Machine Learninig Lab

Exercise 2

1. Import the numpy package under the name `np`

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

2. Print the numpy version and the configuration

In [ ]:
```
print(np.__version__)
np.show_config()
```

3. Create a null vector of size 10

In [ ]:
```
Z = np.zeros(10)
print(Z)
```

4. How to find the memory size of any array

In [ ]:
```
Z = np.zeros((10,10))
print("%d bytes" % (Z.size * Z.itemsize))
```

5. How to get the documentation of the numpy add function from the command line?

In [ ]:
```
%run `python -c "import numpy; numpy.info(numpy.add)"`
```

6. Create a null vector of size 10 but the fifth value which is 1

In [ ]:
```
Z = np.zeros(10)
Z[4] = 1
print(Z)
```

7. Create a vector with values ranging from 10 to 49

In [ ]:
```
Z = np.arange(10,50)
print(Z)
```

8. Reverse a vector (first element becomes last)

In [ ]:
```
Z = np.arange(50)
Z = Z[::-1]
print(Z)
```

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

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

10. Find indices of non-zero elements from [1,2,0,0,4,0]:

In [ ]:

```
nz = np.nonzero([1,2,0,0,4,0])
print(nz)
```

11. Create a 3x3 identity matrix

In [ ]:

```
Z = np.eye(3)
print(Z)
```

12. Create a 3x3x3 array with random values

In [ ]:

```
Z = np.random.random((3,3,3))
print(Z)
```

13. Create a 10x10 array with random values and find the minimum and maximum values

In [ ]:

```
Z = np.random.random((10,10))
Zmin, Zmax = Z.min(), Z.max()
print(Zmin, Zmax)
```

14. Create a random vector of size 30 and find the mean value

In [ ]:

```
Z = np.random.random(30)
m = Z.mean()
print(m)
```

15. Create a 2d array with 1 on the border and 0 inside

In [ ]:

```
Z = np.ones((10,10))
Z[1:-1,1:-1] = 0
print(Z)
```

16. How to add a border (filled with 0's) around an existing array?

In [ ]:

```
Z = np.ones((5,5))
Z = np.pad(Z, pad_width=2, mode='constant', constant_values=0)
print(Z)
```

17. What is the result of the following expression?

In [ ]:

```
print(0 * np.nan)
print(np.nan == np.nan)
print(np.inf > np.nan)
print(np.inf)
print(np.nan - np.nan)
print(np.nan in set([np.nan]))
print(0.3 == 3 * 0.1)
print(np.nan)
print(0.3)
print(3*0.1)
print(np.inf>100000000)
```

18. Create a 5x5 matrix with values 1,2,3,4 just below the diagonal

In [ ]:

```
Z = np.diag(1+np.arange(4),k=-1)
print(Z)
```

19. Create a 8x8 matrix and fill it with a checkerboard pattern

In [ ]:
```
Z = np.zeros((8,8),dtype=int)
Z[1::2,::2] = 1
Z[::2,1::2] = 1
print(Z)
```

20. Consider a (6,7,8) shape array, what is the index (x,y,z) of the 100th element?

In [ ]:
```
print(np.unravel_index(99,(6,7,8)))
```

21. Create a checkerboard 8x8 matrix using the tile function

In [ ]:
```
Z = np.tile( np.array([[0,1],[1,0]]), (4,4))
print(Z)
```

22. Normalize a 5x5 random matrix

In [ ]:
```
Z = np.random.random((5,5))
print(Z)
print(np.mean(Z))
print(np.std(Z))
Z = (Z - np.mean (Z)) / (np.std (Z))
print(Z)
```

23. Given a 1D array, negate all elements which are between 3 and 8, in place.

In [ ]:
```
Z = np.arange(11)
print(Z)
Z[(3 < Z) & (Z <= 8)] *= -1
print(Z)
```

24. Consider an integer vector Z, which of these expressions are legal?

In [ ]:
```
Z=4
Z**Z
print(Z**Z)
print(2 << Z >> 2)
print(Z <- Z)
print(1j*Z)
print(Z/1/1)
print(Z<Z>Z)
```

25. How to find common values between two arrays?

In [ ]:
```
Z1 = np.random.randint(0,10,10)
print(Z1)
Z2 = np.random.randint(0,10,10)
print(Z2)
print(np.intersect1d(Z1,Z2))
```

