MACHINE LEARNING ASSIGNMENT 3

NAME : ASHOK SAI SUDIREDDY
ID : 700734963
Video Link :
https://drive.google.com/file/d/1wwdKh_25JdsVUqCouALGVMLHPKefvaHO/view?usp=share_link
GitHub Link :
https://github.com/AshokSai1999/Machine-Learning.git

Q1) Numpy:

a. Using NumPy create random vector of size 15 having only Integers in the range 1-20.

1. Reshape the array to 3 by 5

2. Print array shape.

3. Replace the max in each row by 0

```
import numpy as np

# Creating random vector of size 15 with integers in range 1-20

arr1 = np.random.randint(1, 21, size=15)

# Reshape array to 3 by 5

arr1 = arr1.reshape(3, 5)

# Print array shape

print(arr1.shape)

# Replace the max in each row by 0

arr1[np.arange(3), np.argmax(arr1, axis=1)] = 0

print(arr1)
```

In [5]:
```
import numpy as np

# Creating random vector of size 15 with integers in range 1-20
arr1 = np.random.randint(1, 21, size=15)

# Reshape array to 3 by 5
arr1 = arr1.reshape(3, 5)

# Print array shape
print(arr1.shape)

# Replace the max in each row by 0
arr1[np.arange(3), np.argmax(arr1, axis=1)] = 0

print(arr1)
```

```
(3, 5)
[[ 0 15 12  5 11]
 [ 0 18  9 15 19]
 [ 2 13  7  0 14]]
```

Create a 2-dimensional array of size 4 x 3 (composed of 4-byte integer elements), also print the shape, type and data type of the array.

import numpy as np

 # Creating 2D array of size 4x3 with 4-byte integer elements

arr1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], dtype=np.int32)

 # Printing array shape, type, and data type

print("Array shape:", arr1.shape)

print("Array type:", type(arr1))

print("Array data type:", arr1.dtype)

In [6]:
```
import numpy as np

# Creating 2D array of size 4x3 with 4-byte integer elements
arr1 = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]], dtype=np.int32)

# Printing array shape, type, and data type
print("Array shape:", arr1.shape)
print("Array type:", type(arr1))
print("Array data type:", arr1.dtype)
```

```
Array shape: (4, 3)
Array type: <class 'numpy.ndarray'>
Array data type: int32
```

b. Write a program to compute the eigenvalues and right eigenvectors of a given square array given below:

[[3 -2]

[1 0]]

 # Eigenvalues and right eigenvectors of the given square array

arr2 = np.array([[3, -2], [1, 0]])

eigenval, eigenvect = np.linalg.eig(arr2)

print("Eigenvalues:", eigenval)

print("Right eigenvectors:")

print(eigenvect)

```
In [8]: # Eigenvalues and right eigenvectors of the given square array
        arr2 = np.array([[3, -2], [1, 0]])
        eigenval, eigenvect = np.linalg.eig(arr2)
        print("Eigenvalues:", eigenval)
        print("Right eigenvectors:")
        print(eigenvect)
```

```
Eigenvalues: [2. 1.]
Right eigenvectors:
[[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

c. Compute the sum of the diagonal element of a given array.

[[0 1 2]

[3 4 5]]

 # The sum of the diagonal elements of the given array

arr3 = np.array([[0, 1, 2], [3, 4, 5]])

diag_sum = np.trace(arr3)

print("Diagonal sum:", diag_sum)

```
In [4]: # The sum of the diagonal elements of the given array
        arr3 = np.array([[0, 1, 2], [3, 4, 5]])
        diag_sum = np.trace(arr3)
        print("Diagonal sum:", diag_sum)
```

```
Diagonal sum: 4
```

d. Write a NumPy program to create a new shape to an array without changing its data.

Reshape 3x2:
[[1 2]
[3 4]
[5 6]]
Reshape 2x3:
[[1 2 3]
[4 5 6]]

 # Reshape 3x2:

arr4 = np.array([[1, 2], [3, 4], [5, 6]])

reshaped_arr4 = arr4.reshape((2, 3))

print("Reshaped 3x2 array:")

print(reshaped_arr4)

#Reshape 2x3:

arr5 = np.array([[1, 2, 3], [4, 5, 6]])

reshaped_arr5 = arr5.reshape((3, 2))

print("Reshaped 2x3 array:")

print(reshaped_arr5)

In [9]:
```
# Reshape 3x2:
arr4 = np.array([[1, 2], [3, 4], [5, 6]])
reshaped_arr4 = arr4.reshape((2, 3))
print("Reshaped 3x2 array:")
print(reshaped_arr4)

#Reshape 2x3:
arr5 = np.array([[1, 2, 3], [4, 5, 6]])
reshaped_arr5 = arr5.reshape((3, 2))
print("Reshaped 2x3 array:")
print(reshaped_arr5)
```

```
Reshaped 3x2 array:
[[1 2 3]
 [4 5 6]]
Reshaped 2x3 array:
[[1 2]
 [3 4]
 [5 6]]
```

2. Matplotlib

1. **Write a Python programming to create a below chart of the popularity of programming Languages.**
2. **Sample data:**
   Programming languages: Java, Python, PHP, JavaScript, C#, C++
   Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

```
import matplotlib.pyplot as plt
# Given sample data
lang = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
# Creating a pie chart
plt.pie(popularity, labels=lang, autopct='%1.1f%%')
# Title of pie chart
plt.title('Popularity of Programming Languages')
# To show the plot
plt.show()
```

In [10]:

```
import matplotlib.pyplot as plt

# Given sample data
lang = ['Java', 'Python', 'PHP', 'JavaScript', 'C#', 'C++']
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]

# Creating a pie chart
plt.pie(popularity, labels=lang, autopct='%1.1f%%')

# Title of pie chart
plt.title('Popularity of Programming Languages')

# To show the plot
plt.show()
```