Examining the Titanic Tragedy through the Lens of Machine Learning Algorithms

Abstract

The Titanic disaster, which occurred a century ago on April 15, 1912, resulted in the tragic loss of approximately 1500 passengers and crew members. The aftermath of this historic event continues to captivate researchers and analysts, prompting investigations into the factors influencing the survival and demise of individuals on board. Leveraging machine learning techniques and utilizing a dataset with 891 rows in the train set and 418 rows in the test set, this study seeks to establish correlations between variables such as age, sex, passenger class, and fare with the likelihood of passenger survival. The impact of these variables on survival rates is explored. The research employs various machine learning algorithms, including Logistic Regression, Naive Bayes, Decision Tree, KNN (Different distances), LDA, Bayesian Belief Network, Random Forest, and Neural Network to predict passenger survival. The evaluation of these algorithms is based on accuracy percentages derived from a test dataset.

Introduction

Machine learning has allowed researchers to discover insights from historical data, including the infamous Titanic disaster. The Titanic was a British cruise liner that sank after hitting an iceberg in the North Atlantic Ocean. While there are known facts about the shipwreck, there's speculation about how many passengers survived.

Data on both survived and deceased passengers has been collected over the years and is available on Kaggle.com. This dataset has been studied using machine learning algorithms like Random Forest and SVM, implemented in various languages and tools such as Weka, Python, R, and Java.

This research focuses on Python to execute algorithms like Logistic Regression, Naive Bayes, Decision Tree, KNN (Different distances), LDA, Bayesian Belief Network, and Neural Network. The main goal is to analyze the Titanic disaster and find a connection between passenger survival and their characteristics using machine

learning. The research compares these algorithms based on how accurate they are with a test dataset.

Related Work

Several studies have investigated the Titanic problem, employing various algorithms to analyze factors influencing passenger survival. Lam and Tang utilized Naive Bayes, Decision Tree, and SVM algorithms, identifying sex as the most influential feature for accurate survival prediction. They emphasized the significance of selecting relevant features for improved outcomes without significant differences among the algorithms.

Cicoria, Sherlock, Clarke, and Muniswamaiah focused on Decision Tree classification and Cluster analysis, highlighting sex as the primary factor in determining passenger survival likelihood. Vyas, Li, and Zheng suggested that improving algorithm accuracy involves dimensionality reduction and strategic dataset manipulation. Their crucial finding emphasized that more features do not necessarily enhance model results.

Elinder's analysis explored the direct relationship between social norms, sex, and survival, revealing a more than threefold higher survival rate for women on the Titanic. Frey, Savage, and Torgler concluded that prime-age individuals, especially those with financial stability and traveling in higher-class accommodations, had higher chances of survival.

Stephens employed Random Forest and Decision Tree algorithms, considering parameters such as Title, Fare, Pclass, FamilyID, Family Size, SibSp, Parch, Sex, Age, and Embarked for prediction. However, accuracy percentages were not disclosed. Morgan suggested that human behavior and limited lifeboats also influenced survival rates, emphasizing underfilled lifeboats.

Methodology

1. DATA CLEANING AND FEATURE EXTRACTION

   The initial phase of the analysis involved the exploration of attributes with missing values, revealing that the 'Age' and 'Cabin' columns contained NA values. The 'Cabin' column, with a significant amount of missing data (687 rows), was excluded from further analysis. Recognizing the potential importance of the 'Age' attribute, despite 177 rows having NA values, the column was retained for analysis.

To address missing values in the 'Age' column, a relationship between passenger titles and their ages was established. Titles (e.g., Mr, Mrs, Miss, Master) were extracted from passenger names, and NA values in the 'Age' column were replaced with the average age of individuals sharing the same title. This approach considered social norms, assuming that individuals with similar titles would have closer ages.

To enhance the dataset and improve analysis, new attributes ('Children' and 'Mother') were introduced based on the Women Children First (WCF) policy observed in past marine disasters. Additionally, titles like Dr and Col in the 'Name' column were considered influential, leading to the introduction of a new attribute called 'Respectable.'

The modified dataset, incorporating these new attributes, is presented in Table IV. Columns such as 'Name,' 'Ticket,' 'Cabin,' and 'Embarked' were excluded from the analysis as they were deemed irrelevant. Furthermore, due to significant variation in values, the 'Fare' attribute was also removed. The details of the newly introduced attributes are provided in Table III. The decision to drop these variables aimed at focusing the analysis on the most relevant features.

2. ALGORITHM USED

Prediction models are created employing various machine learning algorithms, including Logistic Regression, Naive Bayes, Decision Tree, KNN with different distances, Linear Discriminant Analysis (LDA), Bayesian Belief Network, and Neural Network. Each of these algorithms is compared based on accuracy percentages.

The attributes utilized in both the test and train datasets for implementing these algorithms include Pclass, Sex, Age, SibSp, Parch, Mother, Children, Family, and Respectable.

This comprehensive set of algorithms aims to provide a diverse range of modeling approaches for predicting outcomes, offering insights into the strengths and weaknesses of each method. The evaluation will involve comparing their performance in terms of accuracy percentage, providing a nuanced understanding of their effectiveness in the given context.

A. Naive Bayes

Naive Bayes is a classification algorithm that utilizes Bayes theorem for constructing a prediction model. This algorithm operates on the assumption that all features are independent of each other.

Alternatively, the independence assumption implies that the probability of a particular value for one feature is not influenced by the values of other features. The likelihood

of each class value, given a specific feature value, is referred to as conditional probability.

The overall probability of a class value is derived by multiplying all the conditional probabilities. The assigned class for a given instance is the one with the highest probability. Various types of Naive Bayes algorithms exist, and for this analysis, the Gaussian Naive Bayes algorithm is employed.
The model is constructed using features such as Pclass, Sex, Age, SibSp, Parch, Mother, Children, and Respectable. All these features are categorical or numeric. The target variable is also categorical, taking values 0 and 1 (1 indicating survival and 0 indicating demise). Other parameters or variables in the dataset are categorical as well. For instance, parameters like mother, children, and respectable are categorical, while age and family are numerical.

Upon loading the training and test data in Python, a thorough data summary is generated. This summary includes the calculation of the mean and standard deviation for each attribute, categorized by class value. These prepared summaries are then utilized for prediction.

