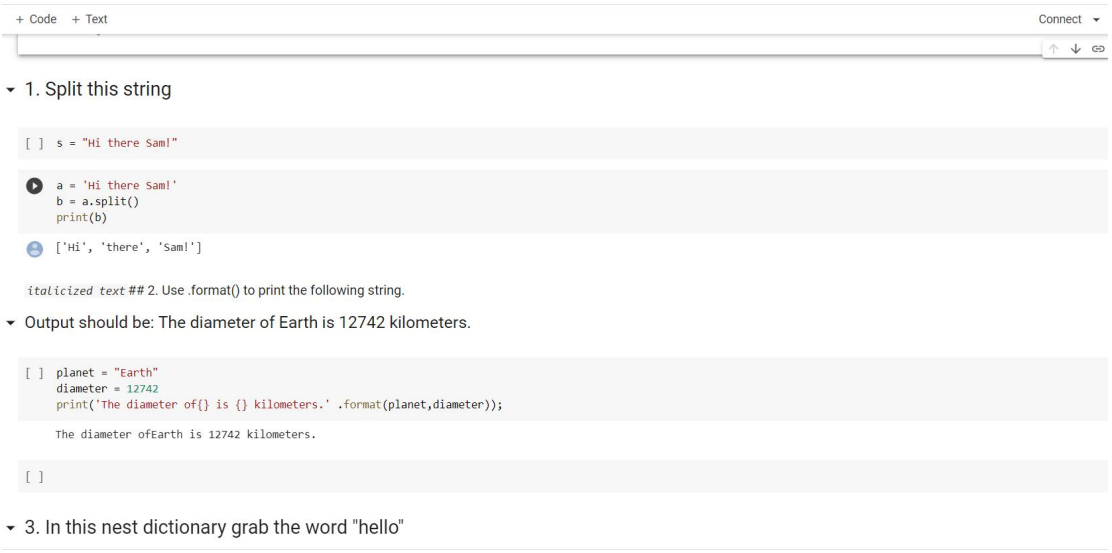
Assignment 1
Statistical Machine Learning Approaches
To Liver Disease Prediction

Student Name	Kamali R
Student Roll no	621319104019
Maximum Marks	2 Marks

Basics of Python:

1. Split this string
2. Output should be:The diameter of Earth is 12742 kilometers
3. In this nest dictionary grab the word “hello”



4. Numpy

4.1 Create an array of 10 zeros?

4.2 Create an array of 10 fives?

```
[ ] d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

    d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
    print(d['k1'][3]["tricky"][3]['target'][3])

    hello
```

+ Code + Text

Numpy

```
[ ] import numpy as np
```

4.1 Create an array of 10 zeros?

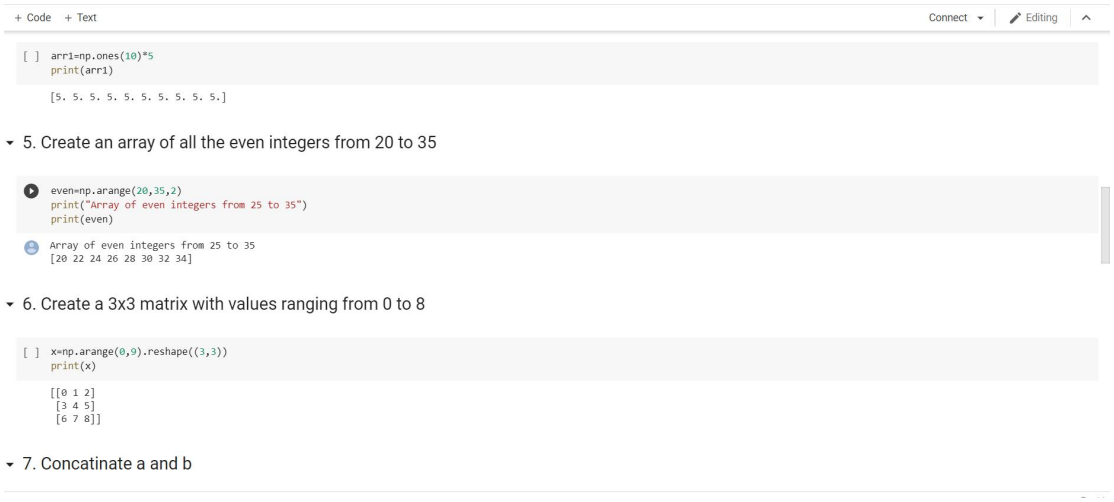
4.2 Create an array of 10 fives?

```
[ ] arr=np.zeros(10)
    print(arr)

    [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

[ ] arr1=np.ones(10)*5
```

