## MACHINE LEARNING – MINI PROJECT

**ELAKYA K (21ITR026)**

**KAVYA S (21ITR052)**

**MAMTHA B (21ITR058)**

# TITANIC SURVIVAL PREDICTION

## ABSTRACT

The Titanic dataset presents a captivating narrative of human tragedy and resilience, inviting exploration through the lens of data science. In this study, we embark on a journey to uncover hidden patterns and insights within the dataset, focusing primarily on predicting passenger survival. Through rigorous data preprocessing, feature selection, and engineering, complemented by the application of advanced machine learning algorithms such as Logistic Regression, Decision Trees, Random Forest, and K-Nearest Neighbors, we endeavor to construct predictive models with high accuracy and robustness. Our analysis delves deep into the socio-demographic factors, cabin classes, and other passenger attributes to discern their impact on survival likelihood. Preliminary findings showcase promising performance, with accuracy rates consistently surpassing 80% across multiple models. By shedding light on the intricacies of survival dynamics aboard the Titanic, this study not only pays homage to the lives lost but also underscores the invaluable insights gleaned from historical data through modern computational techniques. Through this endeavor, we aim to contribute to the growing body of knowledge in data-driven historical analysis and predictive modeling, demonstrating the enduring relevance of data science in unraveling complex narratives and informing future endeavors.

## INTRODUCTION:

Water quality assessment is crucial for maintaining environmental sustainability and public health. Traditional methods often lack real-time monitoring capabilities, leading to delayed responses to water quality issues. Machine learning (ML) algorithms offer a promising solution by enabling the analysis of large datasets and the prediction of water quality parameters with improved accuracy and efficiency. In this report, we explore the application of ML techniques in water quality prediction, aiming to evaluate their effectiveness in addressing the challenges of conventional monitoring methods.

ML algorithms have demonstrated significant potential in various domains, including environmental science and engineering. By leveraging historical and real-time data, these algorithms can identify complex patterns and relationships, allowing for the prediction of water quality parameters such as pH, dissolved oxygen levels, and pollutant concentrations. Through the integration of sensor networks and data analytics, ML enables continuous monitoring of water bodies, facilitating early detection of contaminants and timely intervention measures to mitigate potential risks.

The adoption of ML in water quality prediction holds immense promise for improving the efficiency and effectiveness of water resource management practices. However, challenges such as data quality, model interpretability, and scalability need to be addressed to fully realize the benefits of ML in this domain. Continued research and innovation in ML techniques, coupled with robust data collection and validation processes, are essential for advancing the state-of-the-art in water quality monitoring and ensuring the sustainability of water resources for future generations.

## DATASET DESCRIPTION:

1. **Passenger Class (Pclass):** Indicates the class of travel for each passenger, categorized as 1st, 2nd, or 3rd class. This variable reflects socio-economic status, with higher classes typically associated with greater privileges and amenities.

2. **Name:** Names of the passengers onboard the Titanic. While primarily used for identification purposes, analysis of names may reveal demographic information or familial relationships.

3. **Sex:** Gender of the passengers, categorized as male or female. This variable provides insights into the gender distribution among Titanic passengers.

4. **Age:** Age of the passengers in years. Understanding the age distribution helps assess the demographic composition of the passengers and may influence survival outcomes.

5. **Siblings/Spouses Aboard (SibSp):** Indicates the number of siblings or spouses accompanying each passenger on the Titanic. This variable reflects the presence of family members onboard.

6. **Parents/Children Aboard (Parch):** Represents the number of parents or children accompanying each passenger. Similar to SibSp, Parch indicates family relationships and dependencies.

7. **Ticket:** Ticket number assigned to each passenger. While primarily used for ticket identification, analysis of ticket numbers may reveal patterns or group bookings.

8. **Fare:** Fare paid by each passenger for their ticket. Fare amounts may vary based on passenger class, cabin type, and other factors, reflecting socio-economic disparities among passengers.

9. **Cabin:** Cabin number assigned to each passenger, indicating their accommodation on the Titanic. Cabin data may provide insights into the spatial distribution of passengers onboard.

10. **Embarked:** Port of embarkation for each passenger, categorized as C (Cherbourg), Q (Queenstown), or S (Southampton). This variable indicates the embarkation point and may influence survival outcomes based on socio-economic factors and passenger demographics.

11. **Survived:** Binary variable indicating whether a passenger survived the Titanic disaster (1) or not (0). This variable serves as the target variable for predictive modeling, with survival status being the outcome of interest.

## METHODOLOGY:

1. **Data Collection:**

   Gather a comprehensive dataset containing passenger information from the Titanic, including attributes such as passenger class, name, age, gender, siblings/spouses aboard, parents/children aboard, ticket details, fare, cabin, port of embarkation, and survival status.

2. **Data Preprocessing:**

   **a. Handle Missing Values:** Check for missing values in the dataset and apply appropriate techniques such as imputation or removal to address them.

   **b. Outlier Detection:** Identify and handle outliers in the data to ensure model robustness and accuracy.

   **c. Feature Engineering:** Extract relevant features from the dataset or create new features to improve model performance.

