# Module 3 – Technique Practice

## Navigating Survival: Analytical Insights and Predictive Modeling of Titanic Passenger Outcomes

**Professor - Justin Grosz**

**Submitted by - Kumar Saransh**

**Northeastern University: College of Professional Studies**

**ALY 6040: Data Mining Applications**

**9th March, 2025**

# Abstract

This report analyzes the Titanic dataset to understand factors contributing to passenger survival. Through data preprocessing, exploratory data analysis (EDA), principal component analysis (PCA), and modeling, we identify key variables influencing survival rates. Our findings highlight the importance of demographic factors and provide actionable insights for improving future survival predictions.

# Introduction

The sinking of the Titanic is one of the most infamous maritime disasters in history, resulting in significant loss of life. Analyzing the survival patterns of passengers can provide valuable insights into how demographic and socioeconomic factors influence survival rates during emergencies. This study aims to explore these factors using machine learning techniques and provide recommendations for enhancing safety protocols.

# Persona

This analysis targets stakeholders interested in understanding survival patterns during emergencies, such as policymakers, safety experts, and researchers. The insights gained can inform strategies for improving survival rates in similar scenarios.

# Data Overview

The Titanic dataset used in this analysis contains information about passengers on the ill-fated voyage of the RMS Titanic. The dataset includes a total of 891 entries across 12 columns, providing a comprehensive view of passenger demographics, travel details, and survival outcomes.

# Data Preprocessing

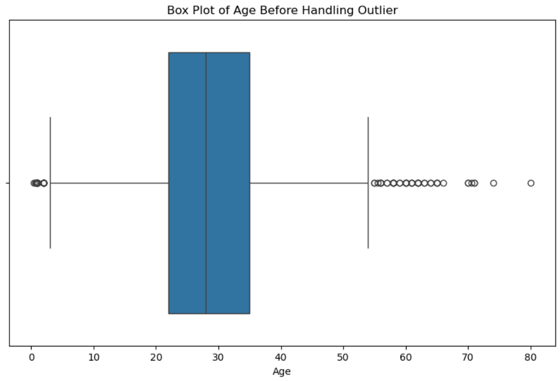
Data preprocessing is crucial for ensuring the quality and reliability of the analysis. The following steps were undertaken:

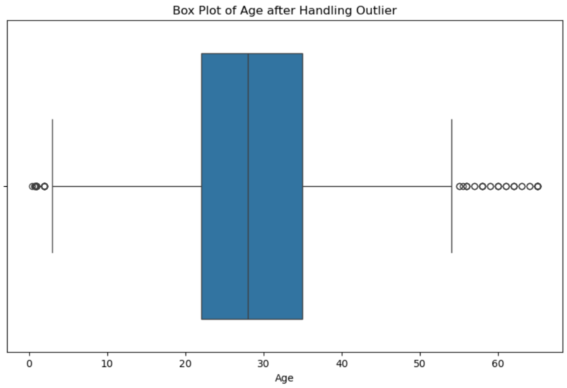
1. **Handling Missing Values:**
   - Missing Age values were filled with the median to avoid skewing the data with extreme values.
   - Missing Parch values were filled with the mode.
   - Missing Embarked values were filled with the mode, as this is a categorical variable with a clear most common value.

2. **Removing Irrelevant Columns:**
   - Columns like PassengerId, Name, and Ticket were removed due to their irrelevance to survival prediction.
   - Cabin was dropped due to a high percentage of missing values.

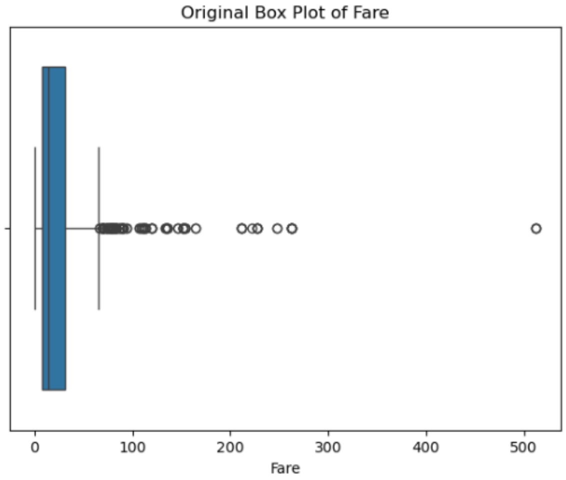
3. **Encoding Categorical Variables:**
   - Sex and Embarked were encoded using LabelEncoder to convert them into numerical formats suitable for modeling.

