Assignment 2: Supervised Learning with Feature Engineering

Jay Shukla
College of Engineering
Northeastern University
Toronto, ON
Shukla.j@northeastern.edu

Abstract: -

In this supervised learning project, I have used titanic dataset which is a very popular dataset on Kaggle. But we can do all the feature engineering precisely on this type of dataset. This is dataset link from Kaggle: https://www.kaggle.com/datasets/yasserh/titanic-dataset .

Description of Dataset: -

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 on board, 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, we ask you to build a predictive model that answers the question: “what sorts of people were more likely to survive?” using passenger data (i.e. name, age, gender, socio-economic class, etc.)

Methodology: -

The methods used to create this project are listed below:

Language: I have used Python Programming language to write the code
Jupyter: I have used this IDE to write my code in python language.
Pandas: used to create a data frame and generate excel file of the data extracted.

What is feature engineering using Supervised Learning?

Feature engineering is a crucial step in the data preprocessing phase of a machine learning pipeline, particularly in supervised learning. Supervised learning involves building a model that makes predictions based on input features and is trained using a labeled dataset, where the target outcomes are already known.

Feature engineering, in the context of supervised learning, involves creating new features or modifying existing ones to improve the performance of a machine learning model. The goal is to provide the model with inputs that are more informative and relevant to the task at hand, thereby improving its ability to learn the relationship between the features and the target variable. Here are some key aspects of feature engineering in supervised learning:

1. **Feature Creation**: This involves generating new features from the existing data. This could be as simple as combining two variables to create a new one (e.g., combining 'height' and 'weight' to create a 'body mass index' feature) or more complex transformations based on domain knowledge.
2. **Feature Transformation**: Transforming features can help in various ways, such as normalizing the scale of variables (e.g., log transformation, scaling), converting categorical variables into numeric format (e.g., one-hot encoding), or making the feature distribution more suitable for a model (e.g., normalization or standardization).
3. **Feature Selection**: Not all features in a dataset may be useful or relevant for making predictions. Feature selection involves identifying and selecting those features that contribute most to the prediction variable. This

can help in reducing the dimensionality of the dataset, improving model performance, and reducing overfitting.

4. **Dimensionality Reduction**: Techniques like Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) can be used to reduce the number of features while retaining the most important information. This is particularly useful in datasets with a high number of features, where many features may be redundant or irrelevant.
5. **Dealing with Missing Values**: Handling missing data is an important aspect of feature engineering. Strategies can include imputing missing values based on other data points, using model-based imputation, or creating indicator features to mark data as missing.
6. **Domain-specific Feature Engineering**: Involves using knowledge specific to the domain from which the data is derived to create features that are particularly informative for the problem at hand. For example, in a financial application, creating features based on moving averages or other financial indicators.

Feature engineering is often an iterative process that involves hypothesis generation about what features might be useful, creating those features, training models using those features, and then evaluating model performance to inform further feature engineering efforts. The effectiveness of feature engineering is highly dependent on the domain, the specific problem, and the type of data, as well as the choice of model.

Understanding The Problem

The Titanic dataset is a classic dataset used in data science and machine learning, often for binary classification problems. The primary problem to solve with the Titanic dataset is to predict whether a passenger survived the Titanic disaster based on various features.

This problem falls under the category of supervised learning, specifically binary classification, because the target variable (whether a passenger survived or not) has two classes: survived (1) or did not survive (0).

Definition of Project Objectives and Goals

The main objective of the Titanic survival prediction project is to build a predictive model that can accurately determine the survival outcomes of passengers based on a set of features. These features might include:

- Passenger demographics (e.g., age, sex)
- Socio-economic status (e.g., passenger class)
- Family information (e.g., number of siblings/spouses aboard, number of parents/children aboard)
- Ticket information (e.g., fare paid, cabin number, embarkation port)

The goal is not only to create a model with high predictive accuracy but also to understand the factors that contributed to the likelihood of survival, which can provide insights into the disaster and the social dynamics of the time.

Identification of the Target Variable

In the context of the Titanic dataset, the target variable is clear: it is the "Survived" column, which indicates whether a passenger survived the disaster. This variable is typically represented as a binary outcome:

- 1: The passenger survived.
- 2: The passenger did not survive.

With these components in place, you can proceed to the next steps of the project, which involve data exploration, preprocessing, feature engineering, model selection, training, evaluation, and interpretation of the results. Understanding the problem, defining the objectives, and identifying the target variable are crucial for guiding these subsequent steps and ensuring the project stays focused on its goals.

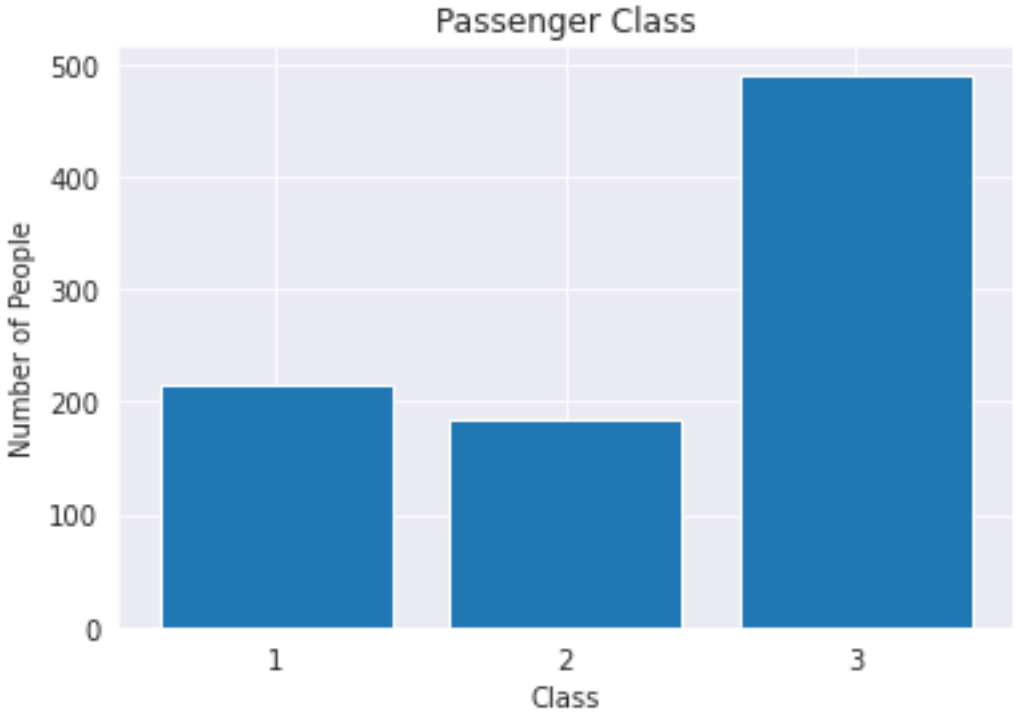
Data Collection and Preparation: -

As the data was highly unbalanced I cleaned the data by handling missing values and removed the irrelevant features

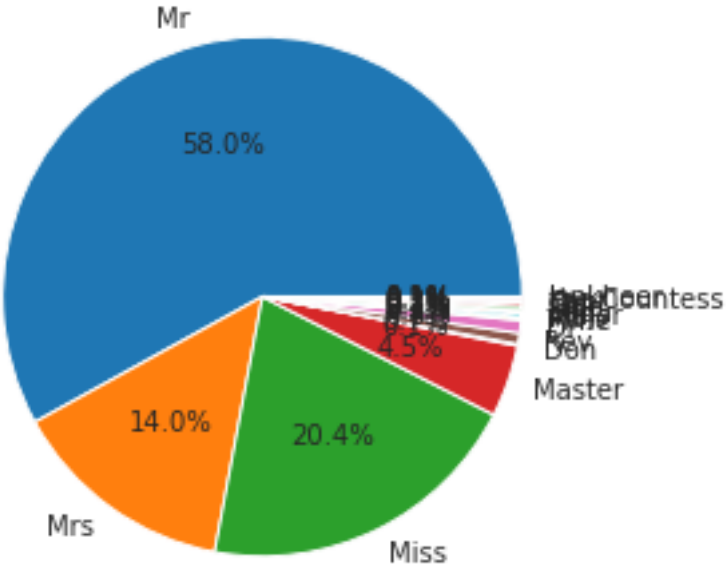
and divided dataset into training and testing dataset.

Output of all the files: Now we are going to implement feature engineering and extract all the features from the dataset.

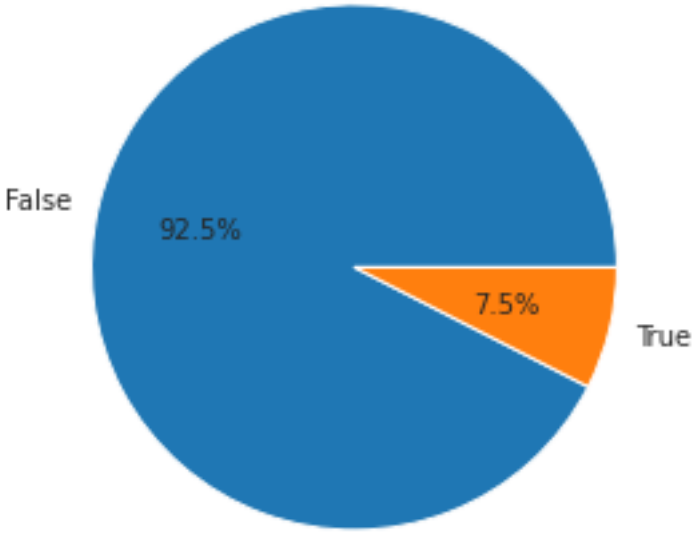
Images are given below:-
Let's look at our first feature: Pclass. For Passenger Class, there are three options: 1, 2, or 3. One corresponds to the upper, highest class, two corresponds ot the middle class, and three corresponds to the lower, lowest class.



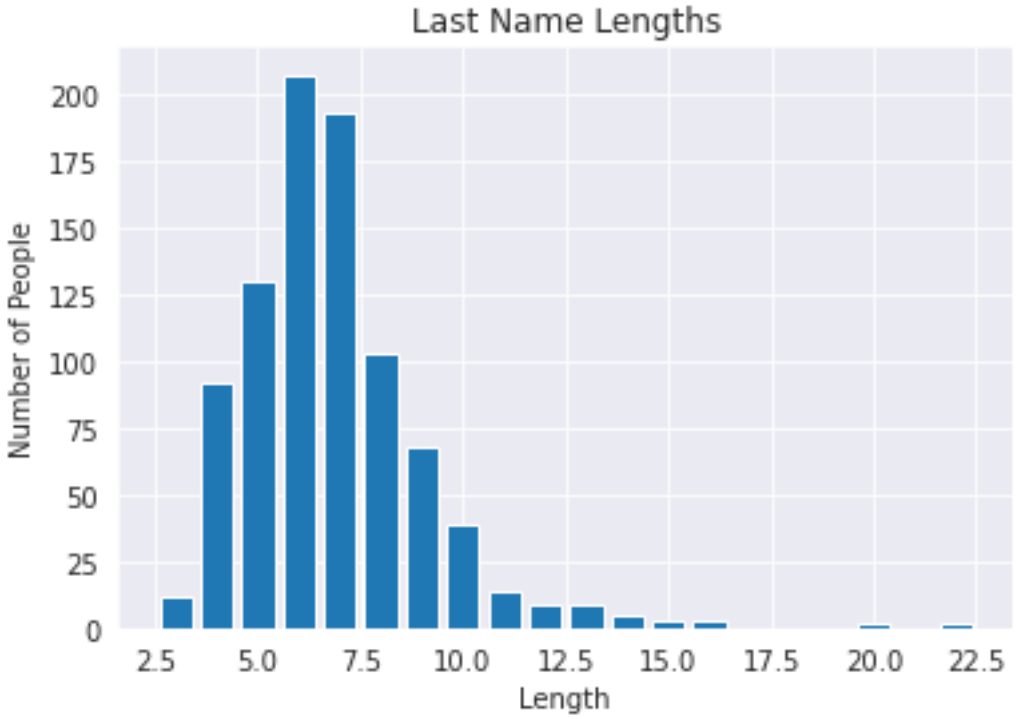
Wow there are some weird ones in there! Who knows what a "jonkheer" is. Let's take a look at the distribution of these titles as well, since some of these are very odd and rare titles.



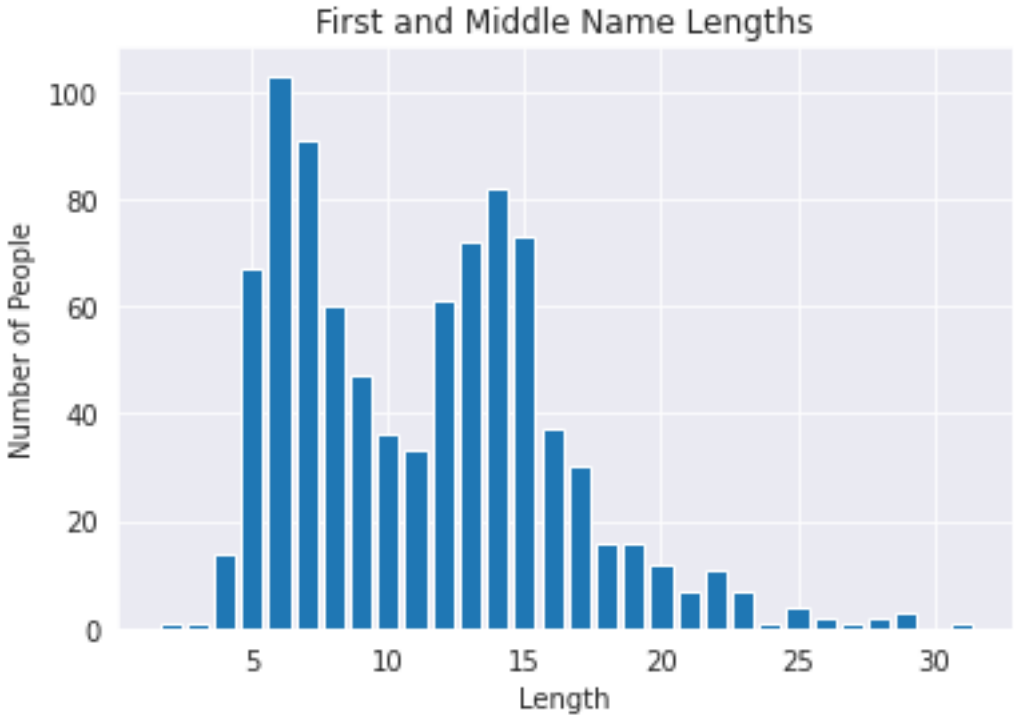
It looks like the bulk of titles are either Mr, Mrs, or Miss, which are the standard titles, and the rest are what I will call "special titles". From this, I believe we can make another feature. I will make a boolean feature, where True will indicate that this element is a "special title" whereas False will indicate it is one of the three common titles.