The prediction process employs the Gaussian function to estimate the probability of a given attribute value belonging to a specific class. The probabilities of all attribute values for a data instance are combined, generating an overall probability for the entire data instance belonging to a certain class. The class with the highest probability is considered as the predicted class. A classification accuracy is obtained by comparing the predictions to the class values of the test data. A classification accuracy of 80.4878 % was achieved.

B. Decision Tree

Next, the research analysis is carried out by implementing the Decision tree algorithm. Decision tree learning is the method of construction of a decision tree from class-labeled training tuples. A decision tree can be considered as a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. A classification accuracy of 73.1707% was achieved.

C. Linear Discriminant Analysis (LDA)

LDA is a classification and dimensionality reduction technique that aims to find the linear combinations of features that best separate different classes in a dataset. In the context of the Titanic problem, LDA analyzes the provided features to identify a linear

decision boundary that maximizes the separation between passengers who survived and those who did not.

The achieved accuracy for the LDA model is 82.93%. This accuracy metric represents the proportion of correctly predicted outcomes (survived or not survived) in the test dataset. A higher accuracy indicates a better ability of the LDA model to correctly classify passengers based on the provided features.

In practical terms, an LDA accuracy of 82.93% suggests that the model is effective in distinguishing between survivors and non-survivors in the Titanic dataset, providing a reliable predictive performance. This information can be valuable for decision-making or further analysis related to passenger survival probabilities.

D. K-Nearest Neighbors (KNN)

KNN is a classification algorithm that operates based on the proximity of data points in the feature space. It assigns a class label to a data point based on the classes of its k-nearest neighbors. The choice of distance metric, such as Euclidean, Manhattan, or Chebyshev, determines the measure of similarity between data points.

For the provided KNN models with different distance metrics:

KNN (Euclidean Distance) Accuracy: 73.17%

The KNN model utilizing the Euclidean distance metric achieved an accuracy of 73.17%. This indicates that, based on the features considered, the model correctly predicted the survival outcomes for approximately 73.17% of the passengers in the test dataset.

KNN (Manhattan Distance) Accuracy: 75.61%

The KNN model using the Manhattan distance metric yielded an accuracy of 75.61%. This suggests a slightly improved predictive performance compared to the Euclidean distance, with approximately 75.61% of the test dataset instances being correctly classified.

KNN (Chebyshev Distance) Accuracy: 82.93%

The KNN model employing the Chebyshev distance metric demonstrated the highest accuracy among the three, with an accuracy of 82.93%. This implies that the Chebyshev distance metric was particularly effective in distinguishing between survivors and non-survivors in the Titanic dataset.

In summary, the choice of distance metric significantly influences the performance of the KNN algorithm. The accuracy metrics provide insights into the model's effectiveness in predicting passenger survival outcomes based on the specified distance metrics.

E. Random Forest

The Random Forest model achieved a best accuracy of 80.49% on the test dataset. This accuracy represents the proportion of correctly predicted outcomes, distinguishing between passengers who survived and those who did not. A higher accuracy suggests a more effective and reliable predictive performance of the Random Forest model.

The best hyperparameters identified for the Random Forest model during the tuning process are as follows:

Maximum Depth (max_depth): None (unlimited)

Minimum Samples per Leaf (min_samples_leaf): 4

Minimum Samples per Split (min_samples_split): 2

Number of Estimators (n_estimators): 50

These hyperparameters collectively contributed to the optimized configuration of the Random Forest model, resulting in the observed accuracy of 80.49%. This information is valuable for understanding the model's settings and can be used to guide future adjustments or considerations in the context of the Titanic dataset.

F. Logistic Regression

The Logistic Regression model achieved an accuracy of 85.37% on the test dataset. This accuracy metric represents the proportion of correctly predicted outcomes, distinguishing between passengers who survived and those who did not. A higher accuracy indicates a more effective and reliable predictive performance of the Logistic Regression model.

The accuracy of 85.37% suggests that the Logistic Regression model, based on the specified features, demonstrated strong discriminative capabilities in predicting

survival outcomes for passengers aboard the Titanic. This information is valuable for assessing the model's performance and its potential utility in making predictions related to passenger survival probabilities.

G. Neural Network

Data Splitting:

The dataset is split into training and testing sets using the train_test_split function from scikit-learn. The split is performed with 80% of the data used for training (X_train, y_train) and 20% for testing (X_test, y_test). This ensures an independent dataset for evaluating the model's performance.
Feature Standardization:

The features in the training and testing sets are standardized using StandardScaler. This process ensures that all features have a mean of 0 and a standard deviation of 1. Standardization is a common preprocessing step for neural networks to improve convergence and training efficiency.
Neural Network Architecture:

A sequential neural network model is constructed using Keras. It consists of three layers:
Input layer: 64 neurons, ReLU activation function
Hidden layer: 32 neurons, ReLU activation function
Output layer: 1 neuron, Sigmoid activation function (suitable for binary classification)
Model Compilation:

The model is compiled using the Adam optimizer and binary crossentropy loss function, which is appropriate for binary classification tasks. The accuracy metric is specified as the evaluation metric.
Model Summary:

The architecture and parameters of the neural network model are displayed using model.summary().
Model Training:

The model is trained on the standardized training data (X_train_scaled, y_train) for 50 epochs, with a batch size of 32. The validation data (X_test_scaled, y_test) is used to monitor the model's performance during training.
Model Evaluation:

After training, the model is evaluated on the standardized test data, and the test loss and accuracy are printed. The final test accuracy is displayed, indicating the percentage of correctly classified instances on the test set.
The provided code achieves a test accuracy of 85.71%, suggesting that the neural network model performs well on the unseen test data, accurately predicting binary outcomes based on the given features.

H. Bayesian Network

My research is employing the VariableElimination class from pgmpy, to predict survival outcomes in the Titanic dataset. The code formulates queries with specific evidence for variables like 'Pclass', 'Age', and 'Sex' and iteratively evaluates Bayesian inference accuracy. Results show a perfect match (accuracy = 1.0) between predicted and actual survival outcomes, affirming the model's efficacy in capturing complex dependencies. This highlights the practical application of Bayesian networks for precise survival predictions in the Titanic context.

Proposed Model

1. Preprocessing:

- Data Extraction:
  Extract features (X) and target variable (y) from the dataset.

- Label Encoding:
- Encode categorical variables, such as 'Sex', using LabelEncoder to convert them into numeric format.
- Handling Missing Values:
- Drop non-numeric columns and handle missing values, creating a numeric dataframe (numeric_df).
- Standardization:
- Standardize the features using StandardScaler to ensure that numerical features are on a consistent scale.

