**Titanic Dataset Preprocessing Pipeline**

This project demonstrates step-by-step data preprocessing and feature engineering for the Titanic dataset using Python and key libraries such as **pandas**, **numpy**, **seaborn**, and **scikit-learn**.

**Dataset**

* Dataset used: Titanic-Dataset.csv
* Task: Prepare the dataset for predictive modeling (e.g., survival prediction)

**Steps Covered**

**Step 1: Import Libraries and Load Dataset**

* Imported essential libraries
* Loaded the dataset using pandas
* Displayed basic info (.shape, .info(), .head())

**Step 2: Handle Missing Values**

* **Age**: Filled with **median**
* **Embarked**: Filled with **mode**
* **Cabin**: Dropped due to excessive missing values

**Step 3: Encode Categorical Features**

* Applied **One-Hot Encoding** to:
  + Sex
  + Embarked
* Used drop\_first=True to avoid dummy variable trap

**Step 4: Convert Numerical Features into Groups**

* **Age**:
  + Binned into groups: Child, Teen, Young Adult, Middle-aged, Senior (Labels: 0–4)
* **Fare**:
  + Binned into **quartiles** (Labels: 0–3)
* Dropped original Age and Fare columns (optional step)

**Step 5: Outlier Detection and Removal**

* Plotted **boxplots** for AgeGroup and FareGroup
* Removed **top 1% FareGroup** outliers

**Step 6: Feature Engineering**

* Worked on a safe copy of the dataframe
* **Extracted Title** from Name (e.g., Mr, Miss, etc.)
  + Rare titles grouped into 'Rare'
* **Family Features**:
  + FamilySize = SibSp + Parch + 1
  + IsAlone = 1 if FamilySize == 1, else 0
* Re-processed Age and Fare if they exist
* **One-Hot Encoded** the Title feature

**Final Feature Examples**

Columns (if present after preprocessing):

* AgeGroup (0–4)
* FareGroup (0–3)
* FamilySize
* IsAlone (0 or 1)
* One-hot encoded Sex, Embarked, Title

**Libraries Used**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import StandardScaler

**Notes**

* Data cleaning and feature engineering are critical before applying machine learning models.
* Make sure to verify column existence after each transformation.
* Always validate the final dataset shape and content before model training.