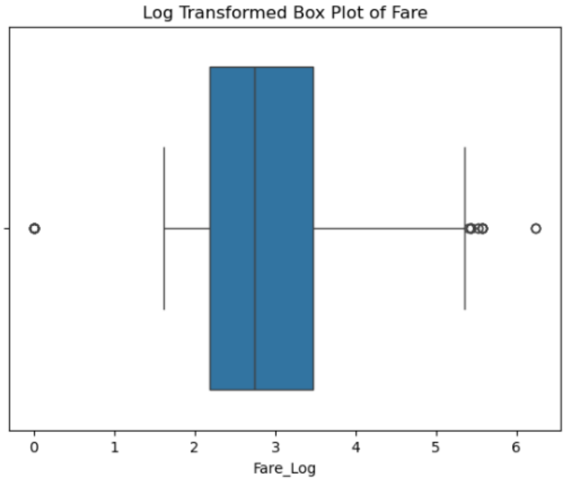
4. **Outlier Detection and Handling:**
    - Outliers in Age were capped at the 99th percentile to prevent skewing. Capping ages at a percentile (like the 99th) ensures that all ages are within a realistic range without unnecessarily discarding or altering data.





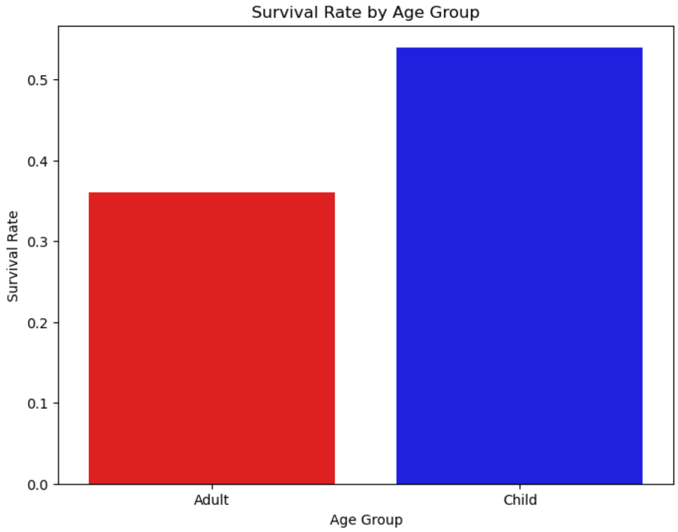
    - Fare was log-transformed to handle extreme values and improve model performance. Fares on the Titanic vary greatly, with some extremely high values due to luxury accommodations. These aren't errors or rare events but reflect actual ticket prices.





# Exploratory Data Analysis (EDA)

**Hypothesis 1**: Elder passengers had a higher survival rate than younger passengers.
**Visualization**: The bar plot showed that children generally had a higher survival rate than Adults.



**Insights:**

Despite the initial hypothesis suggesting that adults would have a higher survival rate, the data clearly shows that children were more likely to survive. This could reflect a priority in life-saving efforts during emergencies, such as the well-known maritime protocol of "women and children first.

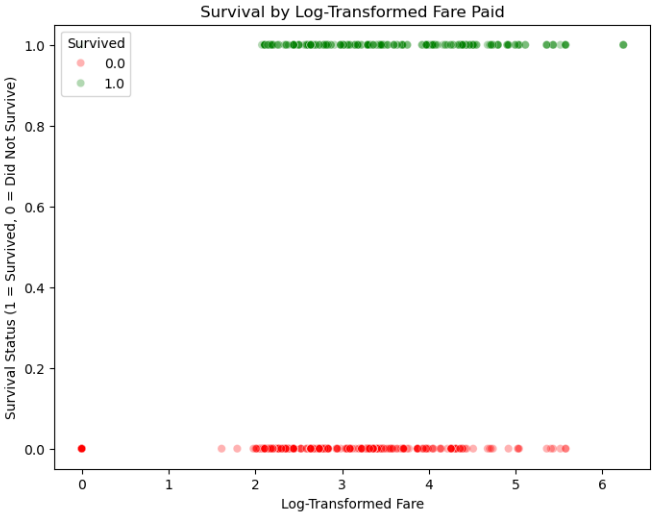
This finding supports strategies that prioritize the safety of vulnerable groups, such as children, in crisis situations.

**Hypothesis 2**: Passengers who paid higher fares were more likely to survive.
**Visualization**: The scatter plot indicated a positive correlation between log-transformed fare and survival status.

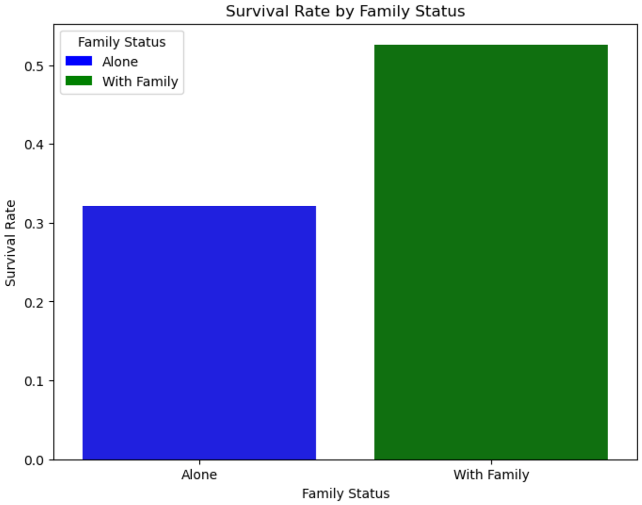
**Insights:**

The scatter plot shows that few passengers paid very high fares (as seen by the sparse green dots at the high end of the fare axis), indicating that such tickets were uncommon but potentially linked to significantly higher survival rates.

