INTRODUCTION

The Titanic, a 883ft. long ship with seven decks, was built in 1912 and claimed to be "practically unsinkable". Unfortunately, during its maiden (first) voyage on April 15, 1912, the ship collided with an iceberg and sank. There were not enough life boats aboard the ship, and 1502 out of the 2224 passengers and crew on board died [1]. Who survived the catastrophe? Were some passengers more likely to survive than others? What factors increased survivability? The aim of this study is to investigate passenger information to help answer these questions.

DATASET

Description

The Titanic dataset used in this study was obtained through Kaggle [1]. The original dataset includes 1308 instances, or passengers, and 13 features associated with each. The features include information about the passenger, their relations with other passengers, and their ticketing information. The features are listed below in Table 1.

Feature	Description	Data Type
pclass	Ticket class (1=1st, 2=2nd, 3=3rd)	float64
name	Name of the passenger	object
sex	Sex of the passenger	object
age	Age of the passenger	object
sibsp	Number of siblings/spouses aboard	float64
parch	Number of parents/children aboard	float64
ticket	Ticket number	object
fare	Ticket fare	float64
cabin	Cabin number	object
embarked	Port of embarkation	embarked
boat	Lifeboat number	object
body	Body number	float64
home.dest	Passenger home location	object

Table 1. Dataset features and data types

The provided target for each passenger is whether he/she survived (1) or did not survive (0) the sinking of the ship.

Preprocessing

Since the features boat and body leak information to the target variable, if the passenger survived or not, they were removed from the dataset. The embarked and home.dest features were removed as they are broad and not particularly useful in distinguishing passengers. There were 187 unique cabin feature values with 1015 missing values. Due to the breadth of this feature, it's large number of missing values, and potentially collinearity with pclass, this feature was removed.

There was one missing value in many of the remaining features, but 264 missing values in the feature. The passengers with features with only one missing value were removed from the dataset. The median age was imputed for the missing age values.

The sex feature was transformed into a binary variable in which ‘female’ was assigned to 0 and ‘male’ was assigned to 1. The sibsp and parch features were transformed into the feature alone. If the passenger had a sibling, spouse, parent, or child on the boat, alone was assigned to 0. If the passenger did not have relations on the boat, alone was assigned to 1. The pclass feature was converted into two binary indicator variables, pclass_2.0 and pclass_3.0. A value of 0 for both pclass_2.0 and pclass_3.0 indicates that the passenger’s ticket was first class.

The dataset post-preprocessing consisted of 1308 passengers and 6 features (pclass_2.0, pclass_3.0, sex, age, fare, and alone).

EXPLORATORY DATA ANALYSIS

The distributions of the features and target were visualized using histograms. It was noted that a large number of the passengers held third class tickets (55%), there were more passengers who did not survive than survived (808 vs. 500), there were more male than female passengers aboard (842 vs. 466), passenger ages ranged from 12.8 to 80 with a mean of 29.5, the average ticket fare was $33.30, and more passengers were alone than not alone (789 vs. 519). See Figure 1.