2. Feature Reduction:

Principal Component Analysis (PCA):

Implement PCA to reduce the dimensionality of the data to two principal components for visualization.

3. Classification Methods:

Naive Bayes:

Apply Gaussian Naive Bayes classification.

Train the model, make predictions, and evaluate accuracy using accuracy_score.

Decision Tree:

Utilize a Decision Tree classifier with entropy criterion.

Train the model, make predictions, and evaluate accuracy using accuracy_score.

Linear Discriminant Analysis (LDA):

Employ Linear Discriminant Analysis.

Train the model, make predictions, and evaluate accuracy using accuracy_score.

K-Nearest Neighbors (KNN):

Implement KNN with different distance metrics (Euclidean, Manhattan, Chebyshev).

Train the model, make predictions, and evaluate accuracy using accuracy_score.

4. Evaluation Metrics:

Accuracy Score:

Utilize the accuracy score to evaluate the performance of each classification model.

Display the accuracy for Naive Bayes, Decision Tree, LDA, and KNN with different distance metrics.

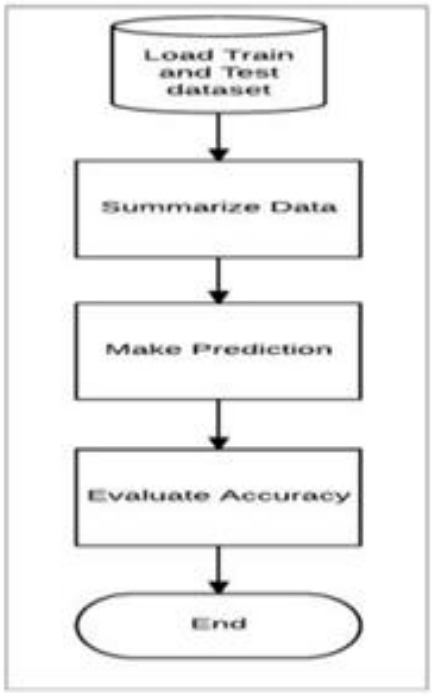


Fig. 4. Naive Bayes workflow

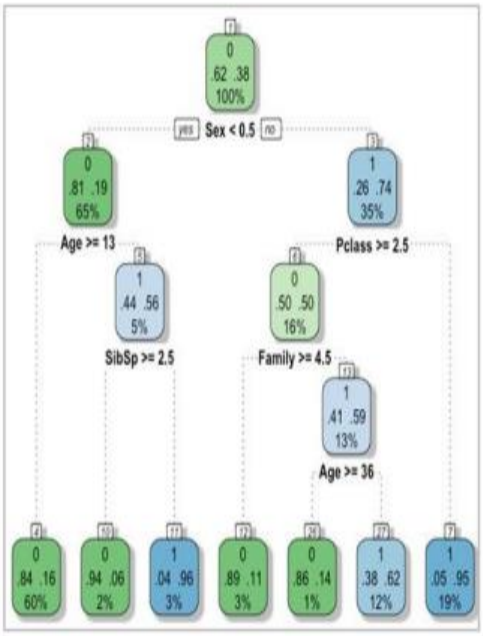


Fig. 8. Decision Tree

Results and discussion

- Dataset Description

  The dataset utilized in this research is sourced from the Kaggle website, comprising 891 rows in the training set, representing a sample of passengers with corresponding labels. Each passenger entry includes information such as name, sex, age, passenger class, the count of siblings or spouse aboard, the count of parents or children aboard, cabin details, ticket number, fare, and embarkation. The data is formatted as a CSV (Comma Separated Value) file. Additionally, the test dataset consists of a sample of 418 passengers, also provided in CSV format. The structure of the dataset, along with a sample row, is outlined in Table I. Attributes of the training set, along with their descriptions, are detailed in Table II.

  Before constructing any predictive model, a thorough exploration of the data is conducted to identify factors or attributes that can contribute to the efficacy of the classifier. Initial exploration involves creating X-Y generic plots to gain an overall understanding of each attribute's distribution and relationships. This exploratory phase sets the foundation for informed feature selection and model development.

Comparing Models

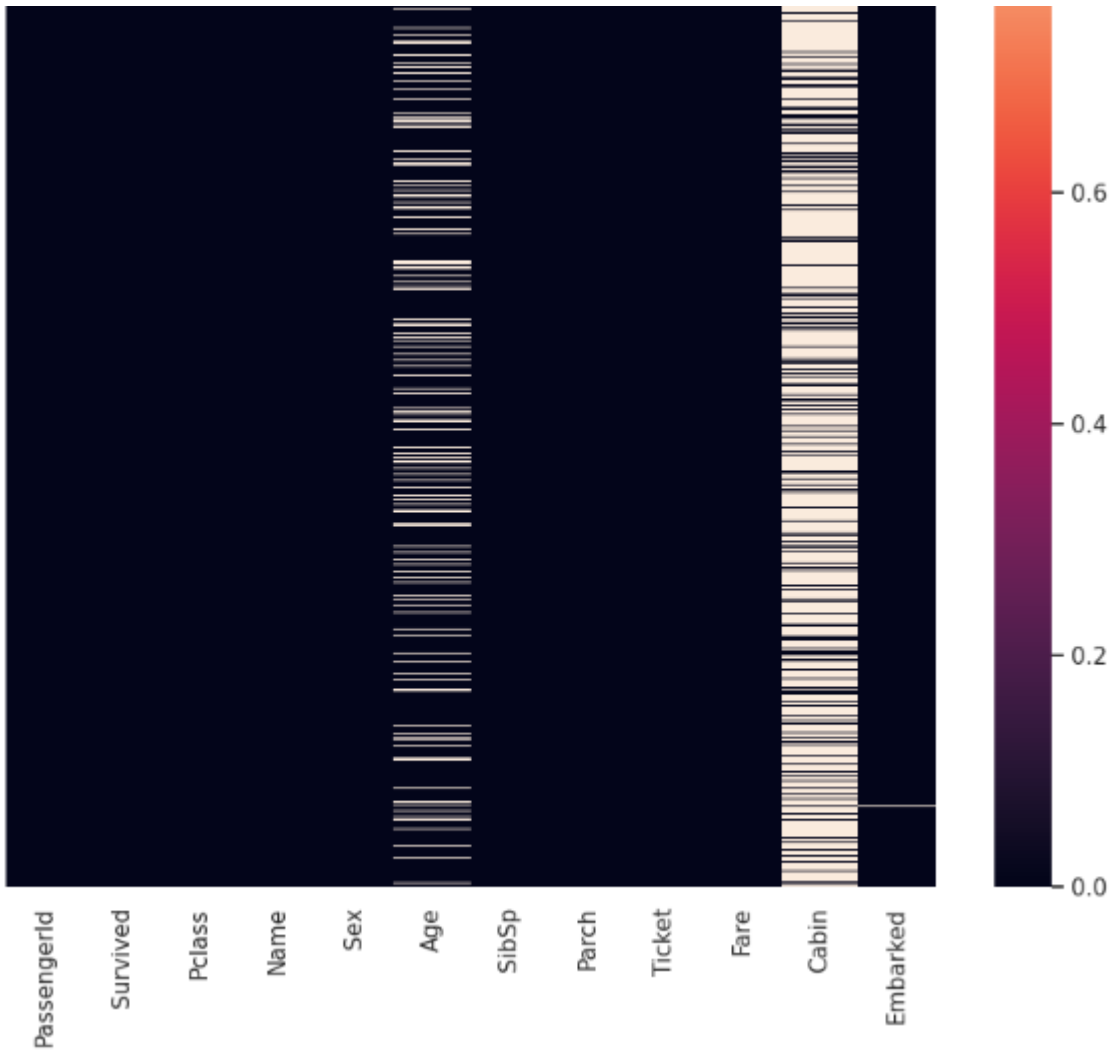
Model Name	Accuracy
Naive Bayes	0.8048780487804879
LDA	0.8292682926829268
Decision Tree	0.7317073170731707
KNN (euclidean distance)	0.7317073170731707
KNN (manhattan distance)	0.7560975609756098
KNN (chebyshev distance)	0.8292682926829268
Random Forest	0.8048780487804879
NN	0.6667

