

Titanic-accident-analysis

학번: 2018086

이름: 주동철

Github address: dongcheolju

1. 안전 관련 머신러닝 모델 개발의 목적

- a. RMS 타이타닉은 북대서양 횡단 여객선으로 1912 년 영국에서 미국으로 향해 중 빙산과 충돌하여 침몰하였다. 이 침몰로 발생한 사망자는 총 1514 명으로 탑승자의 68%가 사망한 대형 사고이다. 이러한 동종 사고를 예방하고자 타이타닉 탑승객의 데이터를 분석하는 머신러닝 모델을 개발하게 되었다.
- b. 학습 모델 활용 대상: 선박이나 항공기 업계에서 사고를 예방하기 위한 학습 지표가 될 수 있다.
- c. 개발의 의의: 학습 모델 개발 시 어떠한 가치를 생성하는지 타이타닉 사고 분석 모델은 탑승객의 정보가 담긴 데이터를 각 칼럼별로 시각화 및 분석하고 예측하여 동종사고 발생을 예방하는 것에 의미를 둔다.

2. 안전 관련 머신러닝 모델의 네이밍의 의미

타이타닉 사고 분석 모델은 타이타닉호 침몰 당시 탑승한 탑승객의 인적 정보가 담긴 데이터 셋을 분석하여 사망자와 생존자의 수를 다양한 방식으로 분류 및 예측하는 모델이다.

3. 개발 계획

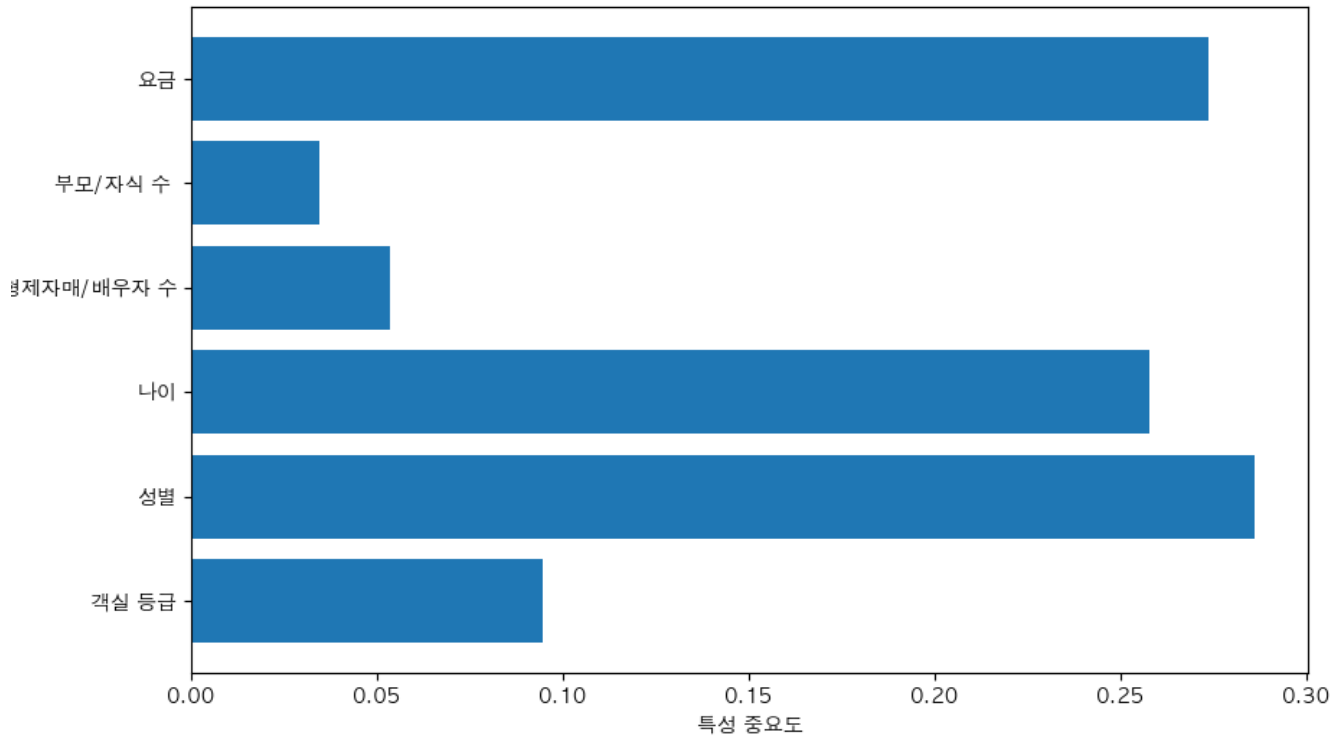
- a. 데이터에 대한 요약 정리 및 시각화

데이터는 <https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv> 에서 가져온 csv 파일로 각 요소의 컬럼명들은 영문으로 되어있었고, 한국어로 번역을 진행하였고 각 요소는 다음과 같다.

