**Aim:** To preprocess (Imputation, Label encoding and data cleaning) and prepare data using NumPy and Pandas in Python for effective analysis and modeling.

**Software Used:** Google Python Colab

**Theory:**

**Data Preprocessing:**

Data preprocessing is the process of preparing raw data for analysis by cleaning and transforming it into a usable format. In data mining it refers to preparing raw data for mining by performing tasks like cleaning, transforming, and organizing it into a format suitable for mining algorithms.

**Key Aspects of Data Preparation:**

1. **Data Collection:**  
   The first step in data preparation is acquiring data from diverse sources such as databases, CSV files, sensors, or through web scraping. This data is then imported and loaded into a working environment where it can be processed further.
2. **Data Cleaning:**  
   Raw data is often messy and filled with errors, missing values, duplicates, or inconsistencies. Data cleaning is the process of identifying and fixing these issues to ensure the dataset is accurate and complete. Common techniques include using functions like dropna() for removing missing values, fillna() for imputing them, and duplicated() for finding and removing repeated entries.
3. **Data Transformation:**  
   Once cleaned, the data may still not be ready for analysis. Data transformation involves converting data into a more usable format through techniques such as feature engineering, normalization, scaling, and label encoding. This ensures the data is better aligned with the requirements of the algorithms that will analyze it.
4. **Data Integration:**Often, datasets are obtained from multiple sources with different formats and structures. In such cases, data integration is necessary to combine them into a single, unified dataset. This may involve resolving naming conflicts, unit mismatches, or format differences.
5. **Data Reduction:**  
   When working with large datasets, it becomes important to reduce the size of the data without losing significant information. Techniques such as dimensionality reduction (e.g., using PCA) help in reducing the number of variables, thus simplifying the model and improving computation time.
6. **Data Sampling:**  
   In scenarios involving very large datasets, sampling methods like random sampling or stratified sampling are used to select a representative subset of the data for quicker and efficient analysis, especially during exploratory phases.
7. **Data Splitting:**  
   For modeling purposes, the data is generally split into training and testing sets. This is a crucial step in evaluating how well a model performs on unseen data. Common splitting ratios include 80:20 or 70:30 for training and testing respectively.
8. **Feature Selection:**  
   Not every column in a dataset contributes equally to the prediction task. Feature selection involves identifying and using the most relevant features while removing redundant or irrelevant ones. This improves model performance and reduces overfitting.
9. **Data Formatting:**  
   Consistency in data types, date-time formats, and data structures is vital. Formatting ensures that all columns follow a uniform structure, which is important for applying algorithms and visualization tools correctly.
10. **Data Validation and Verification:**  
    Validation checks whether the dataset truly represents the problem it is meant to solve, while verification ensures that all steps taken in cleaning and transforming the data were executed correctly. These help in maintaining the quality and trustworthiness of the dataset.
11. **Data Documentation:**

Maintaining proper documentation of the dataset, such as metadata, data dictionaries, and transformation logs, ensures transparency, reproducibility, and easier understanding for future reference or for team collaboration.

**Handling Missing Data(Imputation)**  
Imputation is the process of replacing missing or null values in a dataset with appropriate substitute values. This helps maintain the dataset’s completeness and avoids errors during analysis or modeling.

**Mean/Median/Mode Imputation**

* Use for: Numerical columns
* Mean: When data is normally distributed
* Median: When data has outliers
* Mode: For categorical or repeated values

**Label Encoding**

Label Encoding is a technique that is used to convert categorical columns into numerical ones so that they can be fitted by machine learning models which only take numerical data. It is an important pre-processing step in a machine-learning project. It assigns a unique integer to each category in the data, making it suitable for machine learning models that work with numerical inputs.

When Label Encoding is Not Ideal:

* If categories have no natural order, label encoding can mislead algorithms into thinking one value is greater than another (e.g., "Red" → 0, "Green" → 1, "Blue" → 2).
* In such cases, use one-hot encoding instead.

### **Libraries Used**

**NumPy (Numerical Python)**

Purpose:NumPy is a core library in Python used for numerical computations. It provides support for handling multi-dimensional arrays and matrices, along with a large collection of mathematical functions to operate on these arrays efficiently.