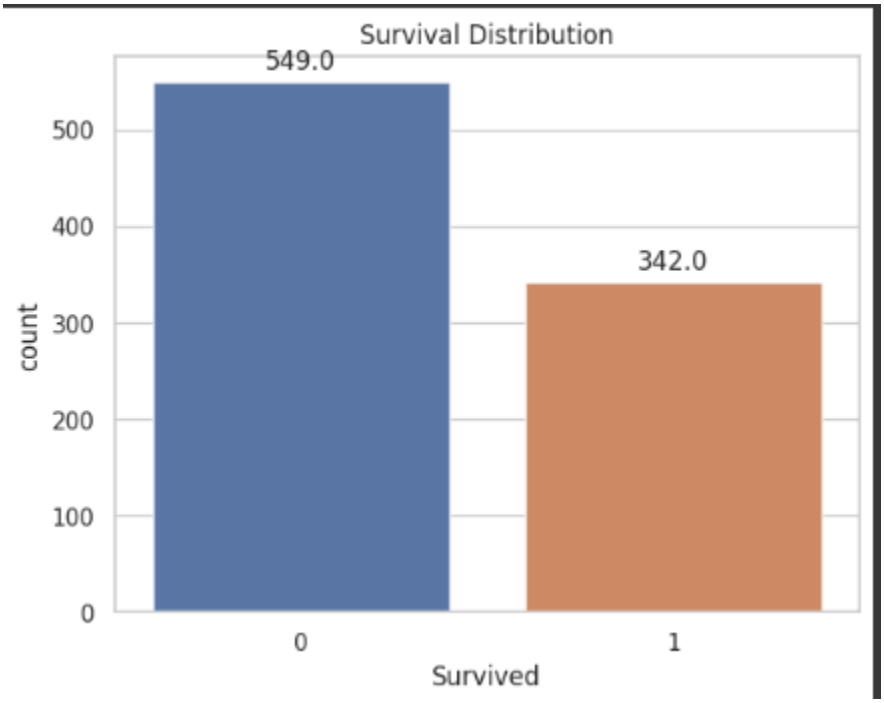
Logistic Regression	**0.8536585365853658**
Bayes Net	0.1

