Titanic

Team Name: UniCoders

Member 1: Hussein Hodroj
Member 2: Hamza Ibraheem
Member 3: Seedra Yamani

Submission Date: May 28, 2024

Abstract

This project aimed to analyze and predict survival and death counts of passengers aboard the well-known Titanic according to various factors. Data went through several stages of cleaning, visualization and preparation for model training, lastly followed by testing the model on different validation sets. The work process included handling missing values, imputing categorical values and feature engineering to create more straightforward data for model training. As we employed several machine learning models including Random Forest, Supportive Vector Classifier(SVC) and K-Nearest Neighbors (KNN).As a result, KNN recorded the highest accuracy value, defeating other models in recall,F1-score and precision.

1 Introduction

The disaster of RMS Titanic sinking in 1912 caused the death of over 1500 passengers and crew who were aboard it, and this tragic event represent an axis of research and intensive analysis that aims to understand the factors that supported survival rates. Passengers information provides a good role to employ different machine learning models to predict the result of survival counts according to different proprieties.

The main goal of this project is analysing and predicting the survival and death counts aboard Titanic using the famous Titanic dataset.By taking advantage of machine learning models, we aim to specify main factors that define the probability of survival.

This project includes a comprehensive process for data cleaning,visualizing using pandas and seaborn libraries for creating HeatMap, Pie Chart and bar plot, feature engineering to prepare data for model training to ensure the accuracy of our trained model predictions.

2 Literature Review

We reviewed a project named Titanic Classifier by Can Wang and Rigpea Wangchuck that was supervised by prof.Babis Tsourakakis and TF Tianyi Chen, this project aimed to predict passenger survival in the Titanic disaster using machine learning, after carefully reviewing this project, we found that this project involves prepossessing the Titanic Dataset handling missing values, encoding categorical variables, and normalizing distributions. Key features analyzed include sex, class, embarkation point, and family relationships.
In this study, students used several machine learning models including logistic regression, linear regression, decision tree, and random forest, all these were used to predict survival rates, and they also used visual representations like HeatMaps.

Now , in the research report the results showed that logistic regression is the most effective model for binary classification tasks with an accuracy of 0.83 percent, which is close to what we had 0.83 percent, for their other Models scikit-learn's achieved 84.4 percent. The decision tree model implemented from scratch achieved 82.2 percent accuracy, higher than the scikit-learn version's 76.7 percent. Random forest and linear regression models both achieved around 82 percent accuracy.

The study concludes that while no single model universally outperforms others, logistic regression is particularly well suited for binary classification for this project . The project highlights the importance of matching models to specific data characteristics and computational resources, enhancing understanding of predictive modeling techniques.
Now after reviewing This research we found out that the titanic data set is well suited for data science and machine learning models and it gives a solid understand for analysis and predictions techniques although they might have used different strategies than what we used we still achieved a good results in our project thanks to the hard work we put in understanding and analysing the titanic dataset.

Titanic Classifier Paper [Rigpea, 2024].

3 Methodology

- **Data Collection:** We started looking for data to work on, suddenly, Titanic Dataset caught our attention. We found data on kaggle, Which is a public website, as we made sure the data was an open source, free to use, and matches our goal in this project by its ability for training and model testing.
  The data was rich of information that qualified it for training model and testing by including Age, sex, passenger class, embarking port, and so on...
  As it contained Three files: testing file, training file, and gender file.