Key Features:

* Offers the powerful ndarray object, which is faster and more memory-efficient than Python lists.
* Supports element-wise operations, linear algebra, Fourier transforms, and statistics
* Works as the foundation for many other data science libraries (like Pandas, SciPy, and scikit-learn)

**Pandas (Python Data Analysis Library)**

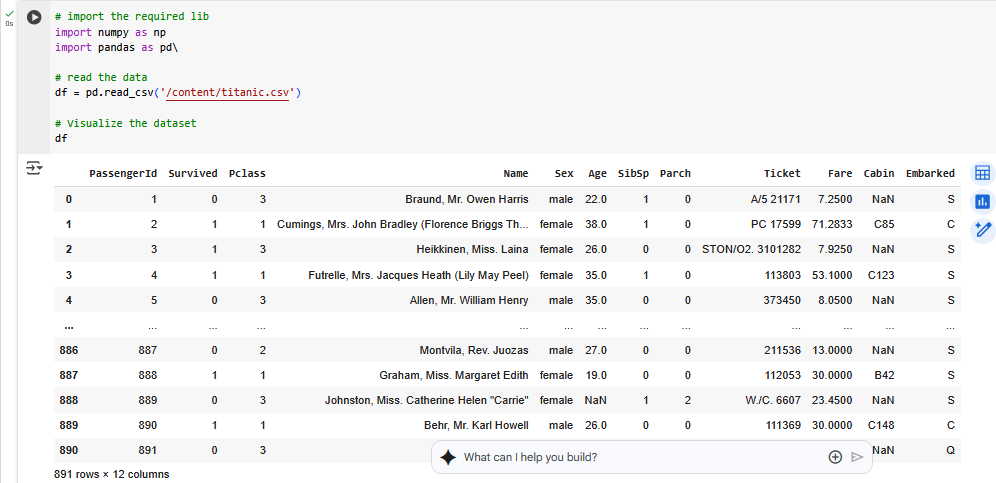
Purpose:  
Pandas is a powerful, flexible, and easy-to-use library for data manipulation and analysis. It is built on top of NumPy and provides tools to clean, transform, and analyze structured data.

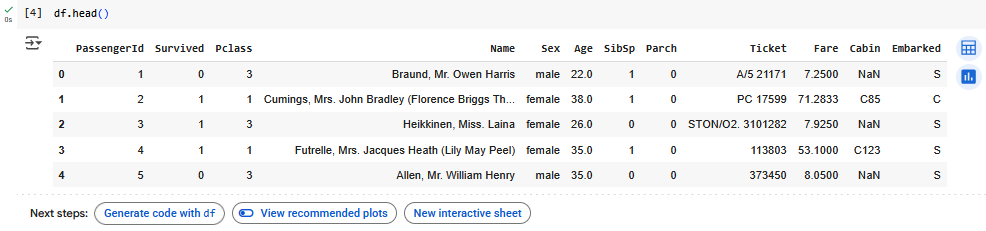
Key Features:

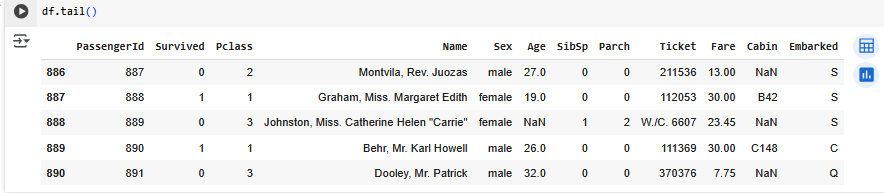
* Easy handling of missing data, filtering, and aggregation
* Supports importing and exporting data from various formats (CSV, Excel, SQL, etc.)
* Integrates well with other libraries like Matplotlib, scikit-learn, and NumPy

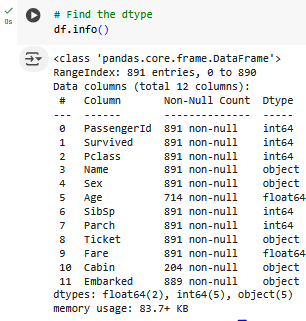
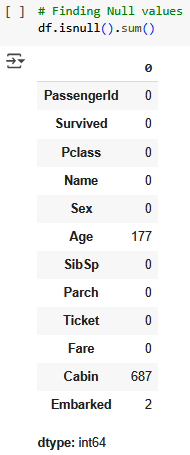
**Outputs:**

