Student Name:_____Roll No:_____Section:_____

CS334 - Machine Learning

Lab 02

Instructor: Dr. Maham Ashraf
E-mail: mashraf@uit.edu

Semester: Fall, 2023

Objective

The purpose of this lab session is to introduce data preparation techniques for Machine Learning(ML) projects.

Instructions

You have to perform the following tasks yourselves. Raise your hand if you face any difficulty in understanding and solving these tasks. **Plagiarism** is an abhorrent practice and you should not engage in it.

How to Submit

- Submit lab work in a single .py file on Microsoft Teams. (No other format will be accepted)
- Lab work file name should be saved with your roll number (e.g. 19a-001-SE_LW01.py)
- Submit home work in a single .py file on Microsoft Teams. (No other format will be accepted)
- Home work file name should be saved with your roll number (e.g. 19a-001-SE_HW01.py)

1 Summarize and Visualize your data

1.1 Data summary

Descriptive statistics is packed with information and insights. A pair of experienced eyes will take a look at every single data point and extract valuable information from the summary table. Let's first see what a table of summary statistics looks like for a given dataset. We will use a built-in dataset that comes with *seaborn* library in Python.

```
import seaborn as sns
import pandas as pd
df = sns.load_dataset('tips')
df.head()
```

Listing 1: Load data in data frame.

Each observation (row) in this dataset represents dining in a restaurant. The columns names here are self-explanatory. Among the numeric columns, 'total_bill' refers to how much bills the diners paid and 'tip' represents the amount of tip they paid.
Just a simple method call `df.describe()` gives you the summary statistics for the numeric columns (I'll touch upon categorical columns towards the end).

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Figure 1: top 5 records of tip dataset

	count	mean	std	min	25%	50%	75%	max
total_bill	244.0	19.785943	8.902412	3.07	13.3475	17.795	24.1275	50.81
tip	244.0	2.998279	1.383638	1.00	2.0000	2.900	3.5625	10.00
size	244.0	2.569672	0.951100	1.00	2.0000	2.000	3.0000	6.00

Figure 2: Summary of tip dataset

1.2 Visualize your data

you can visualize your data by generating different types of graph. Here's an example of what seaborn can do:

```
\# Import seaborn
import seaborn as sns

\# Apply the default theme
sns.set\_theme()

\# Load an example dataset
tips = sns.load\_dataset("tips")

\# Create a visualization
sns.relplot(
    data=tips,
    x="total\_bill", y="tip", col="time",
    hue="smoker", style="smoker", size="size",
)
```

Listing 2: Visualize the data

Figure 3: Dot plot of tip dataset

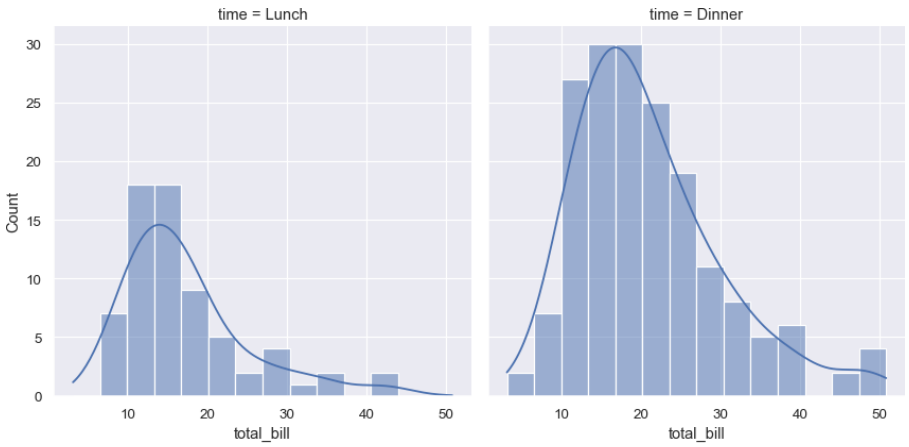

Figure 4: Histogram plot of tip dataset

```
df.describe(include='category').T
```

	count	unique	top	freq
sex	244	2	Male	157
smoker	244	2	No	151
day	244	4	Sat	87
time	244	2	Dinner	176

Figure 5: Describing categorical data of tip dataset

1.3 Informative distributional summaries

Statistical analyses require knowledge about the distribution of variables in your dataset. The seaborn function displot() supports several approaches to visualizing distributions. These include classic techniques like histograms and computationally-intensive approaches like kernel density estimation:

```
sns.displot(data=tips, x="total_bill", col="time", kde=True)
```

1.4 Describing categorical data

So far we have investigated descriptive statistics for numeric variables. Python pandas also offer a summary for categorical variables.

```
df.describe(include='category').T
```

2 Fill Missing Values With Imputation

In this section, you will discover how to identify and fill missing values in data.
Real-world data often has missing values. Data can have missing values for a number of reasons, such as observations that were not recorded and data corruption. Handling missing data is important as many machine learning algorithms do not support data with missing values. Filling missing values with data is called data imputation and a popular approach for data imputation is to calculate a statistical value for each column (such as a mean) and replace all missing values for that column with the statistic.

Dataset: The horse colic dataset describes medical characteristics of horses with colic and whether they lived or died. It has missing values marked with a question mark '?'.

```
# statistical imputation transform for the horse colic dataset
from numpy import isnan
from pandas import read_csv
from sklearn.impute import SimpleImputer
# load dataset
url = 'https://raw.githubusercontent.com/jbrownlee/Datasets/master/horse-colic.csv'
dataframe = read_csv(url, header=None, na_values='?')
# split into input and output elements
data = dataframe.values
ix = [i for i in range(data.shape[1]) if i != 23]
X, y = data[:, ix], data[:, 23]
# print total missing
print('Missing: %d' % sum(isnan(X).flatten()))
# define imputer
imputer = SimpleImputer(strategy='mean')
# fit on the dataset
imputer.fit(X)
# transform the dataset
Xtrans = imputer.transform(X)
# print total missing
print('Missing: %d' % sum(isnan(Xtrans).flatten()))
```

Listing 3: Example of imputing missing values.

Your Tasks

Task 1 Run the example as given in Listing - 1. Print top 10 rows of the data and also print the data summary as given in Figure - 2.

Task 2 Repeat Task 1 on *pima-indians-diabetes.csv* dataset.

Task 3 Generate dot plot and distribution plot on *heart.csv* dataset (https://www.kaggle.com/fedesoriano/heart-failure-prediction?select=heart.csv).

Task 4 Run the example as given in Listing - 3 and review the number of missing values in the dataset before and after the data imputation transform.

Task 5 Repeat Task 4 on *heart.csv* dataset.