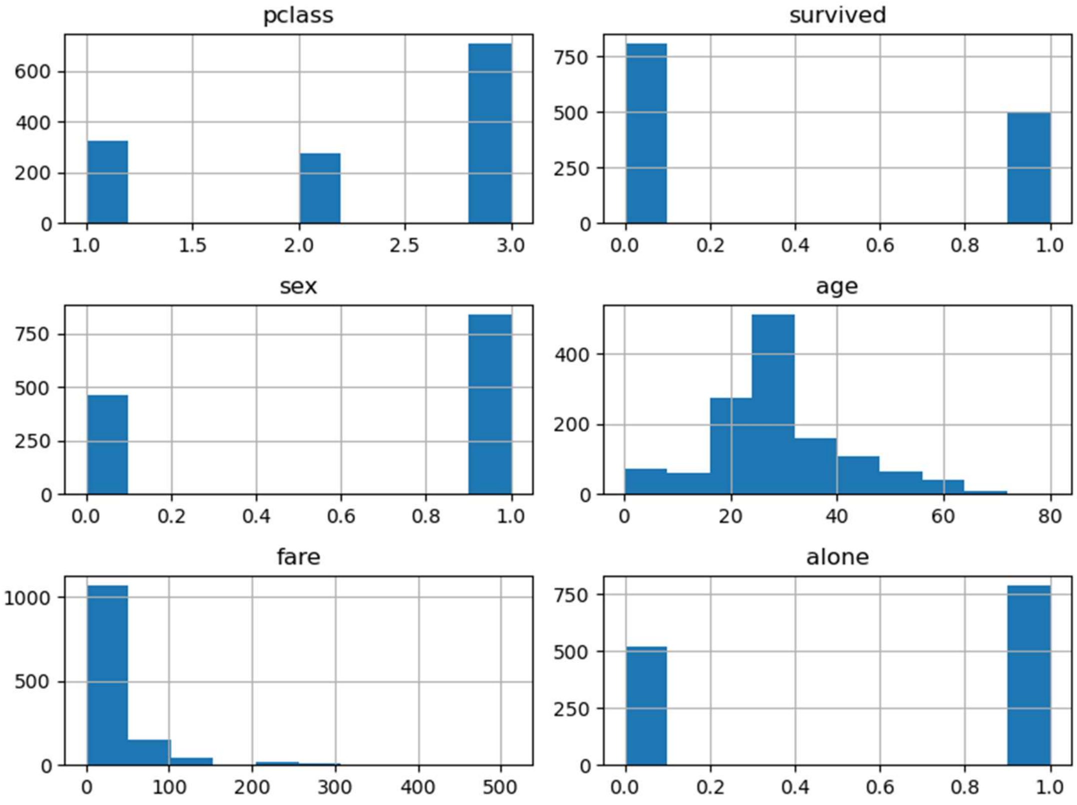

Figure 1. Histogram of features and target

The relationships between the features and the two classes were also examined using count plots. It was noted that among the three ticket classes, most passengers who did not survive held third class tickets (65%). Additionally, among those holding third class tickets, 74% did not survive (Figure 2). This indicates that ticket class likely has an impact on whether the passenger survived, favoring those in first and second class.

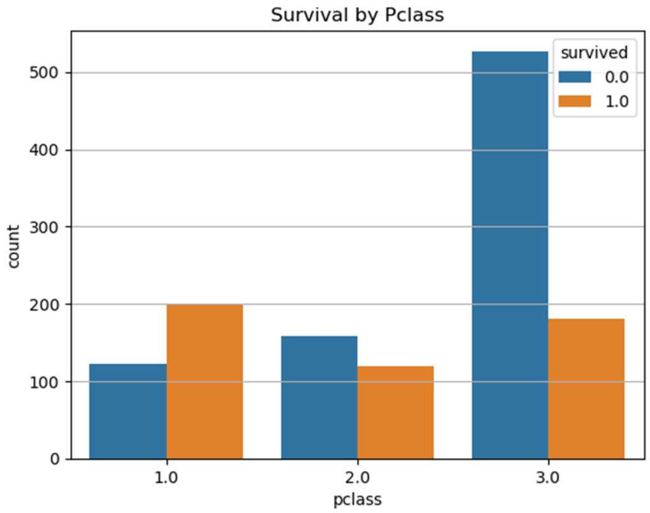

Figure 2. Count plot of survival by ticket class

It was also noted that among those who did not travel alone, the number of those surviving and not surviving were even, while the majority of those traveling alone did not survive (70%) (Figure 3). This indicates that traveling alone likely had a negative impact on if a passenger survived.

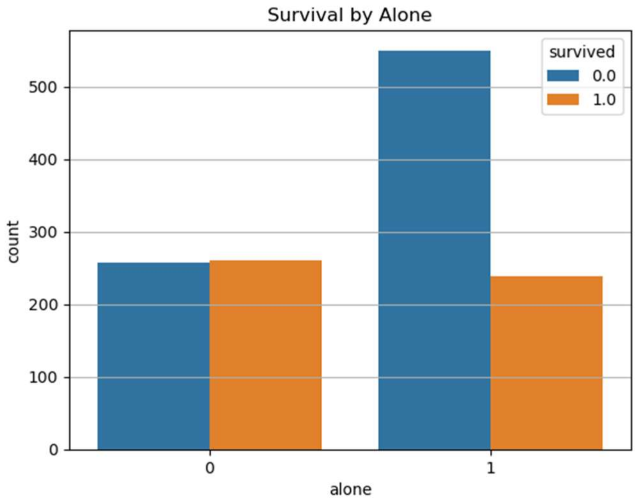

Figure 3. Count plot of survival by traveling alone

Finally, it was noted that more females survived than males, even though there were more males present on the boat. Also, most females survived (73%), while most males did not survive (81%)

