Name : Vaishnavi Eknath Avhad
Class : D15C
Roll No. : 41

Practical 2

Aim : To clean and preprocess a dataset by handling missing values, removing duplicates, encoding categorical variables, correcting data types, and addressing outliers for accurate analysis or modeling.

This practical involves working with a dataset obtained from Kaggle, which is aimed at practicing and applying various data preprocessing techniques. Data preprocessing is a crucial step in any data analysis or machine learning pipeline, as it helps ensure the quality of the data before further analysis or model training. In this practical, we will perform several key operations on the dataset to prepare it for analysis or machine learning tasks.

Key Operations :

1. Handling Missing Values :
   One of the most common issues in real-world datasets is missing or incomplete data. In this practical, we focus on handling missing values in a specific column — Age. Missing values can lead to biased results, so we will fill these missing entries using a statistical approach like replacing them with the median age, which is less sensitive to outliers compared to the mean.

2. Removing Duplicates :
   Duplicates can occur in datasets due to data entry errors, repeated records, or merging datasets. These duplicates can distort the analysis and cause overfitting in machine learning models. In this step, we will identify and remove any duplicate records, ensuring that each entry in the dataset is unique.

3. Encoding Categorical Variables :
   Many machine learning algorithms require numerical input, but real-world data often contains categorical variables (e.g., gender, education level, etc.). In this practical, we

will encode categorical variables into numerical formats using techniques like one-hot encoding or label encoding. This will enable machine learning models to process categorical data effectively.

4. Fixing Data Types :

   Often, columns in datasets may not have the correct data types, such as numeric columns being stored as strings or dates being stored as objects. In this practical, we will ensure that each column has the correct data type. For example, the Salary column should be stored as a float to allow for numerical operations, and date columns should be converted to datetime format.

5. Handling Outliers :

   Outliers are data points that differ significantly from the rest of the data and can skew the results of analysis. For example, an unusually high age (e.g., 120 years) might be an error. In this practical, we will identify outliers in the Age column, especially any values greater than 100, and either correct or remove them to ensure that the dataset is more representative of real-world values.

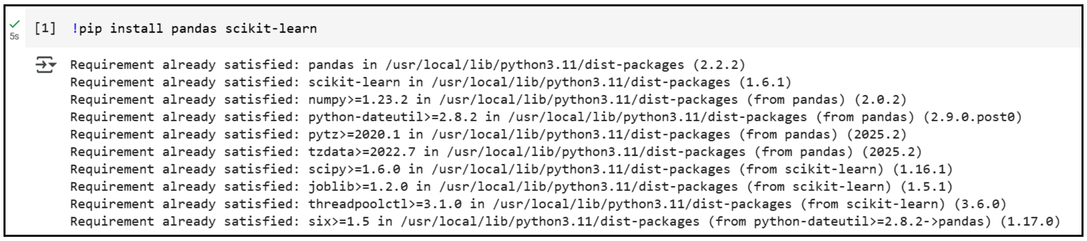
Objective :

The objective of this practical is to ensure the dataset is clean, consistent, and ready for further analysis or machine learning tasks. By handling missing values, removing duplicates, encoding categorical variables, fixing data types, and managing outliers, we create a reliable dataset that can lead to more accurate insights and predictions.

Importance of Data Preprocessing :

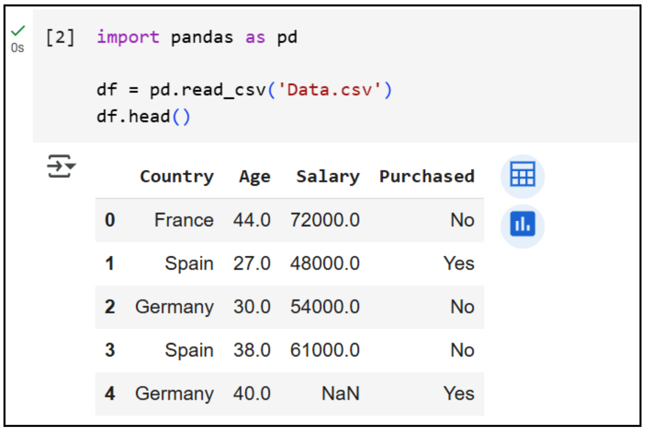
Data preprocessing is a foundational step in any data science or machine learning project. A dataset that is clean and well-structured will lead to more accurate analyses and model performance. Data preprocessing helps reduce bias, improve the quality of insights, and optimize machine learning algorithms, making it a crucial skill for any data professional.

