

(B.TECH) Semester-VI AY 2024-25
ML-1: Lab Assignment No. 01

Problem Statement: Implement various pre-processing techniques on a given dataset.

Objectives:

1. To learn python programming with different modules/libraries.
2. To understand the concept of exploratory data analysis.

Theory:

Data Preprocessing: [Elaborate each point]

- Data Quality
- Major Tasks in Data Preprocessing
 - Data Cleaning
 - Data Integration
 - Data Reduction
 - Data Transformation and Data Discretization

Various types of data:

- **Numerical**

It represents quantitative measurement. Ex.: Height of a people, stock prices.

- **Discrete Data**

Integer based, often counts of something. Ex.: How many times did I toss “Heads”?

- **Continuous Data**

It has an infinite number of possible values. Ex.: How much rainfall on a given day?

- **Categorical Data**

Qualitative data, Ex.: Gender, Yes/No, etc. Assign some number to categorical data but they don't have any mathematical meaning

- **Ordinal Data**

Mixture of numerical and categorical data. Categorical data has mathematical meaning. For example: Movie rating on a scale of 1–5. Rating must be 1,2,3,4,5. They have mathematical meaning. E.x. movie ratings, etc.

Label encoding:

In label encoding, each category is mapped to a number or a label. The labels chosen for the categories have no relationship. So, categories that have some ties or are close to each other lose such information after encoding. It supports the pandas dataframe as input and can transform data.

One-Hot Encoding:

A one hot encoding allows the representation of categorical data to be more expressive. Many Machine Learning algorithms cannot work with categorical data directly. The categories must be converted into numbers.

Label Encoding			One Hot Encoding			
Food Name	Categorical #	Calories	Apple	Chicken	Broccoli	Calories
Apple	1	95	1	0	0	95
Chicken	2	231	0	1	0	231
Broccoli	3	50	0	0	1	50

Operations to be performed on dataset:

Steps in Preprocessing of Data

1. Importing Python Modules/Libraries
2. Importing data
3. Displaying data
4. Creating the Independent and Dependent variables
5. Replacing missing value with meaningful value
6. Encoding categorical data
7. Splitting the data into training and test set
8. Doing feature scaling on data
9. Use any 3-4 graphs/plots

Program code:

Dataset used: (source link & description)

Output:

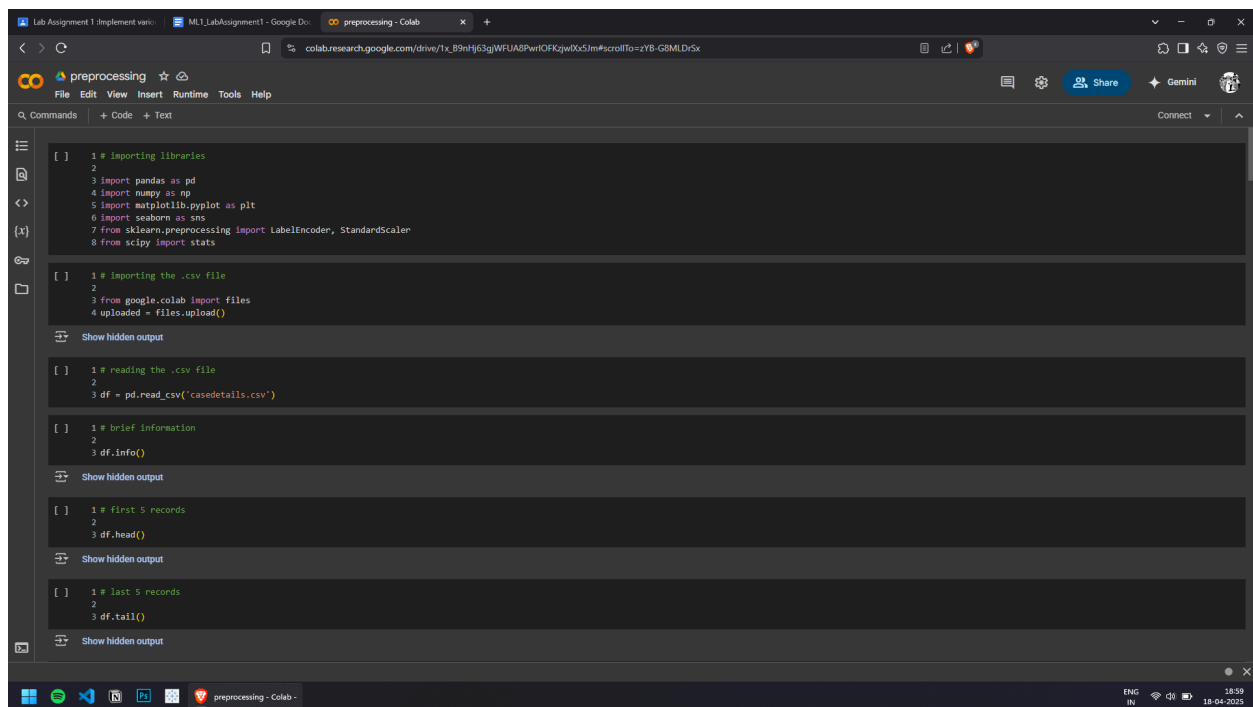
FAQs:

- 1) List two common libraries for data manipulation. Give an example for each library.
- 2) Give an example on how ordinal data is handled in a Machine Learning algorithm.
- 3) Can one hot encoding be used for continuous data. If yes, give an example.
- 4) Why is it necessary to encode strings?
- 5) State the significance of exploratory data analysis.
- 6) 'Handling missing values of data is an important step in Data preprocessing.' Comment on the statement.
- 7) State any 4 graphical techniques/plots used for exploratory data analysis.
- 8) Describe the *box-and-whisker* plot
- 9) Explain Central Tendency functions.

Conclusion:

Data collection, data preparation, handling various data types was studied and exploratory data analysis was performed.

Code & Output:



```
[ ] 1 # importing libraries
2
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import seaborn as sns
7 from sklearn.preprocessing import LabelEncoder, StandardScaler
8 from scipy import stats

[ ] 1 # importing the .csv file
2
3 from google.colab import files
4 uploaded = files.upload()

Show hidden output

[ ] 1 # reading the .csv file
2
3 df = pd.read_csv('casedetails.csv')

[ ] 1 # brief information
2
3 df.info()

Show hidden output

[ ] 1 # first 5 records
2
3 df.head()

Show hidden output

[ ] 1 # last 5 records
2
3 df.tail()

Show hidden output
```

```
preprocessing
File Edit View Insert Runtime Tools Help
Q Commands + Code + Text
[ ] 1 # number of duplicated records
2
3 df.duplicated().sum()

[ ] 1 # number of null records
2
3 nullrecords = df.isnull().sum()
4 nullrecords

[ ] 1 # dropping confirmed_date, recovered_date, symptomatic_date and case_name columns since they are mostly empty and even if not,
2 # they have rubbish values which serve no significance for further processes
3
4 df.drop(['symptomatic_date', 'recovered_date', 'confirmed_date', 'case_name', 'source', 'case_id', 'place_id'], axis = 1, inplace = True)

[ ] 1 df.isnull().sum()

age      464
sex       77
nationality 1297
current_status 0
dtype: int64

[ ] 1 # graphical eda on 'nationality'
2
3 df['nationality'].value_counts().plot(kind='bar', title='nationality')
4 plt.gca().spines[['top', 'right']].set_visible(False)
5 plt.show()

nationality
```