생존, 객실 등급, 이름, 성별, 나이, 형제자매/배우자 수, 부모/자식 수, 요금

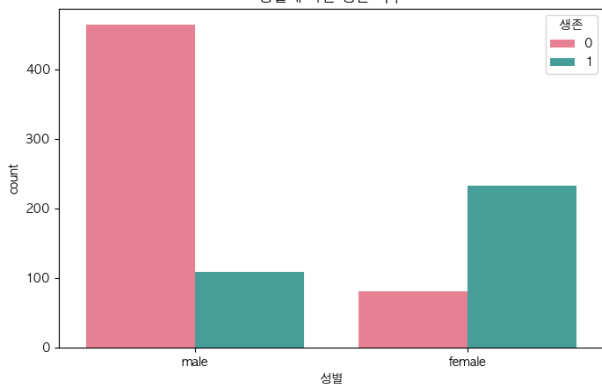
이러한 정보를 시각화하기 위하여 각 요소가 생존에 미치는 중요도를 시각화 하여 막대그래프를 아래와 같이 그렸다.

생존 여부의 특성 중요도

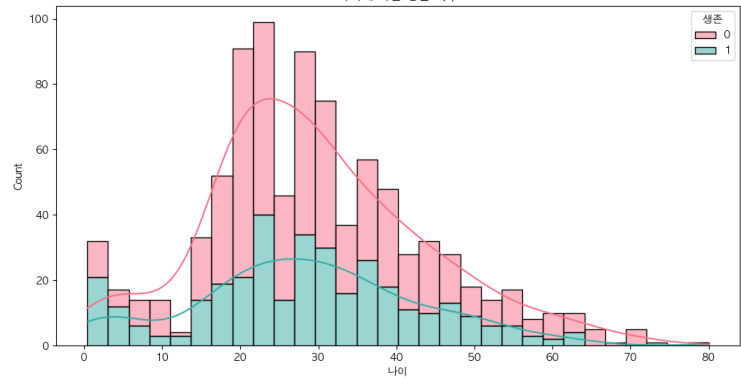


위 그래프를 살펴보면 생존 여부에 가장 크게 작용하는 요소는 성별이고, 다음으로 요금과 나이가 뒤를 잇는다. 이를 좀 더 세부적으로 분석하기 위하여 요금, 나이, 성별에 따른 생존률에 어떤 관계가 있는지 확인하기 위해 다음 그래프를 그렸다.

