Predicting Titanic Survivability Using Logistic Regression

Abdur-RasheedAde

Aug 1, 2022
https://github.com/Abdur-RasheedAde/The-Titanic-Prediction

This is a Machine Learning Prediction of the Titanic Dataset using Logistic Regression while the Exploratory Data Analysis is done with Power BI.

THE CHALLENGE

The sinking of the Titanic is one of the most infamous shipwrecks in history.

On April 15, 1912, during her maiden voyage, the widely considered “unsinkable” RMS Titanic sank after colliding with an iceberg. Unfortunately, there weren’t enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.

While there was some element of luck involved in surviving, it seems some groups of people were more likely to survive than others.

In this challenge, I built a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (ie name, age, gender, socio-economic class, etc).

DATA SOURCE

The data is gotten from the popular Titanic Competition on kaggle.com https://www.kaggle.com/competitions/titanic/data. The description of the data is also well explained on the page. here is a brief summary;

Variable Notes pclass: A proxy for socio-economic status (SES) 1st = Upper 2nd = Middle 3rd = Lower

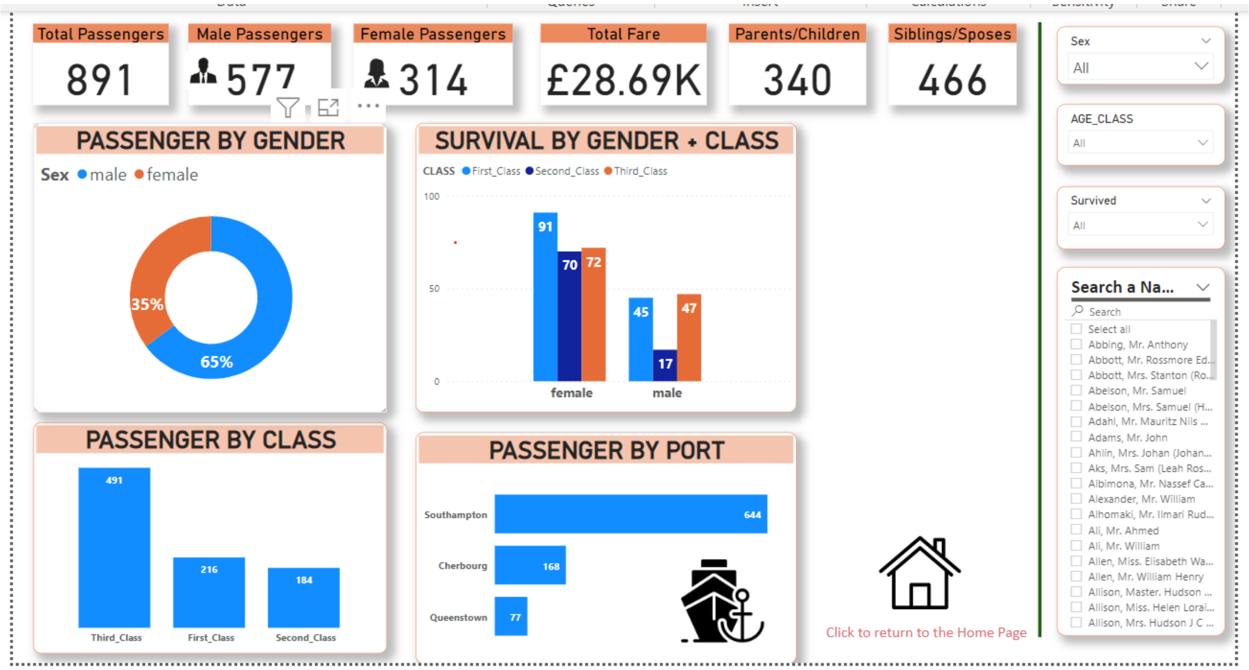
age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way... Sibling = brother, sister, stepbrother, stepsister Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way... Parent = mother, father Child = daughter, son, stepdaughter, stepson Some children travelled only with a nanny, therefore parch=0 for them.

