# [Titanic – Machine Learning from Disaster](https://www.kaggle.com/competitions/titanic/overview)

Table of Contents

[Titanic – Machine Learning from Disaster 1](#_Toc188986446)

[Thought Process – updated 28/01/2025 1](#_Toc188986447)

## Thought Process – updated 28/01/2025

1. ~~Import relevant packages (add where necessary)~~
   1. Pandas – For data manipulation and analysis
   2. NumPy – For numerical computations
   3. matplotlib and seaborn – For data visualization
   4. scikit-learn – For machine learning and data preprocessing
2. ~~Import relevant datasets~~
   1. Train.csv – training data
   2. Test.csv – testing data
   3. Gender\_submission.csv – submission file
3. **~~Data Cleaning & Preprocessing~~**
   1. Remove null values or impute them (use techniques like mean/mode imputation for numerical or categorical variables, or even predictive imputation if applicable).
   2. Check for duplicate rows and remove them if found.
   3. Ensure data types are correct (e.g., converting categorical variables to appropriate types like category in pandas).
   4. Create input\_features df, removing irrelevant columns (not for training)
4. **~~Exploratory Data Analysis (EDA)~~**
   1. **Visualize relationships**: Use visualizations to examine correlations between features like gender, age, class, and survival.
   2. **Identify outliers**: Detect anomalies that could skew model predictions.
   3. **Feature relationships**: For example, explore how survival rates vary by Pclass, Sex, and Embarked.
5. **~~Feature Engineering~~**
   1. Create new features like family size (SibSp + Parch + 1) or categorize Age into age groups.
   2. Convert categorical variables (like Sex, Embarked) into numerical features using one-hot encoding or label encoding.
   3. Drop irrelevant columns like Name, Ticket, or Cabin, unless you plan to extract meaningful information (e.g., extracting titles from Name).
6. **~~Feature Scaling (Standardization)~~**
   1. **Scale numerical features**: Normalize or standardize features like Age and Fare to ensure comparable ranges.
   2. **Optional**: If needed, use StandardScaler or MinMaxScaler from scikit-learn.
7. **~~Model Building~~**
   1. Start with a simple model (e.g., Logistic Regression).
   2. Experiment with more complex models (e.g., Decision Trees, Random Forests, Gradient Boosting models).
   3. Evaluate models using cross-validation.
8. **~~Hyperparameter Tuning~~**
   1. Use Grid Search or Randomized Search to optimize model performance.
9. **Make Predictions**
   1. Use the trained model to predict the survival of passengers in test.csv.
   2. Ensure your submission file follows the required format (PassengerId and Survived).
10. **Evaluate and Iterate**
    1. Assess model performance using metrics such as accuracy, precision, recall, and F1-score.
    2. Iterate on features, model choice, or hyperparameters to improve performance.

## Input Parameters

(Drop all other unnecessary columns)

* Pclass: Ticket class, 1 = 1st, 2 = 2nd, 3 = 3rd (proxy for socio-economic status)
* Sex (strong survival predictor)
* Age
* Fare (could impact survival odds)
* Title
* Family / Party Grouping
* Ticket\_Prefix
* Embarked (Port of embarkation)

Engineered Features

* Title
* First\_Name
* Last\_Name
* Name\_Additional
* Ticket\_Prefix
* Ticket\_Number
* Family and Party Grouping

## Recommendations

* **EDA First**: Start with an EDA step after loading the data to get an overview of null values, outliers, and feature distributions. This can guide your cleaning process.
* **Imputation Plan**: Decide how to handle nulls (e.g., for Age, consider grouping by Pclass and Sex to impute more meaningful values).
* **Validation Strategy**: Use a validation set or K-Fold cross-validation to avoid overfitting on training data.
* **Baseline Model**: Submit a simple model (like predicting survival based on Sex and Pclass) to set a benchmark, then improve from there.
* Add new feature: group by surname? In Name column, first value before “,”
  + Compare survivability of families / parties traveling together
    - Do they have similar / varying survivability?