**1) Titanic.csv:**

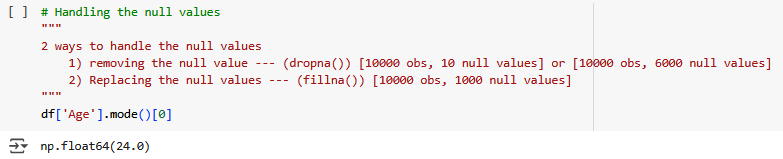


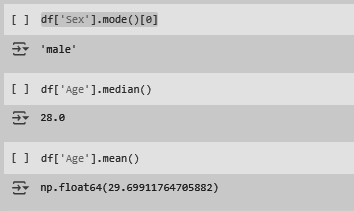


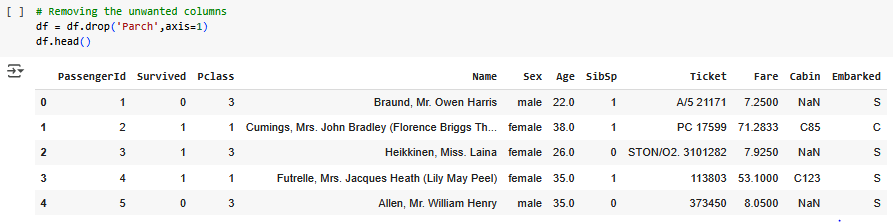


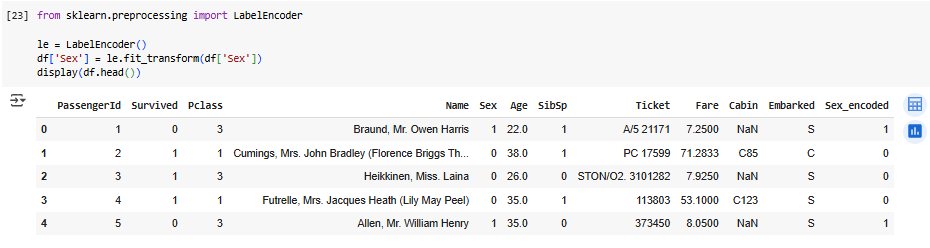
 

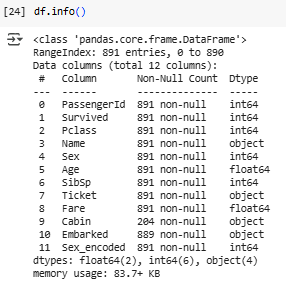




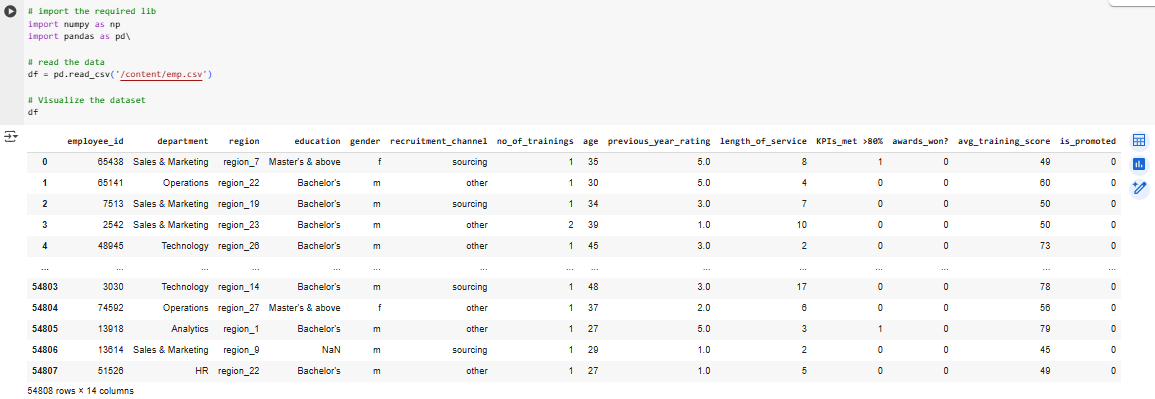


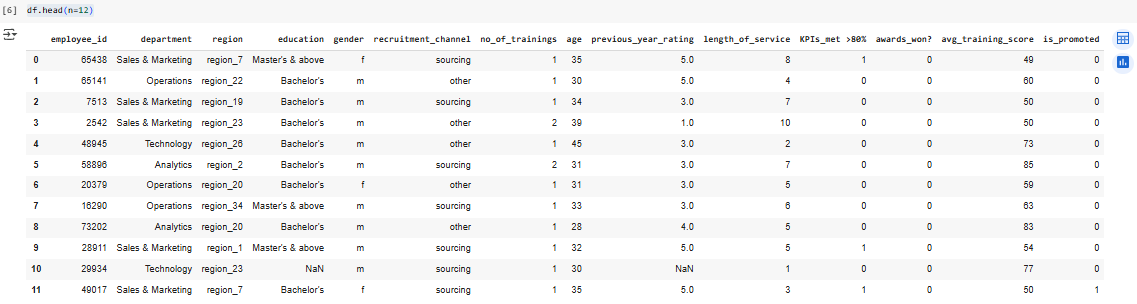


