Dhanwin_221_Lab9

September 16, 2023

[ ]:
```
# Q1. Write a program to distinguish between Array Indexing and Fancy Indexing.
import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print(arr)
simple_index = arr[3]
# We use a single valued index in the array
print("Simple Indexing:", simple_index)
fancy_index = arr[[1, 3, 5, 7]]
# We use an array of indices in the array
print("Fancy Indexing:", fancy_index)
```

```
[ 1  2  3  4  5  6  7  8  9 10]
Simple Indexing: 4
Fancy Indexing: [2 4 6 8]
```

[ ]:
```
# Q2. Execute the 2D array Slicing.

import numpy as np
arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8]])
print(arr)
print("\n")
print(arr[0:2, 2:5])
print(arr[0, 1:3])
print(arr[0:1, 2])
```

```
[[1 2 3 4]
 [5 6 7 8]]


[[3 4]
 [7 8]]
[2 3]
[3]
```

[ ]:
```
# Q3. Create the 5-Dimensional arrays using 'ndmin'.

import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9], ndmin=5)
print(arr)
```

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

[ ]:
```
# Q4. Reshape the array from 1-D to 2-D array.

import numpy as np
arr = np.array([1, 2, 3, 4, 5, 6, 7, 8])
newarr = arr.reshape(2, 4)
print(newarr)
```

```
[[1 2 3 4]
 [5 6 7 8]]
```

[ ]:
```
# Q5. Perform the Stack functions in Numpy arrays - stack(), hstack(),␣
 ↪vstack(), and dstack().

import numpy as np
arr1 = np.array([0 ,2, 3, 4])
arr2 = np.array([7, 2, 2, 1])
print("Array I :",arr1)
print("Array II :",arr2)
arr = np.stack((arr1, arr2), axis=0) #stack()
print("stack()\n",arr)
arr = np.hstack((arr1, arr2)) #hstack()
print("hstack()\n",arr)
arr = np.vstack((arr1, arr2)) #vstack()
print("vstack()\n",arr)
arr = np.dstack((arr1, arr2)) #dstack()
print("dstack()\n",arr)
```

```
Array I : [0 2 3 4]
Array II : [7 2 2 1]
stack()
 [[0 2 3 4]
 [7 2 2 1]]
hstack()
 [0 2 3 4 7 2 2 1]
vstack()
 [[0 2 3 4]
 [7 2 2 1]]
dstack()
 [[[0 7]
  [2 2]
  [3 2]
  [4 1]]]
```

[ ]:
```
# Q6. Perform the searchsort method in Numpy array.

import numpy as np
```

```
arr = np.array([5, 6, 8, 9])
x = np.searchsorted(arr, 7)
print(x)
x = np.searchsorted(arr, 1)
print(x)
```

```
2
0
```

[ ]:
```
# Q7. Create Numpy Structured array using your domain features.
# My domain is Regional Social Networking App so The Numpy Structured Array is␣
 ↪based on User Details

import numpy as np
arr = np.array([('Dhanwin KB', 'dkbiscool', 22, 'dkb@gmail.com'), ('Anirudh␣
 ↪Ravichander', 'rajniismyuncle', 32, 'ani@gmail.com'), ('Mohanlal',␣
 ↪'iamcompleteactor', 69,'a10@gmail.com')],
      dtype=[('uname', (np.str_, 10)), ('password', (np.str_, 25)), ('age', np.
 ↪int32), ('mailid', (np.str_, 50))])
print(arr)
```

```
[('Dhanwin KB', 'dkbiscool', 22, 'dkb@gmail.com')
 ('Anirudh Ra', 'rajniismyuncle', 32, 'ani@gmail.com')
 ('Mohanlal', 'iamcompleteactor', 69, 'a10@gmail.com')]
```

[ ]:
```
# Q8. Create Data frame using List and Dictionary.

import pandas as pd
l1 = ['DKB', 'ARR', 'ANI', 'A10']
d1 = {'upcoming':'DKB','goat':'ARR','next_goat':'ANI','vintage':'A10'}
df = pd.DataFrame(l1, index =['upcoming', 'goat', 'next goat',␣
 ↪'vintage'],columns =['abbr. names'])
print("list :",l1,'\n\n',df)
df = pd.DataFrame(list(d1.items()),columns =['titles','abbr. names'])
print("\ndictionary :",d1,'\n\n',df)
```

```
list : ['DKB', 'ARR', 'ANI', 'A10']

           abbr. names
upcoming           DKB
goat               ARR
next goat          ANI
vintage            A10

dictionary : {'upcoming': 'DKB', 'goat': 'ARR', 'next_goat': 'ANI', 'vintage':
'A10'}

       titles abbr. names
```

```
0   upcoming          DKB
1       goat          ARR
2  next_goat          ANI
3    vintage          A10
```

[ ]:
```
# Q9. Create Data frame on your Domain area and perform the following␣
 ↪operations to find and eliminate the
# missing data from the dataset.

import pandas as pd
usr_db = {
  "unames": ['Dhanwin KB','Anirudh Ravichander','Mohanlal','Mammootty'],
  "passwords":␣
 ↪['dkbiscool','rajniismyuncle','iamcompleteactor','iamyoungerthandq'],
  "age": [22,32,np.nan,72],
  "mailid": ['dkb@gmail.com','rajniismyuncle','a10@gmail.com','ikka@gmail.com']
}
df = pd.DataFrame(usr_db, index = ["1", "2", "3", "4"])
```