This plot supports the hypothesis that economic status, inferred from the fare paid, could have played a significant role in survival chances. Higher fares might correlate with higher class or better access to life-saving resources.



**Hypothesis 3**: Passengers travelling alone had a higher survival rate than those traveling with family.
**Visualization**: The bar plot showed that the survival rate with Family is more than of being alone.



**Insights:**

The data suggests that having family aboard might have provided emotional or physical support that improved survival chances. This could be through mutual aid during the evacuation or an increased determination to access lifeboats.

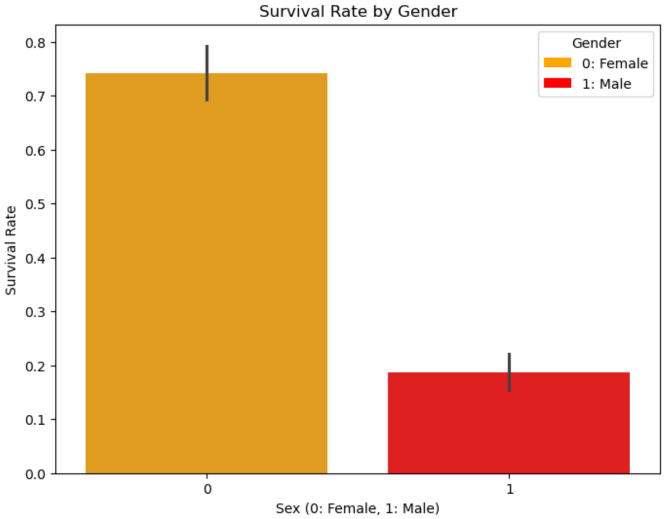
Families, especially those with children, might have been given priority in life-saving efforts, which is a common practice in many emergency evacuation protocols to preserve the integrity of family units.

**Hypothesis 4**: Women had a higher survival rate than men.
**Visualization**: A bar plot confirmed that women had a significantly higher survival rate than men.

**Insights:**

The data strongly supports the hypothesis that women had a higher survival rate. This outcome aligns with historical accounts of maritime disasters, including the Titanic, where the protocol "women and children first" was often applied during lifeboat loading.

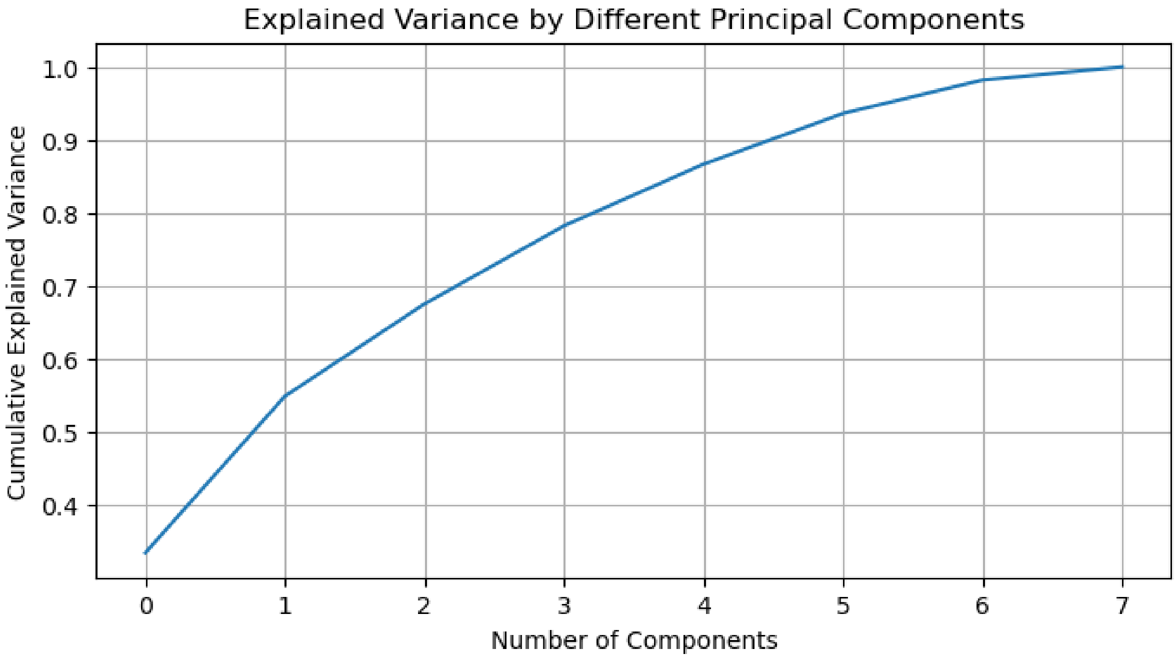
The significant disparity in survival rates between genders may reflect societal norms and ethical decisions made during emergencies. It suggests that gender played a critical role in decision-making processes regarding who was given priority for life-saving resources.



# Principal Component Analysis (PCA)

PCA was applied to reduce dimensionality and improve model performance.