EXPLORATORY DATA ANALYSIS

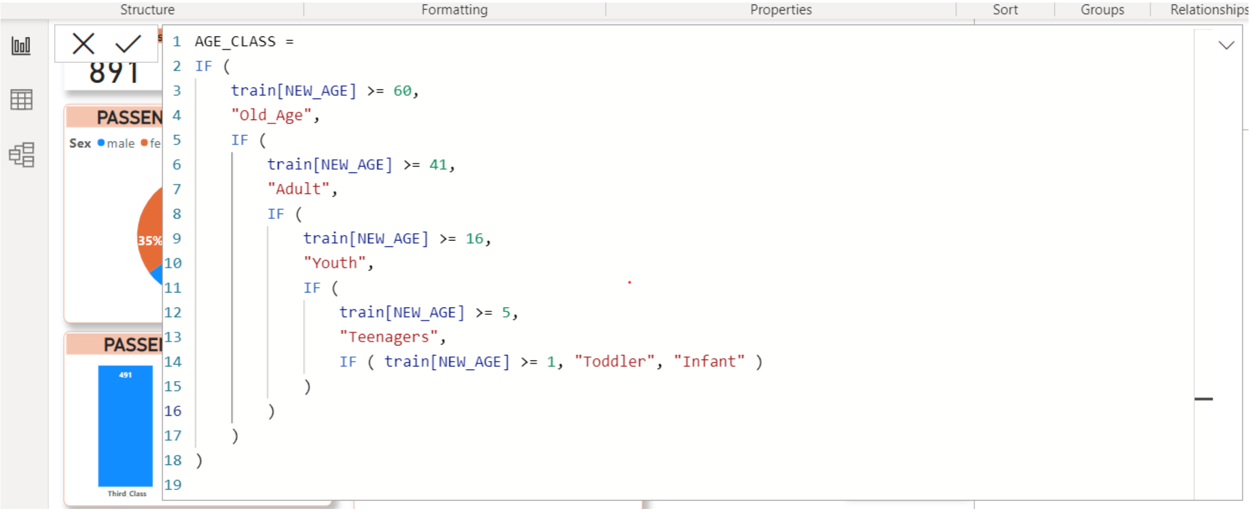
Exploring a dataset is majorly used by Data Scientist to understand the data better before proceeding in the Data Cleaning Process. I therefore used Power BI Desktop to explore the data by building a visualization Report from the Dataset. Pushed it to PowerBI Service and and import it in the Jupyter Notebook. please note that only the training dataset was explored in this visual because most of the Preprocessing are majorly from the train dataset. click to view the Report dashboard https://app.powerbi.com/view?r=eyJrIjoiZDE1MzQyMWQtZWU1ZS00NjM5LWI1MGItNWE2MDgwMGNhZDc3IiwidCI6IjMyNzk2YmUyLTYwZmItNGRhMi04ZDI2LTA2ZTU5MzhlNmU2YiIsImMiOjh9&pageName=ReportSection



PREPROCESSING

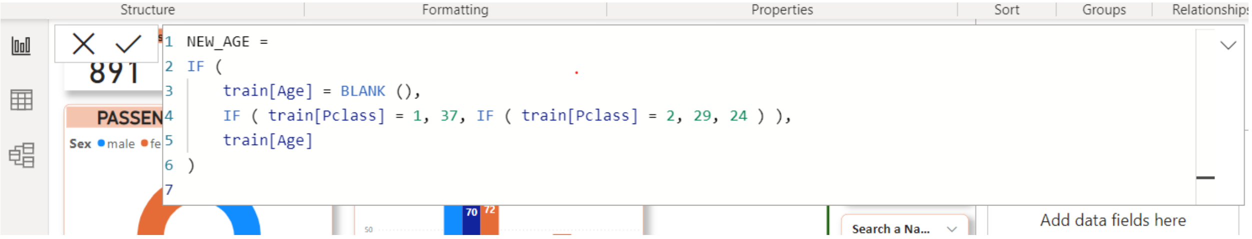
In Power BI, different measures were created to prepare the data for better exploration and visualization