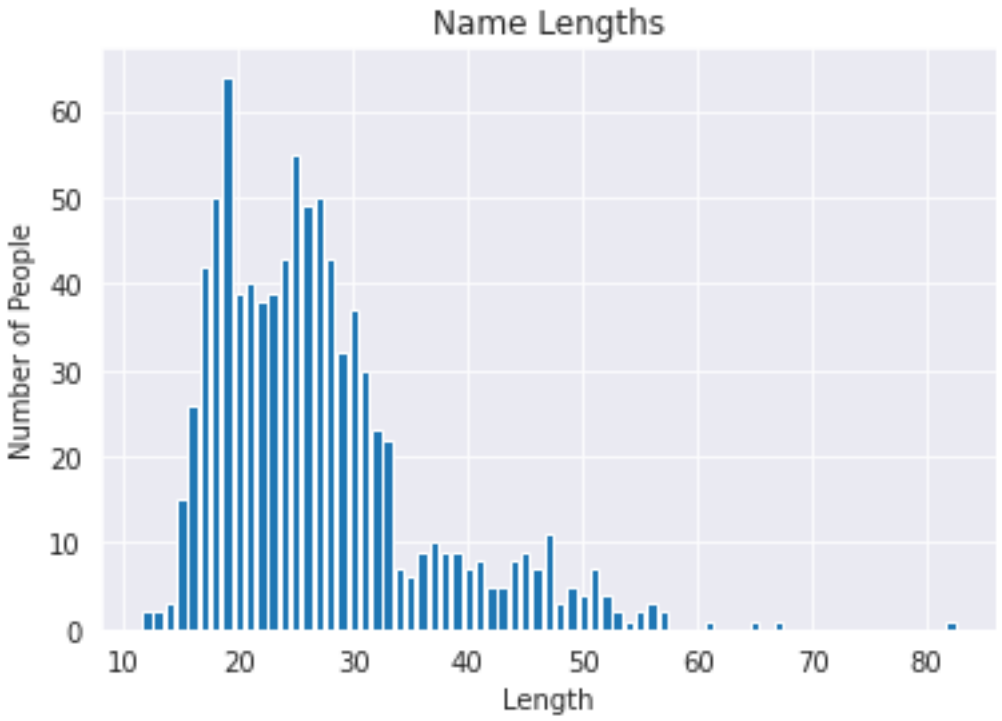
With text, there is also another handy feature to make: The Length. Although this may not be that helpful in predicting the Titantic dataset, it is a handy tool that could be helpful especially for NLP tasks or other classification tasks. Either way, I will create three different features: Last Name Length, First and Middle Name Length, and Total Name Length.



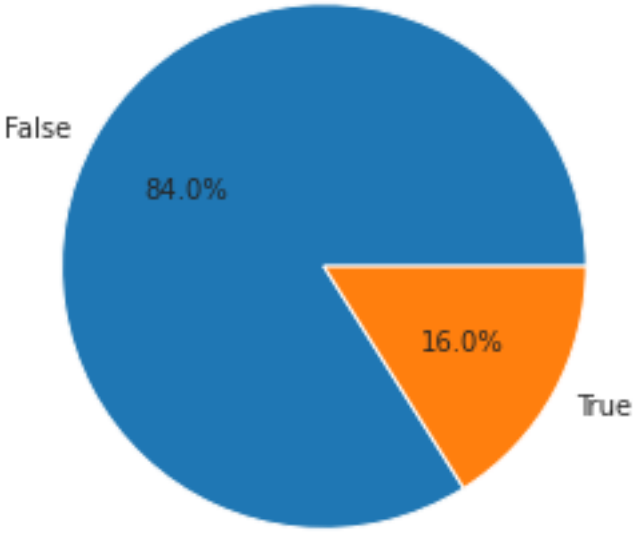
On the graph, we can see there are several outliers towards the 22 and 20 side, which we may have to disinclude or change later on. But as we can see, the bulk of people are centered around the 6-7 characters long.



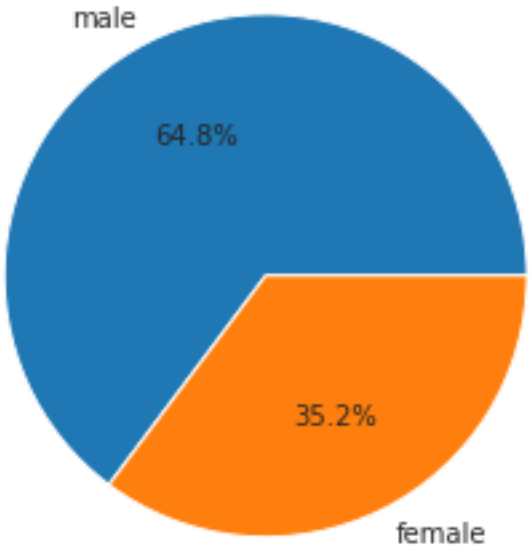
Different from the first graph, we can see two peaks in the graph, one around 6-7 and the other around 14-15. This is an interesting trend that we will have to watch out for.



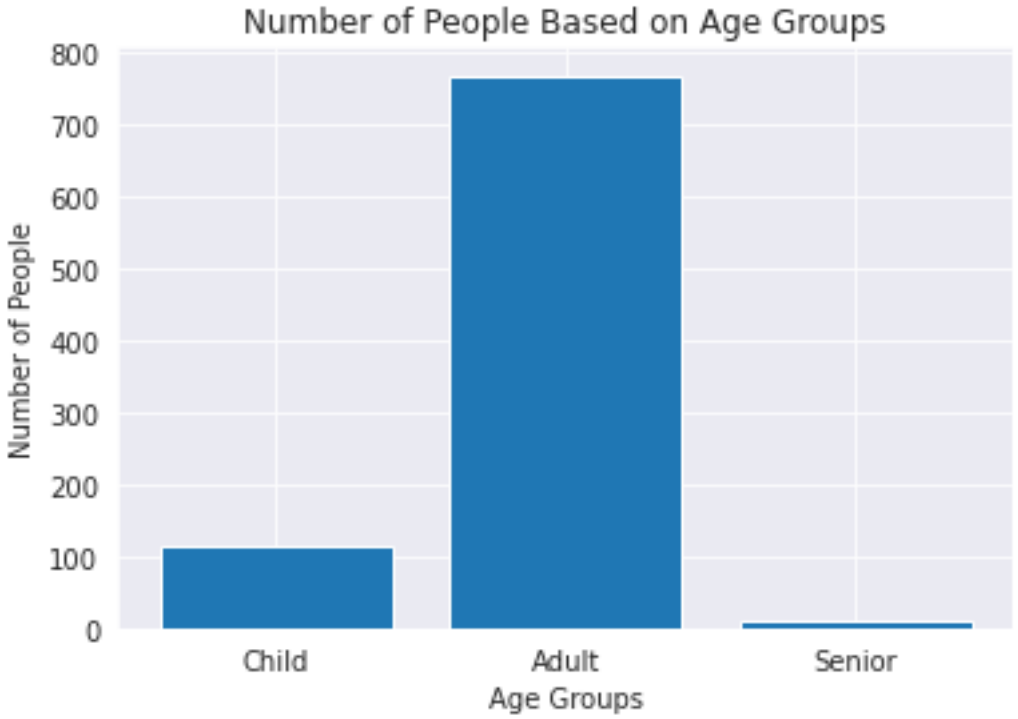
Another interesting feature that I noticed in the name feature was the presence of a "second name", which was denotated by the brackets. This may be an important feature, indicating either higher rank or greater prestige, so I will create a Boolean feature to show whether this person has a "second name" or not.



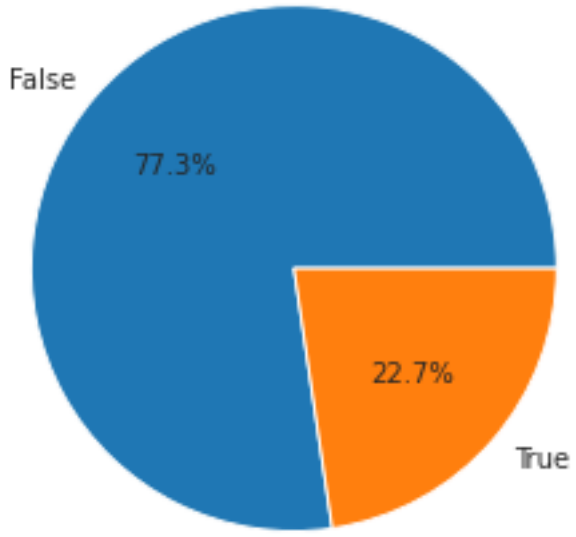
For sex, there is not much we can probe from. The most we can do is assign categorical codes for it. From the pie chart though, it is interesting to see how most of the passengers were male. Maybe the less population of females affected how many of them survived.



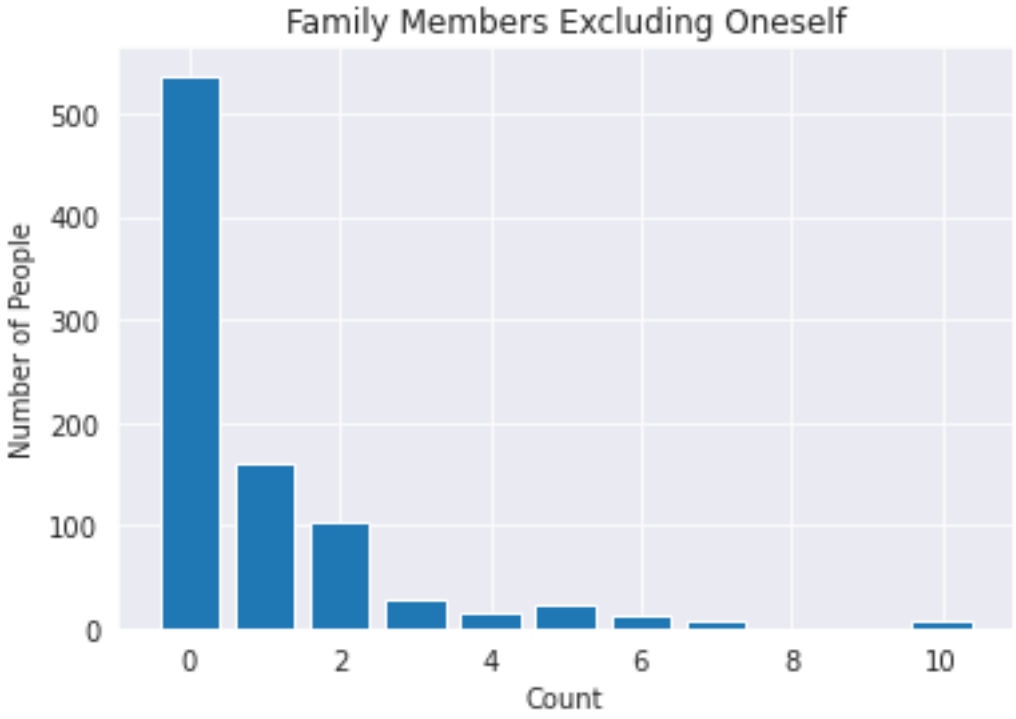
Number of People based on age group.