Screenshots :



```
!pip install pandas scikit-learn
```

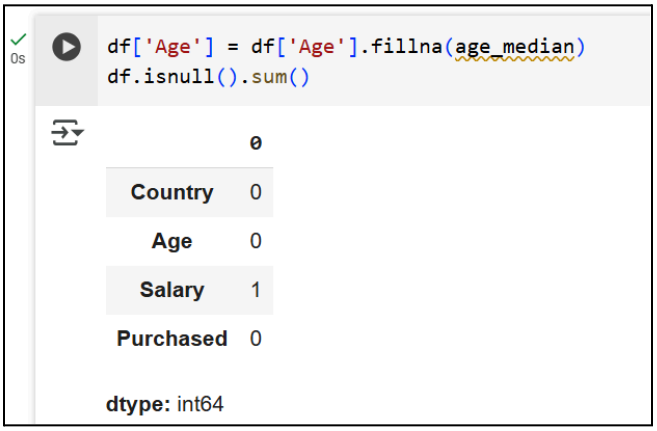
```
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.16.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```



```
import pandas as pd

df = pd.read_csv('Data.csv')
df.head()
```

	Country	Age	Salary	Purchased
0	France	44.0	72000.0	No
1	Spain	27.0	48000.0	Yes
2	Germany	30.0	54000.0	No
3	Spain	38.0	61000.0	No
4	Germany	40.0	NaN	Yes

1. Handling the missing values (on age with median)

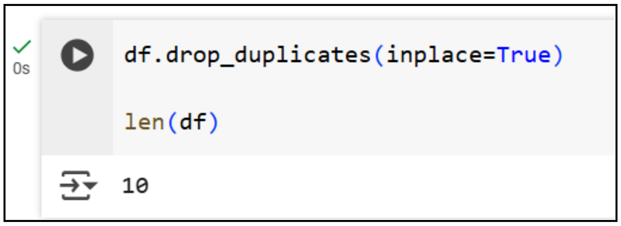


```
df['Age'] = df['Age'].fillna(age_median)
df.isnull().sum()
```

	0
Country	0
Age	0
Salary	1
Purchased	0

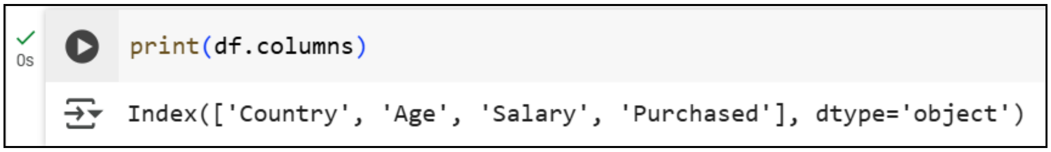
dtype: int64

2. Remove duplicates



```
df.drop_duplicates(inplace=True)