(Figure 4). This is a huge indicator that being female had a strong positive impact on if the passenger survived.

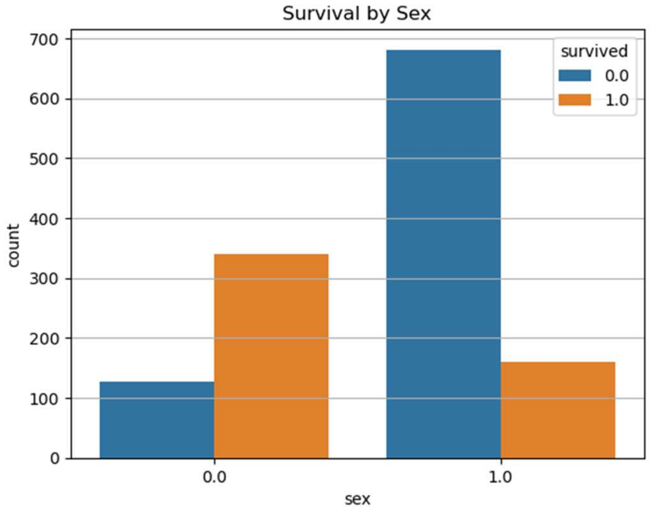

Figure 4. Count plot of survival by sex (female = 0, male = 1)

Density estimation plots were created to investigate the distribution of age and fare in relation to if the passenger survived. The age plot shows slightly higher survival in passengers under age 10 and slightly lower survival in passengers older than 55 (Figure 5). This feature may have an impact on determining if the passenger survived.

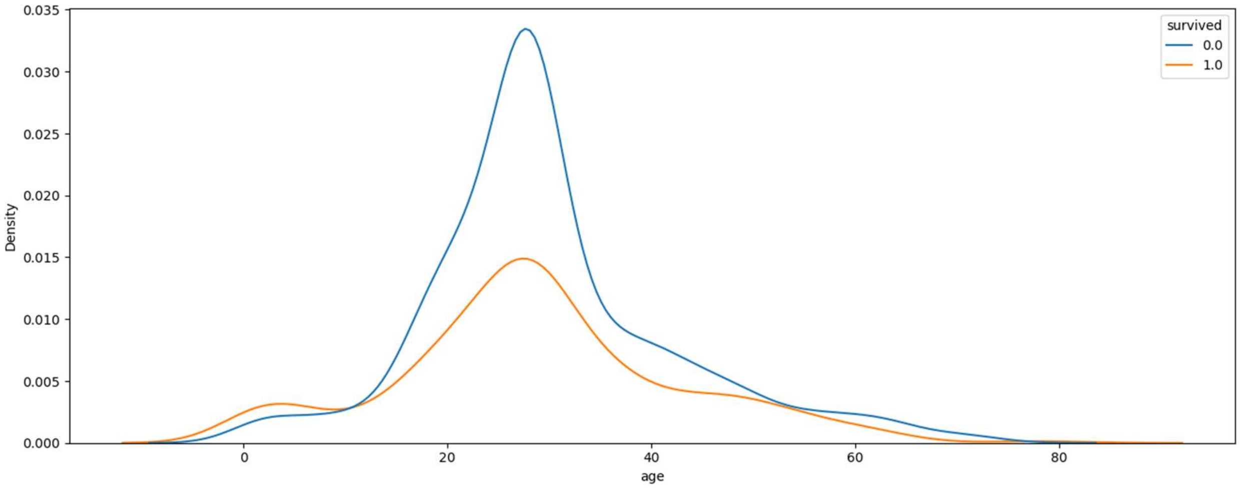

Figure 5. Density estimation of age by survival

The density estimation plot for fare shows slightly higher survival for passengers with fare prices over $50 (Figure 6). Note that the estimation is presented for fares below $0. The minimum fare charged was $0, so values under $0 will be ignored. It is unknown if passengers were truly charged $0 or if the data is missing. For purpose of this study, it is assumed that some passengers did not pay a fare. Based on this plot, it is possible that paying a higher fare may have a positive impact on survival.