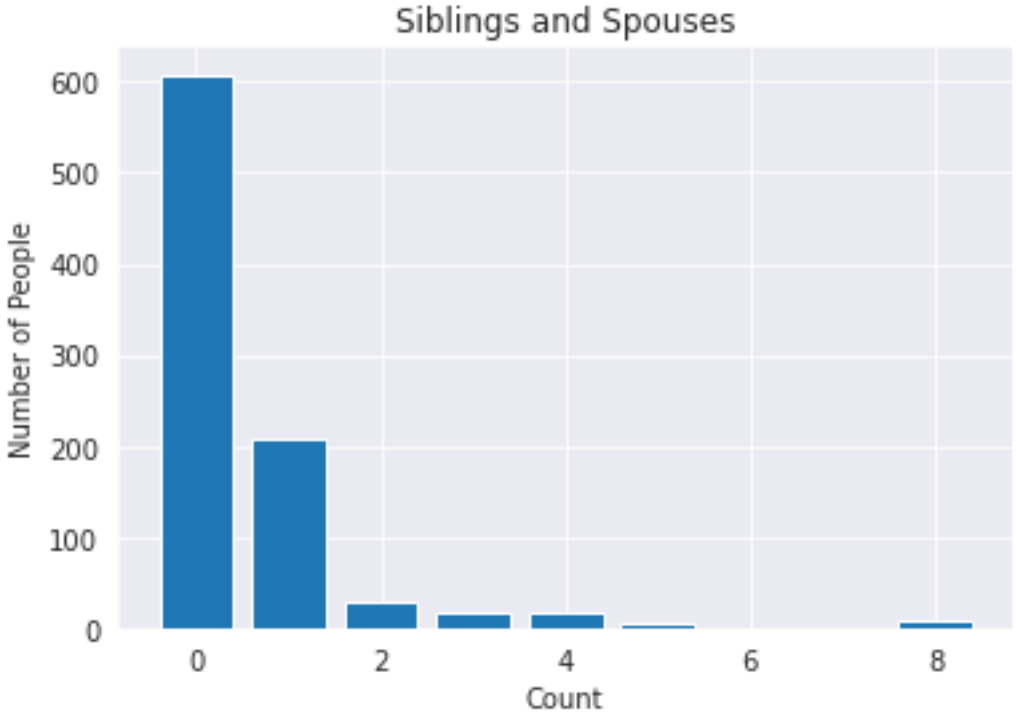
Finally, I will create an 'estimated age' feature. In the documentation of this data, it is stated that if the age ends in .5, the age was estimated. This will be another feature that we can extract that may prove to be useful in the future.



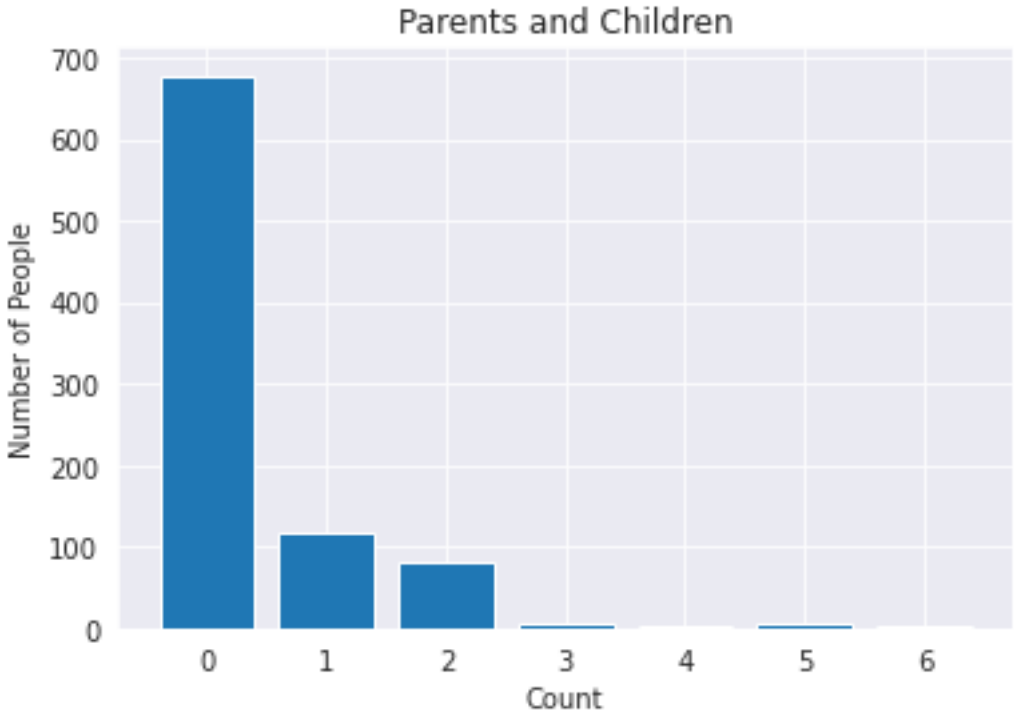
Family member excluding oneself: -



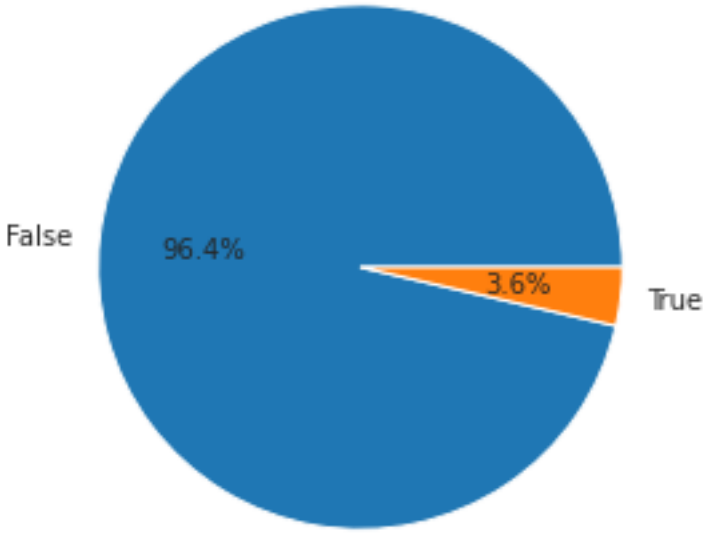
Sibling ans Spouses: -



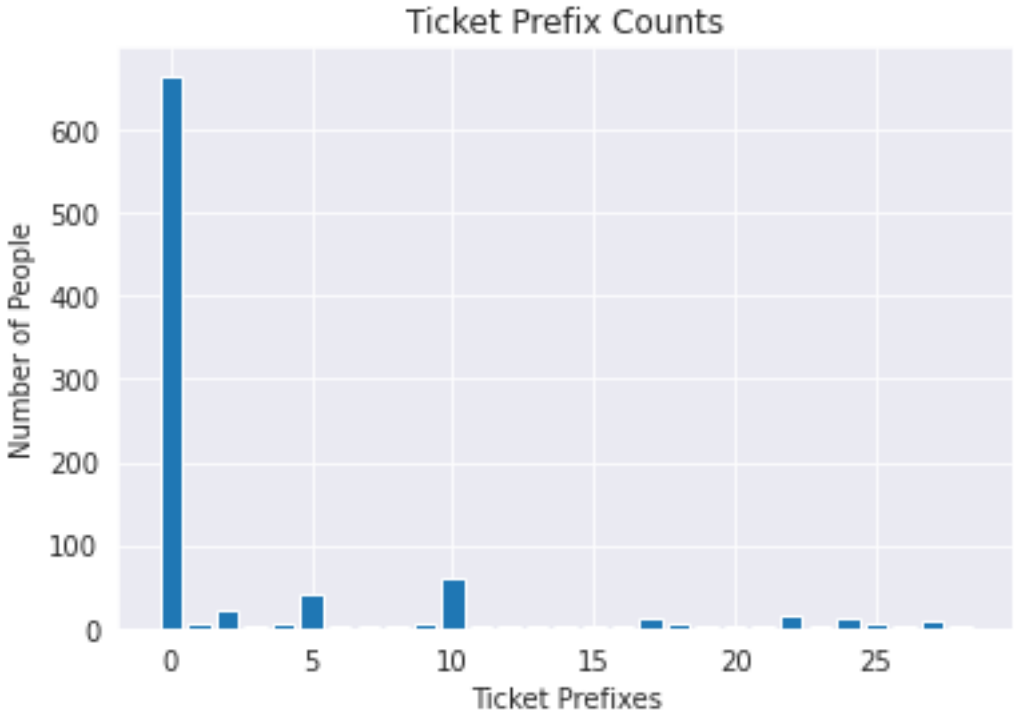
Parents and Children: -



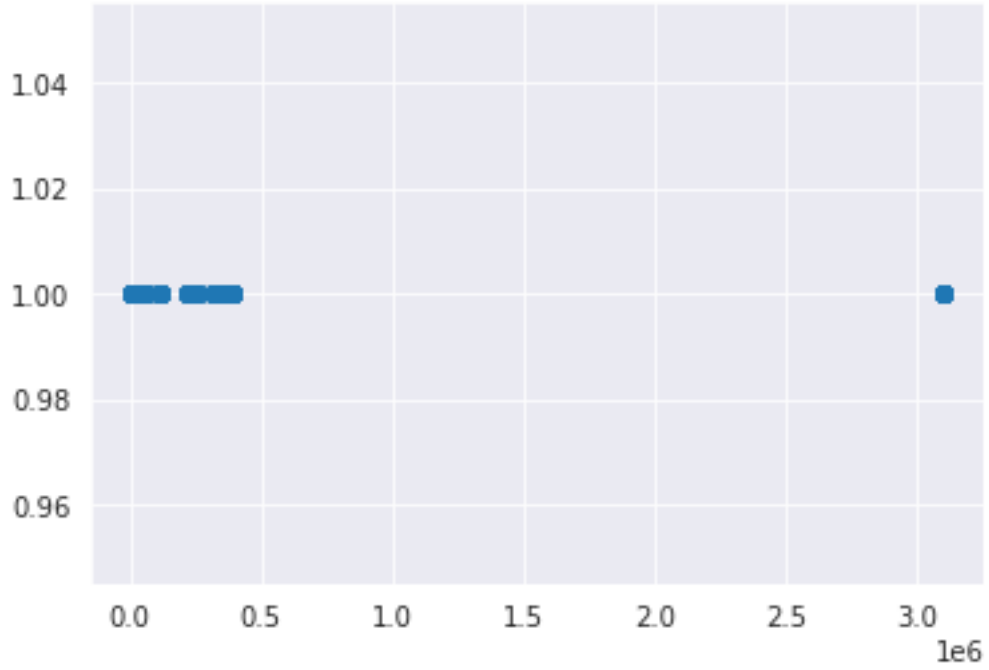
One last interesting feature I can create is from the documentation from the data. It stated how children who have parch as 0 may be travelling with a nanny. It would be helpful to identify these kids, as they may be more likely to be clueless to the events around them without a proper parent taking care of them.



Ticket Prefix Count: -

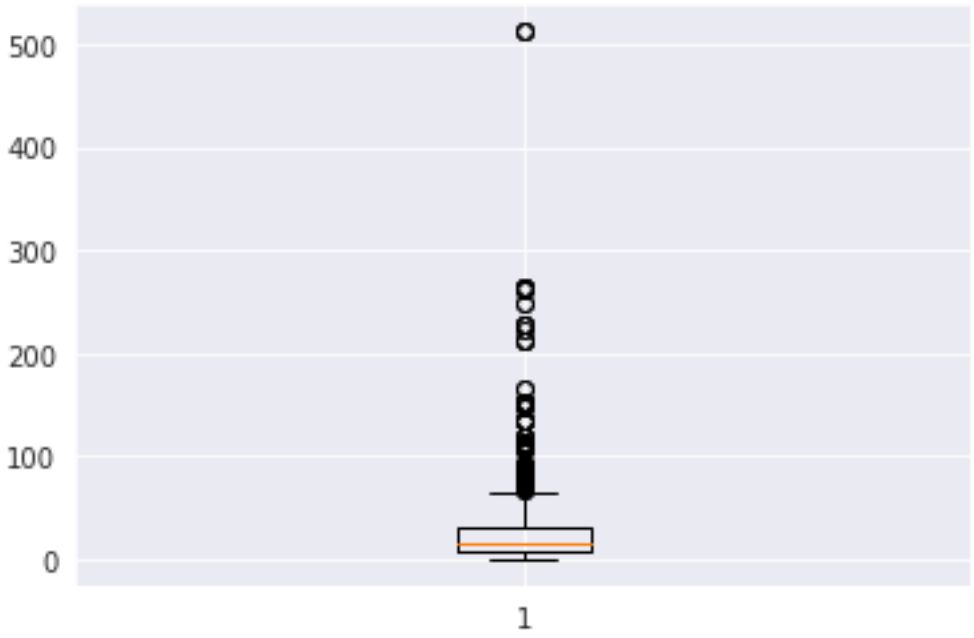


This may not be the best feature for us to use because most of the data is considered null (or 0). But they seems to be clustered around 5 and a little past 10, which could indicate similar types of people.

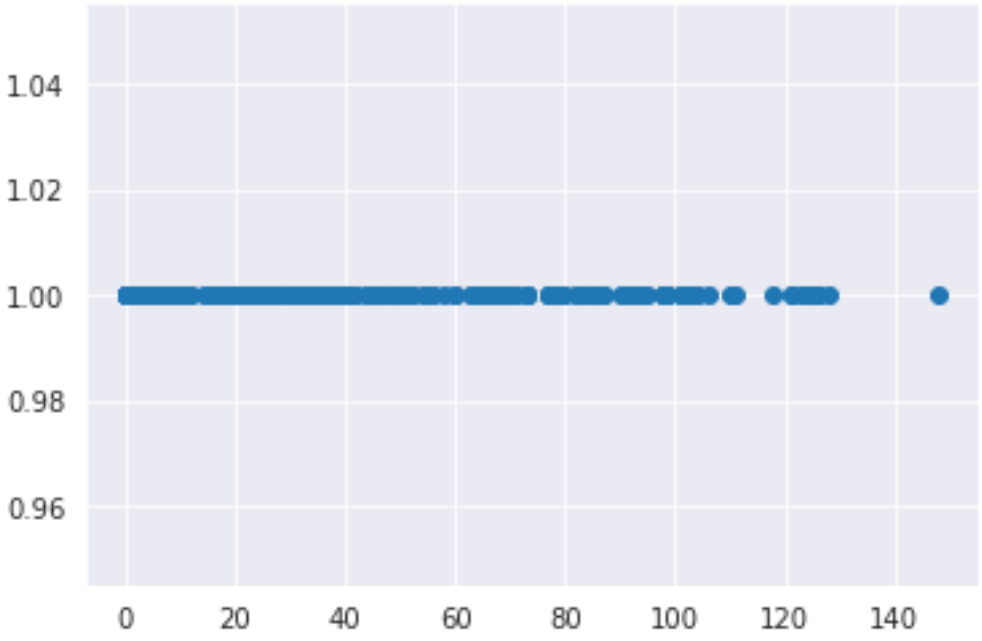


Finally! We have an easier feature to handle. Let's fill in the null values with the mode of the "Fares" feature and visualize the data.