[ ]:
```
# • isnull()

print(df)
df.isnull()
```

```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor   NaN   a10@gmail.com
4             Mammootty  iamyoungerthandq  72.0  ikka@gmail.com
```

[ ]:
```
   unames  passwords    age  mailid
1   False      False  False   False
2   False      False  False   False
3   False      False   True   False
4   False      False  False   False
```

[ ]:
```
# • notnull()

print(df)
df.notnull()
```

```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor   NaN   a10@gmail.com
4             Mammootty  iamyoungerthandq  72.0  ikka@gmail.com
```

[ ]:

	unames	passwords	age	mailid
1	True	True	True	True
2	True	True	True	True
3	True	True	False	True
4	True	True	True	True

[ ]:
```
# • dropna()

print(df)
df.dropna()
```

	unames	passwords	age	mailid
1	Dhanwin KB	dkbiscool	22.0	dkb@gmail.com
2	Anirudh Ravichander	rajniismyuncle	32.0	rajniismyuncle
3	Mohanlal	iamcompleteactor	NaN	a10@gmail.com
4	Mammootty	iamyoungerthandq	72.0	ikka@gmail.com

[ ]:

	unames	passwords	age	mailid
1	Dhanwin KB	dkbiscool	22.0	dkb@gmail.com
2	Anirudh Ravichander	rajniismyuncle	32.0	rajniismyuncle
4	Mammootty	iamyoungerthandq	72.0	ikka@gmail.com

[ ]:
```
# • fillna(value)

print(df)
df.fillna(0)
```

	unames	passwords	age	mailid
1	Dhanwin KB	dkbiscool	22.0	dkb@gmail.com
2	Anirudh Ravichander	rajniismyuncle	32.0	rajniismyuncle
3	Mohanlal	iamcompleteactor	NaN	a10@gmail.com
4	Mammootty	iamyoungerthandq	72.0	ikka@gmail.com

[ ]:

	unames	passwords	age	mailid
1	Dhanwin KB	dkbiscool	22.0	dkb@gmail.com
2	Anirudh Ravichander	rajniismyuncle	32.0	rajniismyuncle
3	Mohanlal	iamcompleteactor	0.0	a10@gmail.com
4	Mammootty	iamyoungerthandq	72.0	ikka@gmail.com

[ ]:
```
# • fillna(method)

print(df)
df.fillna(method='bfill')
# df.fillna(method='ffill')
```

	unames	passwords	age	mailid
1	Dhanwin KB	dkbiscool	22.0	dkb@gmail.com
2	Anirudh Ravichander	rajniismyuncle	32.0	rajniismyuncle
3	Mohanlal	iamcompleteactor	NaN	a10@gmail.com

```
4            Mammootty  iamyoungerthandq  72.0  ikka@gmail.com

C:\Users\USER\AppData\Local\Temp\ipykernel_9128\4214753442.py:4: FutureWarning:
DataFrame.fillna with 'method' is deprecated and will raise in a future version.
Use obj.ffill() or obj.bfill() instead.
  df.fillna(method='bfill')
```

[ ]:
```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor  72.0   a10@gmail.com
4            Mammootty  iamyoungerthandq  72.0  ikka@gmail.com
```

[ ]:
```
# • replace()

print(df)
df.replace('iamyoungerthandq','iamolderthandq')
```

```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor   NaN   a10@gmail.com
4            Mammootty  iamyoungerthandq  72.0  ikka@gmail.com
```

[ ]:
```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor   NaN   a10@gmail.com
4            Mammootty    iamolderthandq  72.0  ikka@gmail.com
```

[ ]:
```
# • interpolate()

print(df)
df.interpolate()
```

```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor   NaN   a10@gmail.com
4            Mammootty  iamyoungerthandq  72.0  ikka@gmail.com

C:\Users\USER\AppData\Local\Temp\ipykernel_9128\2466800549.py:4: FutureWarning:
DataFrame.interpolate with object dtype is deprecated and will raise in a future
version. Call obj.infer_objects(copy=False) before interpolating instead.
  df.interpolate()
```

[ ]:
```
                unames         passwords   age          mailid
1           Dhanwin KB         dkbiscool  22.0   dkb@gmail.com
2  Anirudh Ravichander    rajniismyuncle  32.0  rajniismyuncle
3              Mohanlal  iamcompleteactor  52.0   a10@gmail.com
```

```
4            Mammootty  iamyoungerthandq  72.0  ikka@gmail.com
```

[ ]:
```python
# Q10. Perform the Hierarchical Indexing in the above created dataset.

df.set_index(['unames', 'age'], inplace=True) #multi index made
print(df)
```

```
                                passwords          mailid
unames              age
Dhanwin KB          22.0         dkbiscool   dkb@gmail.com
Anirudh Ravichander 32.0    rajniismyuncle  rajniismyuncle
Mohanlal            NaN   iamcompleteactor   a10@gmail.com
Mammootty           72.0  iamyoungerthandq  ikka@gmail.com
```