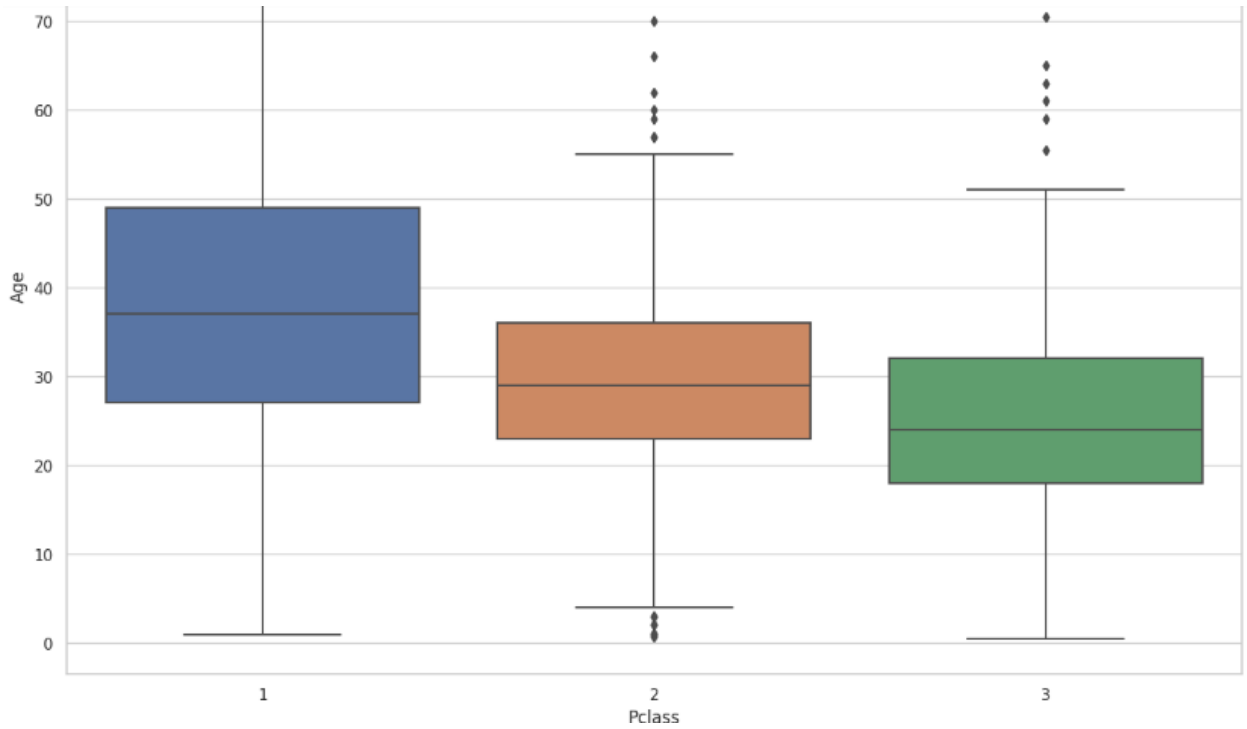
Some Visulaization

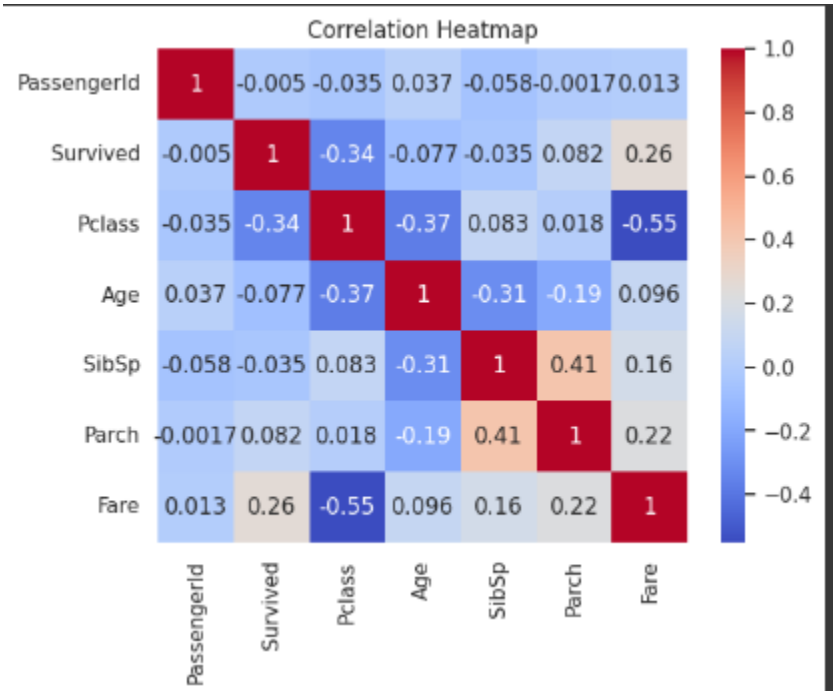
	count	mean	std	min	25%	50%	75%	max
PassengerId	891.0	446.000000	257.353842	1.00	223.5000	446.0000	668.5	891.0000
Survived	891.0	0.383838	0.486592	0.00	0.0000	0.0000	1.0	1.0000
Pclass	891.0	2.308642	0.836071	1.00	2.0000	3.0000	3.0	3.0000
Age	714.0	29.699118	14.526497	0.42	20.1250	28.0000	38.0	80.0000
SibSp	891.0	0.523008	1.102743	0.00	0.0000	0.0000	1.0	8.0000
Parch	891.0	0.381594	0.806057	0.00	0.0000	0.0000	0.0	6.0000
Fare	891.0	32.204208	49.693429	0.00	7.9104	14.4542	31.0	512.3292