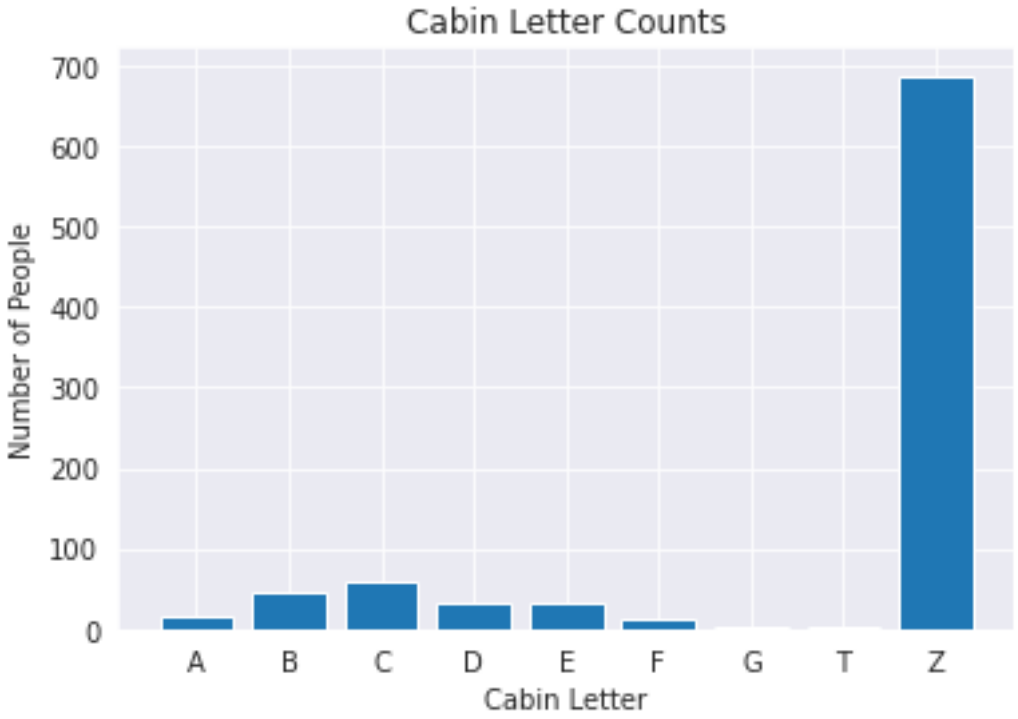
Note 1: I am using the mode because typically, fares are set based on the ticket purchased, so there are many repeated values that would fit each fare. Note 2: I am not making a separate "FareNull" because there are a very small number of null values (Less than 10), which makes the null fare column indiscriminat and unhelpful.



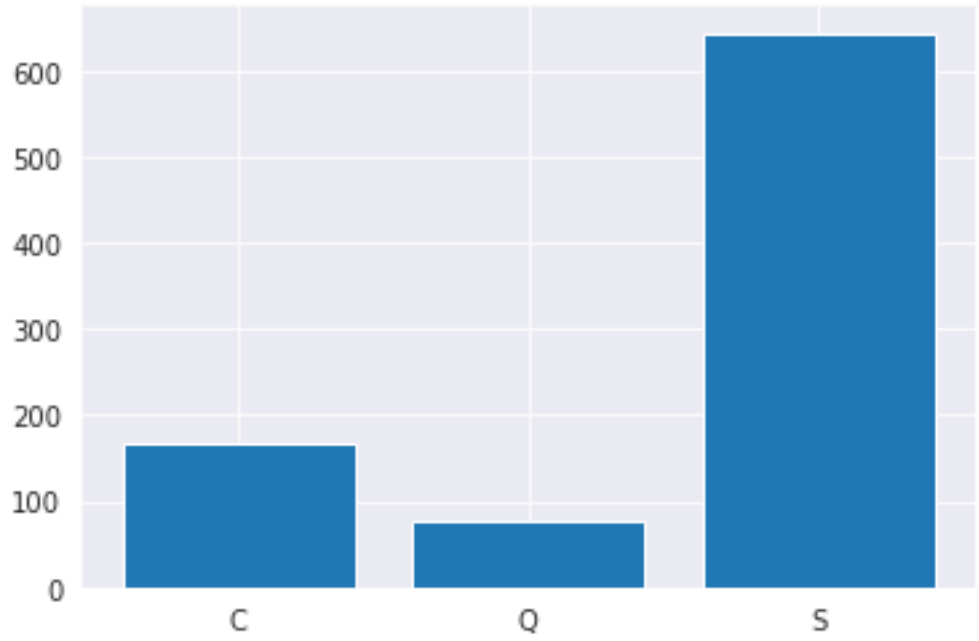
Let's visualize the Cabin Number distribution of the data.



We can see a decently spread out distribution of the Cabin Numbers from the dataset. This may indicate that there is a wide variety of variations in the numbers and may be a strong classification point. Also the location of the Cabin may actually have a huge impact on the probability of the person surviving.
Let's also visualize the count of each prefix in a bar graph.



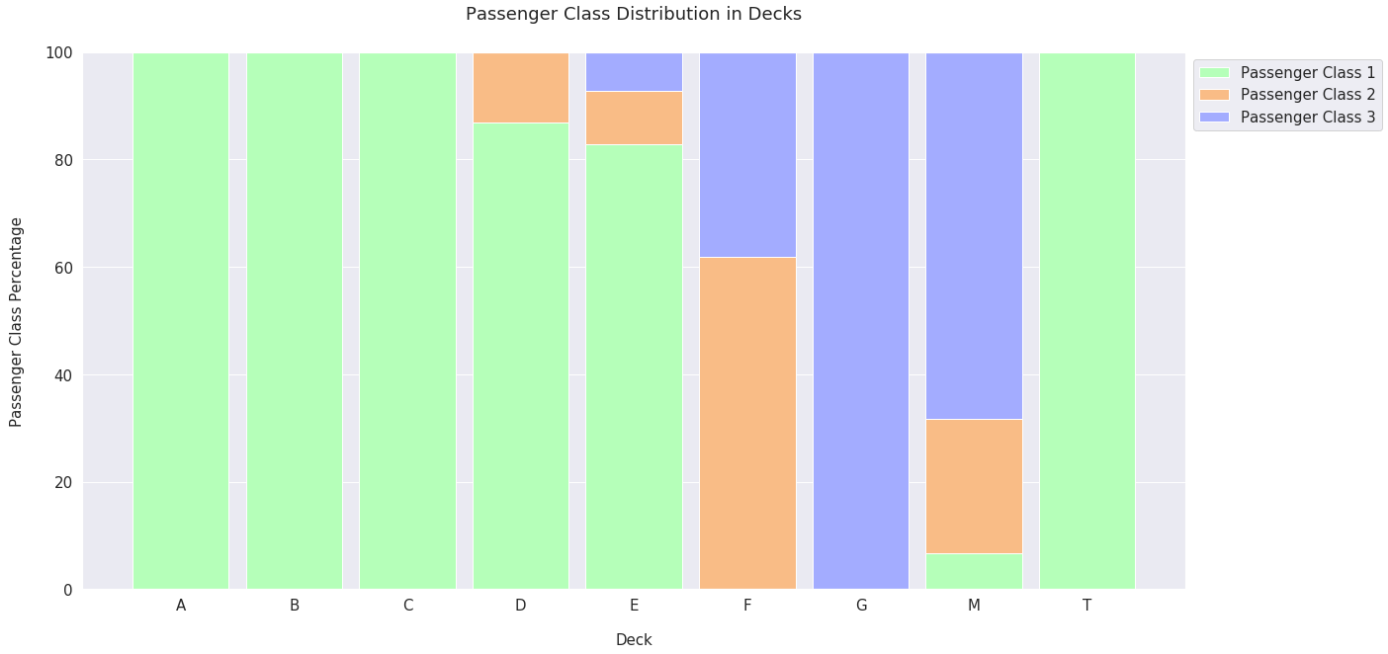
Let's quickly visualize the data before wrapping up.



Now We are going to do Advanced Feature Engineering: -

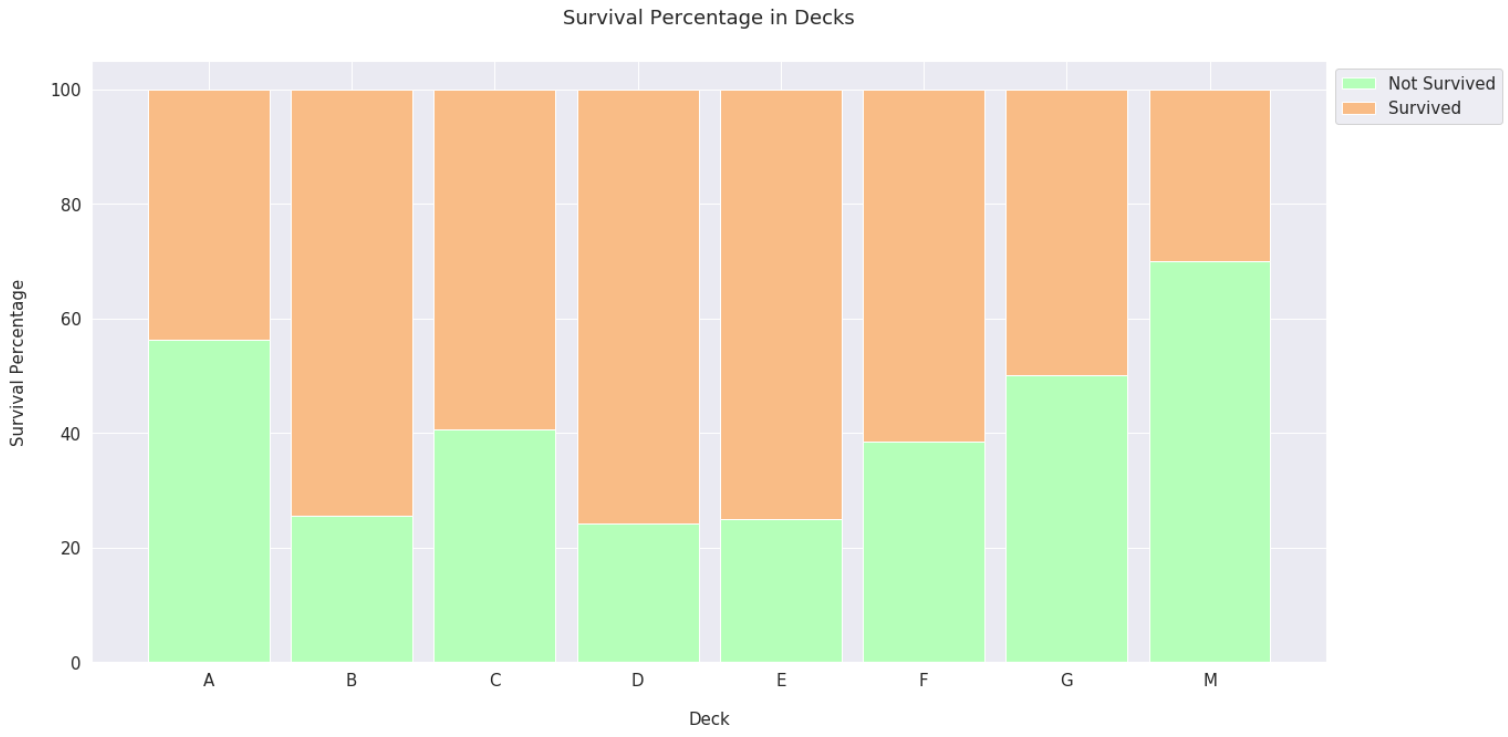
Exploratory Data Analysis: -

First of all, we are going to see passenger distribution in decks.