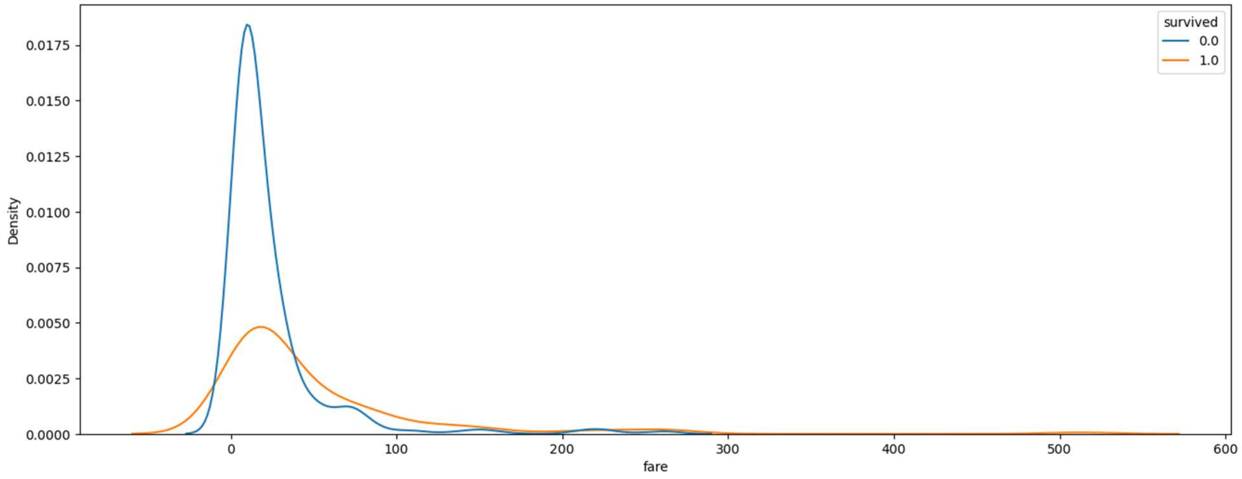

Figure 6. Density estimation of fare price by survival

The relationship between fare and pclass were investigated using a boxplot. It was observed that passengers with first class tickets paid much higher fares than those with tickets in second and third class (Figure 7). This could potentially indicate collinearity between the fare and pclass features.

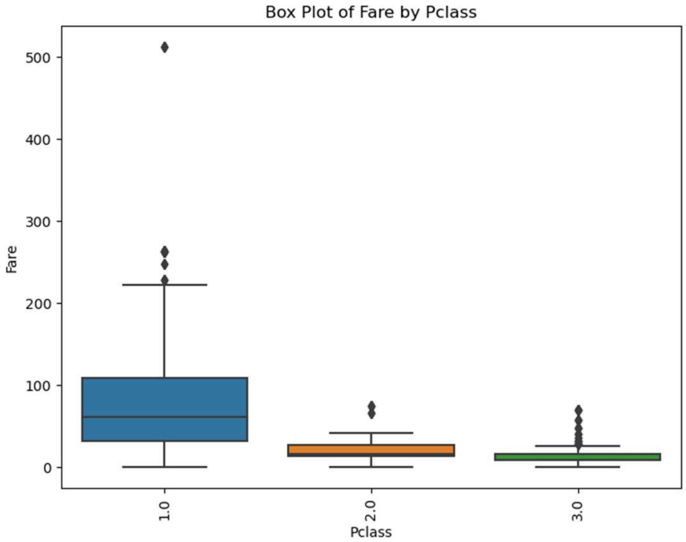

Figure 7. Boxplot of paid fare by ticket class

A one-way ANOVA test was performed to determine if there is collinearity between pclass and fare. The resulting p-value was 8.90e-129, which is less than the threshold of 0.05. This indicates that there is significant collinearity between the features, and one will be removed during feature selection.