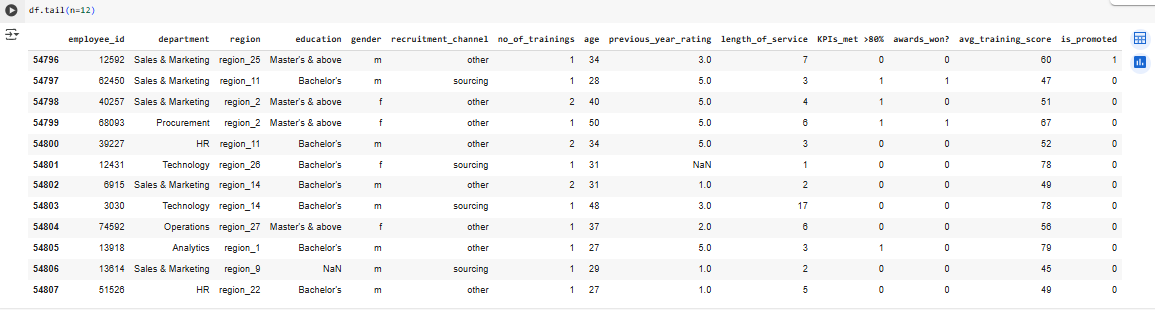


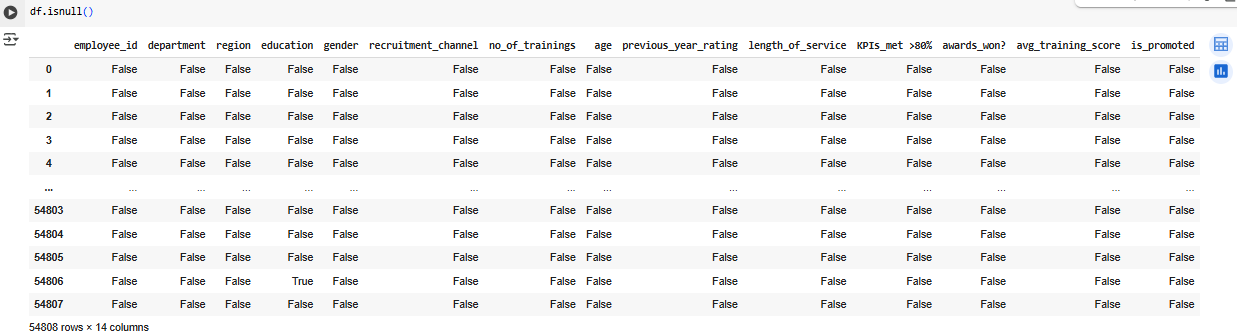


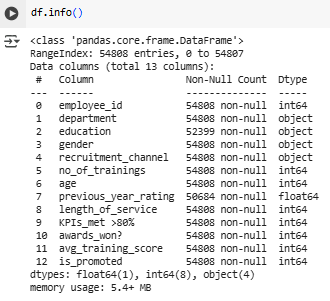
**2)Emp.csv**

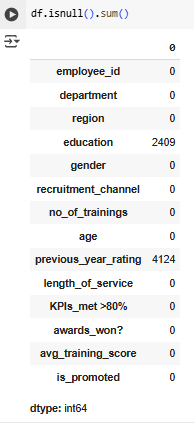


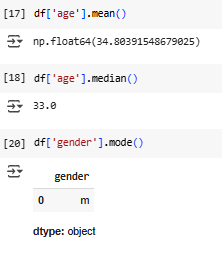


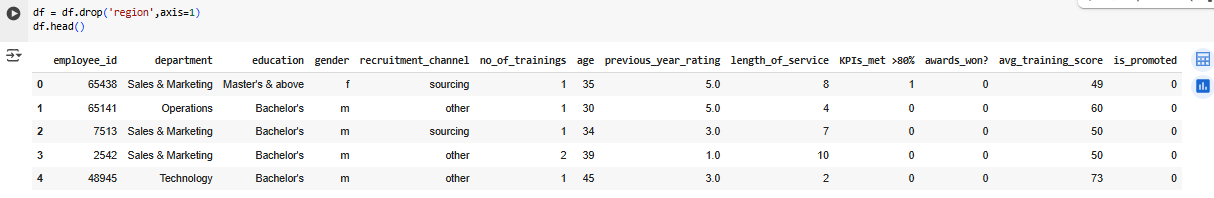


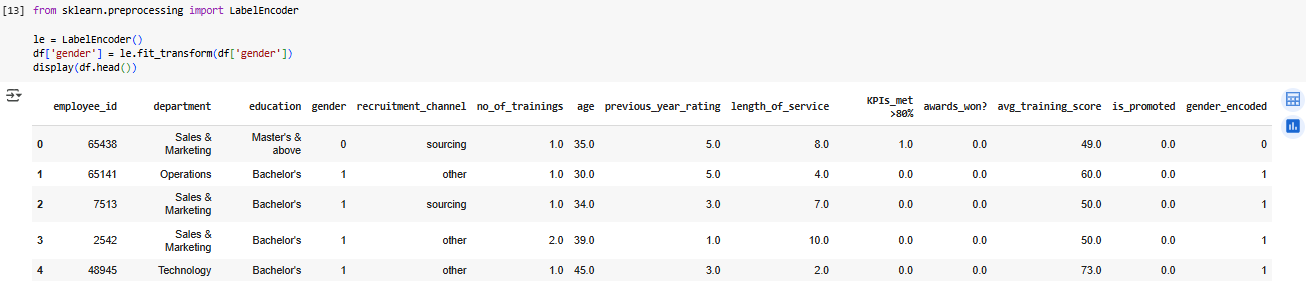


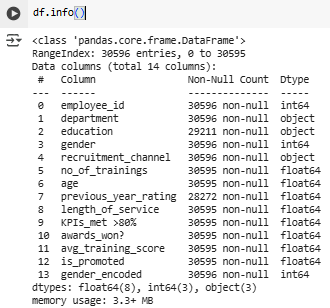