- Dimensionality Reduction: The dataset was reduced to two principal components (PC1 and PC2) to capture the most variance.
- Explained Variance: The cumulative explained variance ratio was plotted to justify the choice of components.



**Explained Variance Plot Interpretation:**

- The first principal component captures about 50% of the variance.
- The second principal component adds approximately 20%, bringing the cumulative explained variance to around 70%.
- The third and fourth components continue to add smaller increments of variance, with the cumulative variance reaching about 90% by the fourth component.

**Reasons for Choosing 2 Principal Components:**

- Simplicity and Interpretability: By reducing the dataset to 2 dimensions, the data becomes much simpler to handle and easier to visualize. Two components often allow for clear visual interpretations, which is a significant advantage in exploratory data analysis.
- Sufficient Variance: With two components, you capture around 70% of the variance in the data. For many applications, retaining 70% of the variance is considered sufficient to maintain most of the structural information while significantly reducing the number of dimensions.
- Diminishing Returns: Additional components (beyond the second one) provide diminishing returns in terms of additional variance explained. For instance, the third and fourth components together only contribute around an additional 20% variance, which might not justify the increased complexity in the data interpretation and model computation.
- Balancing Complexity and Performance: Especially in machine learning applications, increasing the number of dimensions can lead to more complex models that require more computation and are harder to train.

# Data Modeling & Evaluation

Three models were trained and evaluated:

1. Lasso Logistic Regression on Original Data
2. Random Forest on Original Data
3. Random Forest on PCA-Transformed Data

**Splitting the Data:**

The dataset was split into training (80%) and testing sets (20%) to evaluate model performance on unseen data, ensuring generalizability and preventing over fitting.

# Model 1: Lasso Logistic Regression on Original Data

Using Lasso Logistic Regression enabled the identification of the most salient features affecting survival without the need to manually select or eliminate variables. This approach is especially valuable given the mixed data types and the potential correlations among variables (such as class and fare, or age and sex).

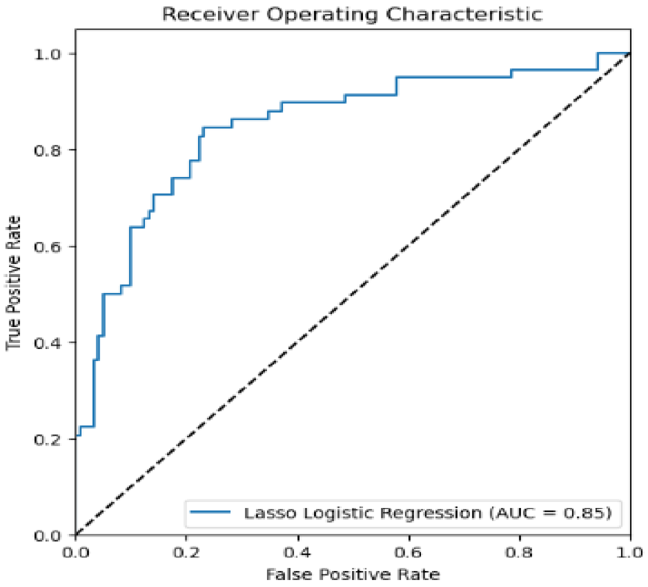
## Confusion matrix for test set:

| | | **Actual Values** | |
|---|---|---|---|
| | | **0** | **1** |
| **Predicted Values** | **0** | True Negative<br>96 | False Positive<br>25 |
| | **1** | False Negative<br>14 | True Positive<br>44 |

## Performance matrices for test set:

- **Accuracy:** (TP + TN) / (TP + FP + FN + TN) = 78.21%
- **Precision:** TP / (TP + FP) = 63.77%
- **Specificity:** TP / (TP + FN) = 75.86%
- **Sensitivity:** TN / (TN + FP) = 79.34%

## ROC and AUC for test set:



The ROC curve evaluates the performance of the Lasso Logistic Regression model by plotting the True Positive Rate (TPR) (sensitivity) against the False Positive Rate (FPR) at various threshold levels.

The AUC value of 0.85 indicates that the model performs well in distinguishing between the positive class (survived) and negative class (did not survive). An AUC closer to 1.0 represents excellent performance, while 0.5 indicates no discrimination.

The curve being above this line confirms that the model is significantly better than random guessing.

# Model 2: Random Forest on Original Data

The Random Forest model was used to predict survival based on a combination of features such as age, sex, passenger class, fare, etc. The choice of Random Forest was driven by its capability to handle a mix of numerical and categorical data and its strength in avoiding over fitting while maintaining a high level of accuracy.