- **Data Cleaning and Feature Engineering:**

  1. Handling Missing Values:There was two missing (NAN) values in the embarked columns in the training dataset, as we looked at the data we discovered that there is 3 categories S='Southampton' which had the highest count '644' the others were: C='Cherbourg' '168', and last one Q='Queenstown' '70', so we replaced the two missing values by filling them with S because it had the highest count.

```
df['Embarked'].fillna('S', inplace=True)
count_S = (df['Embarked'] == 'S').sum()
print("Number of occurrences of 'S' in the 'Embarked'
    column:", count_S)
```

  2. Feature Engineering: We created new features that simplified and categorized data and made it more straightforward to model training that included :

     (a) Age mapping: we encountered 177 missing age values in the training dataset, we will list how we fixed this issue; because it is critical to the model training as when we did the visualization, we figured out that the age is so critical to the deaths count and it represent a pattern that this data was representing,
     1. Handling Missing Age Values:
     we started off by filling the missing age columns by -1
     2. Defining Age Bins and Labels:
     then we defined age groups using specific bins and labels to categorize different age ranges (showed in the table 1).
     3. Categorizing Ages:
     now using the defined bins and labels, we categorized the Age column into age groups for both datasets (Training and testing).
     4. Combining Datasets:
     after that we combined both datasets into a list to ensure uniform processing.
     5.Extracting and Standardizing Titles:
     Titles were extracted from the 'Name' column, and rare and royal titles were standardized also replacing Miss/Ms/Mlle with Miss, and Mme with Mrs for consistency and to group similar titles together.
     6. Displaying Title Data:
     A cross-tabulation of titles and sex, as well as the survival rates for each title, were calculated and displayed.
     7. Mapping Titles to Numerical Values:
     then we made sure the Titles were mapped to numerical values to facilitate machine learning model training.
     8. Handling Missing Age Groups:

Missing 'AgeGroup' values were filled based on the corresponding 'title', using a predefined mapping as we listed before.
9. Converting Age Groups to Numerical Values:
Age group labels were converted into numerical values to make it ready for machine learning.
10. Dropping the 'Age' Column:
The original 'Age' column was removed from both datasets to avoid redundancy.

"Age	Age Category"
"-1	Unknown"
"0 to 5	Baby"
" 5 to 12	Child"
"12 to 18	Teenager"
"18 to 24	Student"
"24 to 35	Young Adult"
"35 to 60	Adult"
"60 to infinite	Senior"

Table 1: Age Categories

(b) Sex mapping: For this feature we classified genders into two numerical classes using a dictionary : (0) for Males, (1) for females, then assigned 'Sex' column values with the new gender values.

(c) Embarking ports mapping: As the previous feature we classified embarking ports using a dictionary as follows : (Southampton (S)= 1,Cherbourg (C)= 2, Queenstown (Q)=3), then assigned new values to the original 'Embarked' column.

(d) Ticket Fare:
Firstly, we filled missing values in 'Fare' column according to passenger class using a lambda function that calculates the mean fare for each class, then rounds it to four decimal digits
Secondly, categorized 'Fare'‘values into quarterlies by creating a new column called 'FareBand' by using 'pd.qcut' function that is used to divide the 'Fare' column in to four equal bins, labeling them with 1,2,3, and 4. Lastly, we removed 'Fare' column from both data sets and kept the 'FareBand' column for model training.

(e) Cabin: we checked if there is any NAN values and returned a series where 1 represents non-Null value, and 0 represents Null values, in result we created a new column called 'CabinBool' contained bool values, then dropped 'Cabin' column because its not needed anymore.

```
df["CabinBool"] = (df["Cabin"].notnull().astype('int'))
dft["CabinBool"] = (dft["Cabin"].notnull().astype('int'))

df = df.drop(['Cabin'], axis=1)
dft = dft.drop(['Cabin'], axis=1)
```

- **Analysis Techniques:**

  1. Descriptive statistics: for describing data, we used the '.discribe()' method that is used to compute Central Tendency measures (Mean, Median) and Dispersion (Standard Deviation), and Distribution insights (Min, Max, Quartiles).

