

School of Computer Science Engineering and Technology

Course- BTech

Course Code- CSET211

Year- Second

Date- 09/08/2022

Type- AI Core-1

Course Name- Statistical Machine Learning

Semester- ODD

Batch- CSE 3rd Semester

Lab Assignment (10th Aug to 12th Aug 2022)

Lab 1 - NumPy

Q1: Import numpy as np and see the version

Sol:

```
import numpy as np

print(np.__version__)
```

Q2: Write program for Creating a 1D array

Sol:

```
arr = np.arange(10)

arr

#> array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

Q3: Write program for Creating a boolean array?

Sol:

```
np.full((3, 3), True, dtype=bool)

#> array([[ True,  True,  True],

#>        [ True,  True,  True],

#>        [ True,  True,  True]], dtype=bool)

# Alternate method:
```

```
np.ones((3,3), dtype=bool)
```

Q4: Write program to extract items that satisfy a given condition from 1D array?

Sol:

```
# Input
```

```
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
# Solution
```

```
arr[arr % 2 == 1]
```

```
#> array([1, 3, 5, 7, 9])
```

Q5: Write program for Replace items that satisfy a condition with another value in numpy array?

Sol:

```
arr[arr % 2 == 1] = -1
```

```
arr
```

```
#> array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

Q6: Write program for reshape an array?

Convert a 1D array to a 2D array with 2 rows

Sol:

```
arr = np.arange(10)
```

```
arr.reshape(2, -1) # Setting to -1 automatically decides the number of cols
```

```
#> array([[0, 1, 2, 3, 4],
```

```
#>      [5, 6, 7, 8, 9]])
```

Q7: Write program for stacking two arrays vertically?

Sol:

```
a = np.arange(10).reshape(2,-1)
b = np.repeat(1, 10).reshape(2,-1)
```

Answers

Method 1:

```
np.concatenate([a, b], axis=0)
```

Method 2:

```
np.vstack([a, b])
```

Q9: How to remove from one array those items that exist in another?

Q. From array a remove all items present in array b

Sol:

```
a = np.array([1,2,3,4,5])
b = np.array([5,6,7,8,9])

# From 'a' remove all of 'b'

np.setdiff1d(a,b)

#> array([1, 2, 3, 4])
```

Q10: Write program to get the positions where elements of two arrays match?

Q. Get the positions where elements of a and b match

Sol:

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

```
b = np.array([7,2,10,2,7,4,9,4,9,8])
```

```
np.where(a == b)
```

```
#> (array([1, 3, 5, 7]),)
```

Q11: Write program to import a dataset with numbers and texts keeping the text intact in python numpy?

Q. Import the [iris dataset](#) keeping the text intact.

Sol:

```
# Solution
```

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
```

```
iris = np.genfromtxt(url, delimiter=',', dtype='object')
```

```
names = ('sepalength', 'sepalwidth', 'petallength', 'petalwidth', 'species')
```

```
# Print the first 3 rows
```

```
iris[:3]
```

```
#> array([[b'5.1', b'3.5', b'1.4', b'0.2', b'Iris-setosa'],
```

```
#>          [b'4.9', b'3.0', b'1.4', b'0.2', b'Iris-setosa'],
```

```
#>          [b'4.7', b'3.2', b'1.3', b'0.2', b'Iris-setosa']], dtype=object)
```

Since we want to retain the species, a text field, I have set the `dtype` to `object`. Had I set `dtype=None`, a 1d array of tuples would have been returned.

Q12: How to normalize an array so the values range exactly between 0 and 1?

Q. Create a normalized form of `iris's sepalength` whose values range exactly between 0 and 1 so that the minimum has value 0 and maximum has value 1.

Sol:

```
# Input
```

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
```

```
sepalength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])
```

```
# Solution
```

```
Smax, Smin = sepalength.max(), sepalength.min()
```

```
S = (sepalength - Smin)/(Smax - Smin)
```

```
# or
```

```
S = (sepalength - Smin)/sepalength.ptp() # Thanks, David Ojeda!
```

```
print(S)
```

```
#> [ 0.222  0.167  0.111  0.083  0.194  0.306  0.083  0.194  0.028  0.167
```

```
#>  0.306  0.139  0.139  0.      0.417  0.389  0.306  0.222  0.389  0.222
```

```
#>  0.306  0.222  0.083  0.222  0.139  0.194  0.194  0.25  0.25  0.111
```

```
#>  0.139  0.306  0.25  0.333  0.167  0.194  0.333  0.167  0.028  0.222
```

```
#>  0.194  0.056  0.028  0.194  0.222  0.139  0.222  0.083  0.278  0.194
```

```
#>  0.75  0.583  0.722  0.333  0.611  0.389  0.556  0.167  0.639  0.25
```

```
#>  0.194  0.444  0.472  0.5  0.361  0.667  0.361  0.417  0.528  0.361
```

```
#>  0.444  0.5  0.556  0.5  0.583  0.639  0.694  0.667  0.472  0.389
```

```
#>  0.333  0.333  0.417  0.472  0.306  0.472  0.667  0.556  0.361  0.333
```

```
#>  0.333  0.5  0.417  0.194  0.361  0.389  0.389  0.528  0.222  0.389
```

```
#>  0.556  0.417  0.778  0.556  0.611  0.917  0.167  0.833  0.667  0.806
```

```
#> 0.611 0.583 0.694 0.389 0.417 0.583 0.611 0.944 0.944 0.472
#> 0.722 0.361 0.944 0.556 0.667 0.806 0.528 0.5 0.583 0.806
#> 0.861 1. 0.583 0.556 0.5 0.944 0.556 0.583 0.472 0.722
#> 0.667 0.722 0.417 0.694 0.667 0.667 0.556 0.611 0.528 0.444]
```