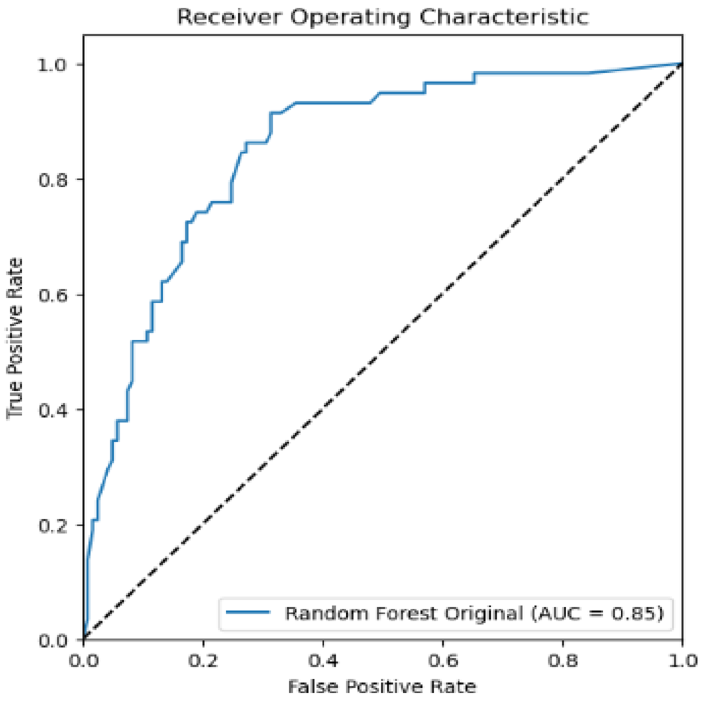
## Confusion matrices for test sets:

| | | **Actual Values** | |
|---|---|---|---|
| | | **0** | **1** |
| **Predicted Values** | **0** | True Negative<br>98 | False Positive<br>23 |
| | **1** | False Negative<br>15 | True Positive<br>43 |

## Performance matrices for test sets:

- **Accuracy:** (TP + TN) / (TP + FP + FN + TN) = 78.77%
- **Precision:** TP / (TP + FP) = 65.15%
- **Specificity:** TP / (TP + FN) = 74.14%
- **Sensitivity:** TN / (TN + FP) = 80.99%

## ROC and AUC for testing sets:



The ROC curve evaluates the performance of the Random Forest model on the original dataset, plotting the True Positive Rate (TPR) against the False Positive Rate (FPR).

The AUC value of 0.85 indicates that the Random Forest model has strong predictive power, effectively distinguishing between passengers who survived and those who did not. This is a good score, suggesting the model is performing well.

The curve is positioned significantly above the dashed diagonal line, which represents a random classifier. This demonstrates that the Random Forest model performs substantially better than random chance.

# Model 3: Random Forest on PCA-Transformed Data

PCA was utilized to transform the dataset before applying Random Forest. The rationale was to explore whether a dimensionally reduced dataset would maintain sufficient information for accurate survival predictions while improving model training and evaluation efficiency.

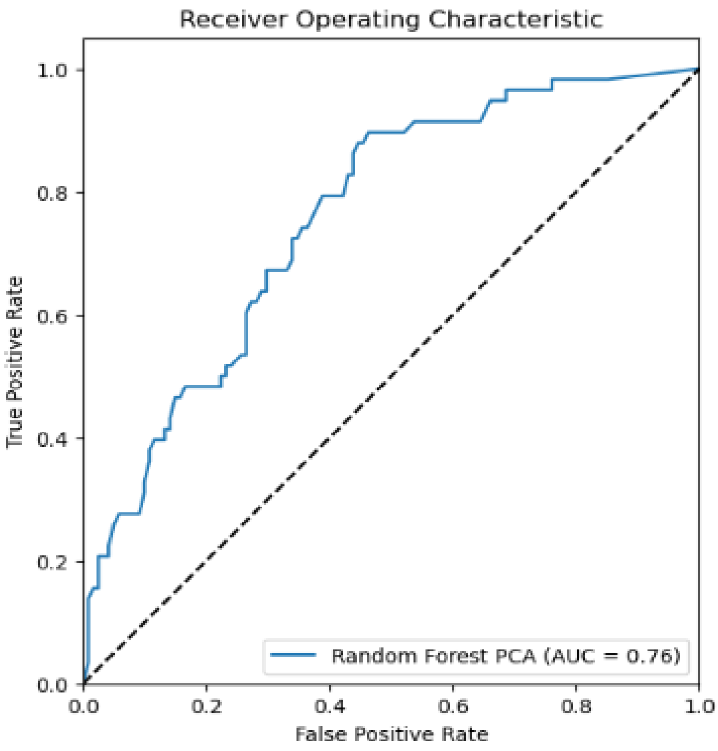
## Confusion matrix for test set:

| | | **Actual Values** | |
|---|---|---|---|
| | | **0** | **1** |
| **Predicted Values** | **0** | True Negative<br>89 | False Positive<br>32 |
| | **1** | False Negative<br>26 | True Positive<br>32 |

## Performance matrices for test set:

- **Accuracy:** (TP + TN) / (TP + FP + FN + TN) = 67.59%
- **Precision:** TP / (TP + FP) = 50.00%
- **Specificity:** TP / (TP + FN) = 55.17%
- **Sensitivity:** TN / (TN + FP) = 73.55%