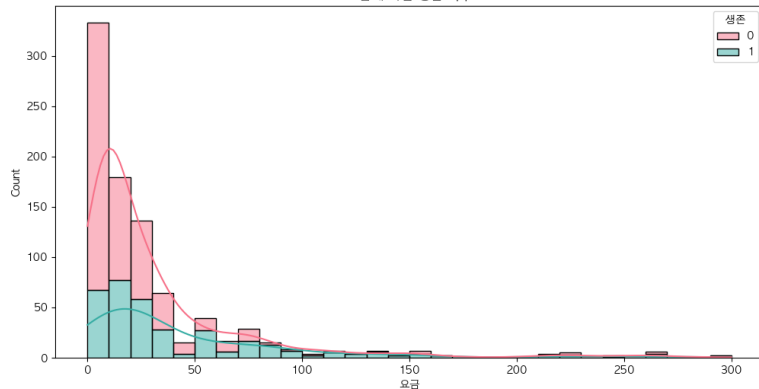
성별에 따른 생존 여부



나이에 따른 생존 여부

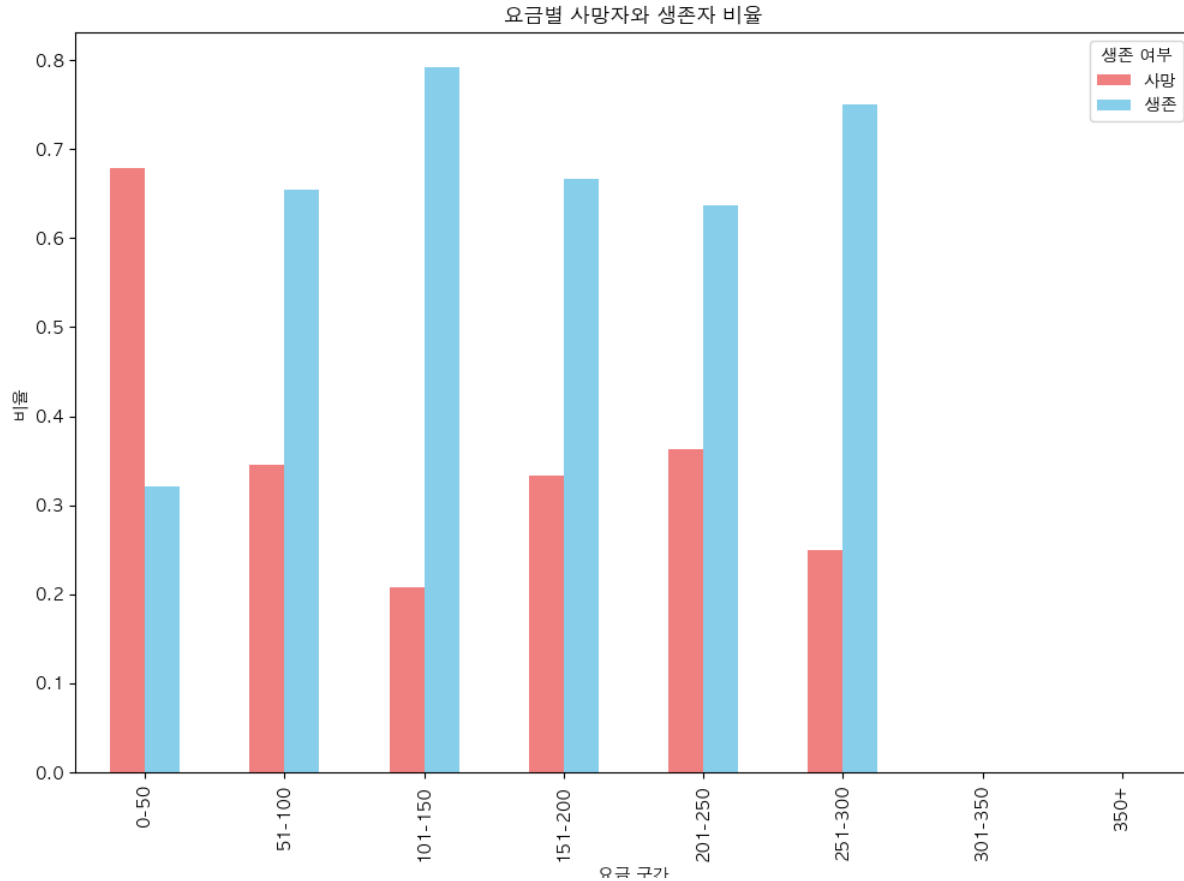


요금에 따른 생존 여부



위 세 그래프를 통해 각 요소별 생존률에 어떻게 작용하는지 간단하게 살펴볼 수 있다.

하지만 요금에 따른 생존 여부는 그래프의 요소 사이의 편차가 커 쉽게 분석하기가 어렵다. 이를 해결하기 위해 요금에 따라 생존의 비율이 어떻게 보이는지 알아보기 위해 다음 그래프를 그렸다.



위 그래프를 통해 요금별 생존률이 어떤 차이를 가지는지 알아볼 수 있다.

b. 데이터 전처리 계획

- i. 'pandas' 라이브러리를 이용하여 csv 파일 불러오기

```
titanic_data = pd.read_csv(url, encoding='cp949')
```

이 때, encoding='cp949'은 한국어로 되어있는 컬럼명을 오류 없이 불러오기 위해 추가했다.

- ii. 필요한 열만 선택하여 데이터프레임을 재구성

```
titanic_data = titanic_data[['객실 등급', '성별', '나이', '형제자매/배우자 수', '부모/자식 수', '요금', '생존']]
```

- iii. 'dropna' 함수를 사용하여 특정 열에 누락된 값이 있는 행 삭제

```
titanic_data = titanic_data.dropna(subset=['나이', '형제자매/배우자 수', '부모/자식 수'])
```

- iv. 성별을 숫자로 매핑하여 범주형 변수를 수치형으로 변환

```
titanic_data['성별'] = titanic_data['성별'].map({'male': 0, 'female': 1})
```

- v. 독립 변수(X)와 종속 변수(y) 정의

```
X = titanic_data.drop('생존', axis=1)
y = titanic_data['생존']
```

- vi. 'train_test_split' 함수를 사용하여 데이터를 학습 데이터와 테스트 데이터로 분리

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

c. 어떠한 머신러닝 모델을 사용할 것인지 (해당 머신러닝 모델의 이론 추가)

이번 코드에서 사용된 머신러닝 모델은 'Random Forest Classifier'이다.