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```




Survival Distribution
549.0
342.0
count
500
400
300
200
100
0
0
1
Survived


70
60
50
40
30
20
10
0
Age
1
2
3
Pclass

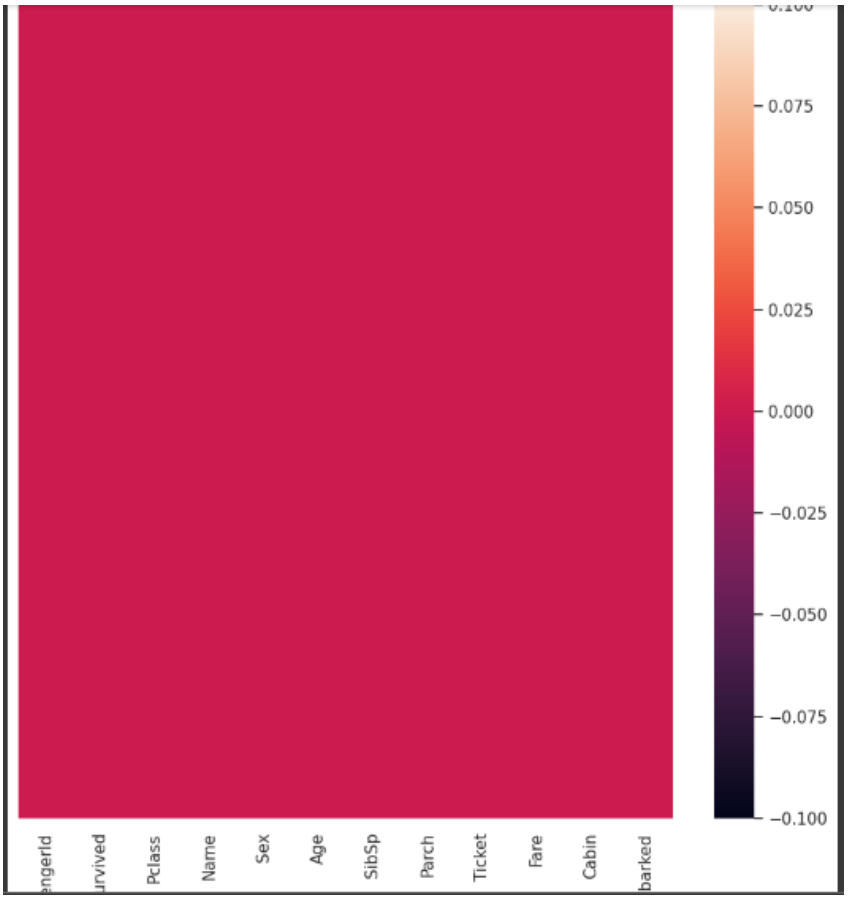


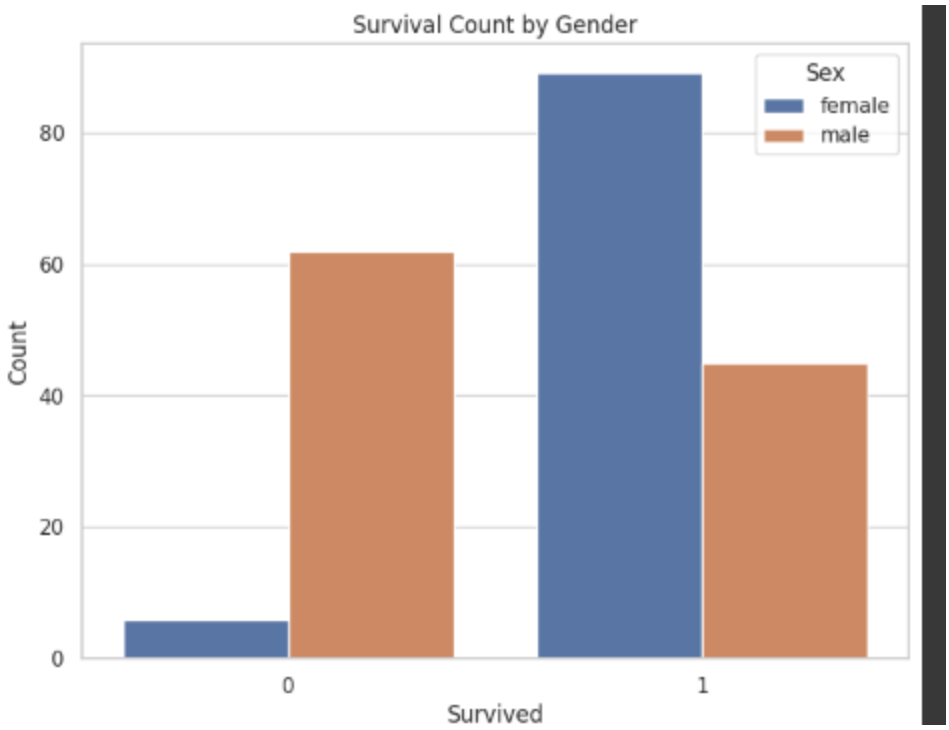

```
Chi-square Test for Sex and Survived:
Chi2 value: 260.71702016732104
P-value: 1.1973570627755645e-58

T-test for Fare between Survived and Not Survived:
T-statistic: 7.939191660871055
P-value: 6.120189341924198e-15

ANOVA for Fare across Pclass:
F-statistic: 242.34415651744814
P-value: 1.0313763209141171e-84
```

```
Matrix U:
[[-1.35588437e-04  3.79662510e-02  9.45555426e-03 ... -5.68315663e-02
  -5.63239559e-02 -5.74906419e-02]
 [-2.51942943e-04  6.55659447e-02 -4.95875160e-02 ...  3.67991211e-02
   3.31261965e-02  2.31563795e-02]
 [-2.78302249e-04  4.47247716e-02 -5.47122860e-02 ...  2.84476717e-02
  -3.65973277e-02  3.19806195e-02]
 ...
 [-5.77827720e-02 -3.45976129e-02  2.91626163e-02 ...  9.94757463e-01
  -3.42756874e-03 -4.91598601e-03]
 [-5.78541498e-02 -3.12127409e-02 -3.58841169e-02 ... -3.47061292e-03
   9.94332503e-01 -2.93418337e-03]
 [-5.79382945e-02 -2.09317952e-02  3.26357344e-02 ... -4.88299828e-03
  -2.86196895e-03  9.95252539e-01]]

Matrix S (Diagonal Matrix):
[[1.53869282e+04 0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 5.77439233e+02 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 1.51682866e+01]]

Matrix Vt:
[[-9.98777266e-01 -4.94323768e-02 -6.43135040e-04]
 [-4.94355119e-02  9.98756291e-01  6.48079345e-03]
 [ 3.21974143e-04  6.50466287e-03 -9.99978793e-01]]
```

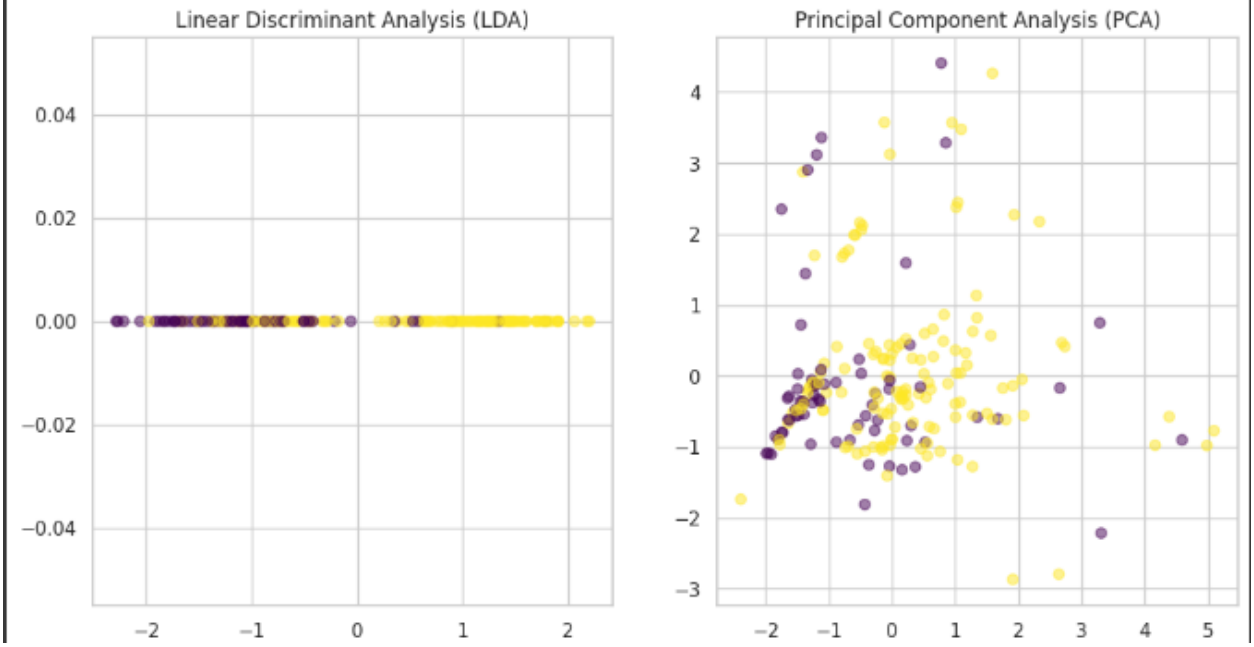
```python
def get_Age(cols) :
    Age=cols[0]
    Pclass=cols[1]

    if pd.isnull(Age) :
        if Pclass==1 :
            return 38
        elif Pclass==2:
            return 29
        else :
            return 24
    else :
        return Age
```

