



**University of Engineering and
Technology ,Taxila**

Department of Computer Engineering

Lab Report 01

For the Course of Machine Learning lab

Submitted By: Muhammad Ibrahim (21-CP-26)

Section: Omega

Lab Instructor: Sir Sharyar Khan

Course Instructor: Dr. Waqar Ahmad

Date: 12-01-24.

Course Title: Machine Learning Lab

Tasks

1. Run all Example 40 in the Lab Manual 01 and save the code and Output?

Examples 1-5

```
79  #exp1
80  import numpy
81  arr=numpy.array([1,2,3,4,5])
82  print(arr)
83  #exp2
84  import numpy as np
85  arr=numpy.array([1,2,3,4,5])
86  print(arr)
87  print(np.__version__)
88  #exp 3
89  print(type(arr))
90  #exp 4
91  arr=np.array((1,2,3,4,5))
92  print("exp 4")
93  print(arr)
94  #exp5
95  print(arr[0])
```

```
C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
[1 2 3 4 5]
[1 2 3 4 5]
1.26.0
<class 'numpy.ndarray'>
exp 4
[1 2 3 4 5]
1

Process finished with exit code 0
```

```
#test your skills
print(arr[2])

b=arr[2]
c=arr[3]
print(a+b)
```

```
C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
3
7

Process finished with exit code 0
|
```

Examples 6-9

```

#exp6
import numpy as np
arr2=np.array([[1,2,3,4,5],[6,7,8,9,10]])
print('2nd elemnet on 1st row',arr2[0,1])

#exp 7
arr3=np.array([[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]])
print('Access the third element of the second array of the first array:', arr3[0,1,2])

#exp 8
print('Last element from 2nd dim:',arr2[1,-1])

#exp 9 array slicing
import numpy as np
arr=np.array([1,2,3,4,5,6,7])
print(arr[1:5])

```

```

C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
[5 6 7]
start to 4 not include [1 2 3 4]

Process finished with exit code 0

```

```

C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
2nd elemnet on 1st row 2
Access the third element of the second array of the first array: 6
Last element from 2nd dim: 10
[2 3 4 5]

Process finished with exit code 0

```

Examples 10-15

```

127 #exp 10
128 import numpy as np
129 arr=np.array([1,2,3,4])
130 print(arr.dtype)
131 #exp 11
132 arr=np.array(['apple','banana'])
133 print(arr.dtype)
134 #exp 12
135 arr=np.array([1,2,3,4,5])
136 x=arr.copy()
137 arr[0]=42
138 print(arr)
139 print(x)
140 #exp 13
141 arr=np.array([1,2,3,4,5])
142 x=arr.view()
143 arr[0]=42
144 print(arr)
145 print(x)
146 #exp 14 sorting
147 arr=np.array([3,2,0,1])
148 print(np.sort(arr))

```

Unable to save settings
Failed to save settings. Please restart PyCharm

```
#exp 15
arr = np.array(['banana', 'cherry', 'apple'])
print(np.sort(arr))
```

```
C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
int32
<U6
[42  2  3  4  5]
[1 2 3 4 5]
[42  2  3  4  5]
[42  2  3  4  5]
[0 1 2 3]
['apple' 'banana' 'cherry']

Process finished with exit code 0
|
```