1. The Age was categorized as Infant, Toddler, Teenager, Youth, Adult and Old Age



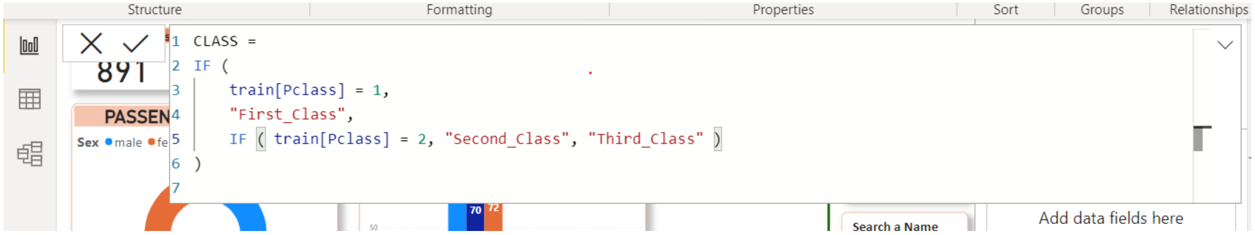
```
AGE_CLASS =
IF (
    train[NEW_AGE] >= 60,
    "Old_Age",
    IF (
        train[NEW_AGE] >= 41,
        "Adult",
        IF (
            train[NEW_AGE] >= 16,
            "Youth",
            IF (
                train[NEW_AGE] >= 5,
                "Teenagers",
                IF ( train[NEW_AGE] >= 1, "Toddler", "Infant" )
            )
        )
    )
)
```

2. Some Ages were missing in the Age Column, there were therefore filled using the Passenger Class Column, details below;



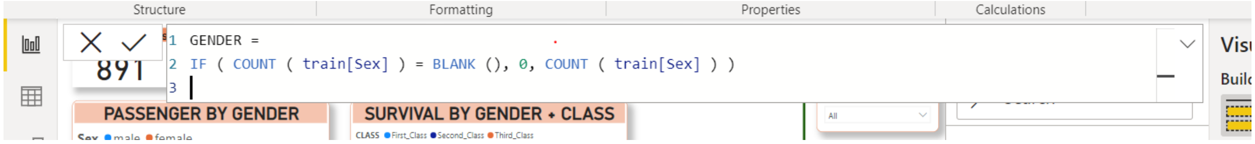
```
NEW_AGE =
IF (
    train[Age] = BLANK (),
    IF ( train[Pclass] = 1, 37, IF ( train[Pclass] = 2, 29, 24 ) ),
    train[Age]
)
```

3. The Class is represented in numbers as 1,2,3 and are mapped to First Class, Second and Third Class respectively using an IF statement



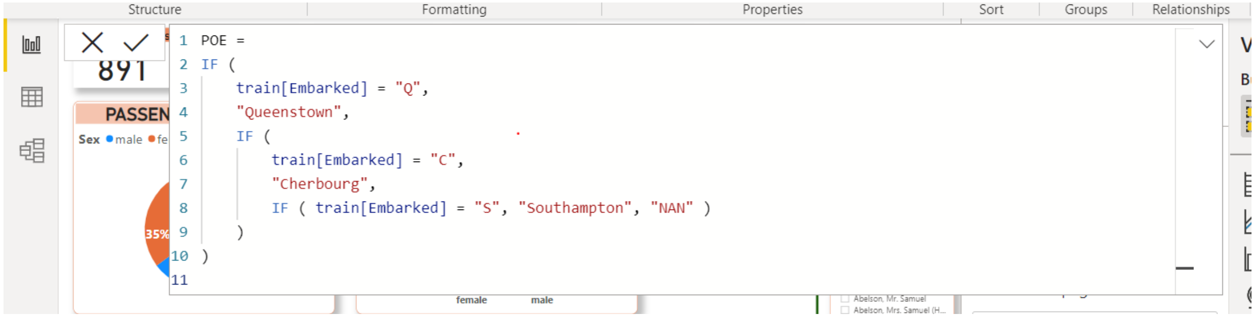
```
CLASS =
IF (
    train[Pclass] = 1,
    "First_Class",
    IF ( train[Pclass] = 2, "Second_Class", "Third_Class" )
)
```

4. Convert Blank to Zero in the Gender Column



```
GENDER =
IF ( COUNT ( train[Sex] ) = BLANK (), 0, COUNT ( train[Sex] ) )
```

5. Point of Embarked were represented with 'S', 'Q', 'U' and re-categorized as Southampton, Queenstown and Cherbourg respectively



```
POE =
IF (
    train[Embarked] = "Q",
    "Queenstown",
    IF (
        train[Embarked] = "C",
        "Cherbourg",
        IF ( train[Embarked] = "S", "Southampton", "NAN" )
    )
)
```

MODEL BUILDING

This is the process in building a Logistic Regression using the whole of the Titanic Dataset (the train and the test dataset). Among other Classification models (K-Nearest Neighbor, Select Vector Machine, Random Forest, Decision Tree…) only Logistic Regression gave a better performance of model prediction when only the train dataset was used to predict, hence it was adopted in this model. This model used both train and test data to predict which passenger in the test data will survive or not. The model predicted 78% of the Passengers accurately and had 22% wrong prediction