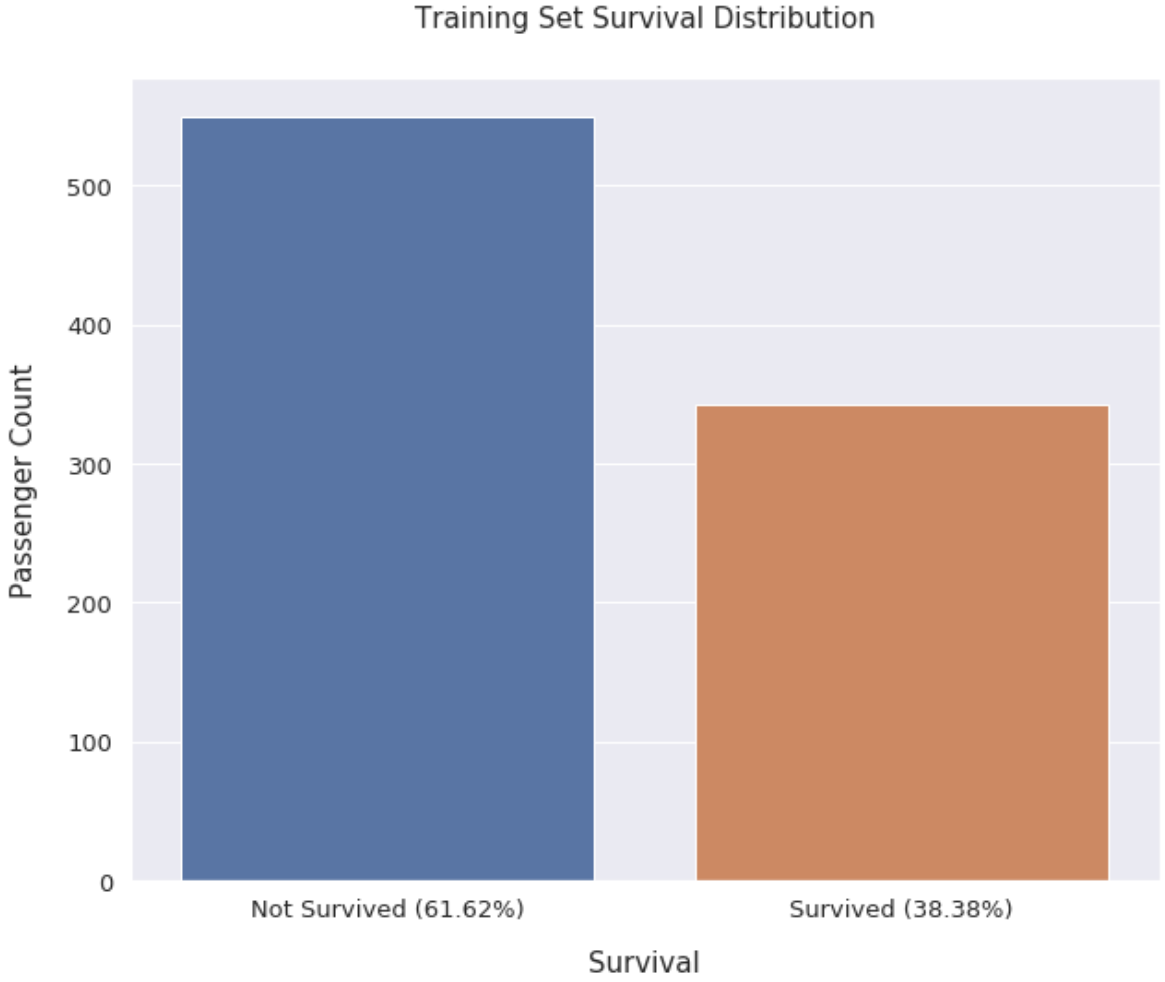


- **100%** of **A**, **B** and **C** decks are 1st class passengers.
- Deck **D** has **87%** 1st class and **13%** 2nd class passengers.
- Deck **E** has **83%** 1st class, **10%** 2nd class and **7%** 3rd class passengers.
- Deck **F** has **62%** 2nd class and **38%** 3rd class passengers.
- **100%** of **G** deck are 3rd class passengers.
- There is one person on the boat deck in **T** cabin and he is a 1st class passenger. **T** cabin passenger has the closest resemblance to **A** deck passengers, so he is grouped with **A** deck
- Passengers labeled as **M** are the missing values in Cabin feature. I don't think it is possible to find those passengers' real Deck so I decided to use **M** like a deck

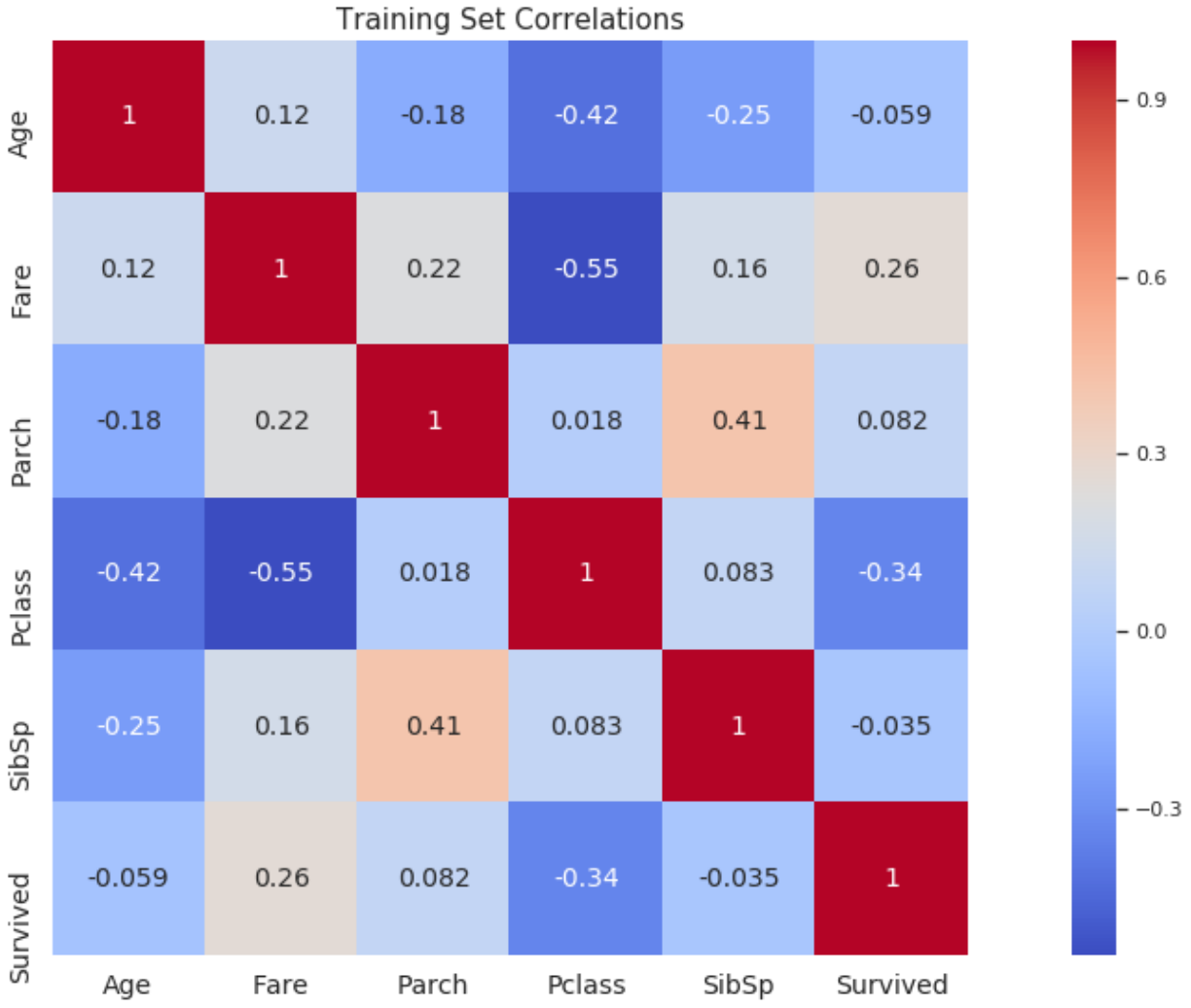
Survival percentage of deck:

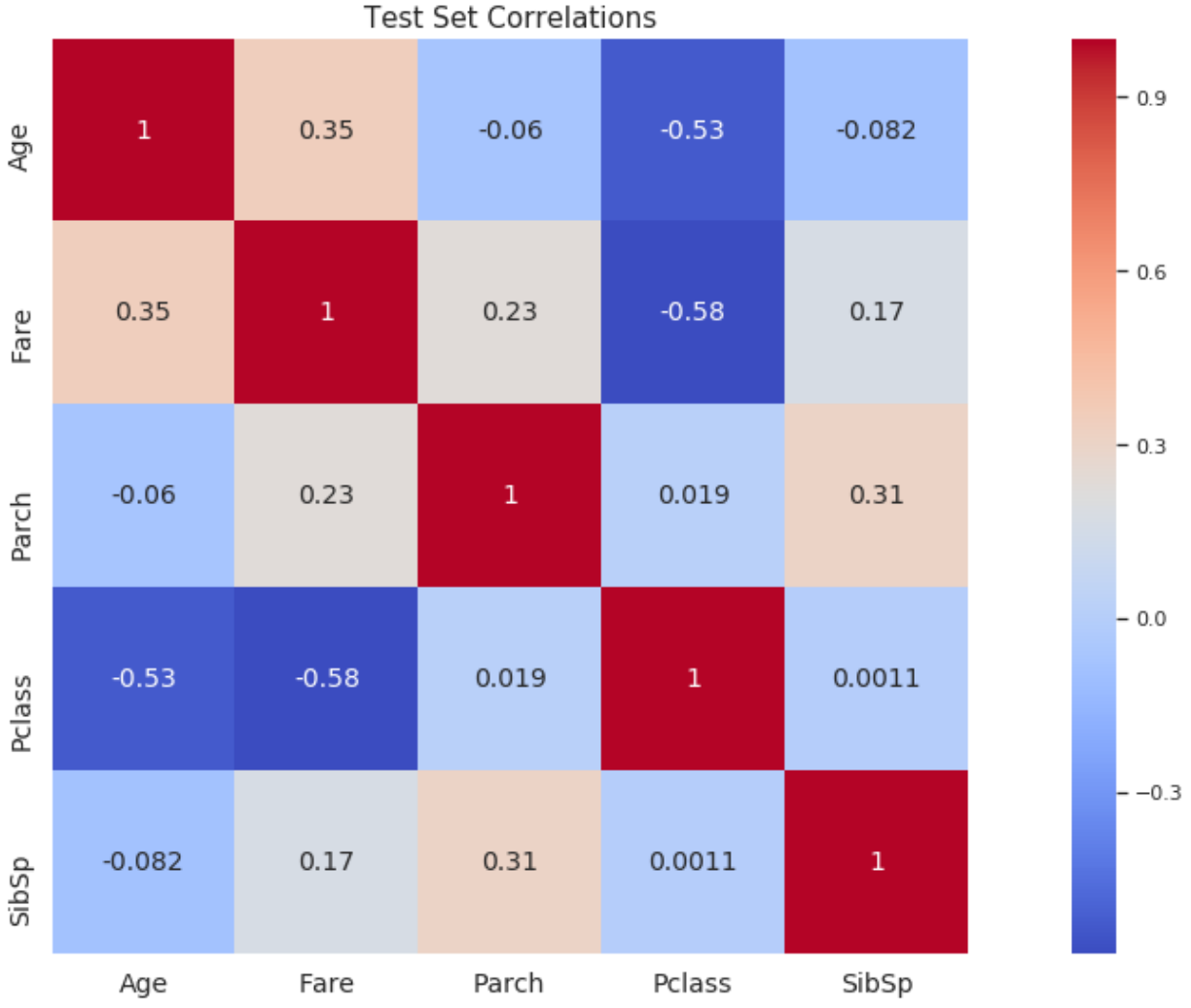


Training set Survival percentage:

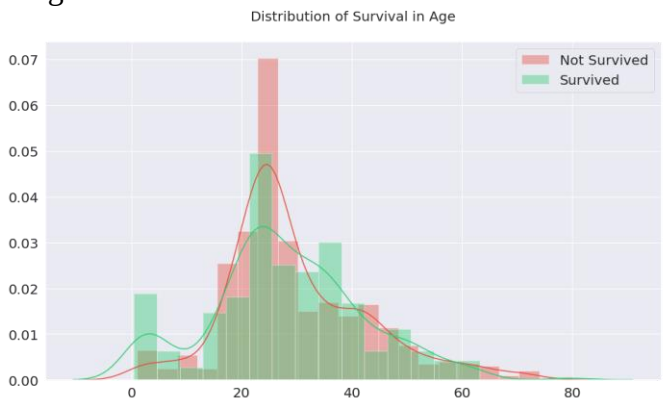


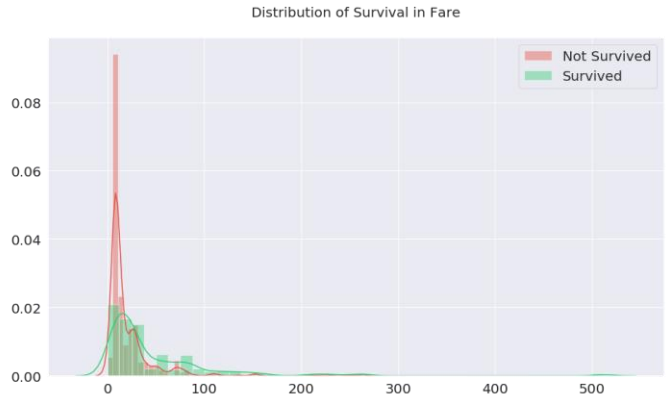
Training Set and Testing Set Correlation: -

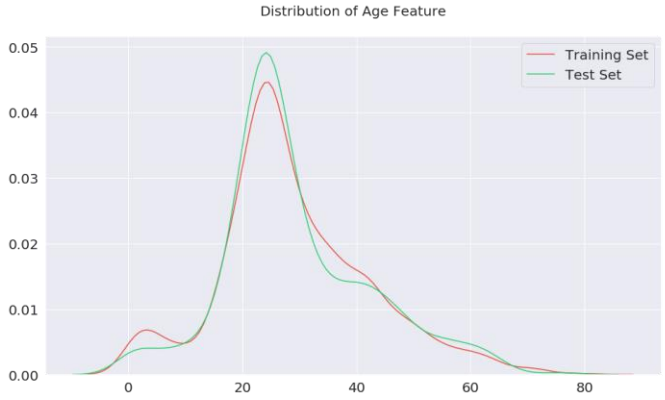

Training Set Correlations
Age
Fare
Parch
Pclass
SibSp
Survived
1 0.12 -0.18 -0.42 -0.25 -0.059
0.12 1 0.22 -0.55 0.16 0.26
-0.18 0.22 1 0.018 0.41 0.082
-0.42 -0.55 0.018 1 0.083 -0.34
-0.25 0.16 0.41 0.083 1 -0.035
-0.059 0.26 0.082 -0.34 -0.035 1
Age Fare Parch Pclass SibSp Survived
0.9
0.6
0.3
0.0
−0.3