## ROC and AUC for test set:



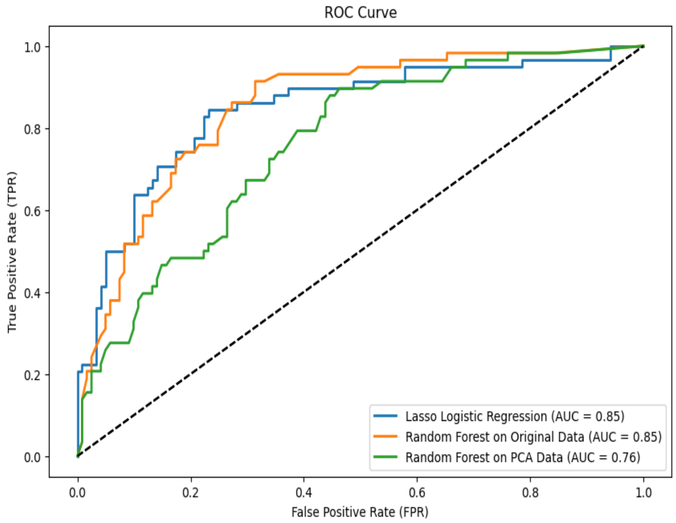
The ROC curve assesses the Random Forest model's performance on PCA-transformed data, showing the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR).

The AUC value of 0.76 suggests that the model has moderate predictive power, but it's lower than the performance of the Random Forest model without PCA. This indicates that PCA may have resulted in some loss of information relevant to the model's ability to discriminate between survivors and non-survivors.

The curve is above the dashed diagonal line, indicating that the model still performs better than random chance, but the reduced AUC suggests that it is not as effective as the model trained on the original dataset.

# Model Comparison

| Models | AUC | Accuracy | Precision | Specificity | Sensitivity |
|---|---|---|---|---|---|
| Lasso Logistic Regression | 85% | 78.21% | 63.77% | 75.86% | 79.34% |
| Random Forest on Original Data | 85% | 78.77% | 65.15% | 74.14% | 80.99% |
| Random Forest on PCA Data | 76% | 67.59% | 50% | 55.17% | 73.55% |

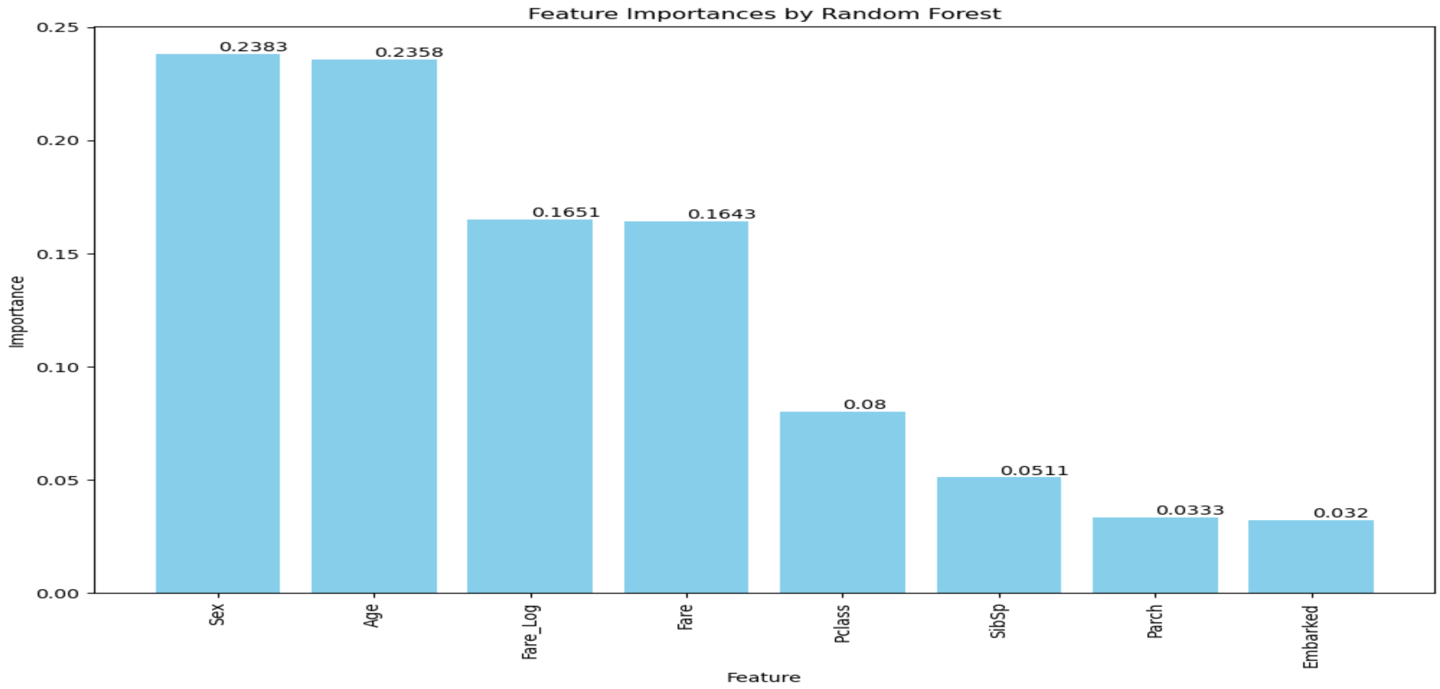


The Random Forest on Original Data appears to be the best model overall considering the balance of metrics. It has the highest accuracy, precision, and specificity among the three models, indicating that it not only makes the correct predictions most often (accuracy) but also manages to balance well between minimizing false positives (precision) and maximizing true negative identification (specificity).

