

EXERCISE - 1

1) Create a Numpy array of size 4 x 5.

```
In [1]: import numpy as np
array=np.array([[1,2,3,4,5],[5,7,8,9,10],[11,12,13,14,15],[16,17,18,19,20]])
print(array)

[[ 1  2  3  4  5]
 [ 5  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]]
```

2) Randomly initialize the array.

```
import numpy as np
narray = np.random.rand(4,5)
print(narray)

[[0.39970999 0.47556514 0.28608306 0.28523947 0.85812895]
 [0.85400706 0.1213114 0.8594821 0.83183588 0.09507433]
 [0.61009624 0.64341525 0.438094 0.18795508 0.57150042]
 [0.36055028 0.54300599 0.1370526 0.69513789 0.53959356]]
```

3)

- Get the Transpose of the Matrix that you created.

```
In [2]: import numpy as np
array=np.array([[1,2,3,4,5],[5,7,8,9,10],[11,12,13,14,15],[16,17,18,19,20]])
print('Original Matrix')
print(array)
print('Transpose of the Matrix')
print(array.T)

Original Matrix
[[ 1  2  3  4  5]
 [ 5  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]]
Transpose of the Matrix
[[ 1  5 11 16]
 [ 2  7 12 17]
 [ 3  8 13 18]
 [ 4  9 14 19]
 [ 5 10 15 20]]
```

- Create a square matrix and find its determinant.

```
In [5]: import numpy as np
array=np.array([[1,2,3],[4,5,6],[7,8,9]])
print(array)
det = np.linalg.det(array)
print('Determinant:')
print(int(det))

[[1 2 3]
 [4 5 6]
 [7 8 9]]
Determinant:
0
```

4) Create another matrix of size 5 x 4 and randomly initialize it.



```
narry = np.random.rand(5,4)
print(narry)
```

```
[[0.56295746 0.08223779 0.27175918 0.82546843]
 [0.08518711 0.60319843 0.09979623 0.81162564]
 [0.52440819 0.13651419 0.86792195 0.50426013]
 [0.88226144 0.32509565 0.17625569 0.1808259 ]
 [0.11145308 0.56187803 0.10698442 0.54848866]]
```

5) Perform Matrix multiplication.

```
In [7]: import numpy as np
array1=np.array([[1,2,3],[4,5,6],[7,8,9]])
array2=np.array([[1,2,3],[4,5,6],[7,8,9]])
print(array1*array2)
```

```
[[ 1  4  9]
 [16 25 36]
 [49 64 81]]
```

6) Perform element wise matrix multiplication.

```
In [9]: import numpy as np
array=np.array([[1,2,3],[4,5,6],[7,8,9]])
result=array*2
print(result)
```

```
[[ 2  4  6]
 [ 8 10 12]
 [14 16 18]]
```

7) Find the mean, median of the numpy array created.



```
array = np.array([[1, 2, 3], [4, 5, 6]])
print('Mean:', np.mean(array))
print('Median:', np.median(array))
```

```
Mean: 3.5
Median: 3.5
```

8) Obtain each row in the second column of the first array.

```
arr = np.array([[1, 2, 3], [4, 5, 6]])  
print(arr)  
arr[:,2]
```

```
[[1 2 3]  
 [4 5 6]]  
array([3, 6])
```

9) Convert Numeric entries(columns) of Iris.csv to Mean Centered Version.

```
data=pd.read_csv('/content/Iris.csv')  
arr= np.array(data)  
iris_array=np.array([arr[:,0],arr[:,1],arr[:,2]])  
iris_centered=iris_array - np.mean(iris_array,axis=0)  
print(iris_centered)
```

```
[[-2.1999999999999997 -1.3000000000000003 -0.6333333333333333  
 0.10000000000000009 0.4666666666666668 0.8999999999999995 2.0  
 2.5333333333333334 3.5666666666666664 4.0 4.300000000000001  
 5.266666666666667 6.066666666666666 6.8999999999999995  
 6.7333333333333325 7.299999999999999 8.233333333333334  
 9.133333333333333 9.5 10.366666666666665 11.066666666666668  
 11.733333333333333 12.6 13.200000000000001 13.933333333333332  
 14.666666666666666 15.200000000000001 15.766666666666666  
 16.466666666666667 17.366666666666667 18.033333333333333 18.4 18.9  
 19.433333333333333 20.666666666666664 21.266666666666666  
 21.666666666666664 22.666666666666664 23.533333333333333  
 23.833333333333332 24.5 25.733333333333334 26.133333333333333 26.5  
 27.033333333333335 28.066666666666666 28.366666666666667 29.4  
 29.666666666666668 30.566666666666666 30.599999999999998  
 31.466666666666665 32.0 33.400000000000006 33.566666666666666 34.5 34.8  
 36.233333333333334 36.166666666666667 37.366666666666666  
 38.333333333333333 38.366666666666666 39.266666666666666  
 39.666666666666667 40.5 40.733333333333334 41.8 42.5 43.2  
 43.966666666666667 44.3 45.033333333333333 45.733333333333334  
 46.366666666666674 46.899999999999999 47.466666666666667  
 48.133333333333334 48.766666666666666 49.7 50.566666666666666  
 51.366666666666666 52.033333333333333 52.5 53.099999999999994  
 53.866666666666666 54.2 54.733333333333334 55.800000000000004  
 56.466666666666667 57.333333333333336 57.966666666666667  
 58.300000000000004 59.2 60.233333333333334 60.566666666666667 61.1 61.8  
 62.3 63.466666666666667 63.833333333333336 64.133333333333333  
 65.166666666666666 65.300000000000001 66.266666666666667 66.833333333333334  
 67.133333333333333 68.866666666666667 68.6 69.6 69.733333333333333  
 70.766666666666667 71.633333333333333 72.066666666666666 73.266666666666667  
 73.800000000000001 74.133333333333333 74.833333333333334 74.833333333333334  
 75.9 77.266666666666668 77.300000000000001 78.533333333333333 78.5]
```

