Introduction

The sinking of the Titanic is one of the most infamous shipwrecks in history. In 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, leading to the death of numerous passengers and crew. This tragic event has since become a haunting tale of the limitations in safety measures of the time and the social stratification that played a role in the survival of certain individuals.

Data Analysis Process

1. Define the objective:

The objective of this analysis is to uncover insights and patterns within the Titanic dataset, using statistical methods and data visualization to draw clear connections between various features and passenger survival. The ultimate goal is to develop a predictive model that accurately forecasts survival outcomes, which can then be evaluated against the competition's test dataset for performance.

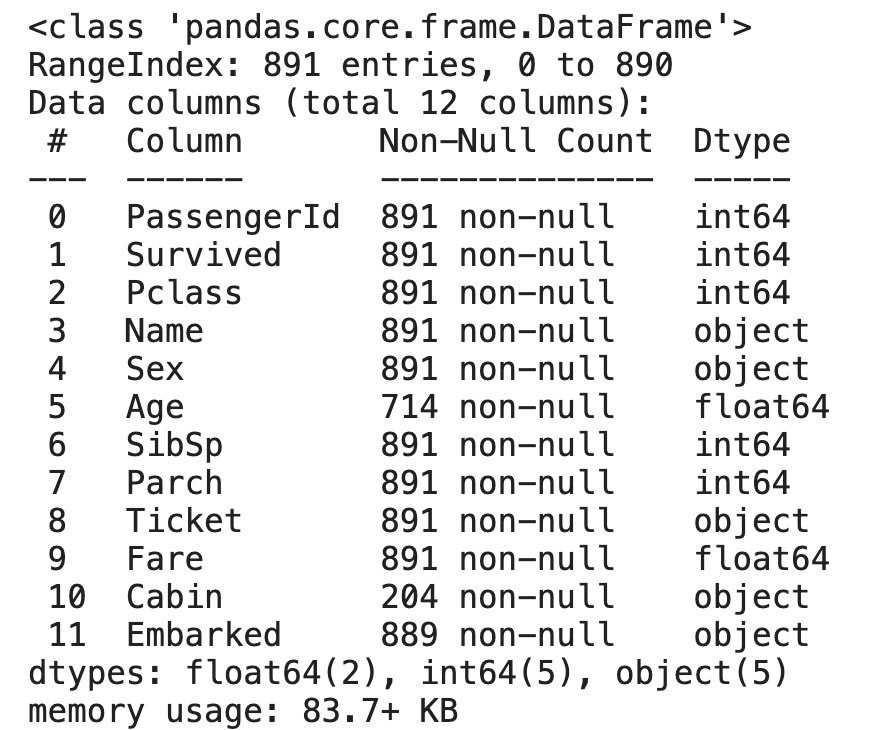
1. Data Collection:

The data for this analysis was obtained from the Kaggle competition website. It consists of two primary CSV files: train.csv for model training and test.csv for model evaluation. These files were imported into pandas DataFrames for further data analysis.

The data has been split into two groups, training set (train.csv) and test set (test.csv). We further split training set into X\_train and X\_test (validation set), X\_test is used to tune the model's hyperparameters and make decisions about which models to choose, among other model selection tasks. The test set is used only after the model development process is complete. It is a separate, untouched dataset that is used to evaluate the final model's performance, providing an unbiased assessment.

1. Data Cleaning:

Clean the unhelpful columns, NaN value, duplicates and inconsistencies.The initial step in the data preparation involved cleaning the dataset by:



The columns 'Age', 'Cabin', and 'Embarked' exhibit instances of missing data. The 'Cabin' column contains 204 non-missing entries, which is significantly lower in comparison to the other columns. In addressing the missing values for 'Age', the median has been selected as the imputation strategy due to its robustness against outliers. For the 'Fare' column, the mean is utilized to impute missing values, as it represents the average fare cost and provides a reasonable estimate in the context of normally distributed data.

