**Data Preprocessing Techniques on Titanic Dataset**

**Project Overview**

Data preprocessing is an essential step in any data analysis or machine learning pipeline. It involves cleaning, transforming, and organizing raw data so that it can be efficiently analyzed or used to train models.

In this project, we applied **three preprocessing techniques** on the Titanic dataset from Kaggle:

1. Binning (Discretization)
2. Transformation
3. Normalization

**1. Binning (Discretization)**

**Description:**  
Binning is the process of converting continuous numerical data into discrete categories (bins). It is commonly used to reduce the effect of minor observation errors and to simplify data for analysis.

**Example:**

* Age of passengers in the Titanic dataset was divided into four categories:
  + Child (0–12)
  + Teen (13–18)
  + Adult (19–60)
  + Senior (61–80)

**Python Demo:**

import pandas as pd

train = pd.read\_csv("data/train.csv")

bins = [0, 12, 18, 60, 80]

labels = ['Child', 'Teen', 'Adult', 'Senior']

train['Age\_Bin'] = pd.cut(train['Age'], bins=bins, labels=labels)

train.to\_csv("data/train\_binned.csv", index=False)

**Result:**

* New column Age\_Bin added to dataset.
* Example:  
  | Age | Age\_Bin |  
  |-----|---------|  
  | 5 | Child |  
  | 17 | Teen |  
  | 25 | Adult |

**GitHub Reference:**  
src/01\_binning.py – <https://github.com/VinukiCharithma/data-preprocessing-titanic.git>

**2. Transformation**

**Description:**  
Transformation applies mathematical functions to change the distribution or scale of the data. Common transformations include logarithmic, square root, and standardization. It helps to reduce skewness and improve model performance.

**Example:**

* The Fare column in Titanic dataset was skewed. We applied a **log transformation**:
* Fare\_Log = log(1 + Fare)

**Python Demo:**

import pandas as pd

import numpy as np

train = pd.read\_csv("data/train.csv")

train['Fare\_Log'] = np.log1p(train['Fare'])

train.to\_csv("data/train\_transformed.csv", index=False)

**Result:**

* Column Fare\_Log added, reducing skewness.
* Example:  
  | Fare | Fare\_Log |  
  |------|----------|  
  | 7.25 | 2.09 |  
  | 71.83| 4.30 |

**GitHub Reference:**  
src/02\_transformation.py [– https://github.com/VinukiCharithma/data-preprocessing-titanic.git](–%20https:/github.com/VinukiCharithma/data-preprocessing-titanic.git)

**3. Normalization**

**Description:**  
Normalization scales numerical features into a fixed range (commonly 0–1). It ensures that features contribute equally to distance-based algorithms and improves convergence in optimization algorithms.

**Example:**

* Age and Fare columns were normalized using Min-Max scaling:
* X\_norm = (X - X\_min) / (X\_max - X\_min)

**Python Demo:**

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

train = pd.read\_csv("data/train.csv")

scaler = MinMaxScaler()

train[['Age\_Norm', 'Fare\_Norm']] = scaler.fit\_transform(train[['Age', 'Fare']])

train.to\_csv("data/train\_normalized.csv", index=False)

**Result:**

* New columns Age\_Norm and Fare\_Norm in range [0,1].
* Example:  
  | Age | Age\_Norm | Fare | Fare\_Norm |  
  |-----|----------|------|-----------|  
  | 22 | 0.27 | 7.25 | 0.01 |  
  | 38 | 0.48 | 71.83| 0.17 |

**GitHub Reference:**  
src/03\_normalization.py – <https://github.com/VinukiCharithma/data-preprocessing-titanic.git>

**Conclusion**

* The preprocessing steps performed help clean, scale, and structure the Titanic dataset for analysis or machine learning.
* Binning simplifies continuous variables, Transformation reduces skewness, and Normalization ensures features are on the same scale.
* These steps are essential before applying predictive modeling algorithms.