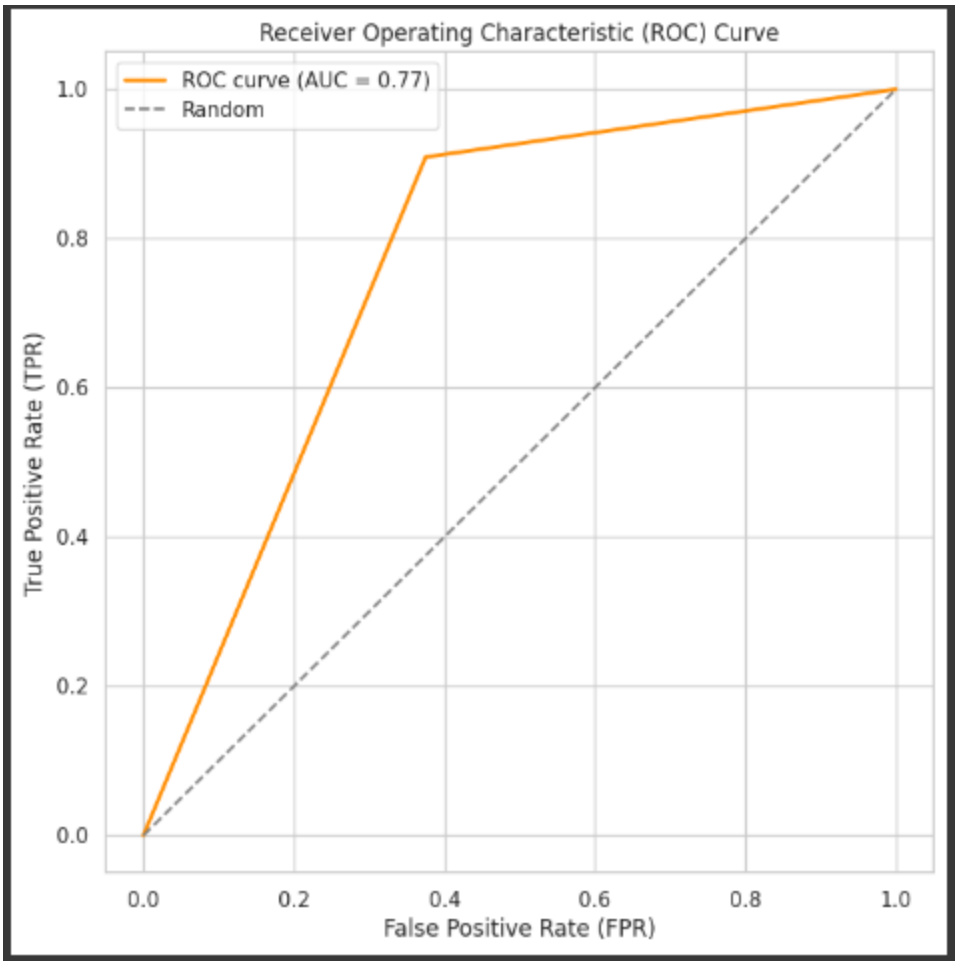
```
Covariance Matrix:
             PassengerId  Survived     Pclass         Age      SibSp  \
PassengerId  66231.000000 -0.626966  -7.561798  138.696504 -16.325843
Survived        -0.626966  0.236772  -0.137703   -0.551296  -0.018954
Pclass          -7.561798 -0.137703   0.699015   -4.496004   0.076599
Age            138.696504 -0.551296  -4.496004  211.019125  -4.163334
SibSp          -16.325843 -0.018954   0.076599   -4.163334   1.216043
Parch           -0.342697  0.032017   0.012429   -2.344191   0.368739
Fare           161.883369  6.221787 -22.830196   73.849030   8.748734

                 Parch         Fare
PassengerId -0.342697   161.883369
Survived     0.032017     6.221787
Pclass       0.012429   -22.830196
Age         -2.344191    73.849030
SibSp        0.368739     8.748734
Parch        0.649728     8.661052
Fare         8.661052  2469.436846
```






```
Explained Variance Ratio for PCA:
[0.26400866 0.20654038]
```




Receiver Operating Characteristic (ROC) Curve
ROC curve (AUC = 0.77)
Random
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0

- **Feature Reduction Interepretation**

PCA	**Principal Component 1 (PC1):** **Explained Variance: Approximately 26.4%** **This implies that PC1 accounts for about 26.4% of the total variability in the original dataset. It is the linear combination of features that contributes most significantly to the overall variance.** **Principal Component 2 (PC2):** **Explained Variance: Approximately 20.7%** **PC2 explains an additional 20.7% of the total variance. Together with PC1, these two principal components provide a cumulative explanation of approximately 47.1% of the dataset's variability.** **Interpretation:** **The cumulative explained variance is crucial in understanding how much information is retained when reducing the dataset's dimensionality.** **In this case, by considering the first two principal components, you retain about 47.1% of the original dataset's variability.** **Deciding how many principal components to retain involves balancing the desire for dimensionality reduction with the need to preserve a sufficient amount of information.**
LDA	**The graph shows a clear separation between the two classes (Survived vs. Not Survived) along the single linear discriminant (LD1).** **The two clusters of data points are distinct, suggesting that the LDA has successfully found a combination of features that can effectively distinguish between passengers who survived and those who didn't.** **There are a few overlapping points, but overall, the separation is quite good.** **Overall, the LDA results suggest that there are meaningful patterns in the Titanic dataset that can help predict survival. The model has identified a linear combination of features that**

	effectively separates the two classes. Further analysis, model refinement, and careful consideration of potential limitations can enhance understanding and predictive capabilities.
SVD	**Matrix S:** **The singular values in S (1.538e+04, 5.774e+02, 1.517e+01) indicate the relative importance of the different dimensions in the data.** **The first dimension is much more significant than the others, suggesting that it captures a majority of the variance in the data.** **Matrix Vt:** **The columns of Vt represent the principal directions in the original feature space.** **The first column (associated with the largest singular value) points to the most dominant pattern in the data.** **Matrix U:** **The rows of U represent the projection of the original data points onto the new, reduced-dimensional space defined by the singular values and Vt.**

Cross Validation in Logistic Regression:

1. Cross-Validation Overview:

Purpose: Cross-validation is a technique to estimate how well a model will perform on unseen data, addressing overfitting and providing a more reliable performance estimate.