Examples 16-40

```
1 #exp 16
2 import numpy as np
3 arr = np.array([1, 2, 3])
4 for x in arr:
5     print(x)
6
7 #exp 17
8 from numpy import random
9 x = random.randint(100)
10 print(x)
11
12 #exp 18
13 from numpy import random
14 x = random.choice([3, 5, 7, 9], p=[0.1, 0.3, 0.6, 0.0], size=(100))
15 print(x)
16
17 #exp 19
18 import matplotlib.pyplot as plt
19 import seaborn as sns
20 sns.distplot([0, 1, 2, 3, 4, 5])
21 plt.show()
22
```

```

23 #exp 20
24 import matplotlib.pyplot as plt
25 import seaborn as sns
26 sns.distplot([0, 1, 2, 3, 4, 5], hist=False)
27 plt.show()
28
29 #exp 21
30 import pandas
31 mydataset = {
32     'cars': ["BMW", "Volvo", "Ford"],
33     'passings': [3, 7, 2]
34 }
35 myvar = pandas.DataFrame(mydataset)
36 print(myvar)
37
38 #exp 22
39 import pandas as pd
40 mydataset = {
41     'cars': ["BMW", "Volvo", "Ford"],
42     'passings': [3, 7, 2]
43 }
44 myvar = pd.DataFrame(mydataset)

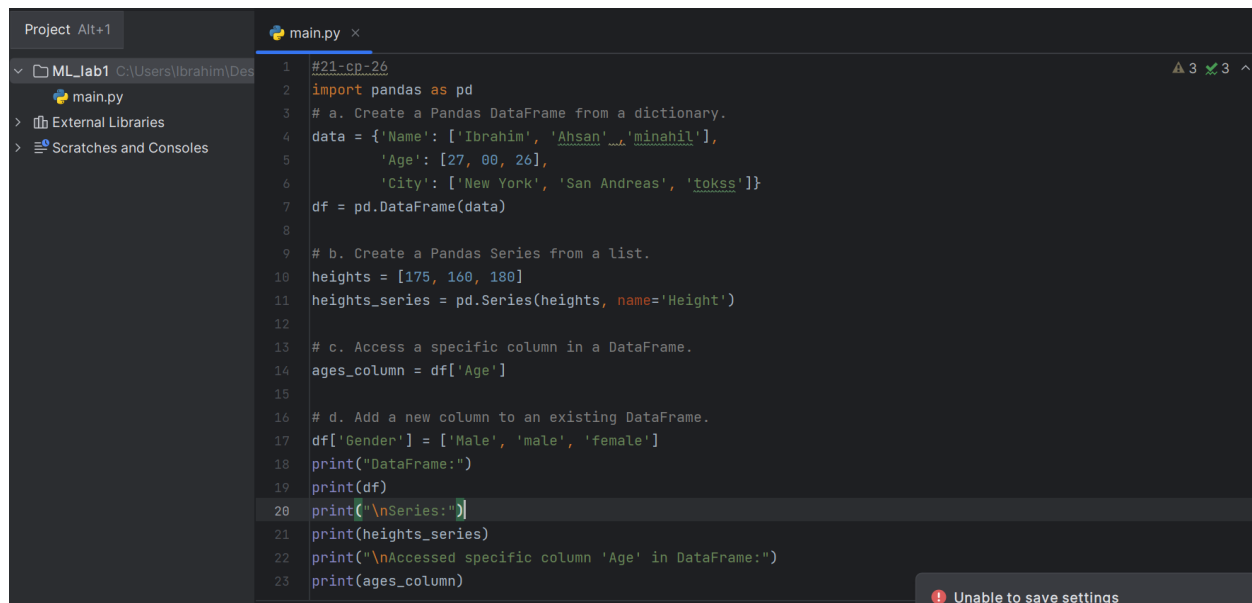
```

Task 2

2. Perform Below Tasks (Pandas)

- Create a Pandas DataFrame from a dictionary.
- Create a Pandas Series from a list.
- Access a specific column in a DataFrame.
- Add a new column to an existing DataFrame.

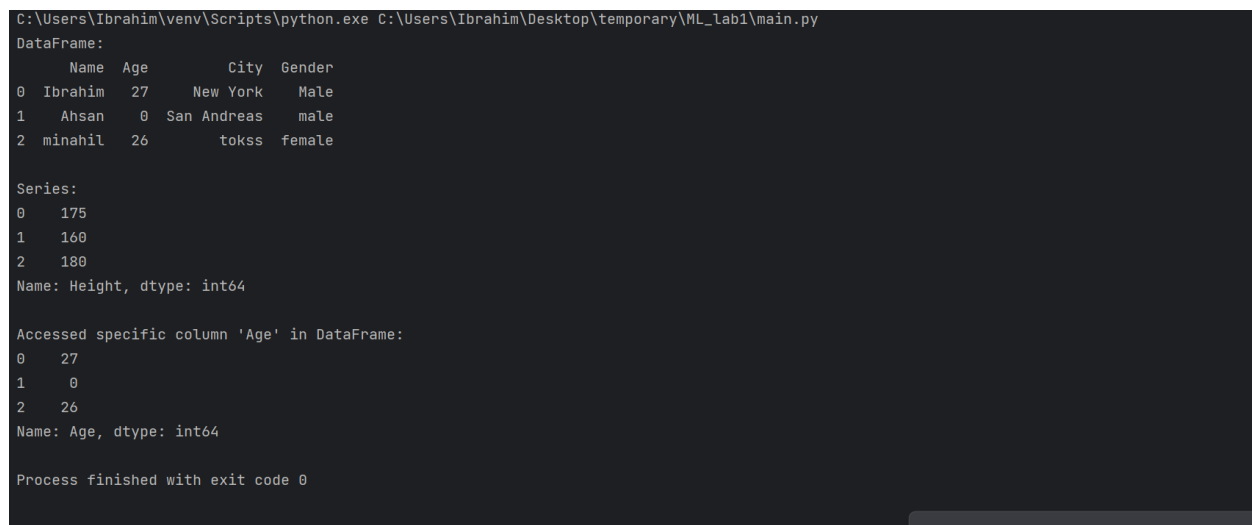
Code:



```
1 #21-CP-26
2 import pandas as pd
3 # a. Create a Pandas DataFrame from a dictionary.
4 data = {'Name': ['Ibrahim', 'Ahsan', 'minahil'],
5         'Age': [27, 00, 26],
6         'City': ['New York', 'San Andreas', 'tokss']}
7 df = pd.DataFrame(data)
8
9 # b. Create a Pandas Series from a list.
10 heights = [175, 160, 180]
11 heights_series = pd.Series(heights, name='Height')
12
13 # c. Access a specific column in a DataFrame.
14 ages_column = df['Age']
15
16 # d. Add a new column to an existing DataFrame.
17 df['Gender'] = ['Male', 'male', 'female']
18 print("DataFrame:")
19 print(df)
20 print("\nSeries:")
21 print(heights_series)
22 print("\nAccessed specific column 'Age' in DataFrame:")
23 print(ages_column)
```

