JoePayyappilly_227_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])
print(arr[2])



i = np.array([2, 4])
print(arr[i])

a = np.array([[0, 2], [1, 3]])
print(arr[a])
```

```
3
[3 5]
[[1 3]
 [2 4]]
```

Q2. Execute the 2D array Slicing.

[ ]:

```
import numpy as np

arr = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])

print(arr[:2])

print(arr[:, -1])

print(arr[1, 2])
```

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

Q3. Create the 5-Dimensional arrays using 'ndmin'.

[ ]:

```
import numpy as np
```

```
arr = np.array([1, 2, 3], ndmin=5)

print(arr)
```

```
[[[[[1 2 3]]]]]
```

Q4. Reshape the array from 1-D to 2-D array.

[ ]:

```
import numpy as np

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

print(arr.reshape((2, 3)))
```

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

Q5. Perform the Stack functions in Numpy arrays – Stack(), hstack(), vstack(), and dstack().

[ ]:

```
import numpy as np

arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])

print("stack them vertically")
print(np.stack((arr1, arr2), axis=0))
print("stack them horizontally")
print(np.hstack((arr1, arr2)))
print("stack them as 3d vertically")
print(np.dstack((arr1, arr2)))
```

```
stack them vertically
[[1 2 3]
 [4 5 6]]
stack them horizontally
[1 2 3 4 5 6]
stack them as 3d vertically
[[[1 4]
  [2 5]
  [3 6]]]
```

Q6. Perform the searchsort method in Numpy array.

[ ]:

```
import numpy as np

arr = np.array([1, 3, 5, 7, 9])

s = np.searchsorted(arr, 5)
print(f"the element is {s}")
```

```
the element is 2
```

Q7. Create Numpy Structured array using your domain features.

[ ]:

```
import numpy as np

array = np.array([(0,0,0),(1,1,0),(2,3,2),(4,10,3)], dtype=[('position', np.
 ↪int64), ('velocity', np.int64), ('acceleration', np.int64)])

print(array)
```

```
[(0,  0, 0) (1,  1, 0) (2,  3, 2) (4, 10, 3)]
```

Q8. Create Data frame using List and Dictionary.

[ ]:

```
import pandas as pd

mydatasetofdict = {
  'Scientist': ["Einstein", "Newton", "Bohr", "Heisenberg"],
  'Year of death': [1900, 1901, 1902, 1903]
}
print("Dataset using dictionary")
dictdata = pd.DataFrame(mydatasetofdict)
print(dictdata)



mydatasetoflist = [['Einstein', 1900], ['Newton',1901],␣
 ↪['Bohr',1902],['Heisenberg',1903]]

listdata= pd.DataFrame(mydatasetoflist, columns=['Scientist', 'Year of Death'])
print('\n')
print("Dataset using list")
print(listdata)
```

```
Dataset using dictionary
    Scientist  Year of death
0    Einstein           1900
1      Newton           1901
2        Bohr           1902
3  Heisenberg           1903


Dataset using list
    Scientist  Year of Death
0    Einstein           1900
1      Newton           1901
2        Bohr           1902
3  Heisenberg           1903
```

Q9. Create Data frame on your Domain area and perform the following operations to find and eliminate the missing data from the dataset. 1. isnull() 2. notnull() 3. dropna() 4. fillna() 5. replace() 6. interpolate()

[ ]:

```
import pandas as pd
import numpy as np
dataset = {
  'Scientist': ["Einstein", "Newton", np.nan, "Heisenberg","Galeleo"],
  'Year of death': [np.nan, 1901,np.nan, 1903,1904]
}
print("Dataset ")
df= pd.DataFrame(dataset)
print(df)

missing = df.isnull()
print("\n")
print(missing)

notmissing =df.notnull()
print('\n')
print(notmissing)

df = df.fillna(value='Unknown')
print('\n')
print(df)


df = df.replace("Newton","Tesla")
print('\n')
print(df)
```

```
Dataset
    Scientist  Year of death
0    Einstein            NaN
1      Newton         1901.0
2         NaN            NaN
3  Heisenberg         1903.0
4     Galeleo         1904.0



   Scientist  Year of death
0      False           True
1      False          False
2       True           True
3      False          False
4      False          False



   Scientist  Year of death
```

```
0       True          False
1       True           True
2      False          False
3       True           True
4       True           True


    Scientist Year of death
0    Einstein       Unknown
1      Newton        1901.0
2     Unknown       Unknown
3  Heisenberg        1903.0
4     Galeleo        1904.0


    Scientist Year of death
0    Einstein       Unknown
1       Tesla        1901.0
2     Unknown       Unknown
3  Heisenberg        1903.0
4     Galeleo        1904.0
```

[ ]:

```
import pandas as pd
import numpy as np
dataset = {
  'Scientist': ["Einstein", "Newton", np.nan, "Heisenberg","Galeleo"],
  'Year of death': [np.nan, 1901,np.nan, 1903,1904]
}
print("Dataset ")
df= pd.DataFrame(dataset)
print(df)

df = df.dropna()
print('\n')
print(df)
```

```
Dataset
    Scientist  Year of death
0    Einstein            NaN
1      Newton         1901.0
2         NaN            NaN
3  Heisenberg         1903.0
4     Galeleo         1904.0


    Scientist  Year of death
1      Newton         1901.0
3  Heisenberg         1903.0
```

```
4     Galeleo         1904.0
```

[ ]:

```python
import pandas as pd
import numpy as np
dataset = {
  'Scientist': ["Einstein", "Newton","Bohr", "Heisenberg","Galeleo"],
  'Year of death': [np.nan, 1901,np.nan, 1903,1904]
}
print("Dataset")
df= pd.DataFrame(dataset)
print(df)

df.interpolate(method ='linear', limit_direction ='backward')
```

```
Dataset
    Scientist  Year of death
0    Einstein            NaN
1      Newton         1901.0
2        Bohr            NaN
3  Heisenberg         1903.0
4     Galeleo         1904.0
```

[ ]:

```
    Scientist  Year of death
0    Einstein         1901.0
1      Newton         1901.0
2        Bohr         1902.0
3  Heisenberg         1903.0
4     Galeleo         1904.0
```

Q10. Perform the Hierarchical Indexing in the above created dataset.

[ ]:

```python
import pandas as pd
import numpy as np

dataset = {
    'Scientist': ["Einstein", "Newton", "Bohr", "Heisenberg", "Galeleo"],
    'Year of death': [np.nan, 1901, np.nan, 1903, 1904]
}


df = pd.DataFrame(dataset)

df.set_index(['Scientist', 'Year of death'], inplace=True)
# Retrieve data for Newton
newton_data = df.loc['Newton']
print(newton_data)
```

```
Dataset
Empty DataFrame
```

```
Columns: []
Index: [1901.0]
```