Assignment 1

Title : Data preparation:

Aim: Download heart dataset from following link.
https://www.kaggle.com/zhaoyingzhu/heartcsv
Perform following operation on given dataset.

a) Find Shape of Data
b) Find Missing Values
c) Find data type of each column
d) Finding out Zero's
e) Find Mean age of patients

f) Now extract only Age, Sex, ChestPain, RestBP, Chol. Randomly divide dataset in training (75%) and testing (25%).

I] Theory

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data.

Data preparation is often a lengthy undertaking for data professionals or business users, but it is essential as a prerequisite to put data in context in order to turn it into insights and eliminate bias resulting from poor data quality.

Data preparation helps in :

- Fix errors quickly — Data preparation helps catch errors before processing. After data has been removed from its original source, these errors become more difficult to understand and correct.
- Produce top-quality data — Cleaning and reformatting datasets ensures that all data used in analysis will be high quality.
- Make better business decisions — Higher quality data that can be processed and analyzed more quickly and efficiently leads to more timely, efficient and high-quality business decisions.

II]Data Preparation Steps

1. Gather data

The data preparation process begins with finding the right data. This can come from an existing data catalog or can be added ad-hoc.

2. Discover and assess data

After collecting the data, it is important to discover each dataset. This step is about getting to know the data and understanding what has to be done before the data becomes useful in a particular context.

3. Cleanse and validate data

Cleaning up the data is traditionally the most time consuming part of the data preparation process, but it's crucial for removing faulty data and filling in gaps. Important tasks here include:

- Removing extraneous data and outliers.
- Filling in missing values.
- Conforming data to a standardized pattern.
- Masking private or sensitive data entries.

Once data has been cleansed, it must be validated by testing for errors in the data preparation process up to this point. Often times, an error in the system will become apparent during this step and will need to be resolved before moving forward.

4. Transform and enrich data

Transforming data is the process of updating the format or value entries in order to reach a well-defined outcome, or to make the data more easily understood by a wider audience. Enriching data refers to adding and connecting data with other related information to provide deeper insights.

5. Store data

Once prepared, the data can be stored or channelled into a third party application—such as a business intelligence tool—clearing the way for processing and analysis to take place.

III] Python Libraries and functions required

numpy : NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy stands for Numerical Python. To import numpy use

```
import numpy as np
```

pandas: pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language. To import pandas use

```
import pandas as pd
```

sklearn : Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. For importing train_test_ split use

```
from sklearn.model_selection import train_test_split
```

IV] Sample Code with comments

```
'''
Created on 21-Aug-2021

@author: Dr.Namita Kale
'''

import pandas as pd
from sklearn.model_selection import train_test_split

# load CSV file to use as a dataframe
filename = 'C:/Users/Dell 1/Desktop/Heart.csv'
data = pd.read_csv(filename)

# Print number of instances (rows)
print('\nNo. of Instances\t:', len(data))

# Print number of dimensions (features/columns)
print('\nNo. of Dimensions\t:', len(data.columns))

# Print dimensions of the dataset,i.e., rows * columns
print('\nShape of dataset\t:', data.shape)

# Print first 5 rows of the dataset

print(data.head())

# Print concise summary of the dataset
data.info()

# Print dimensions of the dataset,i.e., rows * columns
print(data.shape)

# Generate descriptive statistics of dataset Descriptive statistics  include those that summarize
#the central  tendency, dispersion and  shape of a dataset’s distribution,
#excluding NaN values.

summary=data.describe()

print(summary)

# Print the missing values, i.e., NaN values

print("missing values",data.isna().sum())
miss1=data[data.isnull().any(axis=1)]


# Print the missing values, i.e., NaN values
# notnull() returns a dataframe with Boolean values stating True if the value
```

```
# is not null and False if the value is null.
print('\nMissing Values (entire dataframe):')
print(data.notnull())
print('\nMissing Values (single column):')
print(pd.notnull(data['Ca']))

# Print data type of values contained in each column of dataframe
print('\nData Types of Columns:')
print(data.dtypes)

# Print number of zeroes in the columns
# Similarly for all columns
print('\nNumber of zeroes in a column:', (data['Sex'] == 0).sum())
print('\nNumber of zeroes in a column age:', (data['Age'] == 0).sum())
print('\nNumber of zeroes in a column RestBP:', (data['RestBP'] == 0).sum())
print('\nNumber of zeroes in a column Fbs:', (data['Fbs'] == 0).sum())
print('\nNumber of zeroes in a column RestECG:', (data['RestECG'] == 0).sum())
print('\nNumber of zeroes in a column MaxHR:', (data['MaxHR'] == 0).sum())
print('\nNumber of zeroes in a column ExAng:', (data['ExAng'] == 0).sum())
print('\nNumber of zeroes in a column Ca:', (data['Ca'] == 0).sum())
print('\nNumber of zeroes in a column Slope:', (data['Slope'] == 0).sum())
print('\nNumber of zeroes in a column Oldpeak:', (data['Oldpeak'] == 0).sum())


# Print mean of all values in age column
print('\nMean Age:', data['Age'].mean(axis=0, skipna=True))
print('Mean Age rounded to 2 decimal places:', round(data['Age'].mean(axis=0,
skipna=True)))

# Extract only 'Age', 'Sex', 'ChestPain', 'RestBP', 'Chol' columns from dataset
y = data[['Age', 'Sex', 'ChestPain', 'RestBP', 'Chol']]

# Split extracted data into training data and testing data
splits_train, splits_test = train_test_split(y, test_size = 0.25, random_state=20)

# Train Dataset
print('\nTraining Dataset (75%): ')
print('Length of Training Dataset:', len(splits_train))
print(splits_train)

# Test Dataset
print('\nTesting Dataset (75%): ')
print('Length of Testing Dataset:', len(splits_test))
print(splits_test)
```

V] Output of Code:

Note: Run the code and attach your output of the code here.