   **d. Feature Scaling:** Scale numerical features to a similar range to prevent dominance by features with larger magnitudes.

   **e. Encode Categorical Variables:** Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding.

**Train-Test Split:**

Divide the dataset into training and testing sets to evaluate model performance effectively.

**Model Selection:**

a. Choose suitable machine learning algorithms for predicting survival on the Titanic, considering factors such as interpretability, accuracy, and computational efficiency.

b. Experiment with various models such as logistic regression, decision trees, random forests, support vector machines (SVM), k-nearest neighbors (KNN), or gradient boosting to identify the best-performing model.

**Model Training:**

a. Train the selected machine learning models on the training dataset using appropriate training algorithms.

b. Tune hyperparameters using techniques like grid search or randomized search to optimize model performance.

### Model Evaluation:

a. Evaluate the trained models using the testing dataset and performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC).

b. Compare the performance of different models to select the most suitable one for predicting survival on the Titanic.

### Model Interpretation :

a. Interpret the trained model to understand the significance of each feature in predicting survival outcomes.

b. Analyze model predictions to identify factors influencing survival probabilities and potential areas for further investigation.

### Model Deployment :

Deploy the trained machine learning model into production environments for real-time prediction of passenger survival on the Titanic.

### Monitoring and Maintenance:

Implement monitoring mechanisms to track model performance over time and ensure continued effectiveness in predicting survival outcomes. Regularly update the model with new data and retrain as necessary to adapt to changing patterns and improve predictive accuracy.

## ALGORITHM DESCRIPTION:

### 1. Support Vector Machine (SVM):

Data Collection: Gathered Titanic passenger dataset containing attributes like class, name, age, gender, etc.

Data Preprocessing: Handled missing values, outliers, encoded categorical variables, and scaled numerical features.

Train-Test Split: Split the dataset into training and testing sets.

Model Selection: Chose SVM for its effectiveness in handling complex datasets.

Model Training: Trained SVM model on the training set with appropriate kernel functions and hyperparameters.

Model Evaluation: Evaluated SVM model's performance using accuracy, precision, recall, F1-score, and AUC metrics on the testing set.

Model Interpretation (Optional): Interpreted SVM model to understand decision boundaries and support vectors.

Model Deployment (Optional): Deployed trained SVM model for real-time prediction of survival.

Monitoring and Maintenance: Implemented monitoring to track SVM model's performance and updated it with new data.

2. **Linear Regression:**

   Data Collection: Gathered Titanic passenger dataset.

   Preprocessing: Handled missing values, outliers, encoded variables, and scaled features.

   Split Data: Divided dataset into training and testing sets.

   Model Selection: Chose linear regression for simplicity.

   Training: Trained model on the training set.

   Evaluation: Assessed model performance using metrics like MSE, RMSE, MAE, and R-squared.

   Interpretation: Analyzed coefficients for feature importance.

   Deployment: Optionally deployed model for predictions.

   Maintenance: Monitored and updated model with new data as needed.

3. **Naive Bayes:**

   Follow similar steps as SVM but mention specificities of Naive Bayes such as its assumption of independence among features.

4. **Decision Tree:**

   Similar to SVM, but emphasize the tree structure, splitting criteria, and interpretability.

5. **Logistic Regression:**

   Similar to SVM, but highlight logistic regression's probabilistic nature and interpretation of coefficients.

6. **K-Nearest Neighbors (KNN) Classifier:**

   Focus on distance metrics, choice of 'K', and the classification decision based on the majority vote of neighbors.

7. **Gradient Boost:**

   Emphasize boosting technique, ensemble of weak learners, and iterative improvement of model performance.

8. **Random Forest:**

   Highlight ensemble of decision trees, bagging technique, and feature randomization.

9. **K-Nearest Neighbors (KNN) Regressor:**

   Similar to KNN Classifier, but focus on regression problem and predicting continuous outcomes.

**COMPARISION TABLE:**

| **Algorithms / Performance metrics** | **Description** | **Accuracy** | **MSE** |
|---|---|---|---|
| SVM | Finds optimal hyperplane for separation | 0.775 | - |
| Naive Bayes | Gaussian Bayes Algorithm | 0.782 | - |
| Decision Tree | Tree-like model using decision rules | 0.796 | - |
| Logistic Regression | Non parametric, instance- based classification | 0.787 | - |
| KNN Classifier | Non-parametric instance-based classification | 0.805 | - |
| Gradient Boost | Ensemble learning combines multiple weak models | - | 0.12 |
| Random Forest | Multitude of decision trees | 0.776 | - |
| Linear Regression | Model between dependent and independent variables | - | 0.21 |
| KNN Regressor | Non- parametric instance-based classification | - | 0.21 |

## CONCLUSION:

In conclusion, after evaluating multiple supervised learning algorithms on the Titanic dataset, we found that KNN Classifier achieved the highest accuracy of 0.815(82%) among all algorithms tested. While accuracy is an important metric, it's essential to consider other factors such as interpretability, computational efficiency, and scalability when selecting the best algorithm for a specific task. Overall, KNN Classifier showed promising performance and may be a suitable choice for further exploration or deployment in real-world applications.