     a. The mean of a set of values $x_1, x_2, \ldots, x_n$ is given by:

     $$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

     b. The median is the middle value of a dataset that has been arranged in ascending order. If n is odd, the median is the value at position $\left(\frac{n+1}{2}\right)$. If n is even, the median is the average of the values at positions $\left(\frac{n}{2}\right)$ and $\left(\frac{n}{2}+1\right)$.
     For odd n:

     $$\text{Median} = x_{\left(\frac{n+1}{2}\right)}$$

     For even n:

     $$\text{Median} = \frac{x_{\left(\frac{n}{2}\right)} + x_{\left(\frac{n}{2}+1\right)}}{2}$$

     c. The standard deviation is the square root of the variance. For a sample, the sample standard deviation s is:

     $$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

     d. The minimum and maximum values of a dataset are the smallest and largest values, respectively.

     $$\text{Minimum} = x_{\text{min}} = \min(x_1, x_2, \ldots, x_n)$$

     $$\text{Maximum} = x_{\text{max}} = \max(x_1, x_2, \ldots, x_n)$$

     e. Quartiles divide the dataset into four equal parts.The first quartile (Q1),

the second quartile (Q2), which is the median, and the third quartile (Q3) are defined as follows:
- **First Quartile (Q1):** The median of the lower half of the dataset.
- **Second Quartile (Q2):** The median of the dataset.
- **Third Quartile (Q3):** The median of the upper half of the dataset.

2. Counting Embarkation Ports: We used '.value_counts()' to simply count occurrences of passengers embarking at different ports.

3. Missing Values Handling :
a. Embarked Column: Is a categorical column, as we filled missing values using the mode (most frequented value).
b.Fare column: filled missing fare values by using mean fare grouped by passenger class, here is the code and explanation for the equation:

```
dft['Fare'] = dft['Fare'].fillna(dft.groupby('Pclass'
    )['Fare'].transform('mean').round(4))
```

1. Compute the mean μ_X for Fare column X in the DataFrame:

$$\mu_X = \frac{1}{n} \sum_{i=1}^{n} X_i$$

where n is the number of non-missing values in the column.
2. Round μ_X to four decimal places:

$$\mu_X^{\text{rounded}} = \text{round}(\mu_X, 4)$$

3. Replace missing values (NaN) with μ_X^{rounded}:

$$X_i = \begin{cases} X_i & \text{if } X_i \text{ is not NaN} \\ \mu_X^{\text{rounded}} & \text{if } X_i \text{ is NaN} \end{cases}$$

4. Data Visualization: helps identify patterns and relationships between data, making it easier to understand, as we used:
a. Pie Chart: To visualize the percentage of survival rates.
b. Bar Chart: To show death counts according to gender and passenger class.
c. Heatmaps : To clarify correlation and distribution between age and death counts.

5. Age Group Mapping: To analyze age-related trends and patterns. Age groups were created based on predefined bins and labels, that passengers were categorized into. As it allowed more likable analysis of age-related trends.

6. Title Mapping : Titles were extracted from passengers names and categorized into groups as it provides additional information about passengers' social status, which can be related to survival chances.

7. Survival Analysis By Features: Analyzed survival and death counts based on several features such as: gender, age group, passenger class, this helped in identifying important features that supported survival rate.

   By applying these techniques, we gained a comprehensive understanding for the dataset, identified important patterns and relationships, and prepared data for machine learning.

- **Machine Learning Models:** In this project we used three different machine learning models:

  1. **Random Forest Classifier:** Started by importing the required libraries from scikit-learn including:
     a. 'RandomForestClassifier' to create the Random Forest model.
     b. 'train_test_split' to split the dataset.
     c. Evaluation metrics: 'accuracy_score','recall_score','f1_score' and 'confusion' to assess model performance.
     Then, we defined the features (Pclass, sex, AgeGroup,etc...) and the target (whether a passenger survived or not).
     After that, we split the dataset into training and validation sets using the 'train_test_split' function, which allowed us to train the model on one segment of the data and checked its performance on another unseen segment.
     Subsequently, we defined a Random Forest classifier with 100 decision trees and train it using the training data. As the 'random_state' parameter ensured duplication of results. Then, We used the trained model to make predictions on the validation set as it allowed us to compare the predicted survivals with the actual ones and assess model's accuracy.
     Finally, we evaluated the model's performance by calculating the accuracy score that measures the rate of correctly predicted survival outcomes on the validation set, which achieved a score of 0.826 .
  2. **Supportive Vector Machine (SVM) Classifier:** Started by importing required class (SVC) to build the SVM classifier, and 'train_test_split' function to split the dataset into training and validation sets.
     Next, we defined the features and the target (Survived or died).
     After that, we split the dataset into training and validation sets using the 'train_test_split' function, which allowed us to train the model on one segment of the data and checked its performance on another unseen segment.
     Thereafter, we defined an SVM classifier with the 'rbf' kernel, that is suitable for handling non-linear classification problems. As the 'random_state' parameter ensured duplication of results.
     Finally, we used the trained SVM model to make predictions on the validation set, that achieved an accuracy of 0.821 .
  3. **K-Nearest Neighbors (KNN) Classifier:** Started with importing necessary libraries that includes:

a. 'KNeighborsClassifier': To create the KNN model.
b. 'StandardScaler': for standardizing the features.
c.'train_test_split' function : To divide the dataset into training and validation sets.
d. 'accuracy_score': To calculate the model's performance.
As previously, we identified features and target, then split the dataset using 'train_test_split' function.
Consequently, we used 'StandardScaler' to standardize the features by removing the mean and scaling to unit variance, as it ensures that all features contribute equally to the distance calculations used by KNN. Then, it fitted to the training data and then applied to both the training and validation set to maintain consistency.
Next, we initialized and trained a KNN classifier using the scaled training data, as it classifies a data point based on the majority class among its nearest neighbor. The KNN model was ready to make predictions on the scaled validation set, which allowed us to compare its prediction outcomes with the actual outcomes and calculated the model's accuracy.
. last but not least, KNN achieved an accuracy of 0.837 .

4 Data Description

- **Sources:** we first started off by searching on different websites like Kaggel , UCI Machine Learning Repository, Google Dataset Search, OpenML , GitHub , then when we found the titanic dataset on kaggel which caught our attention because this dataset provided us with what we needed and what we were hoping for, [Kaggle, Year]

- **Size:** The overall dataset has 3 excel files , training and testing and third file for gender submission, The training dataset has an overall of 891 rows which is the record for the passengers × 11 columns Survival for features.
for the testing dataset we have 418 rows × 10 columns 1 less than testing dataset because it doesn't have the survival column due to it being a testing dataset for the model we are creating.

- **Attributes: Survived**: Description: Indicates whether the passenger survived the Titanic accident.
Data Type: Categorical (Binary: 0 = No, 1 = Yes).
Relevance: This is the target variable for predictive modeling tasks because predicting survival is often the main objective of analysis.
Pclass (Ticket class):
Description: Represents the ticket class of the passenger.
Data Type: Categorical (Ordinal: 1 = 1st, 2 = 2nd, 3 = 3rd).
Relevance: Ticket class can provide insights into socio-economic status, as passengers in higher classes had better survival chances than those in the third class.

Age:
Description: Represents the age of the passenger in years.
Data Type: Numerical (Continuous).
Relevance: Age can be a significant factor in survival, as children and the elderly might have been given priority during evacuation.
SibSp (Number of siblings/spouses aboard the Titanic):
Description: Indicates the number of siblings or spouses with the passenger.
Data Type: Numerical (Discrete).
Relevance: Provides insights into family size and potential relationships among passengers.
Parch (Number of parents/children aboard the Titanic):
Description: Indicates the number of parents or children with the passenger.
Data Type: Numerical (Discrete).
Relevance: Similar to SibSp, this attribute provides information about family size and relationships among passengers.
Ticket:
Description: Represents the ticket number of the passenger.
Data Type: Categorical (Text/String).
Relevance: The ticket number might not be directly relevant.
Fare:
Description: Represents the fare paid by the passenger.
Data Type: Numerical (Continuous).
Relevance: Fare can be an indicator of socio-economic status.
Cabin:
Description: Represents the cabin number of the passenger.
Data Type: Categorical (Text/String).
Relevance: Cabin location could potentially impact survival, as passengers in certain areas of the ship might have had better access to lifeboats.
Embarked (Port of Embarkation):
Description: Indicates the port from which the passenger embarked.
Data Type: Categorical (Nominal: C = Cherbourg, Q = Queenstown, S = Southampton).
Relevance: While the port of embarkation might not directly affect survival, it could associate with other variables such as socio-economic status or nationality.