10) Study about numpy array attributes and implement it on the first matrix.



```
x=np.array([[4,7,3],[8,10,6]])  
print(x.shape)
```

```
(2, 3)
```

EXERCISE – 2

Perform all of the above preprocessing tasks on any other text dataset.

```
import nltk
from nltk.corpus import twitter_samples
import matplotlib.pyplot as plt
import random
import re # library for regular expression operations
import string # for string operations
from nltk.corpus import stopwords # module for stop words that come with NLTK
from nltk.stem import PorterStemmer # module for stemming
from nltk.tokenize import TweetTokenizer # module for tokenizing strings

nltk.download("twitter_samples")

allPositiveTweets = twitter_samples.strings("positive_tweets.json")
allNegativeTweets = twitter_samples.strings("negative_tweets.json")

print("Number of Positive Tweets", len(allPositiveTweets))
print("Number of Negative Tweets", len(allNegativeTweets))

print('\n\nThe type of all_positive_tweets is: ', type(allPositiveTweets))
print('The type of a tweet entry is: ', type(allNegativeTweets[0]))

# print positive in green
print('\033[92m' + allPositiveTweets[random.randint(0, 5000)])
# print negative in red
print('\033[91m' + allNegativeTweets[random.randint(0, 5000)])
tweet = allPositiveTweets[2277]
print(tweet)

# download the stopwords from NLTK
nltk.download('stopwords')
print('\033[92m' + tweet)
print('\033[94m')
```



```
# remove hyperlinks
tweet2 = re.sub(r'https?:\/\/.*[\r\n]*', '', tweet)
# remove hashtags
# only removing the hash # sign from the word
tweet2 = re.sub(r'#', '', tweet2)
print(tweet2)

print()
print('\033[92m' + tweet2)
print('\033[94m')
# instantiate tokenizer class
tokenizer = TweetTokenizer(preserve_case=False)
# tokenize tweets
tweet_tokens = tokenizer.tokenize(tweet2)
print()
print('Tokenized string:')
print(tweet_tokens)

# Import the english stop words list from NLTK
stopwords_english = stopwords.words('english')
print('Stop words\n')
print(stopwords_english)
print('\nPunctuation\n')
print(string.punctuation)

print()
print('\033[92m')
print(tweet_tokens)
print('\033[94m')
tweets_clean = []
for word in tweet_tokens: # Go through every word in your tokens list
    if (word not in stopwords_english and # remove stopwords
        word not in string.punctuation): # remove punctuation
        tweets_clean.append(word)
print('removed stop words and punctuation:')
```

```

print(tweets_clean)

print()
print('\033[92m')
print(tweets_clean)
print('\033[94m')
# Instantiate stemming class
stemmer = PorterStemmer()
# Create an empty list to store the stems
tweets_stem = []
for word in tweets_clean:
    stem_word = stemmer.stem(word) # stemming word
    tweets_stem.append(stem_word) # append to the list
print('stemmed words:')
print(tweets_stem)

```

```

[nltk_data] Downloading package twitter_samples to /root/nltk_data...
[nltk_data] Package twitter_samples is already up-to-date!
Number of Positive Tweets 5000
Number of Negative Tweets 5000

The type of all_positive_tweets is: <class 'list'>
The type of a tweet entry is: <class 'str'>
@tazzy_7 @5505 only a pleasure, Larry! Shot for listening {}
Gutted for Reynolds. The kind of desserts he has done this season {}
My beautiful sunflowers on a sunny Friday morning off {} #sunflowers #favourites #happy #Friday off_ https://t.co/3tFrm0U1
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
My beautiful sunflowers on a sunny Friday morning off {} #sunflowers #favourites #happy #Friday off_ https://t.co/3tFrm0U1

My beautiful sunflowers on a sunny Friday morning off {} sunflowers favourites happy Friday off_
My beautiful sunflowers on a sunny Friday morning off {} sunflowers favourites happy Friday off_

Tokenized string:
['my', 'beautiful', 'sunflowers', 'on', 'a', 'sunny', 'Friday', 'morning', 'off', ':)', 'sunflowers', 'favourites', 'happy', 'Friday', 'off', '-']
Stop words:
['I', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'you\'re', 'you\'ve', 'you\'ll', 'you\'d', 'your', 'yours', 'yourself', 'yourselves', 'he', 'his', 'his', 'h

Punctuation:
['@5505(){}*.,/:;(){}[]!@!~_[]{}-

['my', 'beautiful', 'sunflowers', 'on', 'a', 'sunny', 'Friday', 'morning', 'off', ':)', 'sunflowers', 'favourites', 'happy', 'Friday', 'off', '-']

removed stop words and punctuation:
['beautiful', 'sunflowers', 'sunny', 'Friday', 'morning', ':)', 'sunflowers', 'favourites', 'happy', 'Friday', '-']

['beautiful', 'sunflowers', 'sunny', 'Friday', 'morning', ':)', 'sunflowers', 'favourites', 'happy', 'Friday', '-']

stemmed words:
['beauti', 'sunflow', 'sunny', 'Friday', 'morn', ':)', 'sunflow', 'favourit', 'happi', 'Friday', '-']

```

