Machine Learning – Assignment 3

CS 5710 (CRN 22002)

Student ID: 700745451

Student Name: Kamala Ramesh

1. Numpy:

a. To create random vector of size 15 having only Integers in the range 1-20 using numpy array

1. Reshape the array to 3 by 5
2. Print array shape.
3. Replace the max in each row by

```python
import numpy as np

#create a random vector of size 15
array = np.random.randint(1,21,size=15)
print("Random vector if size 15:",array)

#reshape the vector to array of size 3X5
reshapeArray = array.reshape((3,5))
print("\nRandom Vector reshaped into array of shape(3,5)\n",reshapeArray)
print("\nShape of array:",reshapeArray.shape)

#replace the max value in each row with zero
modifiedArray = np.where(reshapeArray == np.amax(reshapeArray, axis=1).reshape(-1,1),0,reshapeArray)
print("\nModified Array with zeros instead of Max values:\n",modifiedArray)
```

```
Random vector if size 15: [ 3  2 19 12 20 16 18  2 16 17  9 11 11 14  1]

Random Vector reshaped into array of shape(3,5)
 [[ 3  2 19 12 20]
 [16 18  2 16 17]
 [ 9 11 11 14  1]]

Shape of array: (3, 5)

Modified Array with zeros instead of Max values:
 [[ 3  2 19 12  0]
 [16  0  2 16 17]
 [ 9 11 11  0  1]]
```

To 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.

```
#print the shape, type and datatype of the array
a = np.array([[1,2,3],[4,5,6],[20,11,3],[21,5,6]])
print(a)
print("\nShape of array:",a.shape)
print("Type of array:",type(a))
print("Data type of array:",a.dtype)
```

```
[[ 1  2  3]
 [ 4  5  6]
 [20 11  3]
 [21  5  6]]

Shape of array: (4, 3)
Type of array: <class 'numpy.ndarray'>
Data type of array: int32
```

b. To compute the eigenvalues and right eigenvectors of a given square array

```
#given array
b = np.array([[3,-2],[1,0]])
print("Given square array:\n",b)

#Compute the Eigen values and vectors
eigenVal, eigenVec = np.linalg.eig(b)

print("\nEigen Values:\n",eigenVal)
print("\nEigen Vectors:\n",eigenVec)
```

```
Given square array:
 [[ 3 -2]
 [ 1  0]]

Eigen Values:
 [2. 1.]

Eigen Vectors:
 [[0.89442719 0.70710678]
 [0.4472136  0.70710678]]
```

c. To compute the sum of the diagonal element of a given array.

```
arr = np.array([[0,1,2],[3,4,5]])
print("Given array:\n",g)

diagArr = np.diag(g) #diagonal of the given array
diagSum = np.sum(diagArr) #Sum of the diagonal elements

print("\nDiagonal of the given array:\n",diagArr)
print("\nSum of the diagonal elements:",diagSum)
```

```
Given array:
 [[0 1 2]
 [3 4 5]]

Diagonal of the given array:
 [0 4]

Sum of the diagonal elements: 4
```

d. To create a new shape to an array without changing its data.

```
#array of shape 3 X 2
e = np.array([[1,2],[3,4],[5,6]])
print("Original array:\n",e)

#reshpe the original array to 2 X 3
r = e.reshape((2,3))
print("\nAfter Reshape:\n",r)
```

```
Original array:
 [[1 2]
 [3 4]
 [5 6]]

After Reshape:
 [[1 2 3]
 [4 5 6]]
```

2. Matplotlib

To create a pie chart of the popularity of programming Languages for the given 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
popularity = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
progLang = ['Java', 'Python', 'PHP', 'JavaScript','C#','C++']

#this function creates a wedge for the maximum value in the pie chart
def createWedge(list1):
    length = len(list1)
    list2 = [0] * length
    list2[list1.index(max(list1))] = 0.1
    return list2

explode = createWedge(popularity)

#plot the data in a pie chart
plt.pie(popularity, labels = progLang,explode=explode,autopct='%1.1f%%',
        startangle = 135, shadow=True,
        wedgeprops = {"edgecolor" : "black",
                      'linewidth': 1,
                      'antialiased': True})
plt.show()
```