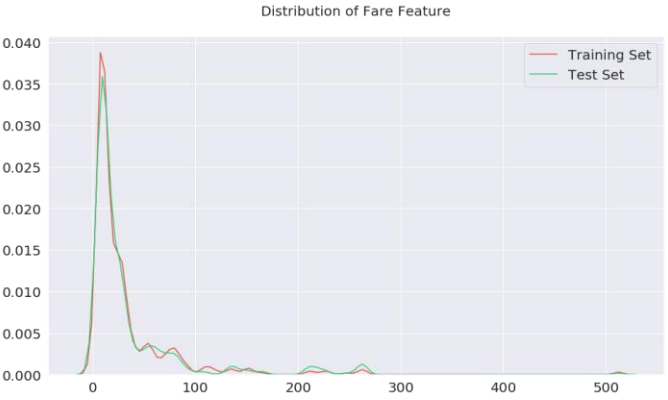

Test Set Correlations
Age
Fare
Parch
Pclass
SibSp
1 0.35 -0.06 -0.53 -0.082
0.35 1 0.23 -0.58 0.17
-0.06 0.23 1 0.019 0.31
-0.53 -0.58 0.019 1 0.0011
-0.082 0.17 0.31 0.0011 1
Age Fare Parch Pclass SibSp
0.9
0.6
0.3
0.0
−0.3