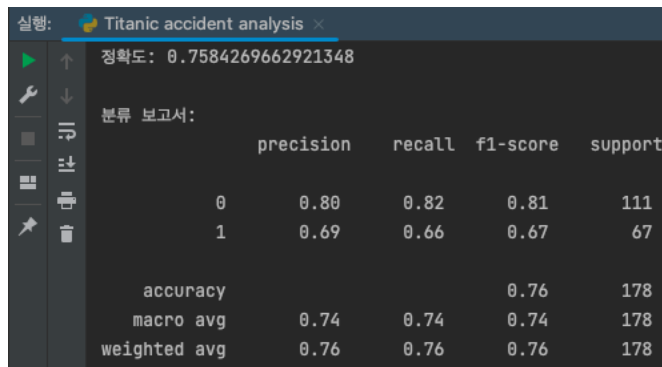
Random Forest 는 여러 개의 의사결정 나무를 사용하여 데이터를 학습하고, 각 나무의 예측을 기반으로 최종 예측을 수행하는 앙상블 학습 방법 중 하나이다.

각 나무는 데이터의 부분집합에 대해 학습되며, 이들의 예측을 종합하여 모델의 정확도를 향상시킨다.

이 코드에서는 'Random Forest Classifier' 모델을 생성하고 학습시킨 후, 테스트 데이터로 예측하고 정확도 및 분류 보고서를 출력하였다.

d. 머신러닝 모델 예측 결과

전체 샘플 중에서 정확하게 예측한 비율은 다음과 같고 약 75.84%의 정확도를 보인다.



	precision	recall	f1-score	support
0	0.80	0.82	0.81	111
1	0.69	0.66	0.67	67
accuracy			0.76	178
macro avg	0.74	0.74	0.74	178
weighted avg	0.76	0.76	0.76	178

classification_report 를 이용하여 분류 보고서를 출력하였다.

분류 보고서의 각 요소는 다음과 같다.

I. Precision (정밀도):

- 생존(1) 클래스의 정밀도: 0.69
- 사망(0) 클래스의 정밀도: 0.80
- 각 클래스에 대한 예측 중 실제로 해당 클래스에 속하는 비율
- 생존 클래스의 경우 69%의 예측이 실제로 생존했고, 사망 클래스의 경우 80%의 예측이 실제로 사망함

II. Recall (재현율):

- 생존(1) 클래스의 재현율: 0.66
- 사망(0) 클래스의 재현율: 0.82
- 각 클래스에 속한 샘플 중 모델이 정확하게 예측한 비율
- 생존 클래스의 경우 66%의 실제 생존한 경우가 모델에 의해 정확하게 예측되었으며, 사망 클래스의 경우 82%의 실제 사망한 경우가 모델에 의해 정확하게 예측됨

III. F1-score (F1 스코어):

- 생존(1) 클래스의 F1 스코어: 0.67
- 사망(0) 클래스의 F1 스코어: 0.81

- 정밀도와 재현율의 조화평균으로 계산되는 스코어로, 두 지표를 한 번에 고려한 모델의 성능을 나타냄

IV. Support (지원):

- 각 클래스에 속한 샘플의 수
- 생존 클래스의 경우 67 개의 샘플이 있고, 사망 클래스의 경우 111 개의 샘플이 있음

e. 사용할 성능 지표

단계에서 언급한 것처럼 정확도(accuracy)와 분류 보고서(Classification Report)가 있다.

f. 성능 검증 방법 계획 등

크게 네 종류로 나뉘었으며 다음과 같다.

i. 학습 데이터와 테스트 데이터 분리

`train_test_split` 함수를 사용하여 전체 데이터를 학습 데이터와 테스트 데이터로 분리

이를 통해 모델이 학습 데이터에서 학습되고, 테스트 데이터에서 성능을 평가할 수 있게 함

ii. 랜덤 포레스트 모델 학습

`RandomForestClassifier` 를 사용하여 랜덤 포레스트 모델 학습

iii. 모델 평가:

테스트 데이터를 사용하여 학습된 모델의 예측 수행

`accuracy_score` 함수를 사용하여 정확도를 계산

`classification_report` 함수를 사용하여 분류 보고서를 생성

이 보고서에는 각 클래스(생존, 비생존)에 대한 정밀도, 재현율, F1 점수 등을 포함함

iv. 특성 중요도 시각화:

학습된 랜덤 포레스트 모델에서는 각 특성의 중요도를 확인할 수 있음

이 중요도는 각 특성이 예측에 얼마나 기여하는지를 나타냄

특성 중요도는 `feature_importances` 속성을 통해 얻을 수 있으며, 이를 막대 그래프로 시각화하여 확인 진행

4. 개발 과정

a. 계획 후 실제 학습 모델 개발 과정을 기록 (*개발 과정 캡처 필수)

i. 필요한 라이브러리 import

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rc
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

ii. 데이터 불러오기 및 전처리

```
# 데이터 불러오기
A = "data/titanic.csv"
titanic_data = pd.read_csv(A, encoding='cp949') # 한국어를 불러올 수 있게 인코딩 지정

# 데이터 전처리
titanic_data = titanic_data[['객실 등급', '성별', '나이', '형제자매/배우자 수', '부모/자식 수', '요금', '생존']]
titanic_data = titanic_data.dropna(subset=['나이', '형제자매/배우자 수', '부모/자식 수']) # 누락된 값이 있는 행 삭제
```

iii. 데이터 학습

```
# 성별을 숫자로 매핑
titanic_data['성별'] = titanic_data['성별'].map({'male': 0, 'female': 1})

# Features와 Labels 정의
X = titanic_data.drop('생존', axis=1)
y = titanic_data['생존']

# 학습 데이터와 테스트 데이터로 나누기
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

iv. RandomForestClassifier 를 이용하여 모델 생성 및 학습

```
# 랜덤 포레스트 모델 생성
model = RandomForestClassifier(n_estimators=100, random_state=42)

# 모델 학습
model.fit(X_train, y_train)
```

v. 데이터 예측과 정확도 및 분류 보고서 출력

```
# 테스트 데이터로 예측
y_pred = model.predict(X_test)

# 정확도 출력
accuracy = accuracy_score(y_test, y_pred)
print("정확도:", accuracy)

# 분류 보고서 출력
print("\n분류 보고서:\n", classification_report(y_test, y_pred))
```

vi. 각 요소의 중요도 시각화

```
# 각 요소의 중요도 시각화
feature_importances = model.feature_importances_
feature_names = X.columns
plt.figure(figsize=(10, 6))
plt.barh(range(len(feature_importances)), feature_importances, align="center")
plt.yticks(range(len(feature_importances)), feature_names)
plt.xlabel("특성 중요도")
plt.title("생존 여부의 특성 중요도")
plt.savefig("titanic_analysis_plots/생존 여부의 특성 중요도")
plt.show()
```

vii. 나이, 성별, 요금에 따른 생존 여부 시각화

```
# 나이에 따른 생존 여부 시각화
plt.figure(figsize=(12, 6))
sns.histplot(x='나이', hue='생존', data=titanic_data, kde=True, multiple='stack', bins=30, palette='husl')
plt.title('나이에 따른 생존 여부')
plt.savefig("titanic_analysis_plots/나이에 따른 생존 여부")
plt.show()

# 성별에 따른 생존 여부 시각화
# 성별 매핑을 다시 원래 값으로 변경
titanic_data['성별'] = titanic_data['성별'].map({0: 'male', 1: 'female'})
plt.figure(figsize=(8, 5))
sns.countplot(x='성별', hue='생존', data=titanic_data, palette='husl')
plt.title('성별에 따른 생존 여부')
plt.savefig("titanic_analysis_plots/성별에 따른 생존 여부")
plt.show()

# 요금에 따른 생존 여부 시각화
plt.figure(figsize=(12, 6))
sns.histplot(x='요금', hue='생존', data=titanic_data, kde=True, multiple='stack', bins=30, palette='husl')
plt.title('요금에 따른 생존 여부')
plt.savefig("titanic_analysis_plots/요금에 따른 생존 여부")
plt.show()
```

viii. 요금 구간별 사망자 비율 시각화