METHODOLOGY

Train/Test Split

The data was split into a training set and testing set. The training set was randomly selected and is composed of 80% of the data, while the testing set contains the remaining 20% of data.

Feature Selection

Feature selection was performed through determining the chi-squared scores and p-values of the features in relation to the target. Based on these scores, all variables except pclass_2.0 were significant with p-values less than 0.05. Since there is collinearity between fare and pclass and the p-value for pclass_3.0 is much stronger than that of fare, the fare feature will be removed from the dataset. The pclass_2.0 feature will also be removed since it is not considered to be significant.

Feature	P-Value
sex	1.30e-25
pclass_3.0	9.47e-08
alone	2.81e-05
age	7.82e-04
fare	0.00
pclass_2.0	0.38

Table 2. Chi-squared analysis of features

Modeling

The following classification models were fitted using the training data (features: sex, pclass_3.0, alone, age) and evaluated using the testing data. Parameter tuning was performed using a grid-search technique on the training data with a 5-fold cross validation splitting strategy and scored using prediction accuracy. The models tuned and trained are listed below:

Model	Tuned Parameters
Logistic Regression	regularization parameter, C = 0.1
K-Nearest Neighbors (KNN)	n-neighbors = 5
Support Vector Machine (SVM)	regularization parameter, C = 10 kernel = radial basis function (rbf)
Decision Tree	criteria = entropy maximum depth = None minimum samples per split = 5
Random Forest	n-estimators = 300 maximum depth = None minimum samples per split = 10
Gradient Boosting	learning rate = 0.01 maximum depth = 10 minimum samples per split = 10 n-estimators = 100

Table 3. Tuned parameters for classification models

After tuning and fitting the model to the training data, the target classes of the test data were predicted and compared to the actual test data targets. Metrics reported include accuracy, precision, recall, and F1-score. Accuracy is defined as the proportion of correctly assigned instances out of total instances. Precision is defined as the ability of the model to correctly identify true positives. Recall is defined as the ability of the model to correctly identify all positive instances. The F1-score is the mean of the precision and recall. Note that precision, recall, and F1-scores reported are macro averages of both variables to ensure that the model performs well for all classes, regardless of sample size.

RESULTS

The coefficients determined by the Logistic Regression model are presented in Table 4.

Feature	Coefficient
sex	-1.9836
pclass_3.0	-0.9622
alone	-0.2080
age	-0.0146

Table 4. Logistic regression coefficients

The testing accuracy of each model is presented in Table 5.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.79	0.79	0.76	0.77
Gradient Boosting	0.78	0.79	0.76	0.76
Decision Tree	0.77	0.77	0.74	0.75
Support Vector Machine (SVM)	0.76	0.75	0.75	0.75
Logistic Regression	0.75	0.74	0.73	0.74
K-Nearest Neighbors (KNN)	0.71	0.70	0.69	0.69

Table 5. Classification model metrics

The confusion matrix of the highest performing model, the Random Forest, is presented in Figure 8.

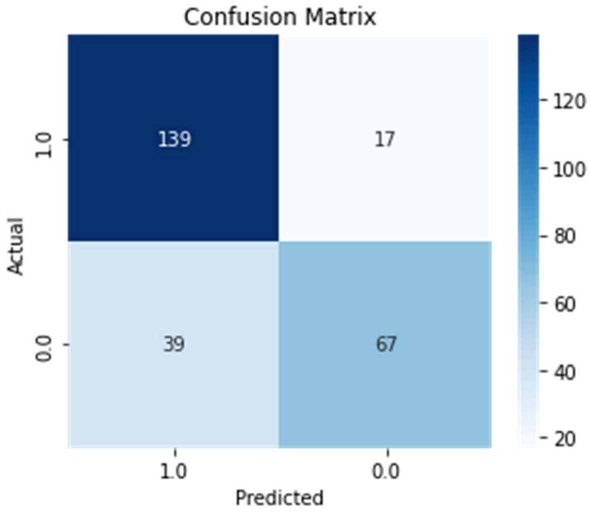

Figure 8. Random Forest model confusion matrix

DISCUSSION

The K-Nearest Neighbors (KNN) model classifies datapoints based on its similarity to the datapoints nearby. The parameter n-neighbors is used to specify how many neighboring datapoints will influence the classification of the datapoint being predicted. Larger numbers of neighbors may not capture the data well enough, while smaller numbers of neighbors may lead to overfitting. The KNN model performed the worst out of all the classification models. KNN tends to work well with decision boundaries that are not complex. The lower performance of this model may indicate that the decision boundary is more complex.

