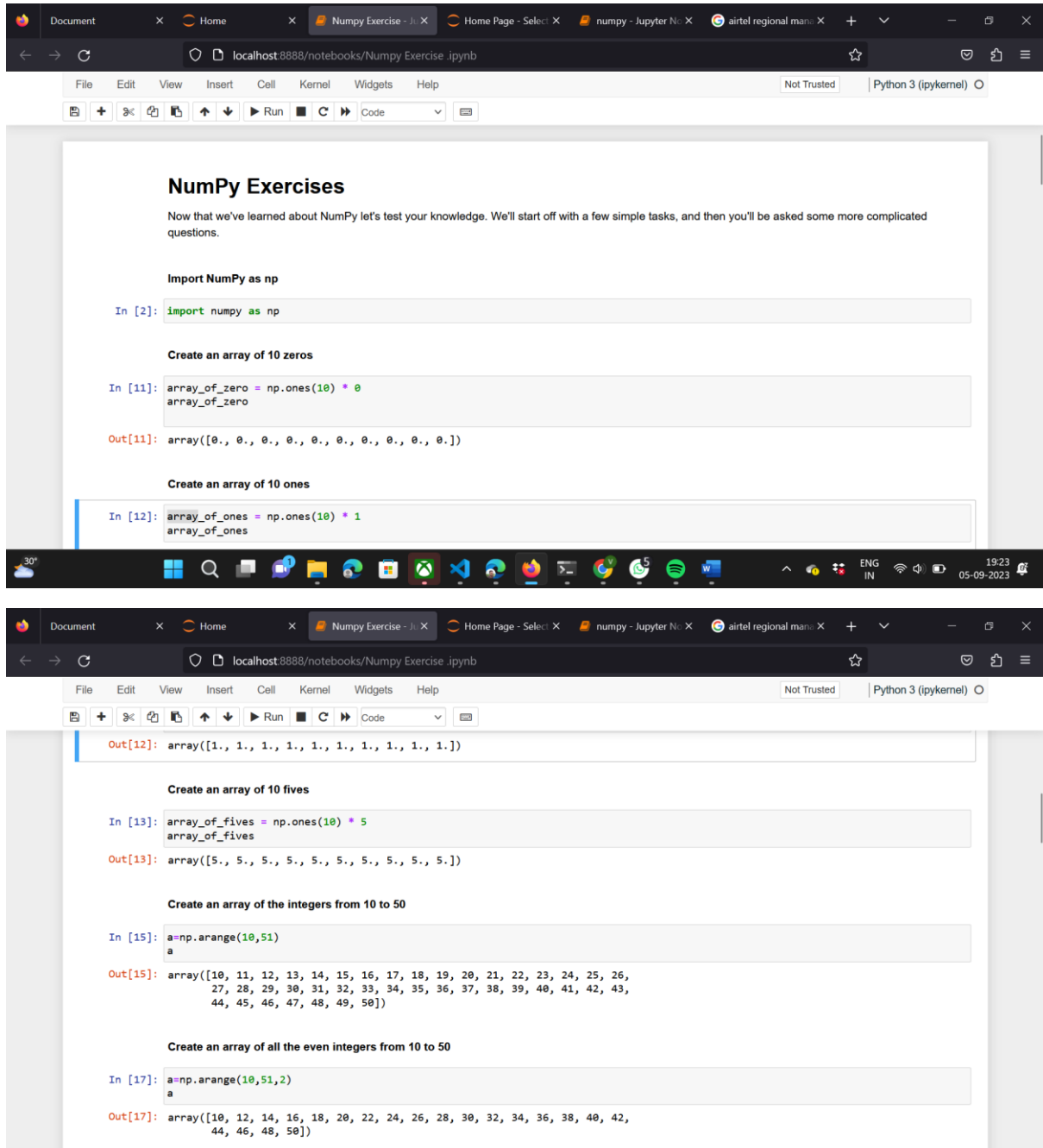


Assignment-1 (Machine learning and artificial intelligence)

Name- Vivek Kumar Tiwari

Email- vivekkumar.tiwari2021@vitstudent.ac.in



The screenshot displays a Jupyter Notebook interface with the following content:

NumPy Exercises

Now that we've learned about NumPy let's test your knowledge. We'll start off with a few simple tasks, and then you'll be asked some more complicated questions.

Import NumPy as np

```
In [2]: import numpy as np
```

Create an array of 10 zeros

```
In [11]: array_of_zero = np.ones(10) * 0
array_of_zero
```

```
Out[11]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

Create an array of 10 ones

```
In [12]: array_of_ones = np.ones(10) * 1
array_of_ones
```

```
Out[12]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

Create an array of 10 fives

```
In [13]: array_of_fives = np.ones(10) * 5
array_of_fives
```

```
Out[13]: array([5., 5., 5., 5., 5., 5., 5., 5., 5., 5.])
```

Create an array of the integers from 10 to 50

```
In [15]: a=np.arange(10,51)
a
```

```
Out[15]: array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50])
```

Create an array of all the even integers from 10 to 50

```
In [17]: a=np.arange(10,51,2)
a
```

```
Out[17]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
44, 46, 48, 50])
```

Document x Home x Numpy Exercise - Jupyter x Home Page - Select x numpy - Jupyter Notebook x airtel regional man: x + - x

localhost8888/notebooks/Numpy Exercise.ipynb

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

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

```
In [19]: matrix = np.arange(9).reshape(3, 3)
print(matrix)
```

