩의 결측치가 남아있습니다. 또한, 분석에서 성이 같으면 모두 가족으로 분류되었는데, 가족이 아니더라도 우연히 겹치는 성 때문에 발생하는 잠재적 오류를 해결할 수 없었습니다. 이 두 가지 양상을 배제하고 생각한다면, 결론적으로 가족 인원수와 단일 승객 여부는 실종률에 영향을 미치지 않았습니다.

3. 결론

모든 결측치가 제거된 최종적인 데이터프레임에서 모델 설계에 필요 없다고 판단되는 변수들을 제거하여 final_df 변수에 저장했습니다. 종속 변수로 설정한 'Transported' 열을 예측할 수 있는 모델을 설계하기 위해 해당 train 데이터셋에서 훈련용 데이터 80%, 모델 성능평가용 데이터 20%를 추출했습니다. 분류를 위해 데이터프레임에 포함된 범주형 변수들을 모두 dummy 변수로 변환하였으며 로지스틱 회귀 분석 방법을 활용했습니다. 계수 값들을 확인해 보면 감금된 여부(CryoSleep)가 실종률에 가장 큰 영향을 미치고 있었음을 확인하였고, 그다음으로는 목적지(Destination)가 'TRAPPIST-1e' 일 때 실종률과 적지 않은 음의 상관관계를 가지고 있었음을 확인할 수 있었습니다. 모델의 성능은 accuracy 지표와 f1-score의 수치를 따져보았을 때 0.73 정도로 나쁘지 않은 편이었습니다.

최종적으로 kaggle에서 제공한 test.csv 데이터셋으로 모델의 성능을 평가해 봤습니다. test 데이터셋도 train과 마찬가지로 결측치가 존재했던 탓에 train 데이터셋의 결측치를 제거했던 로직과 똑같은 방식으로 결측치를 제거하고 필요한 변수들을 생성했습니다. 앞에서 설계했던 동일한 모델로 주어진 test 데이터셋들의 정보를 대입하여 예측된 종속 변수의 값들을 csv 파일로 변환하여 kaggle 사이트에 업로드했습니다. 그 결과, 총점수는 72점을 얻을 수 있었습니다.



이러한 결과를 바탕으로 우리 팀은 재난 상황에서 재난 약자가 될 가능성이 높은 집단 승객들에 대한 안전 및 보안 절차를 개선하기 위해, 필요한 요건들을 정리해 보았습니다.

먼저, 감금 상태(CryoSleep) 대신 승객의 생존을 보장할 수 있는 다른 방안을 연구하거나 감금 상태에서의 위험 상황에 빠르게 대응할 수 있도록 훈련과 장비를 개선해야 합니다. 이를 위해서는 감금 상태의 위험을 더 자세히 분석하고, 해당 승객들의 안전성과 생존 가능성을 고려하여 감금 상태 선택을 결정해야 합니다. 기술적인 개선으로는 감금 상태 승객을 실시간으로 모니터링하고, 감금 상태가 언제든지 위험에 노출될 수 있는 상황을 조기에 감지하기 위한 시스템을 구축하는 방안이 있습니다. 또한, 재난 상황에서 감금 상태가 안전하게 자동 해제되는 절차를 개발하고 승객들에게 대피 훈련을 제공하여 생존을 보장합니다.

그다음, 목적지(Destination)가 실종률에 영향을 미치는 것으로 확인되었으므로 미래의 여행에서는 목적지 선택에 특별한 주의를 기울여야 합니다. 항해 계획을 수립할 때

목적지의 안전성을 정밀하게 평가하고, 목적지와 우주선의 상태를 실시간으로 모니터링하여 위험 상황을 조기에 감지할 수 있는 시스템을 구축해야 합니다.

과거 데이터를 활용하여 재난 상황에서 실종 가능성이 높은 승객들을 사전에 식별하고, 다양한 시나리오를 시뮬레이션하여 잠재적 위험을 사전에 파악한다는 점에서 이 프로젝트는 중요한 의미를 갖습니다. 이를 통해 미래 정책 및 계획을 조정하고, 비슷한 상황에서 대비 능력을 강화할 수 있을 것입니다.