Logistic Regression utilizes the logit/sigmoid function to create a boundary between two classes. It is a linear model that performs well when there is a linear relationship between the features and the target. Regularization adds a penalty to the cost function in order to help prevent overfitting. The regularization parameter, C, is used to control the regularization penalty. Smaller values of C increase the regularization penalty effect. The coefficients determined by the model are presented in Table 4. It is as expected that the coefficients for sex and pclass_3.0 had the largest coefficients, since they were identified as being important to the model in EDA and Feature Selection. The sex feature coefficient was about twice as large as the pclass_3.0 feature coefficient.

The Logistic Regression model performed better than the KNN model, but worse than the remaining models. The lower performance of this model suggests that the data may not have a linear relationship with the target classes.

The Support Vector Machine (SVM) model identifies a hyperplane that classifies the data by maximizing the margin, or distance between the hyperplane and support vectors. The support vectors are datapoints closest to the hyperplane that influence the hyperplane's placement. A regularization parameter, C, is chosen to control the prioritization of the margin. Smaller values of C prioritize a wider margin, and larger values of C prioritize correctly classified datapoints and a narrower margin. The kernel function of the SVM module transforms the data when a non-linear decision boundary is needed linear kernel that was determined for this model is used to capture linear relationships. It is interesting that the linear kernel performed better than the nonlinear kernels in model tuning. The SVM model performed average compared to the other models but was the best performing non-ensemble method.

The Decision Tree model performs classification through splitting data into leaves based on node criteria. The criteria parameter indicates how the quality of the split will be measured, the max depth parameter indicates the maximum number of expansions allowed, and the minimum samples per split indicates the minimum number of samples allowed per leaf. The decision tree model performed the best out of the non-ensemble methods.

Gradient Boosting is an ensemble learning method that works by training tree models sequentially. With each iteration, the model works to correct deficiencies in the previous model. Through this method, the weak tree learners are combined to create a stronger learner. The learning rate determines how quickly the model is adapted in each iteration. A larger learning rate results in quicker changes to the model. The same hyperparameters are utilized for each individual tree in the model. The n-estimators parameter determines how many boosting iterations will be performed. The Gradient Boosting model performed better than the non-ensemble methods, but worse than the Random Forest model.

The Random Forest model is an ensemble learning method consisting of many decision trees. Each of the decision trees are constructed individually on a random subset of the data. The results from the individual trees are compiled as votes to determine the ultimate classification of the data points. Random forests are known to be robust due to their cross-validation-like technique and excel at handling high-dimensional data. The Random Forest model performed better than all other models.

The tree-based models were the top performers among all the classification models. They also all had higher precision than recall scores. This indicates that the tree-based models are better at correctly identifying positive instances but may miss actual positive instances, which is a potential drawback to these models. The non-tree-based models had more even precision and recall scores.

CONCLUSION

Overall, the tree-based models outperformed the non-tree-based models and the ensemble learning methods outperformed the non-ensemble learning methods. However, it is worth noting that the tree-based models had more uneven precision and recall scores (higher precision and lower recall). It was observed that the sex and ticket class features had the largest impact on predicting survival. Being female had a positive impact on survival and a third-class ticket had a negative impact on survival. This is not unexpected, as historically, especially during the early 1900s, it was not uncommon for men to prioritize helping women. Additionally, and unfortunately, it is not too surprising that passengers with third-class tickets were treated differently than those with first- or second-class tickets.

REFERENCES

[1] *The Complete Titanic Dataset*. Kaggle. https://www.kaggle.com/datasets/vinicius150987/titanic3.

APPENDIX

See attached for project code.