```
# 요금 구간별 사망자 비율 시각화
# 요금을 구간별로 나누기
bins = [0, 50, 100, 150, 200, 250, 300, 350, float('inf')]
labels = ['0-50', '51-100', '101-150', '151-200', '201-250', '251-300', '301-350', '350+']
titanic_data['요금 구간'] = pd.cut(titanic_data['요금'], bins=bins, labels=labels, include_lowest=True)
# 요금별 사망자와 생존자의 수 계산
fare_survival_counts = titanic_data.groupby(['요금 구간', '생존'], observed=False).size().unstack()
# 비율 계산
fare_survival_ratio = fare_survival_counts.div(fare_survival_counts.sum(axis=1), axis=0)
# 그래프 그리기
fig, ax = plt.subplots(figsize=(12, 8))
fare_survival_ratio.plot(kind='bar', stacked=False, color=['lightcoral', 'skyblue'], ax=ax)
ax.set_title('요금별 사망자와 생존자 비율')
ax.set_xlabel('요금 구간')
ax.set_ylabel('비율')
ax.legend(title='생존 여부', labels=['사망', '생존'])
plt.savefig("titanic_analysis_plots/요금구간에 따른 생존 비율")
plt.show()
```

b. 각 함수는 어떻게 동작하는 지 구체적으로 설명

코드 파일에 각주로 설명을 달아둬.

c. 에러 발생 지점 및 해결 과정

사용자의 분석을 편하게 진행할 수 있도록 하기 위해 직접 한글로 번역하였다.

그러나 파이썬에서 데이터를 불러오고 시각화 하는 과정에서 한글이 깨지는 일이 발생하여 다음 코드를 추가하여 오류를 잡았다.

```
rc('font', family='AppleGothic')
plt.rcParams['axes.unicode_minus'] = False

titanic_data = pd.read_csv(A, encoding='cp949') # 한국어를 불러올 수 있게 인코딩 지정
```

또, 성별에 따른 생존률 분석 시각화 과정중 앞서 예측을 위해 성별을 0 과 1 로 매핑해둔 것을 글자로 바꿔주기 위해 다음 코드를 추가 했다.

```
# 성별을 숫자로 매핑
titanic_data['성별'] = titanic_data['성별'].map({'male': 0, 'female': 1})
# 성별 매핑을 다시 원래 값으로 변경
titanic_data['성별'] = titanic_data['성별'].map({0: 'male', 1: 'female'})
```

성별을 숫자로 매핑

성별을 글자로 매핑

d. 학습 모델의 성능 평가

3.의 f 항목과 같은 내용이며, 정확도가 75.84%의 성능을 나타내고 있다.

e. 결과 시각화

3.의 a 항목에서 앞서 모두 언급함.

5. 개발 후기

이번 코드는 타이타닉 사고의 사상자를 분석하고 예측하는 모델이다.

시각화된 파일을 살펴보면 남성은 사망자가 생존자보다 월등히 많았으나, 여성은 사망자보다 생존자의 수가 더 많았으며, 성인보다는 나이가 어린 어린이나 유아가 더 높은 비율로 생존했다는 것을 알 수 있다.

추가로 요금에 관한 생존률을 시각화 한 파일을 살펴보면 흥미로운 사실들이 있다. 아래의 두 그래프에서 요금에 따른 생존 여부 그래프를 살펴보면 사망자와 생존자의 수는 요금이 가장 저렴한 0~50 구간에 가장 많이 분포한 것을 알 수 있다. 요금이 저렴한 구간이 요금이 비싼 구간보다 월등히 탑승객이 많았으며 사망자와 생존자가 가장 많다는 것을 알 수 있다.

하지만 요금별 사망자와 생존자의 비율을 나타낸 그래프를 살펴보면 가장 저렴한 0~50 구간에는 생존자에 비해 사망자가 월등히 그 비율이 높은 것을 알 수 있다. 요금이 비싸지면서 생존자의 비율이 사망자의 비율보다 더 높은 것을 알 수 있다.

간단하게 정리하면, 요금이 저렴한 구간에서는 생존자와 사망자의 수가 다른 구간에 비해 많고, 사망자의 비율도 높은 것을 알 수 있으며, 요금이 비싼 구간에서는 생존자와 사망자의 수가 저렴한 구간보다 적고, 생존자의 비율이 사망자의 비율보다 높은 것을 알 수 있다.

과거 타이타닉 침몰이 있던 현장에서도 아이와 여성을 먼저 구조하는 모습들이 보였으며, 돈이 비교적 적은 서민보다는 돈이 비교적 많은 사람들이 더 생존에 유리했다는 것을 알 수 있었다.