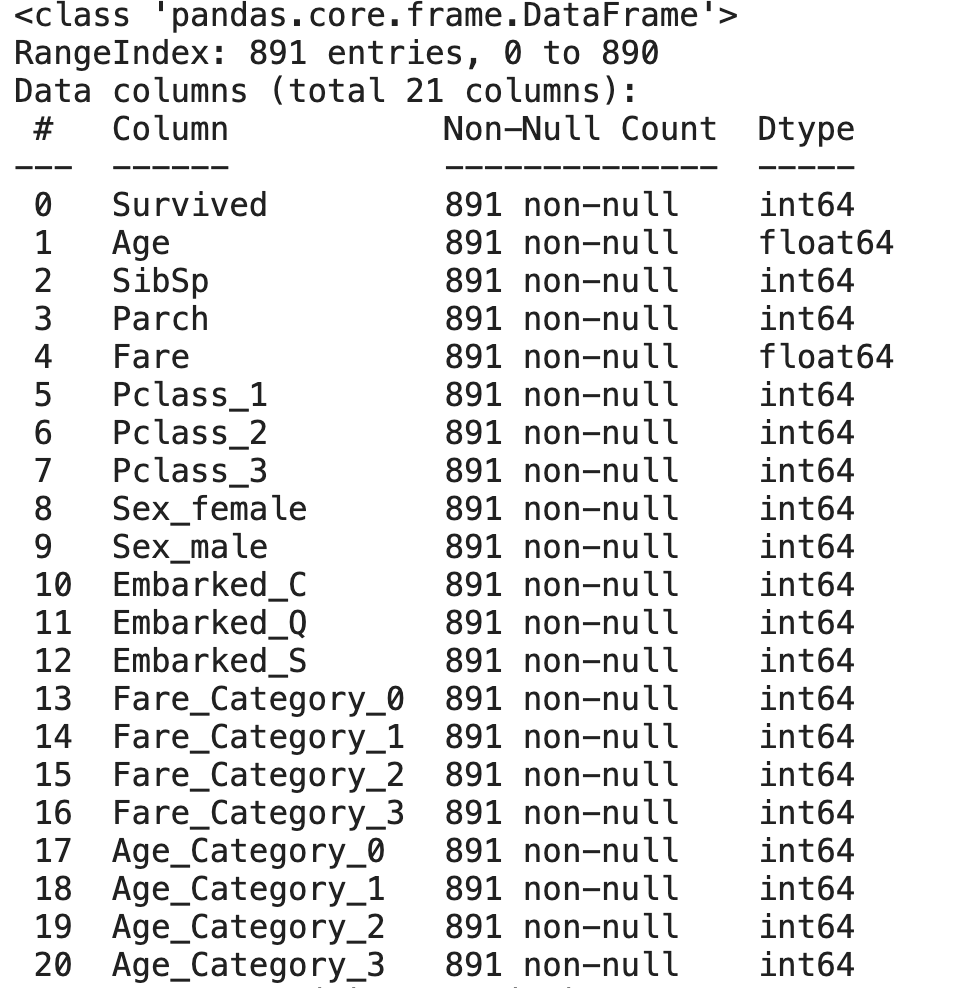
Regarding the 'Embarked' column, the mode—which represents the most frequently occurring embarkation point—is used to estimate the missing entries. Upon further examination, it has been determined that the 'Cabin' variable has a negligible correlation with survival rates. Consequently, in the interest of model simplicity and performance, we have decided to exclude the 'Cabin' column from further analysis, along with 'PassengerId', 'Name', and 'Ticket' fields, as they do not provide substantive predictive value.

1. Data Preprocessing:

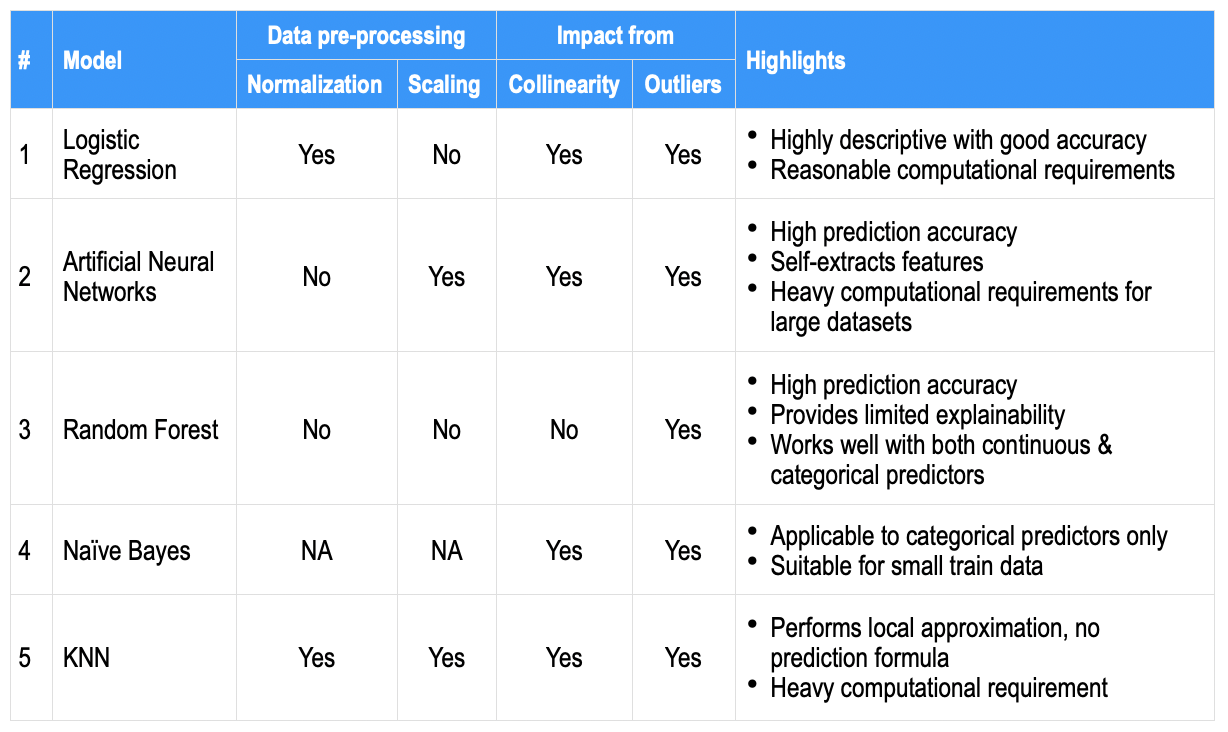
1. Feature engineering:

In order to enhance model performance, new features are devised from existing data points. The initial phase of model development revealed the significant impact of 'Age' and 'Fare' on outcomes. Consequently, additional features have been introduced, such as 'Fare\_category' and 'Age\_category', to encapsulate these variables more effectively.

2. Data Transformation: Standard scaling methods are employed to ensure that the data adheres to a common scale, facilitating more accurate analysis and prediction by the model.



1. Data Modeling:



K Nearest Neighbors (KNN):

The KNN algorithm is utilized due to its simplicity and effectiveness in classification problems where the relationship between features is not linear. It excels in scenarios where the decision boundary is irregular. The algorithm's reliance on feature similarity allows it to make predictions based on how closely data points resemble each other, which can be particularly useful in systems where proximity is a strong indicator of equivalence.

Logistic Regression:

Logistic regression is chosen for its efficiency in binary classification tasks. It provides a probabilistic framework that outputs the probability of the target variable belonging to a specific class. This model is beneficial for its interpretability, as it allows for an understanding of the impact of each feature on the likelihood of outcomes. It is particularly useful when the relationship between the independent variables and the log-odds of the dependent variable is approximately linear.

Random Forest:

The Random Forest algorithm is selected due to its robustness and ability to handle overfitting when dealing with large datasets with many features. It operates by constructing multiple decision trees and voting on the most popular output class for classification or averaging the prediction for regression. This ensemble method improves prediction accuracy and generalizability by reducing the variance that individual decision trees might exhibit.

Deep Neural Network (DNN):

A Deep Neural Network is applied to capture complex patterns and interactions within the data. Its multiple layers and non-linear processing units enable it to learn high-level features in data, which is a significant advantage in tasks requiring feature extraction and recognition. DNNs are particularly useful in dealing with large-scale and high-dimensional data, such as image and speech recognition, where the intricate structure of the data can be leveraged for more accurate predictions.

1. Performance Evaluation:

The performance of various models has been assessed using cross-validation to ensure that the results are robust and not dependent on a particular partitioning of training and test data. The K Nearest Neighbors (KNN) algorithm exhibited the highest accuracy with a score of 0.83. Logistic regression achieved an accuracy of 0.79, although it had previously reached 0.81 during the feature engineering phase. The Random Forest model showed an accuracy of 0.82, while the Deep Neural Network (DNN) achieved an accuracy of 0.827.

1. Model Selection:

Given the performance metrics, the K Nearest Neighbors (KNN) model is chosen as the preferred model due to its superior cross-validated accuracy score of 0.83.

Conclusion

Despite the complexity of models like Random Forest and DNN, which usually excel in capturing complex patterns, KNN's performance indicates that such complexity may not be necessary for the current dataset. Sometimes, simpler models are less prone to overfitting and might provide better performance on new, unseen data. Also, If the dataset contains noise, KNN can be surprisingly robust, provided that there are enough representative instances close to the query point. Furthermore, KNN naturally captures complex interactions between features without the need for explicit modeling, as it uses the raw features to compute distances. For further analysis, please refer to the notebook https://www.kaggle.com/code/jackren000/titanic-machine-learning-from-disaster/edit/run/161900402.