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

Create a 3x3 identity matrix

```
In [46]: identity_mat = np.identity(3)
identity_mat
```

```
Out[46]: array([[1., 0., 0.],
 [0., 1., 0.],
 [0., 0., 1.]])
```

Use NumPy to generate a random number between 0 and 1

```
In [22]: d=np.random.rand(0,1)
d
```

```
Out[22]: array([], shape=(0, 2), dtype=float64)
```

Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

Document x Home x Numpy Exercise - Jupyter x Home Page - Select x numpy - Jupyter Notebook x airtel regional man: x + - x

localhost8888/notebooks/Numpy Exercise.ipynb

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Use NumPy to generate an array of 25 random numbers sampled from a standard normal distribution

```
In [36]: mat = np.random.normal(0, 1, 25)
mat
```

```
Out[36]: array([-1.82994292, -0.08219666,  0.00566356,  0.50197875,  0.82587505,
 -1.14308212,  1.56567132, -0.0158226 , -0.00460834, -0.08368191,
  0.47621118,  1.28558434, -0.55930242, -0.89927104, -0.92492831,
 -0.58173172, -0.37409545, -0.8391357 , -1.02391431, -0.35595839,
  0.47123637,  0.41956119, -0.4957687 ,  0.19373148,  1.14976407])
```

Create the following matrix:

```
In [45]: mat = np.arange(0.01, 1.01, 0.01)
# Reshape the array into a 10x10 matrix
mat = mat.reshape(10, 10)
mat
```

```
Out[45]: array([[0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 ],
 [0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 ],
 [0.21, 0.22, 0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 ],
 [0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 ],
 [0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47, 0.48, 0.49, 0.5 ],
 [0.51, 0.52, 0.53, 0.54, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6 ],
 [0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67, 0.68, 0.69, 0.7 ],
 [0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77, 0.78, 0.79, 0.8 ],
 [0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88, 0.89, 0.9 ],
 [0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1. ]])
```

```
Document x Home x Numpy Exercise - Jupyter x Home Page - Select x numpy - Jupyter N x airtel regional man x + - x x
localhost8888/notebooks/Numpy Exercise.ipynb
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)
[0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99, 1. ]])

Create an array of 20 linearly spaced points between 0 and 1:

In [47]: linear_array = np.linspace(0, 1, 20)
linear_array
Out[47]: array([0.          , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.        ]])

Numpy Indexing and Selection

Now you will be given a few matrices, and be asked to replicate the resulting matrix outputs:

In [39]: mat = np.arange(1,26).reshape(5,5)
mat
Out[39]: array([[ 1,  2,  3,  4,  5],
[ 6,  7,  8,  9, 10],
[11, 12, 13, 14, 15],
[16, 17, 18, 19, 20],
[21, 22, 23, 24, 25]])

In [0]: #first we search for where the 12 value lies first we select the col 2 to n bcz we want till last value after this where is 15
#Lies then we select the row from 1 to m where n==sizeof(col) and m==sizeof(row)
```

```
Document x Home x Numpy Exercise - Jupyter x Home Page - Select x numpy - Jupyter N x airtel regional man x + - x x
localhost8888/notebooks/Numpy Exercise.ipynb
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

In [27]: d=mat[2:,1:]
d
Out[27]: array([[12, 13, 14, 15],
[17, 18, 19, 20],
[22, 23, 24, 25]])

In [0]: #first we select 3 columns which cover till n-1 in which 20 lies (n==sizeof(col) && m==sizeof(row))
# after this we select 4 row in which 20 lies
#mat[3][4]

In [30]: mat[3][4]
Out[30]: 20

In [0]: #same here above we find the range of 2 which lies 0 and last 12 lies 2 but we take till n+1
# and row range here are 1 to 2

In [31]: mat[0:3,1:2]
Out[31]: array([[ 2],
[ 7],
[12]])

In [0]: # here we need only print last row so we take col value is 4 to m ans row values is 0 to 5 where n==sizeof(col) and m==sizeof(row)

In [32]: mat[4:,0:5]
Out[32]: array([[21, 22, 23, 24, 25]])
```

Document x Home x Numpy Exercise - Jupyter x Home Page - Select x numpy - Jupyter N x airtel regional man x + - x

localhost8888/notebooks/Numpy Exercise.ipynb ☆

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [35]: mat[3:,0:5]
Out[35]: array([[16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25]])
```

Now do the following ¶

Get the sum of all the values in mat

```
In [40]: sum_of_mat = np.sum(mat)
sum_of_mat
Out[40]: 325
```

Get the standard deviation of the values in mat

```
In [42]: std_of_mat = np.std(mat)
std_of_mat
Out[42]: 7.211102550927978
```

Get the sum of all the columns in mat

localhost8888/notebooks/Numpy Exercise.ipynb#Now-do-the-following

Document x Home x Numpy Exercise - Jupyter x Home Page - Select x numpy - Jupyter N x airtel regional man x + - x

localhost8888/notebooks/Numpy Exercise.ipynb ☆

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
sum_of_mat
Out[40]: 325
```

Get the standard deviation of the values in mat

```
In [42]: std_of_mat = np.std(mat)
std_of_mat
Out[42]: 7.211102550927978
```

Get the sum of all the columns in mat

```
In [43]: column_sums = np.sum(mat, axis=0)
column_sums
Out[43]: array([55, 60, 65, 70, 75])
```

Type Markdown and LaTeX: α^2