HW1

Baktash Ansari

STU number: 99521082

Q1

I perform element-wise comaprison with just one comparison operation. I compare two numpy array and numpy create same size numpy array which has values 0 and 1 for each index. It says that if the camparison is true the corresponding value will be 1 and reverse.

Q2

We have two kinds of multplications: element-wise and matrix multiplication. I just write a if condition for the method and then write relevant command. For element-wise we can use np.multiply and for matrix we can use np.matmul

Q3

First I check the input shapes and method type and if one of them is not valid, I raise a value error. After that for row wise if we simply add q to p it will be broadcast. So I just write p + q. Yet, for column wise I change the q shape to (n, 1) from (n,) and after that if we add them together, q broadcasts to each column of p.

Q4

I create random matrix with np.random.randint and after that I subtract min element from each element then divided to (max - min). Therefore, we will have values between 0 and 1

Q5

For this Question first I read data.csv with pandas library and then iterate over this dataframe and calculate daily returns for everyday expect first day. Then I save these values with relevant date in a new dataframe name returns. After that, I change the return value column to numpy array and then calculate average and deviation values. Then, I plot the two dataframe And after that I itterate over Close Prices and also Daily Returns values to find maximum and minimum values and corresponding dates. I used matplolib, numpy, and pandas libraries.

Q6

As we see in the notebook, First I use two inner loops for feed forwarding and then I use matmul function for matrix multiplication. As we can see the duration time of vectorization method is much less than inner loops.

Q7

We just need a Comparitive operator and numpy do the rest!

Q8

For this Question:
For method 1: I use two inner loop for checking each element of those matrices.
For method 2: First I implement th output matrix with zeroes. Then, with two inner loop check every two relevant elements and assign True or False for the same index in output matrix.
For method3: I explain my first line code with this line of code:
all(any(all(elem in l1 for elem in l2) for l1 in list1) for l2 in list2)
So we have:

- outer loop iterating over each sublist l2 in list2
- Inner loop iterating over each sublist l1 in list1
- all(elem in l1 for elem in l2): checks if every element in l2 is present in l1
- all(...) checks if all the elements generated by the generator expression are True
- any(all(elem in l1 for elem in l2) for l1 in list1): This checks if there exists at least one sublist l1 in list1 for which all the elements in l2 are present.
- any(...) checks if any of the results from the all(...) evaluations are True.
- all(any(all(elem in l1 for elem in l2) for l1 in list1) for l2 in list2): This outer all(...) checks if this condition holds for all sublists l2 in list2.
- It means it checks if every sublist in list2 has at least one corresponding sublist in list1 such that all the elements in the sublist from list2 are present in the corresponding sublist from list1.