- **Prepossessing Steps:**

1. Data Loading and Inspection:
we first started by Loading the Titanic dataset into pandas DataFrames (train=df,test=dft). Inspected the first 15 rows of the train and test datasets to understand their structure. Then Checked for missing values and inspected the data types of each column.
2. Handling Missing Values:
Identified missing values in the 'Embarked' column then Filled missing 'Embarked' values with the most repeated value 'S'.

Filled missing 'Age' values with (-1) for further processing.
Imputed missing 'Fare' values in the test dataset with the mean fare for each passenger class.
3. Feature Engineering:
Created a binary 'CabinBool' column indicating cabin presence, if a passenger have a cabin the cabinbool list it as 1 otherwise 0.
Binned 'Age' into descriptive age groups based on predefined bins and titles then Extracted titles from names and categorized them into common groups after that we Filled missing 'AgeGroup' values based on titles
Mapped categorical variables ('Sex', 'Embarked') to numerical values. Binned 'Fare' into quartiles and created 'FareBand' for both training and test datasets.
4. Feature Selection:
Dropped irrelevant columns ('Cabin', 'Ticket', 'Name') from both datasets.
Summary and Rationale: The prepossessing steps included data cleaning, missing value handling, feature engineering, and feature selection.

The choice of prepossessing techniques was guided by the dataset characteristics and the goal to have a model able of predicting survival on the Titanic.

5 Results and Discussion

5.1 Results

- Key Findings:
  a. Survivals By Passenger Class: The survival rate was remarkably higher for first-class passengers compared to other classes with a record of 80 deaths.

  b. Survivals By Gender and Age: "Women and children first" was the protocol used in the Titanic when deciding who got to use the limited number of lifeboats, as a result highest survival rates were Females and Children (0-12 years) compared to Males and Adults.

  c. Model Performance: The K-Nearest Neighbor (KNN) achieved the highest accuracy compared to the other models.

Model	Accuracy	Precision	Recall	F1 Score
Random Forest	82.6%	80.2%	77.02%	78.6%
SVC	82.1%	80%	75%	77.7%
KNN	83.7%	80%	81.01%	80.5%

5.2 Discussion

- our main objective was to understand what lies behind the tragic accident and how did it affect the life of passengers aboard, what we actually discovered is that if you come from a wealthy family you probably had a better rate of surviving based on a visualization we did, The questions that we have were does it really matter

