SIT719 Security and Privacy Issues in Analytics

Pass Task 2.1: Basic scripting with python

SECTION 1

OUTPUTS:

```
In [31]: iris.head()
Out[31]:
   sepal_length  sepal_width  petal_length  petal_width        class
0           4.9          3.0           1.4          0.2  Iris-setosa
1           4.7          3.2           1.3          0.2  Iris-setosa
2           4.6          3.1           1.5          0.2  Iris-setosa
3           5.0          3.6           1.4          0.2  Iris-setosa
4           5.4          3.9           1.7          0.4  Iris-setosa

In [32]: iris.isnull()
Out[32]:
     sepal_length  sepal_width  petal_length  petal_width  class
0           False        False         False        False  False
1           False        False         False        False  False
2           False        False         False        False  False
3           False        False         False        False  False
4           False        False         False        False  False
..            ...          ...           ...          ...    ...
144         False        False         False        False  False
145         False        False         False        False  False
146         False        False         False        False  False
147         False        False         False        False  False
148         False        False         False        False  False

[149 rows x 5 columns]

In [33]: iris.iloc[10:29, 2] = np.nan
```

```
In [34]: iris.iloc[10:29, 2]
Out[34]:
10   NaN
11   NaN
12   NaN
13   NaN
14   NaN
15   NaN
16   NaN
17   NaN
18   NaN
19   NaN
20   NaN
21   NaN
22   NaN
23   NaN
24   NaN
25   NaN
26   NaN
27   NaN
28   NaN
Name: petal_length, dtype: float64

In [35]: iris.isnull().sum()
Out[35]:
sepal_length     0
sepal_width      0
petal_length    19
petal_width      0
class            0
dtype: int64
```

```
In [36]: iris = iris.replace(np.nan,10.0)

In [37]: iris.iloc[10:29, 2]
Out[37]:
10    10.0
11    10.0
12    10.0
13    10.0
14    10.0
15    10.0
16    10.0
17    10.0
18    10.0
19    10.0
20    10.0
21    10.0
22    10.0
23    10.0
24    10.0
25    10.0
26    10.0
27    10.0
28    10.0
Name: petal_length, dtype: float64

In [38]: iris.isnull().sum()
Out[38]:
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
class           0
```

CODE :

```
@author: sagar
"""
import pandas as pd

import numpy as np
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
iris = pd.read_csv(url)
iris.head()

#creation of column heads for the dataset
iris.columns = [ "sepal_length" , "sepal_width" , "petal_length" ,"petal_width", "class" ]
iris.head()

#to check if there is missing values
iris.isnull()

#set the values of the rows 10 to 29 of the column 'petal_length' to NaN
iris.iloc[10:29, 2] = np.nan
iris.iloc[10:29, 2]

#, check (NaN)? Count,  missing values.
iris.isnull().sum()

#Substitute the NaN values to 10.0
iris = iris.replace(np.nan,10.0)
iris.iloc[10:29, 2]

#finally to check for NaN
iris.isnull().sum()
```

SECTION 2

->Importance of python libraries for data analysis

Python libraries like Pandas, NumPy, Matplotlib, Scikit Learn, Tensorflow and more are being used widely by data scientists and many others for analysis, data manipulation, wrangling, array processing, visualization of data and much more. The popularity of usage comes from the efficiency, speed and the ease these libraries provide to users

1. Pandas

Pandas provide high-performance, easy-to-use data structures and data analysis tools for the labelled data in Python programming.

2. NumPy

NumPy is a general-purpose array-processing package. It provides high-performance multidimensional array objects and tools to work with the arrays.

4. Matplotlib

The quintessential Python library used to create stories with the data visualized with Matplotlib.

->Functionalities and usages related to data frame manipulation

1.**NaN check**

The occurrence of missing data is frequent in the data analysis, to discover and manipulate these missing values pandas provide few built-in functions like isnull() which return bool value if the data is missing or not.

2.**concat()**

This inbuilt function is used to combining together with various logic for join/merge type operation.

->Sample visualization example using matplotlib library

```python
#Simple matplotlib visualization[subplots]
import matplotlib.pyplot as plt

plt.figure(figsize=(6, 4))
plt.subplot(1, 2, 1)
plt.xticks(())
plt.yticks(())
plt.text(0.5, 0.5, 'subplot(A)', ha='center', va='center',
        size=24, alpha=.5)

plt.subplot(1, 2, 2)
plt.xticks(())
plt.yticks(())
plt.text(0.5, 0.5, 'subplot(B)', ha='center', va='center',
        size=24, alpha=.5)

plt.tight_layout()
plt.show()
```