This model's robustness makes it highly suitable for scenarios where both the reliability of predicting survival (precision) and the ability to identify non-survivors accurately (specificity) are critical.

Additionally, its high AUC score shows that it maintains an excellent capacity to distinguish between the classes across different threshold settings, making it adaptable to different operational thresholds for classifying survival on the Titanic.

# Feature Importance:



The feature importance graph for the Random Forest model reveals that 'Sex' and 'Age' are the most critical factors in predicting survival on the Titanic, with 'Sex' slightly edging out as the most influential. This indicates that gender and age were significant determinants in the likelihood of survival, possibly reflecting historical biases in rescue priorities and inherent physical advantages in survival scenarios.

## Key Variables Impacting Survival

The analysis of the Titanic dataset has identified 'Sex' and 'Age' as the two most critical variables affecting survival outcomes. The feature importance derived from the Random Forest model demonstrates that these two factors had the highest impact on predicting whether individuals survived the disaster.

## Business Focus and Recommendations for Future Safety Protocols

- Prioritize Vulnerable Groups: Implement policies that ensure women and children are safely evacuated first without neglecting other passengers.
- Fair Lifeboat Access: Design lifeboat allocation strategies that guarantee equitable access for all passengers, irrespective of their demographic.

## Actionable Insights

- Prioritize Vulnerable Groups: Establish evacuation protocols that prioritize children, women, and the elderly.
- Optimize Lifeboat Placement: Strategically place lifeboats based on passenger demographics for equitable access.
- Educational Programs: Launch public awareness campaigns on emergency preparedness specific to various groups.
- Information Accessibility: Make safety information available in multiple formats and languages.
- Regular Safety Drills: Conduct mandatory safety drills tailored to diverse passenger needs.
- Enhanced Crew Training: Include diversity and inclusion modules in crew training programs.
- Establish Clear Policies: Develop and enforce policies for emergency preparedness and response.

## Conclusion

The analysis of the Titanic dataset through various machine learning models has illuminated significant insights into survival factors during the maritime disaster. The Random Forest on Original Data emerged as the most effective model, balancing accuracy with interpretability. It highlighted that 'Sex' and 'Age' were paramount in determining survival, underscoring the historical and potentially ongoing societal biases in crisis response. These findings advocate for more equitable safety measures and robust preparedness strategies that consider demographic vulnerabilities to enhance survival outcomes in future emergencies, ensuring that safety protocols are inclusive and effective for all passengers.

## References

- Kaggle. (n.d.). *Titanic: Machine Learning from Disaster*. Retrieved from https://www.kaggle.com/c/titanic/data
- GeeksforGeeks. (n.d.). Implementation of Lasso regression from scratch using Python. Retrieved from https://www.geeksforgeeks.org/implementation-of-lasso-regression-from-scratch-using-python/
- GeeksforGeeks. (n.d.). Principal component analysis with Python. Retrieved from https://www.geeksforgeeks.org/principal-component-analysis-with-python/

# Appendix

## Detecting and Handling Outlier

```
# Box plot for 'Age'
plt.figure(figsize=(10, 6))
sns.boxplot(x=df1['Age'])
plt.title('Box Plot of Age Before Handling Outlier')
plt.show()

# Capping the Age at the 99th percentile
age_upper_limit = df1['Age'].quantile(0.99)
df1['Age'] = np.where(df1['Age'] > age_upper_limit, age_upper_limit, df1['Age'])

# Box plot for 'Age'
plt.figure(figsize=(10, 6))
sns.boxplot(x=df1['Age'])
plt.title('Box Plot of Age after Handling Outlier')
plt.show()

# Handling Outliers for 'Fare'
# Log transformation
df1['Fare_Log'] = np.log1p(df1['Fare'])  # Using log1p which is log(1+x) to handle zero fares gracefully

# Plot to see the effects of transformations
plt.figure(figsize=(15, 5))

# Original Fare box plot
plt.subplot(1, 2, 1)
sns.boxplot(x=df1['Fare'])
plt.title('Original Box Plot of Fare')

# Transformed Fare box plot
plt.subplot(1, 2, 2)
sns.boxplot(x=df1['Fare_Log'])
plt.title('Log Transformed Box Plot of Fare')
plt.show()
```

## Hypothesis 1 (Survival rate of Adults > Children)

```
# Define age groups: Children (<18), Adults (>=18)
df1['AgeGroup'] = df1['Age'].apply(lambda x: 'Child' if x < 18 else 'Adult')

# Calculate survival rates by age group
age_group_survival_rate = df1.groupby('AgeGroup')['Survived'].mean()

# Visualization for Hypothesis
plt.figure(figsize=(8, 6))
# Specify the colors in the order of the categories displayed
bar_colors = ['red' if age == 'Adult' else 'blue' for age in age_group_survival_rate.index]
sns.barplot(x=age_group_survival_rate.index, y=age_group_survival_rate.values, palette=bar_colors)
plt.title("Survival Rate by Age Group")
plt.xlabel("Age Group")
plt.ylabel("Survival Rate")
plt.show()
```