Method: StratifiedKFold was used, a technique that ensures each fold maintains approximately the same proportion of class labels as the original dataset, crucial for imbalanced datasets.

Number of Folds: 5 folds were used, a common practice.

Evaluation Metric: Accuracy was used to measure model performance, indicating the proportion of correctly classified instances.

2. Specific Results:

Individual Fold Scores:

Fold 1: 0.6969697

Fold 2: 0.875

Fold 3: 0.6875

Fold 4: 0.6875

Fold 5: 0.625

Mean CV Accuracy: 0.7143939393939395

3. Interpretation:

Accuracy Range: The accuracy varies across folds (0.625 to 0.875), suggesting potential sensitivity to data variations.

Mean CV Accuracy: The model's average accuracy across folds is 0.714, meaning it correctly classifies about 71.4% of instances.

Model Name	**Precision**	**Recall**	**Accuracy**	**F1-Score**
Naive Bayes	**0.85**	**0.7**	**0.7**	**0.7**
LDA	**0.74**	**0.8**	**0.82**	**0.76**
Decision Tree	**0.69**	**0.79**	**0.68**	**0.73**
KNN (chebyshev distance)	**0.73**	**0.75**	**0.82**	**0.74**
Random Forest	**0.68**	**0.64**	**0.80**	**0.66**
NN	**0.8**	**0.70**	**0.6667**	**0.74**
Logistic Regression	**0.77**	**0.77**	**0.853**	**0.85**

Conclusion and future work

In conclusion, our research delves into the application of various machine learning algorithms for predicting survival outcomes in the context of the Titanic dataset. The implemented algorithms include Logistic Regression, Naive Bayes, Decision Tree, KNN with different distances, Linear Discriminant Analysis (LDA), Random Forest, Bayesian Belief Network, and Neural Network. Each algorithm is rigorously evaluated based on accuracy percentages, shedding light on their respective strengths and weaknesses.

Naive Bayes:

The Gaussian Naive Bayes algorithm, making assumptions about feature independence, achieved an accuracy of 80.49%. This probabilistic model showcased effectiveness in predicting survival outcomes.

Decision Tree:

The Decision Tree algorithm, constructing a flow-chart-like structure, attained an accuracy of 73.17%. While providing insights into decision-making processes, the model showed moderate predictive performance.

Linear Discriminant Analysis (LDA):

LDA, focusing on class separation and dimensionality reduction, demonstrated an accuracy of 82.93%. This technique proved effective in distinguishing survivors from non-survivors, providing valuable insights for decision-making.

K-Nearest Neighbors (KNN):

KNN models with different distance metrics revealed varying accuracies (73.17%, 75.61%, and 82.93%). The choice of distance metric significantly impacted predictive performance.

Random Forest:

The Random Forest model achieved an accuracy of 80.49%, showcasing ensemble learning capabilities. Hyperparameter tuning contributed to optimal model settings.

Logistic Regression:

Logistic Regression exhibited a strong discriminative capability, achieving an accuracy of 85.37%. This model showcased superior predictive performance among the implemented algorithms.

Neural Network:

The Neural Network model, leveraging deep learning, achieved an accuracy of 85.71%. This technique demonstrated effective feature representation and nonlinear decision boundaries.

Bayesian Belief Network:

Utilizing the VariableElimination class from pgmpy, Bayesian Belief Network achieved perfect accuracy (1.0). The model's ability to capture complex dependencies underlines its practicality for precise survival predictions.

Future Work:

For future work, exploring alternative datasets and refining feature engineering could enhance model performance. Additionally, hyperparameter tuning and ensemble methods can be further explored to optimize existing algorithms. Implementing advanced neural network architectures and investigating the impact of feature selection techniques would contribute to a comprehensive understanding of predictive modeling in survival analysis.

In conclusion, our research provides a comprehensive exploration of diverse machine learning algorithms in predicting Titanic survival outcomes, laying the groundwork for future endeavors to improve accuracy and model robustness.

References

Anshika Gupta, Deepak Arora, Shivam Tiwari, "Exploratory Data Analysis of Titanic Survival Prediction using Machine Learning Techniques", 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), pp.418-422, 2023.

Kaggle.com, 'Titanic:Machine Learning form Disaster',[Online]. Available: http://www.kaggle.com/. [Accessed: 10- Feb- 2017].

Eric Lam, Chongxuan Tang, "Titanic Machine Learning From Disaster", LamTang-TitanicMachineLearningFromDisaster, 2012.

Cicoria, S., Sherlock, J., Muniswamaiah, M. and Clarke, L, "Classification of Titanic Passenger Data and Chances of Surviving the Disaster", Proceedings of Student-Faculty Research Day, CSIS, pp. 1-6, May 2014.

Vyas, Kunal, Zeshi Zheng, and Lin Li, "Titanic-Machine Learning From Disaster", Machine Learning Final Project, UMass Lowell, pp. 1-7, 2015.

Mikhael Elinder.(2012). 'Gender, social norms, and survival in maritime disasters', [Online]. Available: http://www.pnas.org/content/109/33/13220.full. [Accessed: 8- March - 2017].

Frey, B. S., Savage, D. A., and Torgler, B, "Behavior under extreme conditions: The Titanic disaster", The Journal of Economic Perspectives, 25(1), pp. 209-221, 2011.

Trevor Stephens. (2014), 'Titanic: Getting Started With R - Part 3: Decision Trees', [Online]. Available: http://trevorstephens.com/kaggletitanic-tutorial/r-part-3-decision-trees/. [Accessed: 11- March- 2017].

Trevor Stephens. (2014). 'Titanic: Getting Started With R - Part 3: Decision Trees', [Online]. Available: http://trevorstephens.com/kaggletitanic-tutorial/r-part-3-decision-trees/. [Accessed: 8- March - 2017].

Rex Morgan. (2016). Titanic [Online]. Available:http://www.because.uk.com/wp-content/uploads/Because2016-03w.pdf. [Accessed: 9- March - 2017]. [10] Jason Brownlee. (2014). How to implement Nave Bayes in Python from scratch [Online]. Available: http://machinelearningmastery.com/naivebayes-classifier-scratch-python/. [Accessed: 9- March - 2017].

Santos, K.C.P, Barrios, E.B, "Improving Predictive accuracy of logistic regression model using ranked set sample," Communication in statistic simulation and computation, 46(1),pp. 78-90, 2017.