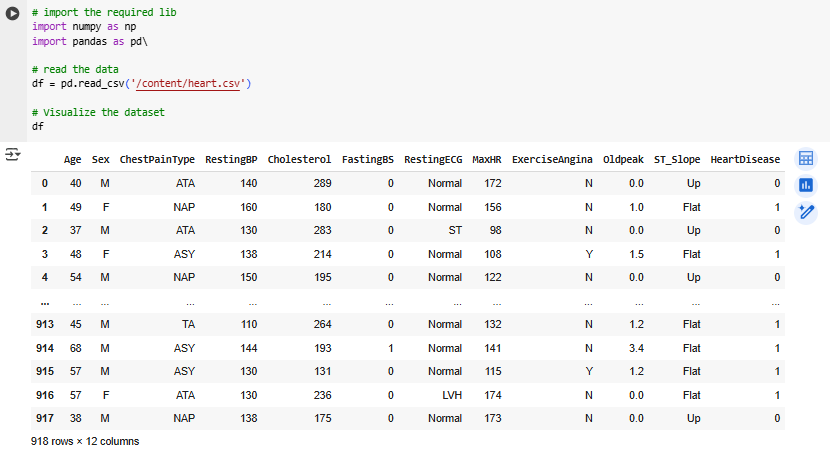


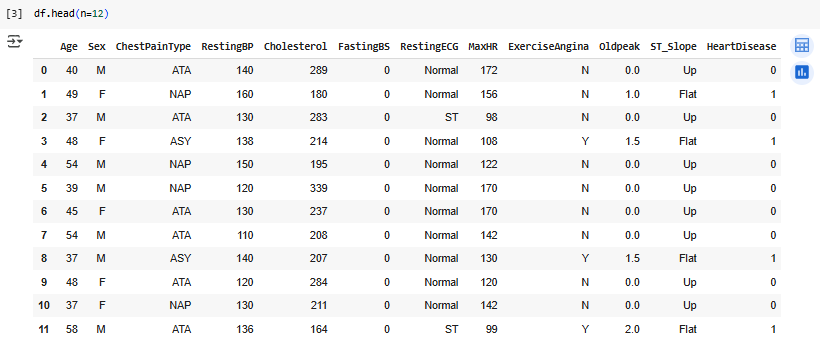


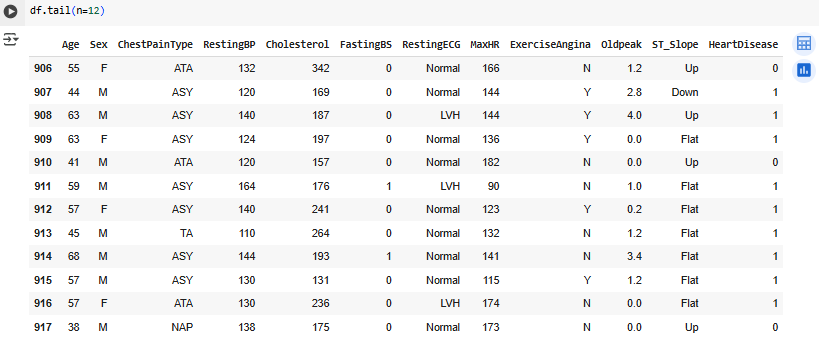


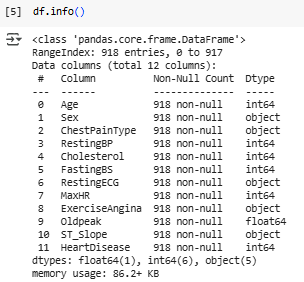


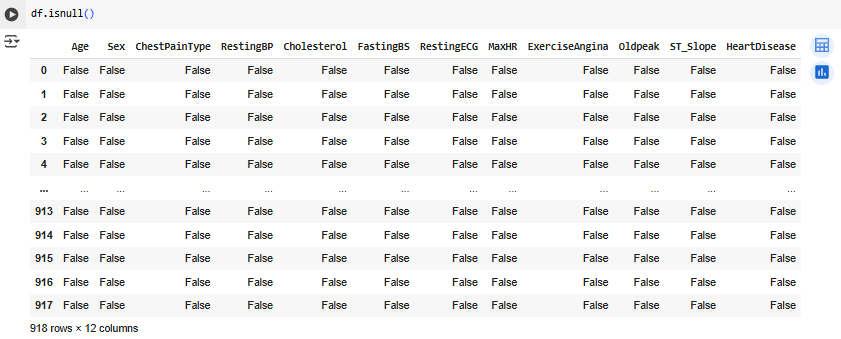
**3)Heart.csv:**

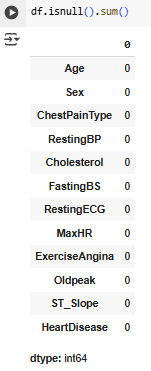


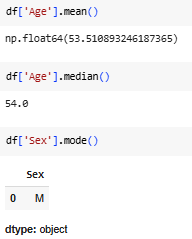


