Matrix Operations

Matrices

```
# Create a matrix using the 'matrix()' function of the 'numpy' module.
import numpy as np
x=np.matrix('1 2 3;4 5 6;7 8 9')
x
```

```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

```
# Convert a nested Python list into a matrix using the 'matrix()' function of the 'nump
import numpy as np
x=np.matrix([[1,2,3],[4,5,6],[7,8,9]])
x
```

```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

Matrix Multiplication

$$\begin{bmatrix} 8 & 7 \\ 3 & -5 \end{bmatrix} \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

To multiply two matrices, you have to use the `matmul()` function of the `numpy` module. It takes two matrices to be multiplied as inputs and returns a new matrix.

```
# Multiplication of the above two matrices.
import numpy as np
x=np.matrix('8 7;3 -5')
y=np.matrix('3;2')
r=np.matmul(x,y)
r
```

```
matrix([[38],
        [-1]])
```

Note: The number of columns in the first matrix should be the same as the number of rows in the second matrix. Otherwise, you will get `ValueError`

Identity Matrix

```
# Create an identity matrix of the dimension 5 and of 'int' type.
import numpy as np
x=np.identity(5,dtype=int)
x

```

```
array([[1, 0, 0, 0, 0],
       [0, 1, 0, 0, 0],
       [0, 0, 1, 0, 0],
       [0, 0, 0, 1, 0],
       [0, 0, 0, 0, 1]])
```

Now let's multiply a square matrix with an identity matrix of the same dimension (say 3).

```
# Multiply a square matrix with an identity matrix of the same dimension.
import numpy as np
x=np.matrix('1 2 3;4 5 6;7 8 9')
i=np.identity(3,dtype=int)
r=np.matmul(x,i)
r


```

```
matrix([[1, 2, 3],
        [4, 5, 6],
        [7, 8, 9]])
```

Multiplicative Inverse of a Matrix^

Eg. let A be a matrix of dimension 3

$$A = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix}$$

Then its inverse is

$$A^{-1} = \begin{bmatrix} 7 & -3 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Both A and A^{-1} have the same dimension.

If you multiply the matrix A with its inverse, you should get the identity matrix of dimension 3.

$$A^{-1}A = \begin{bmatrix} 7 & -3 & -3 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix}$$

$$\Rightarrow A^{-1}A = \begin{bmatrix} 7\times1-3\times1-3\times1 & 7\times3-3\times4-3\times3 & 7\times3-3\times3-3 \\ -1\times1+1\times1+0\times1 & -1\times3+1\times4+0\times3 & -1\times3+1\times3+0 \\ -1\times1+0\times1+1\times1 & -1\times3+0\times4+1\times3 & -1\times3+0\times3+1 \end{bmatrix}$$

$$\Rightarrow A^{-1}A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
#Multiply the above matrix with its inverse.
import numpy as np
x=np.matrix('1 3 3;1 4 3;1 3 4')
y=np.matrix('7 -3 -3;-1 1 0;-1 0 1')
r=np.matmul(x,y)
r

```

```
matrix([[1, 0, 0],
        [0, 1, 0],
        [0, 0, 1]])
```

Determinant of a Matrix^^

$$X = \begin{bmatrix} 8 & 7 \\ 3 & -5 \end{bmatrix}$$

To calculate the determinant value of a square matrix, use the `linalg.det()` function of the `numpy` module.

Syntax: `np.linalg.det(a)`

where `a` is some square matrix.

```
# Calculate the determinant of the matrix X.
import numpy as np
x=np.matrix('8 7;3 -5')
np.linalg.det(x)
```

```
-60.99999999999995
```

Let's calculate the determinant of the matrix

$$A = \begin{bmatrix} 1 & 3 & 3 \\ 1 & 4 & 3 \\ 1 & 3 & 4 \end{bmatrix}$$

using Python.

```
#Calculate the determinant of the above matrix A.
x=np.matrix('1 3 3;1 4 3;1 3 4')
np.linalg.det(x)

```

```
1.0
```

```
# Calculate the inverse of the above matrix A.
x=np.matrix('1 3 3;1 4 3;1 3 4')
np.linalg.inv(x)

```

```
matrix([[ 7., -3., -3.],
        [-1.,  1.,  0.],
        [-1.,  0.,  1.]])
```

```
#Create a dictionary to store weatherdata and convert it to a dtaframe
import pandas as pd
weather={ 'day' : ['19/04/21','20/04/21','21/04/21','22/04/21','23/04/21'],
          'city' :['Ernakulam','Kollam','Kannur','Kottayam','Alappuzha'],
          'temp' : [30,40,45,50,55]
         }

# Display the first five rows of dataframe
df=pd.DataFrame(weather)
df.head()


#Display the last five rows of dataframe
df.tail()


```

	day	city	temp
0	19/04/21	Ernakulam	30
1	20/04/21	Kollam	40
2	21/04/21	Kannur	45
3	22/04/21	Kottayam	50
4	23/04/21	Alappuzha	55