Unable to save settings

Output:



```
C:\Users\Ibrahim\env\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
DataFrame:
   Name  Age   City  Gender
0 Ibrahim  27  New York   Male
1  Ahsan   0 San Andreas  male
2 minahil  26   tokss  female

Series:
0    175
1    160
2    180
Name: Height, dtype: int64

Accessed specific column 'Age' in DataFrame:
0    27
1     0
2    26
Name: Age, dtype: int64

Process finished with exit code 0
```

Unable to save settings

TASK 3

3. Perform below Tasks

- Handle missing values in a DataFrame.
- Remove duplicate rows from a DataFrame.
- Rename columns in a DataFrame.

Code:

```
ML_lab1 C:\Users\Ibrahim\Des
main.py
> External Libraries
> Scratches and Consoles

25 import pandas as pd
26 data = {'Name': ['EBERA', 'ABRA', 'IBRA', 'SHIBRA', 'LINCON'],
27         'Age': [25, None, 22, 30, 25],
28         'City': ['ISB', 'San', 'SGD', 'BHWL', 'ISB']}
29 df = pd.DataFrame(data)
30
31 # a Handle missing values in a DataFrame.
32 df_filled = df.fillna({'Age': df['Age'].mean()}) # FILLING by mean of the column
33
34 # b Remove duplicate rows from a DataFrame.
35 df_no_duplicates = df.drop_duplicates()
36
37 # c Rename columns in a DataFrame.
38 df_renamed = df.rename(columns={'Name': 'Full Name', 'Age': 'Years'})
39
40 print("Original DataFrame:")
41 print(df)
42 print("\nDataFrame with missing values handled:")
43 print(df_filled)
44 print("\nDataFrame with duplicate rows removed:")
45 print(df_no_duplicates)
46 print("\nDataFrame with columns renamed:")
47 print(df_renamed)
```

Unable to save settings

Output

C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py

Original DataFrame:

	Name	Age	City
0	EBERA	25.0	ISB
1	ABRA	NaN	San
2	IBRA	22.0	SGD
3	SHIBRA	30.0	BHWL
4	LINCON	25.0	ISB

DataFrame with missing values handled:

	Name	Age	City
0	EBERA	25.0	ISB
1	ABRA	25.5	San
2	IBRA	22.0	SGD
3	SHIBRA	30.0	BHWL
4	LINCON	25.0	ISB

DataFrame with duplicate rows removed:

	Name	Age	City
0	EBERA	25.0	ISB
1	ABRA	NaN	San
2	IBRA	22.0	SGD
3	SHIBRA	30.0	BHWL
4	LINCON	25.0	ISB


```
DataFrame with columns renamed:
```

	Full Name	Years	City
0	EBERA	25.0	ISB
1	ABRA	NaN	San
2	IBRA	22.0	SGD
3	SHIBRA	30.0	BHWL
4	LINCON	25.0	ISB

```
Process finished with exit code 0
```

TASK 4

4. Do as Directed (Numpy)

- Create a 1D and 2D NumPy array.
- Reshape an array. c. Slice and index arrays.
- Perform element-wise operations (addition, subtraction, multiplication, etc.) on arrays.
- Use NumPy functions like `np.sum()`, `np.mean()`, and `np.max()` on arrays.
- Perform matrix multiplication.
- Calculate mean, median, and standard deviation of an array.
- Calculate correlation and covariance between two arrays.