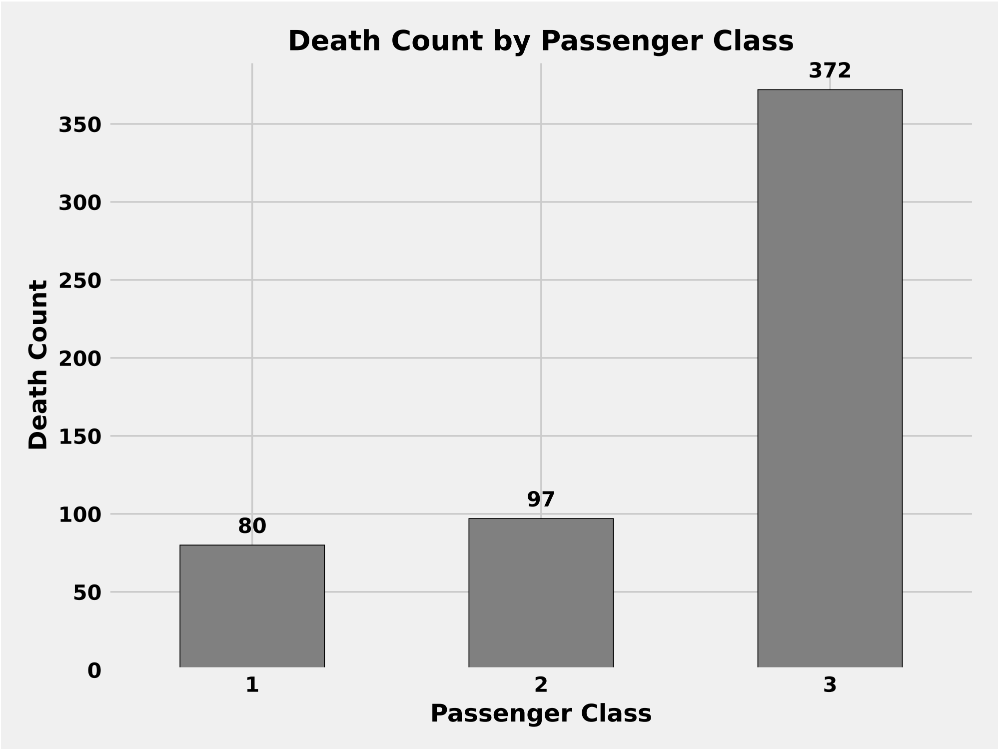

Figure 1: Death Count By Passenger Class

your gender? your age? your traveling class, thankfully all these questions got an answer and it is 'YES' it matters, we tried our best to understand this major event in the history of humanity and we did it.

- what actually surprised us was how small details in passenger life made a huge difference in their percentage of surviving, titanic dataset was a great way to introduce us to the world of data science and machine learning.
- This data contains a lot of useful information's , Features of the passengers where helpful , but not to say we didn't have any hard times working on it, we actually did and we tried day and night to fix them and to bring the best of this dataset, an example to that having a huge number of missing age values making it hard for us to complete the work on the data, and adding to that not having a specific information where exactly the passengers where when the accident happened, eventually we were able to solve these problems without letting them affect our results and project work.
- after carefully analysing the Titanic dataset we revealed several significant insights , as we discussed before the titanic dataset has important key variables.
  passenger Class (Pclass):
  Finding: Passengers in first class had a higher survival rate compared to those in