5. Create an array of all the even integers from 20 to 35
6. Create a 3x3 matrix with values ranging from 0 to 8



```
[ ] arr1=np.ones(10)*5
    print(arr1)

    [5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]
```

5. Create an array of all the even integers from 20 to 35

```
    even=np.arange(20,35,2)
    print("Array of even integers from 25 to 35")
    print(even)

    Array of even integers from 25 to 35
    [20 22 24 26 28 30 32 34]
```

6. Create a 3x3 matrix with values ranging from 0 to 8

```
[ ] x=np.arange(0,9).reshape((3,3))
    print(x)

    [[0 1 2]
     [3 4 5]
     [6 7 8]]
```

7. Concatinate a and b

7. Concatinate a and b
a=np.array([1,2,3]),b=np.array([4,5,6])

Pandas

8.Create a dataframe with 3 rows and 2 columns

+ Code + Text

7. Concatinate a and b

a = np.array([1, 2, 3]), b = np.array([4, 5, 6])

```
[ ] a=np.array([[1,2,3]])
    b=np.array([[4,5,6]])
    np.concatenate((a,b),axis=0)
```

```
array([[1, 2, 3],
       [4, 5, 6]])
```

Pandas

8. Create a dataframe with 3 rows and 2 columns

```
[ ] import pandas as pd
```

```
[ ] df=pd.DataFrame()
    print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

9.Generate the series of dates from 1st Jan,2023 to 10th Feb,2023

+ Code + Text

```
Index: []
```

9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023

```
import datetime

#initializing date
test_date=datetime.datetime.strptime("01-01-2023","%d-%m-%Y")

#initializing K
K=datetime.datetime.strptime("10-2-2023","%d-%m-%Y")

date_generated=pd.date_range(test_date,K)
print(date_generated.strftime("%d-%m-%Y"))
```

```
Index(['01-01-2023', '02-01-2023', '03-01-2023', '04-01-2023', '05-01-2023',
       '06-01-2023', '07-01-2023', '08-01-2023', '09-01-2023', '10-01-2023',
       '11-01-2023', '12-01-2023', '13-01-2023', '14-01-2023', '15-01-2023',
       '16-01-2023', '17-01-2023', '18-01-2023', '19-01-2023', '20-01-2023',
       '21-01-2023', '22-01-2023', '23-01-2023', '24-01-2023', '25-01-2023',
       '26-01-2023', '27-01-2023', '28-01-2023', '29-01-2023', '30-01-2023',
       '31-01-2023', '01-02-2023', '02-02-2023', '03-02-2023', '04-02-2023',
       '05-02-2023', '06-02-2023', '07-02-2023', '08-02-2023', '09-02-2023',
       '10-02-2023'],
      dtype='object')
```

10. Create 2D list to DataFrame

```
lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
```

10.Create 2D list to DataFrame

dtype= object)

10. Create 2D list to DataFrame

lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]

+ Code + Text

```
[ ] lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]
    r= pd.DataFrame(lists,columns=['s.no','letter','number'])
    print(r)
```

```
   s.no letter  number
0     1    aaa      22
1     2    bbb      25
2     3    ccc      24
```

```
[ ]
```