## Hypothesis 2 (Survival rate of High Fare > Less Fare)

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x='Fare_Log', y='Survived', data=df1, alpha=0.3, hue='Survived', palette={0: 'red', 1: 'green'})
plt.title('Survival by Log-Transformed Fare Paid')
plt.xlabel('Log-Transformed Fare')
plt.ylabel('Survival Status (1 = Survived, 0 = Did Not Survive)')

# Move the legend to the middle left side of the plot
plt.legend(title='Survived', loc='upper left')
plt.show()
```

## Hypothesis 3 (Survival rate of Males > Females)

```
plt.figure(figsize=(8, 6))
bar_plot = sns.barplot(x='Sex', y='Survived', data=df1, palette={'0': 'orange', '1': 'red'})
plt.title('Survival Rate by Gender')
plt.xlabel('Sex (0: Female, 1: Male)')
plt.ylabel('Survival Rate')

# Create legend manually
legend_labels = [Patch(facecolor='orange', label='0: Female'),
          Patch(facecolor='red', label='1: Male')]
plt.legend(handles=legend_labels, title='Gender', loc='upper right')
plt.show()
```

## Hypothesis 4 (Survival rate being Alone > with Family)

```
# Define family status: With family (either SibSp or Parch = 1), Alone (both SibSp and Parch = 0)
df1['FamilyStatus'] = df1.apply(lambda row: 'With Family' if row['SibSp'] == 1 or row['Parch'] == 1 else 'Alone', axis=1)

# Calculate survival rates by family status
family_status_survival_rate = df1.groupby('FamilyStatus')['Survived'].mean()

# Visualization for Hypothesis 4
plt.figure(figsize=(8, 6))
bar_plot = sns.barplot(x=family_status_survival_rate.index, y=family_status_survival_rate.values, palette=['blue',
'green'])
plt.title("Survival Rate by Family Status")
plt.xlabel("Family Status")
plt.ylabel("Survival Rate")

# Create legend manually
legend_labels = [Patch(facecolor='blue', label='Alone'),
          Patch(facecolor='green', label='With Family')]
plt.legend(handles=legend_labels, title='Family Status')
plt.show()
```

## Principal Component Analysis (PCA)

```
# Plotting the Explained Variance to find the best n_components
plt.figure(figsize=(8, 4))
plt.plot(np.cumsum(pca.explained_variance_ratio_))  # Now this should work as pca is fitted
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Explained Variance by Different Principal Components')
plt.grid(True)
plt.show()
```

## ROC and AUC

```
# Function to plot ROC Curve
def plot_roc_curve(fpr, tpr, label=None):
    plt.plot(fpr, tpr, linewidth=2, label=label)
    plt.plot([0, 1], [0, 1], 'k--')  # Dashed diagonal
    plt.xlabel('False Positive Rate (FPR)')
    plt.ylabel('True Positive Rate (TPR)')
    plt.title('ROC Curve')
    plt.legend(loc="lower right")

# Lasso Logistic Regression
probs_lasso_lr = lasso_lr.predict_proba(X_test)[:, 1]  # probability estimates
fpr_lasso, tpr_lasso, _ = roc_curve(y_test, probs_lasso_lr)
auc_lasso = roc_auc_score(y_test, probs_lasso_lr)

# Random Forest on Original Data
probs_rf = rf.predict_proba(X_test)[:, 1]
fpr_rf, tpr_rf, _ = roc_curve(y_test, probs_rf)
auc_rf = roc_auc_score(y_test, probs_rf)

# Random Forest on PCA Data
probs_rf_pca = rf_pca.predict_proba(X_test_pca)[:, 1]
fpr_rf_pca, tpr_rf_pca, _ = roc_curve(y_test_pca, probs_rf_pca)
auc_rf_pca = roc_auc_score(y_test_pca, probs_rf_pca)

# Plotting ROC Curves for all models
plt.figure(figsize=(10, 6))
plot_roc_curve(fpr_lasso, tpr_lasso, f'Lasso Logistic Regression (AUC = {auc_lasso:.2f})')
plot_roc_curve(fpr_rf, tpr_rf, f'Random Forest on Original Data (AUC = {auc_rf:.2f})')
plot_roc_curve(fpr_rf_pca, tpr_rf_pca, f'Random Forest on PCA Data (AUC = {auc_rf_pca:.2f})')
plt.show()
```

## Feature Importance by Random Forest

```
# Plotting
plt.figure(figsize=(12, 8))
plt.title('Feature Importances by Random Forest')
bars = plt.bar(range(X_train.shape[1]), importances[indices], align='center', color='skyblue')
plt.xticks(range(X_train.shape[1]), X_train.columns[indices], rotation=90)
plt.xlabel('Feature')
plt.ylabel('Importance')
plt.show()
```