GLS University
Faculty of Computer Applications & Information Technology
iMSCIT SEM VI
221601605 Practicals on Machine Learning
Practical Assignment
Unit 1

1.	Write a Machine Learning program to remove duplicate entries from a customer database using the drop_duplicates() method in pandas. Demonstrate how to remove duplicates based on specific columns, keep either the first or last occurrence.
	data = {
	'Name': ['John', 'Anna', 'Peter', 'John'],
	'Age': [24, 13, 53, 24]
	}

2.	Write a Machine Learning program to handle missing values in a dataset. Demonstrate two approaches: ● Deleting rows or columns with missing values using dropna(). ● Imputing missing values using strategies like mean, median, or a specified constant. data = {'Name': ['John', 'Anna', 'Peter', None], 'Age': [24, 13, None, 33]}
3.	Write a Machine Learning program to standardize inconsistent date formats in a dataset using the to_datetime() method in pandas. data = {'Date': ['2023-01-01', '01/02/2023', '2023.03.03']}
4.	Write a Machine Learning program to filter out irrelevant or erroneous data points from a dataset based on predefined criteria, 1. Age between 25 to 60 2. Salary greater than 10000 data = { 'Name': ['John', 'Anna', 'Peter', 'Linda'], 'Age': [24, 13, 53, 33], 'Salary': [50000, 2000, 100000, 30000]

<table>
<tr><td></td><td>}</td></tr>
<tr><td><b>5.</b></td><td>Write a Machine Learning program to clean textual data by removing HTML tags, special characters, and punctuation. Use Python's re library to demonstrate this process.
text = "<html>Hello! This is <b>clean</b> text.</html>"</td></tr>
<tr><td><b>6.</b></td><td>Write a Machine Learning program to convert categorical variables into numerical representations using one-hot encoding and label encoding techniques. Use pandas and sklearn to demonstrate the encoding process.
data ={'Department': ['HR', 'Legal', 'Marketing', 'Management']}</td></tr>
<tr><td><b>7.</b></td><td>Write a Machine Learning program to scale numerical features in a dataset using Min-Max scaling.
data = {'Income': [15000, 1800, 120000, 10000], 'Age': [25, 18, 42, 51]}</td></tr>
<tr><td><b>8.</b></td><td>Write a Machine Learning program to transform skewed distributions using log or square root transformations. Visualize the effect of these transformations using matplotlib.
Define data as:
data = np.random.exponential(scale=2, size=1000)</td></tr>
<tr><td><b>9.</b></td><td>Write a Machine Learning program to preprocess textual data by applying tokenization, stemming, and lemmatization. Use the NLTK library for implementation.
text = "The striped bats are hanging on their feet for best."</td></tr>
<tr><td><b>10.</b></td><td>Write a Python script to create a Pandas DataFrame with the following data:
<table>
<tr><th>Name</th><th>Location</th><th>Age</th></tr>
<tr><td>John</td><td>New York</td><td>24</td></tr>
<tr><td>Anna</td><td>Paris</td><td>13</td></tr>
</table>
</td></tr>
</table>

	Peter Berlin 53 Linda London 33 1. Display the entire DataFrame. 2. Select and display all rows where the age is greater than 30. 3. Display the details of the first person (row with index 0). 4. Display the details of the first two people (rows with indexes 0 and 1).
11.	Write a Machine Learning program to Given a CSV file named 1.csv, perform the following tasks: 1. Load the CSV file into a Pandas DataFrame and print its contents. 2. Check and print the maximum number of rows that Pandas will display by default. 3. Display the first 5 rows of the DataFrame. 4. Display the last 5 rows of the DataFrame.
12.	Write a Machine Learning program to Create a DataFrame with the following data: ● Income: [15000, 1800, 120000, 10000] ● Age: [25, 18, 42, 51] ● Department: ['HR', 'Legal', 'Marketing', 'Management'] After creating the DataFrame, **scale** the 'Income' and 'Age' columns using **MinMaxScaler**. Print the scaled DataFrame.
13.	Write a Machine Learning program to Use the DataFrame from above Question, **encode the 'Department' column** using OneHotEncoder. Display the result of the encoding.
14.	Write a Machine Learning program to 1. **Create a DataFrame** with the following data: o Name: ['Alex', 'Bob', 'Clarke']

   - Age: [10, 12, 13]
   - Print the DataFrame.

2. **Read a CSV file** named employees_info.csv and display its contents.

3. **Get the general information** of the DataFrame (such as column names, data types, and memory usage).
4. **Access the 'name' and 'gender' columns** of the DataFrame.
5. **Retrieve the first row** of the DataFrame using .loc[].
6. **Get records from row index 0 to 5**, but only select the 'name' and 'job title' columns.
7. **Filter records** where the department is "Accounting", and select the name, job title, and department columns.

8. **Delete the 'time zone' column** from the DataFrame.
9. **Drop duplicates** from the DataFrame and display the result.
10. **Drop duplicates** based on the 'residence' column and show the DataFrame after dropping.
11. **Drop rows with missing values** and display the DataFrame after dropping them.
12. **Drop columns with missing values** and display the resulting DataFrame.
13. **Drop rows/columns with specific thresholds**. Keep rows with at least 2 non-NaN values and display the resulting DataFrame.

14. **Count the missing values** in each column of the DataFrame.
15. **Calculate the percentage of missing values** in each column of the DataFrame.
16. **Fill missing values** in the DataFrame with the default value "Unknown" and display the result.

17. **Standardize the 'name' column** by converting it to title case, then to lowercase, and display the results.
18. **Replace gender values** where 'M' is replaced with "Male" and 'F' with "Female" in the 'gender' column, and display the updated column.
19. **Remove non-numeric characters** from the 'phone' column using regex (specifically remove hyphens) and display the cleaned column.