len(df)
```

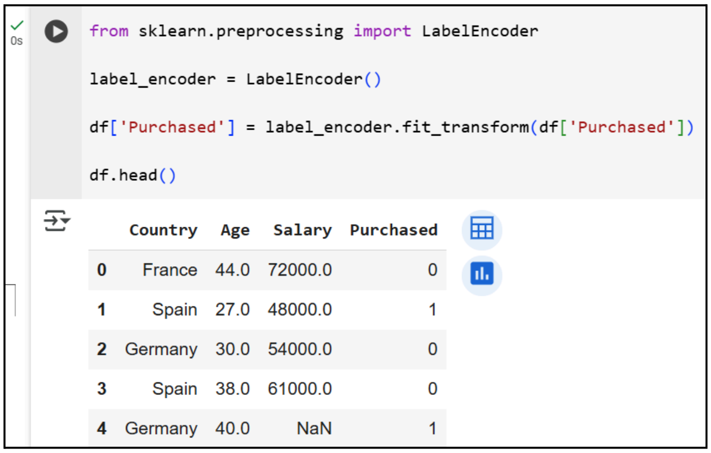
```
10
```



```
print(df.columns)
```

```
Index(['Country', 'Age', 'Salary', 'Purchased'], dtype='object')
```

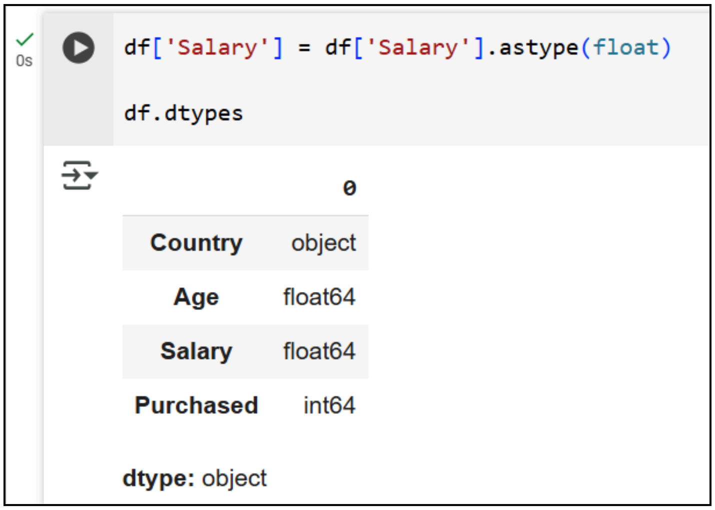
3. Encode categorical variables



```
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()

df['Purchased'] = label_encoder.fit_transform(df['Purchased'])

df.head()
```

```
   Country   Age   Salary  Purchased
0   France  44.0  72000.0          0
1    Spain  27.0  48000.0          1
2  Germany  30.0  54000.0          0
3    Spain  38.0  61000.0          0
4  Germany  40.0      NaN          1
```

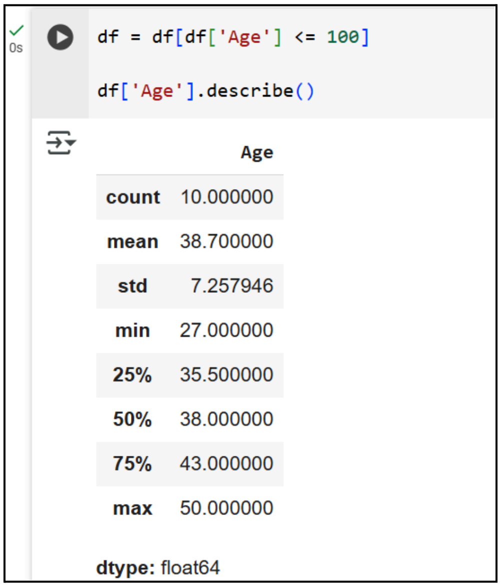
4. Fix datatypes (e.g salary as float)



```
df['Salary'] = df['Salary'].astype(float)

df.dtypes
```

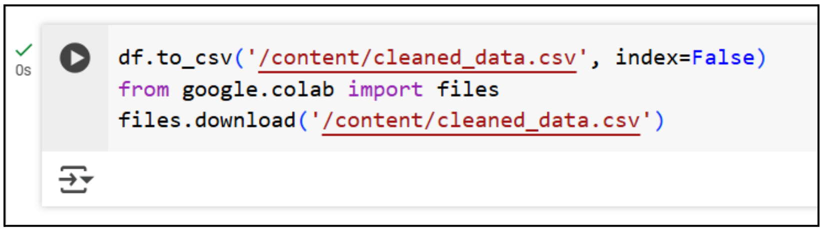
```
                 0
  Country   object
      Age  float64
   Salary  float64
Purchased    int64

dtype: object
```

5. Handle Outliers



```
df = df[df['Age'] <= 100]

df['Age'].describe()
```

```
             Age
count  10.000000
mean   38.700000
std     7.257946
min    27.000000
25%    35.500000
50%    38.000000
75%    43.000000
max    50.000000

dtype: float64
```



```
df.to_csv('/content/cleaned_data.csv', index=False)
from google.colab import files
files.download('/content/cleaned_data.csv')
```

Conclusion :

In this practical, we applied key data preprocessing steps to clean and prepare the dataset for analysis. We handled missing values by replacing them with the median, removed duplicates, encoded categorical variables, fixed data types, and addressed outliers. These steps ensured the dataset was clean, consistent, and ready for further analysis or machine learning tasks. Proper data preprocessing is crucial for accurate insights and improved model performance, demonstrating its importance in the data science pipeline.