26. How to get the dates of yesterday, today and tomorrow?

In [ ]:
```
yesterday = np.datetime64('today', 'D') - np.timedelta64(4, 'D')
today     = np.datetime64('today', 'D')
tomorrow  = np.datetime64('today', 'D') + np.timedelta64(1, 'D')
print(yesterday)
```

27. How to get all the dates corresponding to the month of July 2016?

In [ ]:
```
Z = np.arange('2016-07', '2016-08', dtype='datetime64[D]')
print(Z)
```

28. Extract the integer part of a random array using 5 different methods

In [ ]:

```
Z = np.random.uniform(0,10,10)
print(Z)
print(Z%1)
print (Z - Z%1)
print (np.floor(Z))
print (np.ceil(Z)-1)
print (Z.astype(int))
print (np.trunc(Z))
```

29. Create a 5x5 matrix with row values ranging from 0 to 4

In [ ]:

```
Z = np.zeros((5,5))
Z += np.arange(5)
print(Z)
```

30. Create a random vector of size 10 and sort it

In [ ]:

```
Z = np.random.random(10)
Z.sort()
print(Z)
```

31. Consider two random array A and B, check if they are equal

In [ ]:

```
A = np.random.randint(0,2,5)
B = np.random.randint(0,2,5)
print(A,B)
# Assuming identical shape of the arrays and a tolerance for the comparison of values
equal = np.allclose(A,B)
print(equal)

# Checking both the shape and the element values, no tolerance (values have to be exactly equal)
equal = np.array_equal(A,B)
print(equal)
```

32. Make an array immutable (read-only)

In [ ]:

```
Z = np.zeros(10)
Z.flags.writeable = False
Z[0] = 1
```

33. Print the minimum and maximum representable value for each numpy scalar type

In [ ]:

```
for dtype in [np.int8, np.int32, np.int64]:
   print(np.iinfo(dtype).min)
   print(np.iinfo(dtype).max)
for dtype in [np.float32, np.float64]:
   print(np.finfo(dtype).min)
   print(np.finfo(dtype).max)
   print(np.finfo(dtype).eps)
```

34. How to find the closest value (to a given scalar) in a vector?

In [ ]:

```
Z = np.arange(100)
v = np.random.uniform(0,100)
print(v)
index = (np.abs(Z-v)).argmin()
print(Z[index])
```

35. Consider a random vector with shape (100,2) representing coordinates, find point by point distances

In [ ]:

```
Z = np.random.random((3,2))
X,Y = np.atleast_2d(Z[:,0], Z[:,1])
D = np.sqrt( (X-X.T)**2 + (Y-Y.T)**2)
print(D)

# Much faster with scipy
#Why? Find the reason
#Also find similar type of functions in Scipy which can perform better
import scipy

import scipy.spatial

Z = np.random.random((3,2))
D = scipy.spatial.distance.cdist(Z,Z)
print(D)
```

36. How to convert a float (32 bits) array into an integer (32 bits) in place?

In [ ]:

```
Z = np.arange(10, dtype=np.float32)
print(Z)
Z = Z.astype(np.int32, copy=False)
print(Z)
```

37. How to read the following file?

In [ ]:

```
from io import StringIO

# Fake file
s = StringIO("""1, 2, 3, 4, 5\n
                6,  ,  , 7, 8\n
                 ,  , 9,10,11\n""")
Z = np.genfromtxt(s, delimiter=",", dtype=np.int)
print(Z)
```

38. Subtract the mean of each row of a matrix

In [ ]:

```
X = np.random.rand(5, 10)

# Recent versions of numpy
Y = X - X.mean(axis=1, keepdims=True)

# Older versions of numpy
Y = X - X.mean(axis=1).reshape(-1, 1)

print(Y)
```

39. How to sort an array by the nth column?

In [ ]:

```
Z = np.random.randint(0,10,(3,3))
print(Z)
print(Z[Z[:,1].argsort()])
```

40. How to tell if a given 2D array has null columns?

In [ ]:

```
Z = np.random.randint(0,3,(3,10))
print(Z)
print((~Z.any(axis=0)).any())
```

41. Find the nearest value from a given value in an array

In [ ]:

```
Z = np.random.uniform(0,1,10)
print(Z)
z = 0.5
m = Z.flat[np.abs(Z - z).argmin()]
print(m)
```