Target Distribution in Features:-

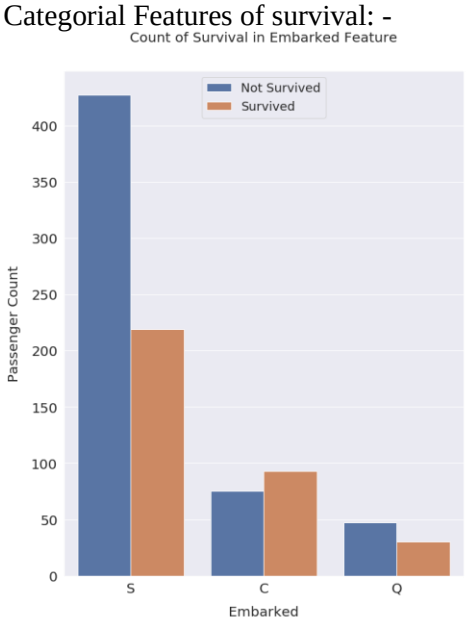


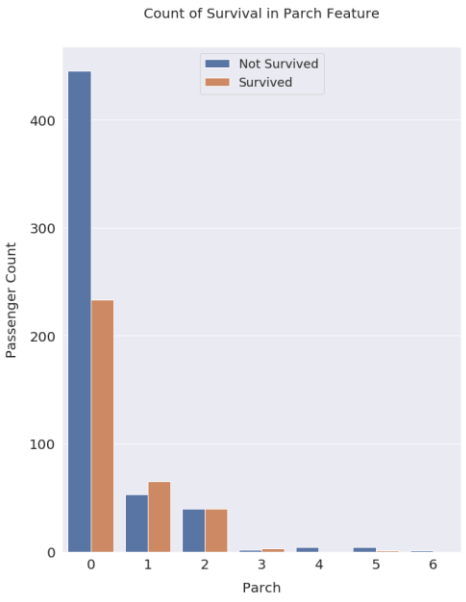


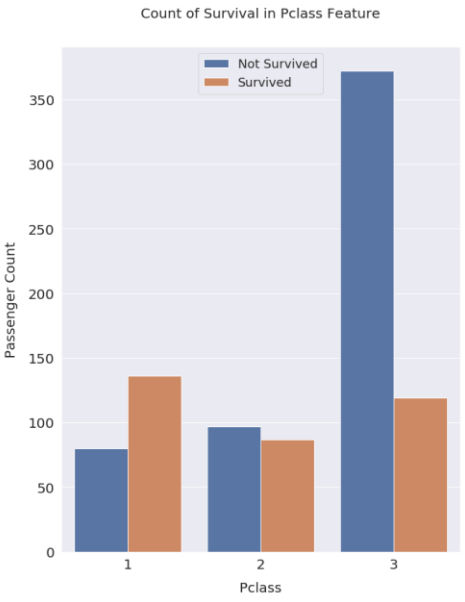


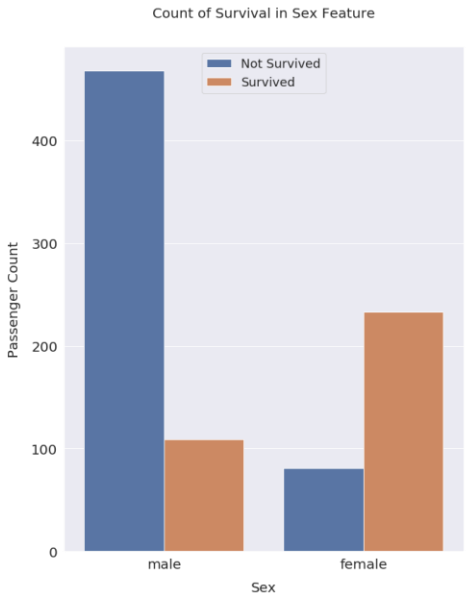


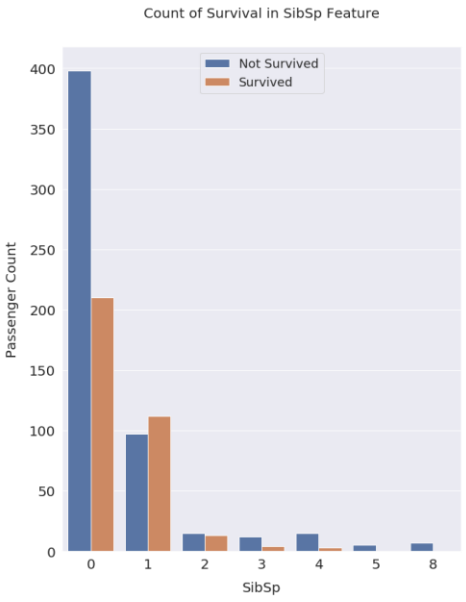
Categorial Features of survival: -

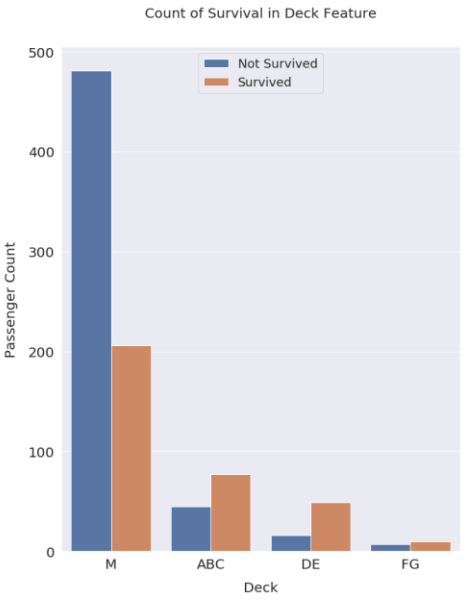




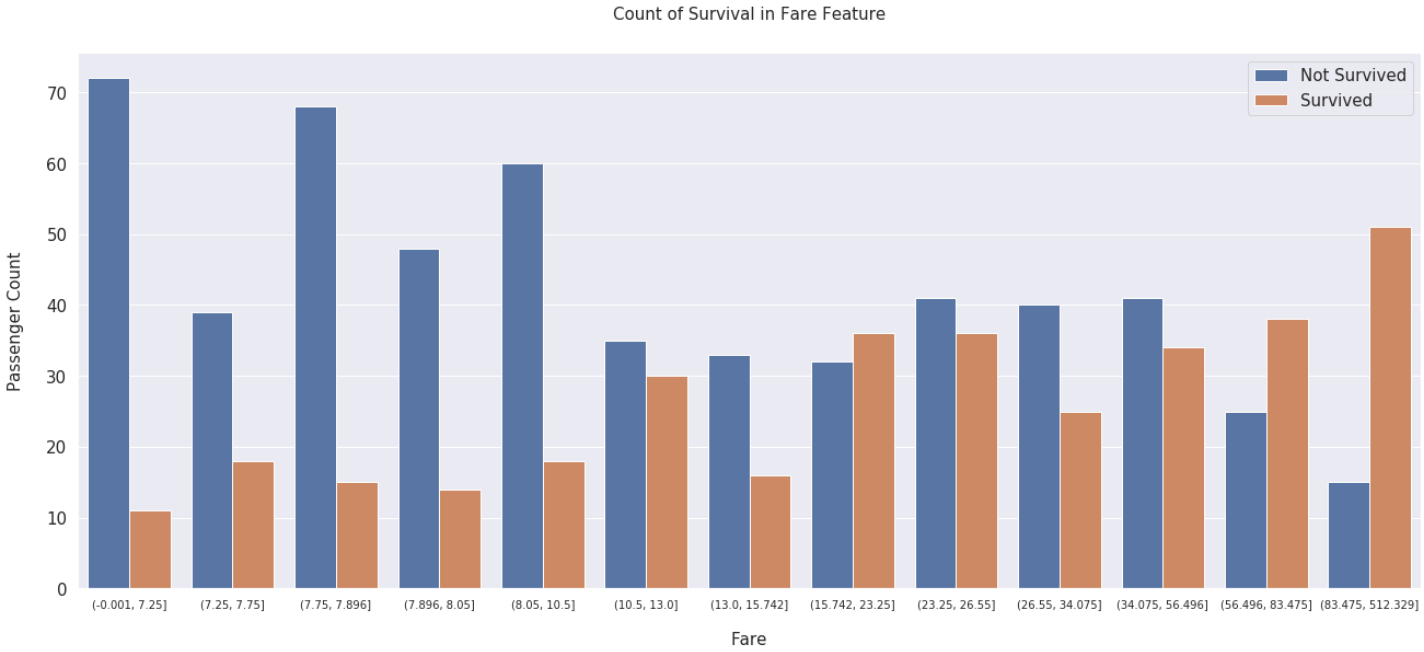




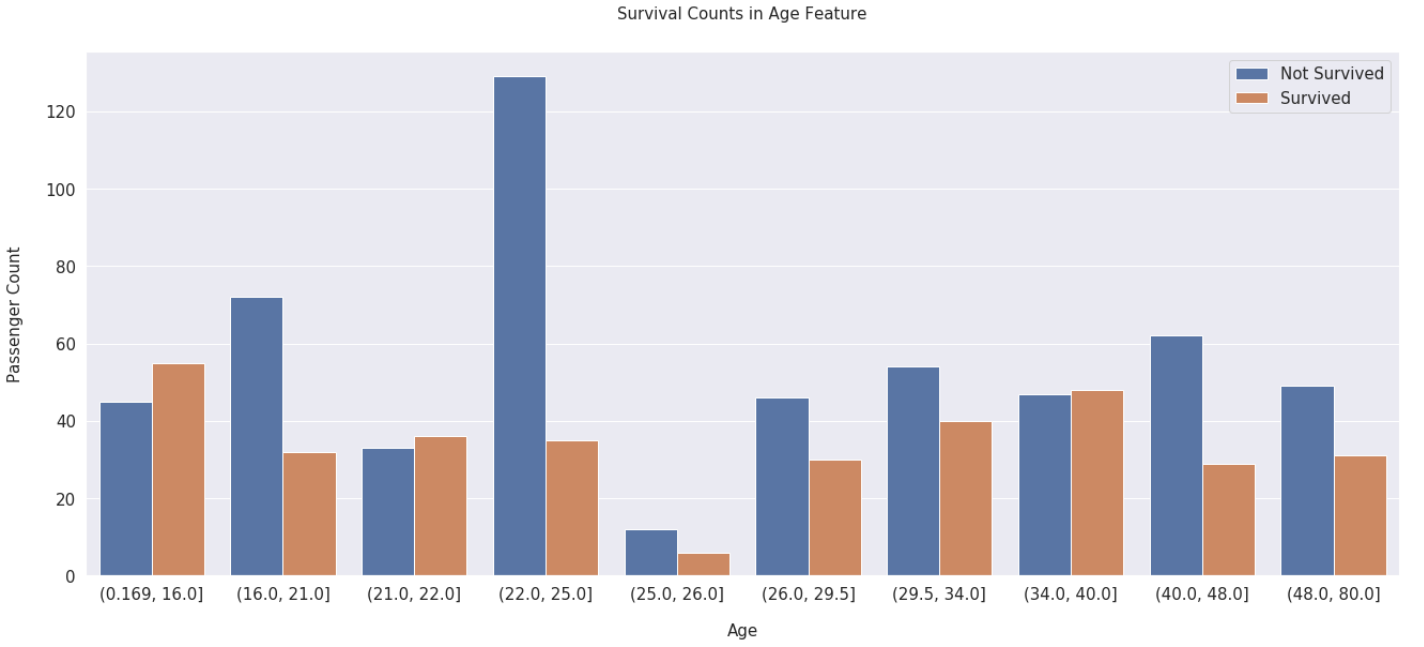




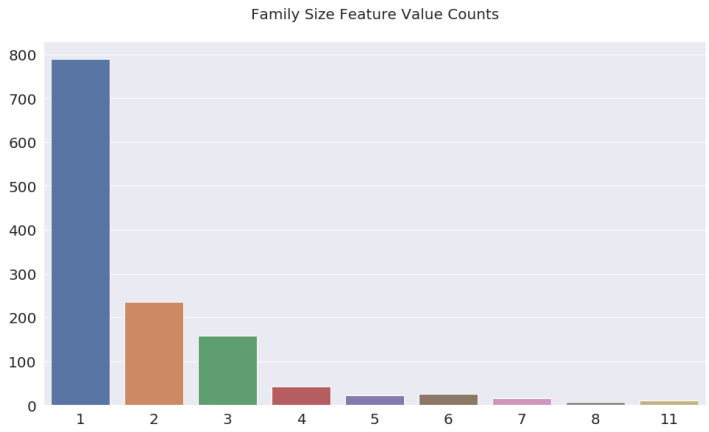
Count of Survival in Fare Features:-

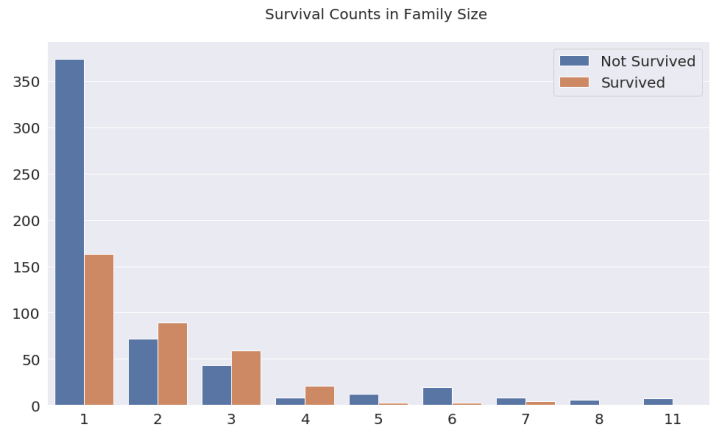


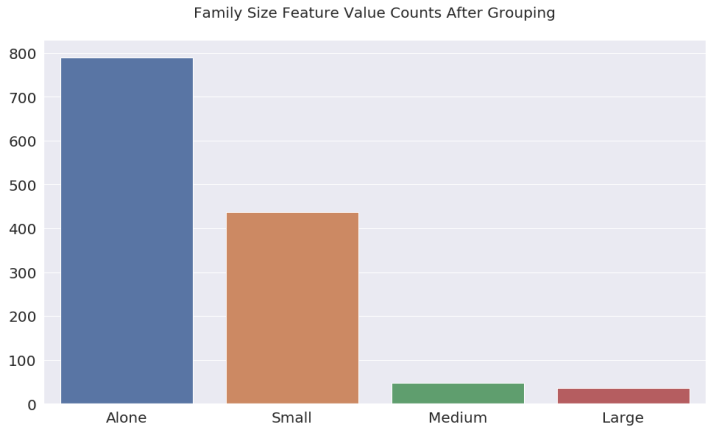
Survival Count by age features: -

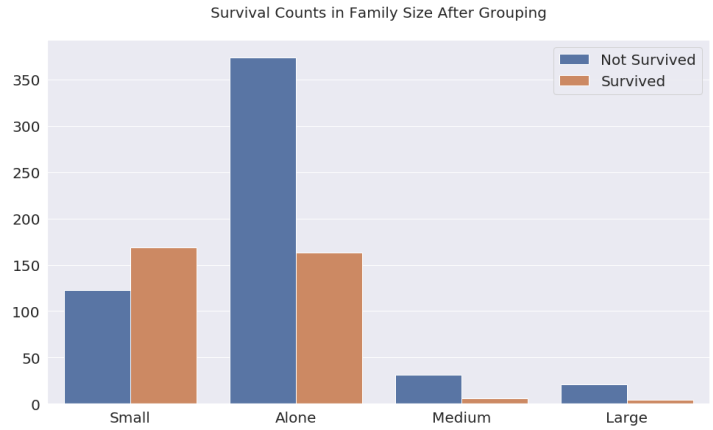


Frequency Encoding: -

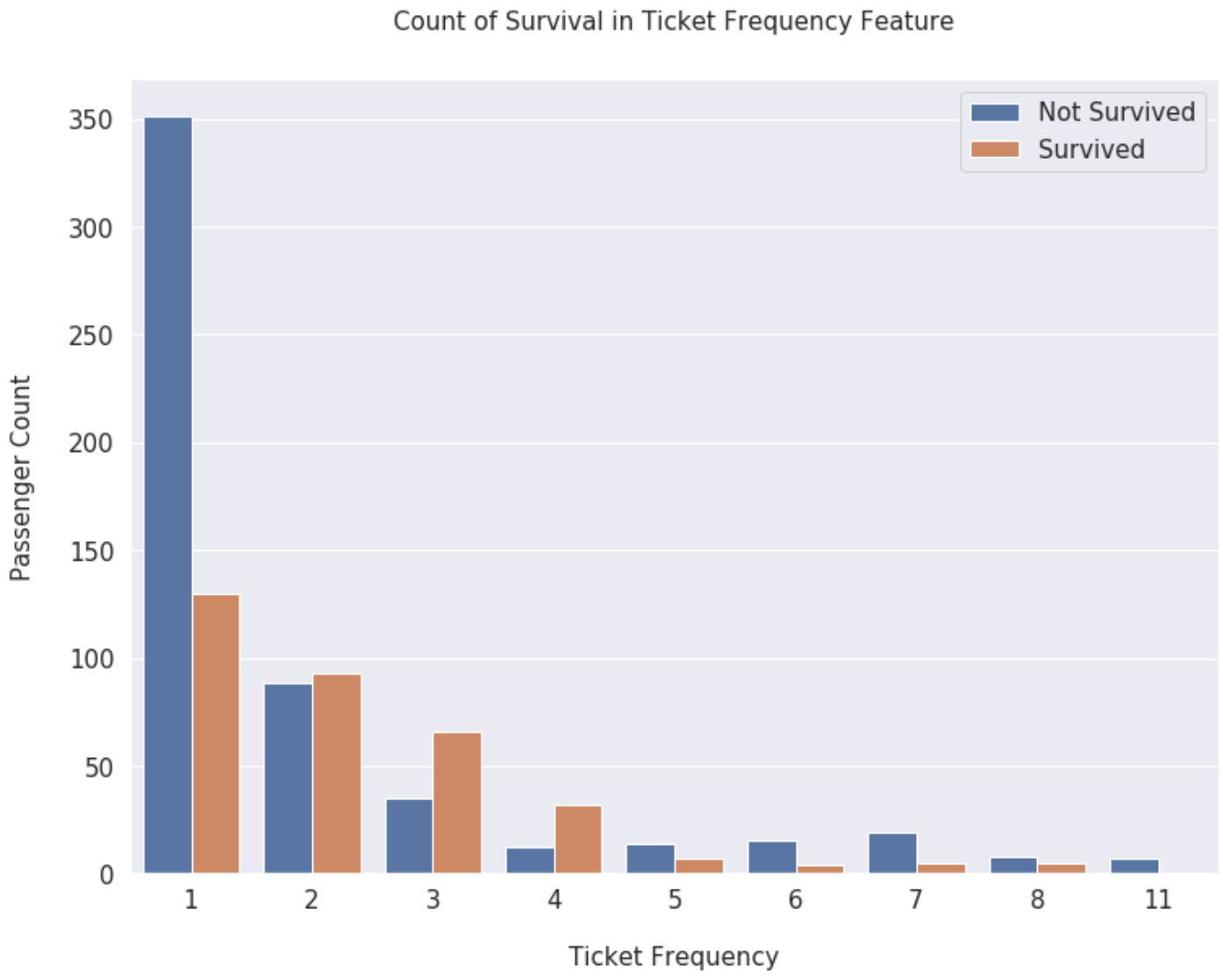




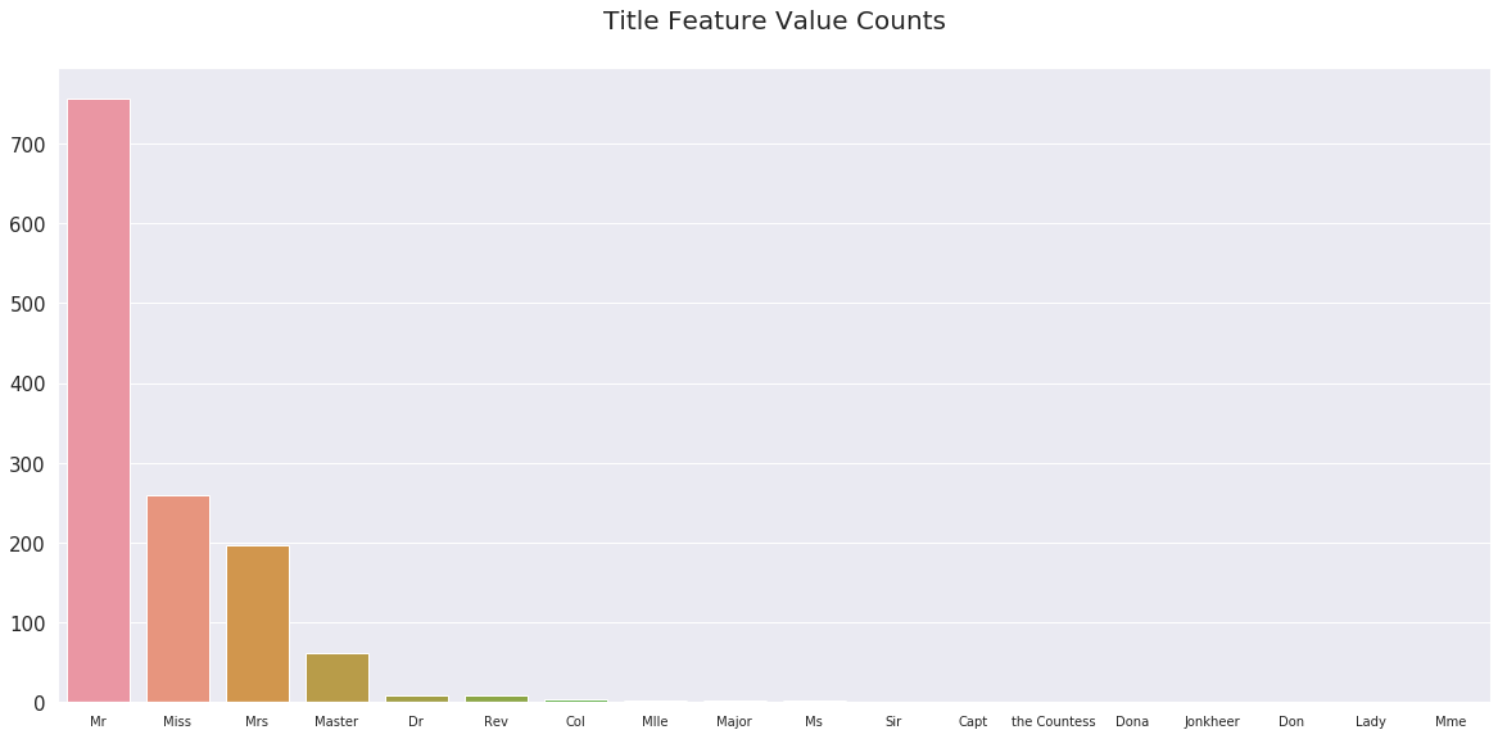


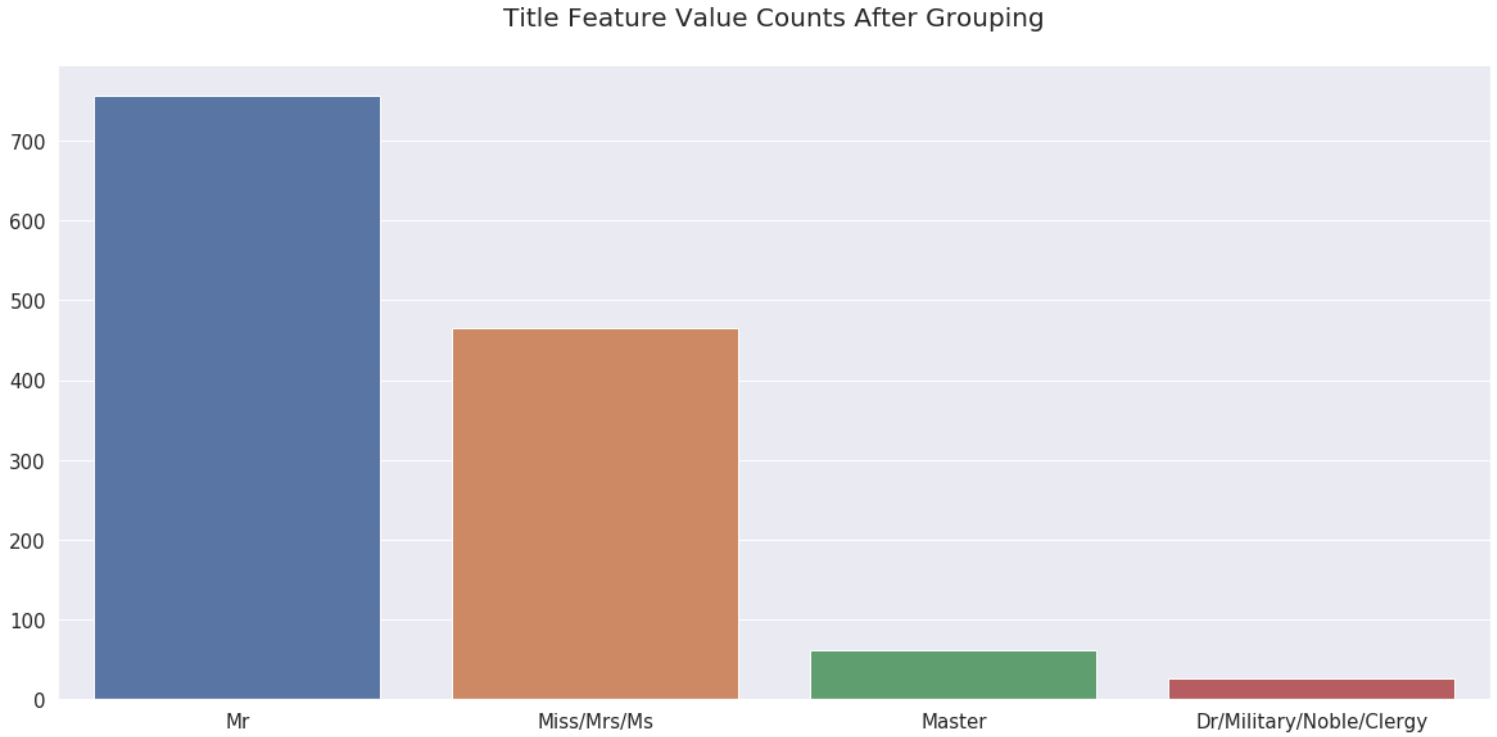


Count of Survival in Ticket Frequency Rate:-



Title and is married: -





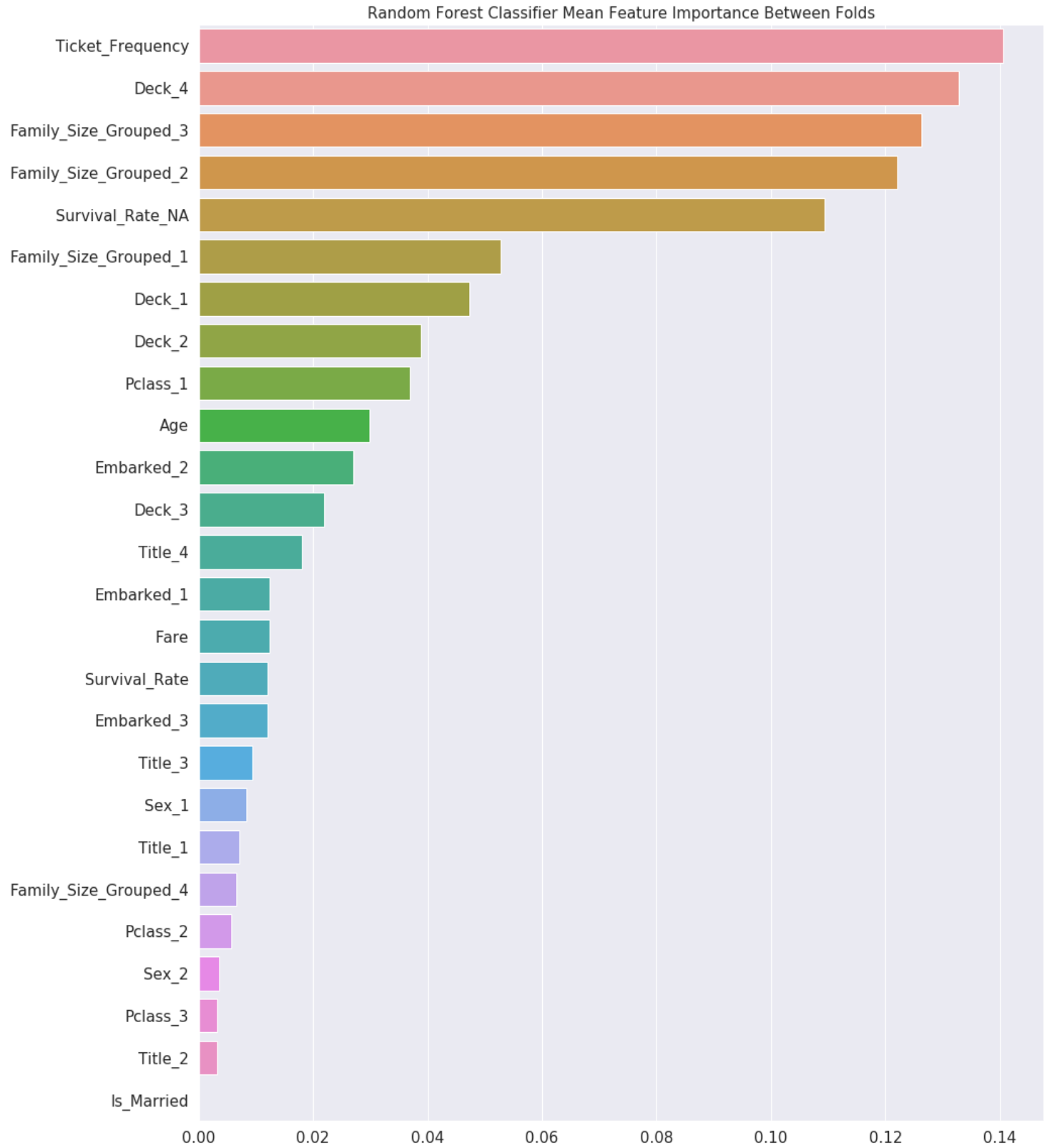
Random Forest Classifier:-

Created 2 RandomForestClassifier's. One of them is a single model and the other is for k-fold cross validation.

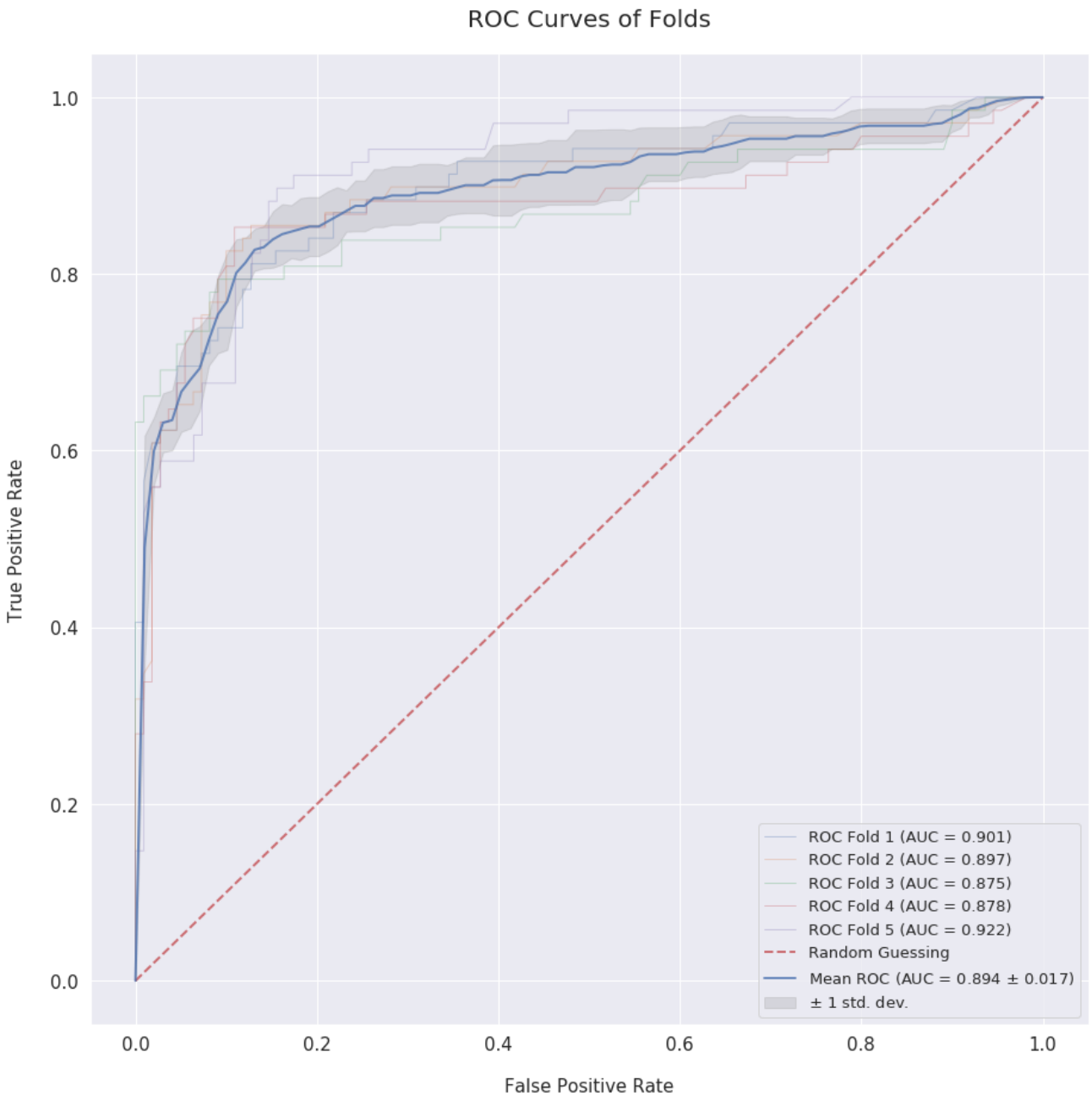
The highest accuracy of the single_best_model is **0.82775** in public leaderboard. However, it doesn't perform better in k-fold cross validation. It is a good model to start experimenting and hyperparameter tuning.

The highest accuracy of leaderboard_model is **0.83732** in public leaderboard with 5-fold cross validation. This model is created for leaderboard score and it is tuned to overfit slightly. It is designed to overfit because the estimated probabilities of X_test in every fold are going to be divided by **N** (fold count). If this model is used as a single model, it would struggle to predict lots of samples correctly.

Feature Importance: -



ROC Curves of fold



Validation Techniques:

- **Holdout Validation**: Split the dataset into training and testing sets, as you've done previously, and evaluate the model on the testing set.
- **Cross-Validation**: More robust than the holdout method, cross-validation (such as k-fold cross-validation) splits the dataset into several segments, using each in turn for testing while training on the remaining segments. This approach provides a more reliable estimate of the model's performance.

Recommendations for future work or improvements to the model:

Data Augmentation

1. **Synthetic Data Generation**: Techniques like SMOTE (Synthetic Minority Over-sampling Technique) could be used to generate synthetic examples, particularly to address class imbalance issues.
2. **External Data**: Incorporating external data sources, such as historical records on lifeboat capacities, crew member duties, or layout specifics of the Titanic, could provide additional context for predictions.

Fairness and Bias Mitigation

1. **Fairness Analysis**: Conduct a thorough fairness analysis to ensure that model predictions do not disproportionately favor or disadvantage any group of passengers based on sensitive attributes.
2. **Bias Mitigation Techniques**: Implement and evaluate bias mitigation techniques at various stages of the modeling process (pre-processing, in-processing, post-processing) to ensure equitable predictions.

Validation and Evaluation

1. **Cross-Validation**: Employ more robust cross-validation techniques, such as stratified k-fold cross-validation, to ensure stable and reliable performance estimates.
2. **Alternative Evaluation Metrics**: Explore alternative metrics that might be more appropriate given the problem context, such as the area under the precision-recall curve (AUPRC) for imbalanced datasets.

Interpretability and Explainability

1. **Model Interpretation**: Use tools like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) to gain deeper insights into how features influence model predictions.
