# Task 1: Data Preprocessing, Transformation, and Loading Pipeline

## 1. Objective

The objective of Task 1 is to build an automated pipeline for data preprocessing, transformation, and loading using tools like pandas and scikit-learn. This forms the foundation of any data science workflow, ensuring the data is clean and ready for analysis or model building.

## 2. Tools and Libraries Used

- pandas  
- scikit-learn  
- seaborn (for sample dataset)  
- numpy (implicit via scikit-learn)

## 3. ETL Pipeline Steps

The ETL process is divided into three main stages:  
- Extract: Load the dataset  
- Transform: Clean, impute, encode, and scale the data  
- Load: Final processed data ready for use in ML models or storage

## 4. Code Implementation

import pandas as pd  
from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import StandardScaler, OneHotEncoder  
from sklearn.compose import ColumnTransformer  
from sklearn.pipeline import Pipeline  
from sklearn.impute import SimpleImputer  
  
def load\_data():  
 import seaborn as sns  
 df = sns.load\_dataset('titanic')  
 return df  
  
def preprocess\_data(df):  
 df = df.drop(columns=['deck', 'embark\_town', 'alive', 'class', 'who', 'adult\_male'])  
 X = df.drop('survived', axis=1)  
 y = df['survived']  
 numeric\_features = X.select\_dtypes(include=['int64', 'float64']).columns  
 categorical\_features = X.select\_dtypes(include=['object', 'category', 'bool']).columns  
  
 numeric\_transformer = Pipeline(steps=[  
 ('imputer', SimpleImputer(strategy='mean')),  
 ('scaler', StandardScaler())  
 ])  
  
 categorical\_transformer = Pipeline(steps=[  
 ('imputer', SimpleImputer(strategy='most\_frequent')),  
 ('onehot', OneHotEncoder(handle\_unknown='ignore'))  
 ])  
  
 preprocessor = ColumnTransformer(  
 transformers=[  
 ('num', numeric\_transformer, numeric\_features),  
 ('cat', categorical\_transformer, categorical\_features)  
 ]  
 )  
  
 X\_processed = preprocessor.fit\_transform(X)  
 return X\_processed, y, preprocessor  
  
def run\_pipeline():  
 df = load\_data()  
 X, y, transformer = preprocess\_data(df)  
 print("Data processing completed.")  
 print("Shape of processed data:", X.shape)  
 return X, y  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 X, y = run\_pipeline()

## 5. Output

The output of the pipeline is a transformed dataset ready for use in machine learning models. The numeric features are scaled, and the categorical features are one-hot encoded. The shape of the processed dataset indicates successful preprocessing.

## 6. Conclusion

This pipeline automates the preprocessing step for any dataset. By using reusable components from scikit-learn and pandas, it allows data scientists to quickly clean and transform data in preparation for modeling tasks.