42. How to accumulate elements of a vector (X) to an array (F) based on an index list (I)?

In [ ]:
```
import numpy as np
X = [1,2,3,4,5,6]
I = [1,3,9,3,4,1]
F = np.bincount(I,X)
print(F)
```

43. How to get the diagonal of a dot product?

In [ ]:
```
A = np.random.uniform(0,1,(5,5))
B = np.random.uniform(0,1,(5,5))

# Slow version
np.diag(np.dot(A, B))

# Fast version
np.sum(A * B.T, axis=1)

# Faster version
np.einsum("ij,ji->i", A, B)
```

44. Consider the vector [1, 2, 3, 4, 5], how to build a new vector with 3 consecutive zeros interleaved between each value?

In [ ]:
```
Z = np.array([1,2,3,4,5])
nz = 3
Z0 = np.zeros(len(Z) + (len(Z)-1)*(nz))
Z0[::nz+1] = Z
print(Z0)
```

45. How to swap two rows of an array?

In [ ]:
```
A = np.arange(25).reshape(5,5)
A[[0,1]] = A[[1,0]]
print(A)
```

46. How to compute averages using a sliding window over an array?

In [ ]:
```
def moving_average(a, n=3) :
    ret = np.cumsum(a, dtype=float)
    ret[n:] = ret[n:] - ret[:-n]
    return ret[n - 1:] / n
Z = np.arange(20)
print(moving_average(Z, n=3))
```

47. Consider 2 sets of points P0,P1 describing lines (2d) and a point p, how to compute distance from p to each line i (P0[i],P1[i])?

In [ ]:
```
def distance(P0, P1, p):
    T = P1 - P0
    L = (T**2).sum(axis=1)
    U = -((P0[:,0]-p[...,0])*T[:,0] + (P0[:,1]-p[...,1])*T[:,1]) / L
    U = U.reshape(len(U),1)
    D = P0 + U*T - p
    return np.sqrt((D**2).sum(axis=1))

P0 = np.random.uniform(-10,10,(10,2))
P1 = np.random.uniform(-10,10,(10,2))
p  = np.random.uniform(-10,10,( 1,2))
print(distance(P0, P1, p))
```

48. Consider 2 sets of points P0,P1 describing lines (2d) and a set of points P, how to compute distance from each point j (P[j]) to each line i (P0[i],P1[i])?

In [ ]:

```
# based on distance function from previous question
P0 = np.random.uniform(-10, 10, (10,2))
P1 = np.random.uniform(-10,10,(10,2))
p = np.random.uniform(-10, 10, (10,2))
print(np.array([distance(P0,P1,p_i) for p_i in p]))
```

49. How to find the most frequent value in an array?

In [ ]:

```
Z = np.random.randint(0,10,50)
print(np.bincount(Z).argmax())
```

50. Extract all the contiguous 3x3 blocks from a random 10x10 matrix

In [ ]:

```
Z = np.random.randint(0,5,(10,10))
n = 3
i = 1 + (Z.shape[0]-3)
j = 1 + (Z.shape[1]-3)
C = stride_tricks.as_strided(Z, shape=(i, j, n, n), strides=Z.strides + Z.strides)
print(C)
```

51. Consider a 16x16 array, how to get the block-sum (block size is 4x4)?

In [ ]:

```
Z = np.ones((16,16))
k = 4
S = np.add.reduceat(np.add.reduceat(Z, np.arange(0, Z.shape[0], k), axis=0),
                                       np.arange(0, Z.shape[1], k), axis=1)
print(S)
```

52. How to get the n largest values of an array

In [ ]:

```
Z = np.arange(10000)
np.random.shuffle(Z)
n = 5

# Slow
print (Z[np.argsort(Z)[-n:]])

# Fast
print (Z[np.argpartition(-Z,n)[:n]])
```

53. Compute bootstrapped 95% confidence intervals for the mean of a 1D array X (i.e., resample the elements of an array with replacement N times, compute the mean of each sample, and then compute percentiles over the means).

In [ ]:

```
X = np.random.randn(100) # random 1D array
N = 1000 # number of bootstrap samples
idx = np.random.randint(0, X.size, (N, X.size))
means = X[idx].mean(axis=1)
confint = np.percentile(means, [2.5, 97.5])
print(confint)
```