Code: a-d

```
Project ▾  main.py ×  
ML_lab1 C:\Users\Ibrahim\Des  
main.py  
External Libraries  
Scratches and Consoles  
49 import numpy as np  
50 # a 1D and 2D NumPy array.  
51 array_1d = np.array([1, 2, 3, 4, 5])  
52 array_2d = np.array([[1, 2, 3], [4, 5, 6]])  
53  
54 # b Reshape an array.  
55 reshaped_array = array_1d.reshape(5, 1)  
56  
57 # c Slice and index arrays.  
58 sliced_array = array_1d[1:4]  
59 indexed_element = array_2d[1, 2]  
60  
61 # d Perform element-wise operations (addition, subtraction, multiplication, etc.) on arrays.  
62 array_addition = array_1d + 2  
63 array_multiplication = array_2d * 3  
64 print("1D Array:")  
65 print(array_1d)  
66 print("\n2D Array:")  
67 print(array_2d)  
68 print("\nReshaped Array:")  
69 print(reshaped_array)  
70 print("\nSliced Array:")  
71 print(sliced_array)
```

```
print(sliced_array)  
print("\nIndexed Element:")  
print(indexed_element)  
print("\nArray Addition:")  
print(array_addition)  
print("\nArray Multiplication:")  
print(array_multiplication)
```

```
C:\Users\Ibrahim\venv\Scripts\python.exe C:\Users\Ibrahim\Desktop\temporary\ML_lab1\main.py
```

```
1D Array:
```

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

```
2D Array:
```

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

```
 [4 5 6]]
```

```
Reshaped Array:
```

```
[[1]
```

```
 [2]
```

```
 [3]
```

```
 [4]
```

```
 [5]]
```

```
Sliced Array:
```

```
[2 3 4]
```

```
Indexed Element:
```

```
6
```

```
Array Addition:
```

```
[3 4 5 6 7]
```

! Unable to save
Failed to save

```
Array Addition:
```

```
[3 4 5 6 7]
```

```
Array Multiplication:
```

```
[[ 3  6  9]
```

```
 [12 15 18]]
```

```
Process finished with exit code 0
```

Code: e-h

```
# sum ,mean function    part e
array_sum = np.sum(array_2d)
array_mean = np.mean(array_2d)
array_max = np.max(array_1d)

# f matrix multiplication.
matrix_a = np.array([[1, 2], [3, 4]])
matrix_b = np.array([[5, 6], [7, 8]])
matrix_multiplication = np.dot(matrix_a, matrix_b)

# g. mean, median tandard deviation of an array.
array_mean_value = np.mean(array_1d)
array_median_value = np.median(array_1d)
array_std_deviation = np.std(array_1d)

# h. correlation and covariance b/w arrays.
array_x = np.array([1, 2, 3, 4, 5])
array_y = np.array([5, 4, 3, 2, 1])
correlation = np.corrcoef(array_x, array_y)
covariance = np.cov(array_x, array_y)
print("\nArray Sum:")
```

```
covariance = np.cov(array_x, array_y)
print("\nArray Sum:")
print(array_sum)
print("\nArray Mean:")
print(array_mean)
print("\nMaximum Value in Array:")
print(array_max)
print("\nMatrix Multiplication:")
print(matrix_multiplication)
print("\nArray Mean Value:")
print(array_mean_value)
print("\nArray Median Value:")
print(array_median_value)
print("\nArray Standard Deviation:")
print(array_std_deviation)
print("\nCorrelation Matrix:")
print(correlation)
print("\nCovariance Matrix:")
print(covariance)
```

Array Sum:

21

Array Mean:

3.5

Maximum Value in Array:

5

Matrix Multiplication:

[[19 22]

[43 50]]

Array Mean Value:

3.0

Array Median Value:

3.0

Array Standard Deviation:

1.4142135623730951

Correlation Matrix:

[[1. -1.]

[-1. 1.]]

Covariance Matrix:

[[2.5 -2.5]