2. **Domain Expert Collaboration**: Collaborate with historians or maritime experts to interpret model findings within the historical context of the Titanic disaster, ensuring that conclusions drawn are plausible and respectful of the event's gravity.

Ethical Considerations

1. **Ethical Review**: Regularly review the ethical implications of the model, particularly considering the historical context and sensitivity surrounding the Titanic disaster.
2. **Transparent Reporting**: Ensure transparent reporting of model limitations, the potential for historical bias in the data, and the steps taken to address these issues.

Ethical Standards

1. **Respect for Privacy**: Ensure that any data used, even if historical, is handled with respect for the individuals it represents. Avoid using personally identifiable information unless it is necessary and publicly available.
2. **Transparency**: Be clear about the limitations of your data and models. Document the sources of your data, the assumptions made during modeling, and the potential limitations of your findings.
3. **Accountability**: Take responsibility for the models you create and their potential impact. This includes being prepared to address any unintended consequences that may arise from the use of your model.
4. **Historical Context**: Recognize the historical context of the Titanic disaster and the real human tragedy it represents. Ensure that any analysis, presentation, or discussion of your work is conducted with sensitivity and respect for those who lost their lives and their descendants.

Fairness Considerations

1. **Bias Detection and Mitigation**: Actively search for and mitigate biases in your data and model. This includes biases related to class, gender, age, or any other sensitive attribute that could lead to unfair or discriminatory outcomes.
2. **Fairness Metrics**: Utilize fairness metrics to evaluate your model's performance across different groups defined by sensitive attributes. Adjust your model as necessary to ensure equitable treatment and outcomes for all groups.
3. **Diverse Perspectives**: Engage with a diverse range of perspectives when developing and reviewing your model. This can include collaborating with team members from different backgrounds or consulting with external experts to identify potential ethical or fairness issues.
4. **Model Interpretability**: Strive for model interpretability, especially for models that might inform decision-making processes. Understanding how and why your model makes certain predictions is key to identifying and correcting biases.
5. **Continuous Monitoring**: Understand that fairness and ethical considerations are not one-time checks but require ongoing monitoring and adjustment as the model is developed, deployed, and updated.
6. **Inclusive Design and Development**: Involve stakeholders, including potentially affected groups, in the design and development process. This can help ensure that the model addresses a wide range of needs and concerns and is more likely to be accepted and used responsibly.
7. **Ethical Review**: Consider setting up an ethical review process for your project, involving individuals who can assess the ethical implications of your work from multiple perspectives.

Adhering to these ethical standards and fairness considerations ensures that your work on the Titanic dataset or any other data science project contributes positively to the field and respects the individuals and stories behind the data.

Conclusion: -

So, I have done the feature engineering on titanic dataset and extracted all the features of dataset and also did EDA using Supervised learning. In short, all the project requirements have been fulfilled.