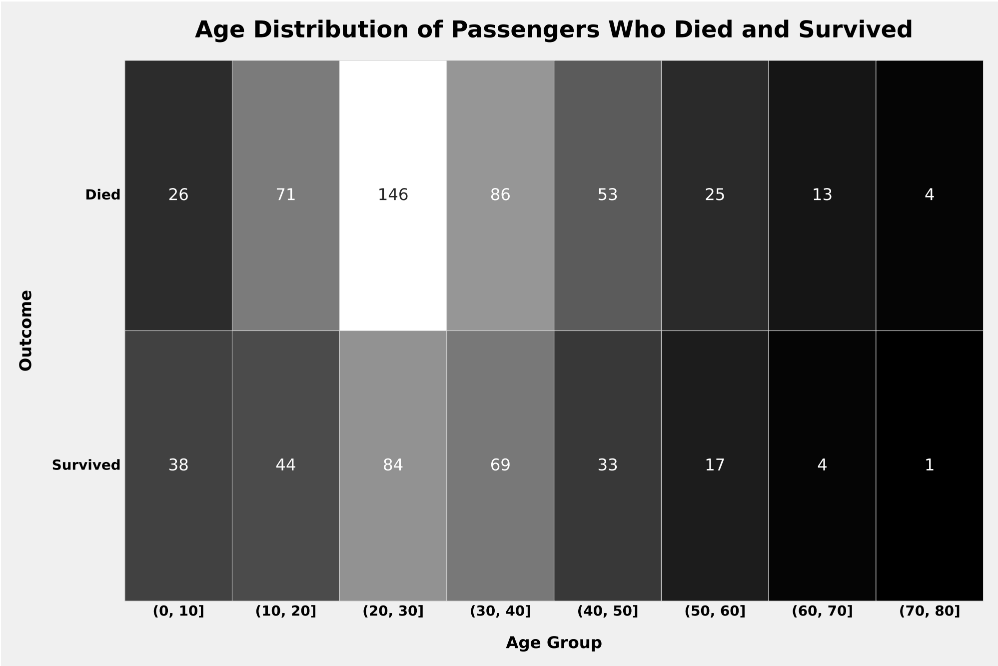

Figure 2: Death Distribution By Age Group

second and third classes.
Implication: This reflects the historical reality of the time, where socio-economic status greatly influenced access to life-saving resources, we could say that being wealthy or coming from a wealthy family at that time could make you gain access to valuable safety resources.

Gender:

Finding:Females had a higher survival rate than males
Implication: as we discussed before The 'Women and children first' protocol appears to have been followed during the evacuation, This finding reveals the importance of emergency protocols and their potential impact on survival outcomes.

Age:

: Finding: Younger passengers, particularly children, had a higher likelihood of survival.
Implication: This supports the prioritization of children in life-saving scenarios, which is important in any life scenario when we need to choose who we can rescue.

Fare:

Finding: Passengers who paid higher fares had better survival rates.
Implication: Fare paid can be a proxy for socio-economic status, linking back to the influence of wealth and class on survival chances, also This could guide future research on how financial resources impact survival in disasters.

after carefully analysing and working on The Titanic data set , The findings ac-

tually opens up several avenues for further research and practical applications we will be listing few of them:
Impact of Family Connections on Survival:
How did family connections influence survival rates? Did passengers traveling with family members have higher survival rates compared to those traveling alone?
Role of Crew Members:
What was the survival rate of crew members compared to passengers?
How did the duties and responsibilities of crew members affect their chances of survival?

Evacuation:
How did the timing of evacuation (early vs.late evacuation) influence survival rates?
What patterns can be observed in the timeline of lifeboat launches and their occupancy?

[Rigpea, 2024]

This is The project that we reviewed , we listed before our detailed overview on it.