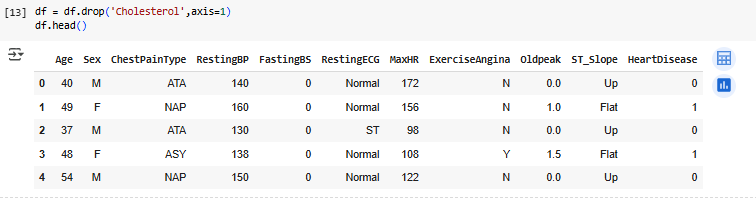


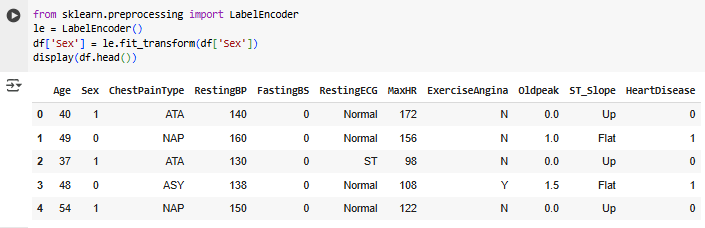


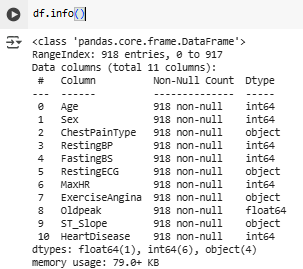












**Conclusion:** From this, we conclude that with the help of libraries like NumPy and Pandas, is highly effective for data preprocessing. It provides powerful tools for efficient data handling, transformation, and analysis.