6 Tables, Figures, and Code Listings

6.1 Figures

Survival Rate

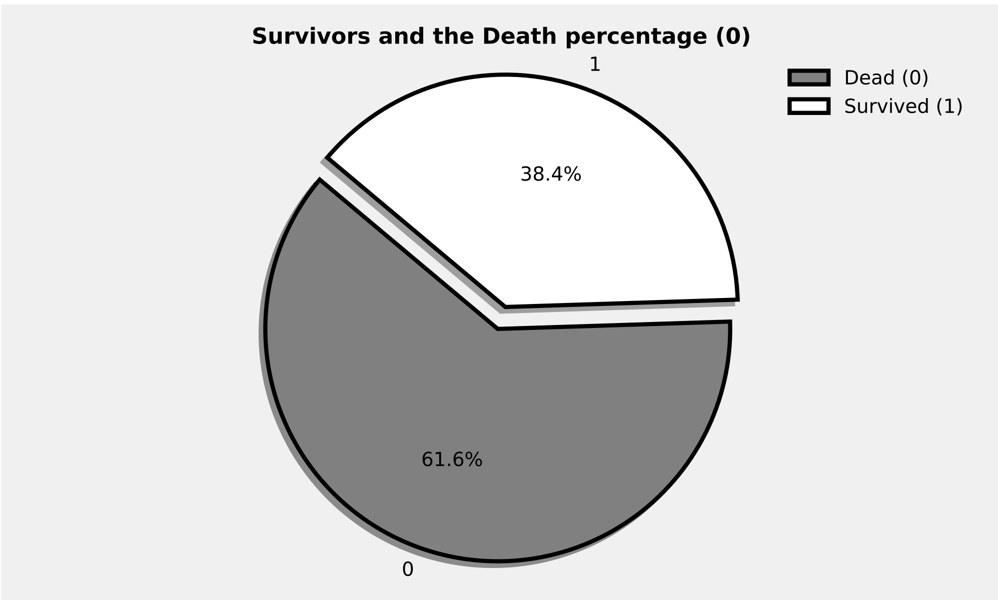

Figure 3: Survivors And Death Percentage

Survival Percentage 3 This Pie Chart illustrates the survival percentage As 0 represents deaths with a value that equals to 61.6% and 1 represents Survivals with a value equals to 38.4%

Death Counts By Passenger Class 1 This Bar Plot illustrates that first-class passengers had the lowest death rates, with a record of 80 deaths. second-class passengers had a death record of 97 deaths, and the highest death rates unfortunately were to the third-class passengers with a record of 372 deaths.

Death And Survival Distribution By Age Group 2 This Heatmap illustrates Death in Survival rates by age groups, as shown, age group (20-30 years) Had the highest death counts.

6.2 Code Listing

```python
for dataset in combine:
    dataset['Title'] = dataset['Name'].str.extract(' ([A-Za-z]+)\.', expand=False)
    dataset['Title'] = dataset['Title'].replace(['Lady', 'Capt', 'Col', 'Don', 'Dr', 'Major', 'Rev', 'Jonkheer', 'Dona'], 'Rare')
    dataset['Title'] = dataset['Title'].replace(['Countess', 'Lady', ' Sir'], 'Royal')
    dataset['Title'] = dataset['Title'].replace(['Mlle', 'Ms'], 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')
    print(pd.crosstab(df['Title'], df['Sex']))
```

In this code, We extracted the titles from passengers names and Standardized them into categories (Rare, Royal, Miss, and Mr), then analyzed title distribution by sex and survival rate, and finally, mapped titles to numerical values and updated the dataset accordingly.

6.3 Tables

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Table 2: Titanic Training Data Sample Before Cleaning And Feature Engineering

PassengerId	Pclass	Sex	SibSp	Parch	Embarked	CabinBool	AgeGroup	Title	FareBand
892	3	0	0	0	3	0	5.0	1	1
893	3	1	1	0	1	0	6.0	3	1
894	2	0	0	0	3	0	7.0	1	2
895	3	0	0	0	1	0	5.0	1	2
896	3	1	1	1	1	0	4.0	3	2

Table 3: Titanic Training Data Sample After Cleaning And Feature Engineering

7 Conclusion

In conclusion, our work on the Titanic dataset leaded to several significant ideas about the key factors that affected survival rates. We found that socio-economic status, gender, and age played a huge role on survival rates, first-class passengers, women and children had higher chances of survivals compared to other groups. According to these results, we understood the social inequalities in disaster managements.

The broader implications of our work indicate that similar social and demographic factors may play crucial roles in other disaster scenarios, thus informing future policies and safety measures For example, Our findings assure the need for fair safety protocols that do not unequally favor wealthier individuals.

Despite the intensity of our analysis, there were limitations.The dataset had missing values, especially in the Age column, which was a critical issue , but as we mentioned before we were able to fix it, In addition to that the historical nature of the dataset shows that it may not include all relevant factors, such as the exact location of passengers during sinking.

Overall, our project not only provides a detailed exploration of the Titanic tragedy, but also offers valuable insights that can inform future research and practical applications in disaster managements and response.

References

Kaggle. (Year). *Titanic: Machine learning from disaster*. https://www.kaggle.com/competitions/titanic/data?select=train.csv

Rigpea. (2024). *Titanic classifier paper*. https://github.com/Rigpea/titanic-classifier/blob/main/365_Paper%20(1).pdf