**Laboratory Practical File**

# on

# Fundamentals of Machine Learning [CSE313]

**Logo

Description automatically generated**

# Semester-5

# Department of Artificial Intelligence

#### Amity School of Engineering and Technology

#### Amity University, Noida, Uttar Pradesh, India

**Submitted by:**

Janya Verma

A023119820023

5AI-1

**Submitted to:**

Dr. Monika Arora

**Table of Contents**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S. No. | Category of Assignment | Code | Exp.  No. | Name of Experiment | Date of Allotment of experiment | Date of Evaluation | Max.  Marks | Marks obtained | Signature of Faculty |
| 1. | **Mandatory Experiment\*** | **LR (10)** | **1** | Perform basic operation on matrices (addition, subtraction, multiplication) and display specific rows and columns of a matrix. | 20-07-22 | 03-08-22 |  |  |  |
| 2. | **Mandatory Experiment\*** | **2** | Perform other matrix operations like converting a matrix's data to absolute values, taking the negative of matrix's values, adding/removing rows/columns, finding maximum and minimum values of a matrix in a row/column, finding the sum of all the elements in a matrix, and concatenating two matrices. | 03-08-22 |  |  |  |  |
| 3. | **Mandatory Experiment\*** | **3** |  |  |  |  |  |  |
| 4. | **Mandatory Experiment\*** | **4** |  |  |  |  |  |  |
| 5. | **Mandatory Experiment\*** | **5** |  |  |  |  |  |  |
| 6. | **Mandatory Experiment\*** | **6** |  |  |  |  |  |  |
| 7 | **Mandatory Experiment\*** | **7** |  |  |  |  |  |  |
| 8. | **Mandatory Experiment\*** | **8** |  |  |  |  |  |  |
| 9. | **Mandatory Experiment\*** | **9** |  |  |  |  |  |  |
| 10. | **Mandatory Experiment\*** | **10** |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 11. | **Design Based Open Ended experiment\*\*** | **PR (10)** |  |  |  |  |  |  |  |
| 12. | **Viva** | **Viva (5)** |  |  |  |  |  |  |  |

# Experiment-1

#### Aim

#### Perform basic operation on matrices (addition, subtraction, multiplication) and display specific rows and columns of a matrix.

#### Software Used

OS (Windows 11), IDE (Visual Studio Code (1.69.2)), and Programming Language (Python (3.10.5))

#### Code

|  |
| --- |
| *import* numpy *as* np  *#Addition on matrices*  m1 = np.random.randn(3,4)  m2 = np.random.randn(3,4)  m3 = np.add(m1,m2) *#m3 = m1 + m2 will also work*  print("Random matrix addition result:",m3)  *#Subtraction on matrices*  m4 = np.random.randn(3,4)  m5 = np.random.randn(3,4)  m6 = np.subtract(m4,m5) *#m6 = m4 - m5 will also work*  print("Random matrix subtraction result:",m6)  *#Multipliction on matrices*  m7 = np.random.randn(3,4)  m8 = np.random.randn(4,3)  m9 = np.dot(m7,m8)  print("Random matrix multiplication result:",m9)  *#Matrix indexing*  arr = np.arange(20)  print("The original array is:",arr)  print("arr[2]",arr[5]) |

**Result**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application, Word

Description automatically generated**

**Conclusion**

The objective of this experiment is efficiently achieved by using the python library “NumPy”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| **Total** | 6 |  |  |

# Experiment-2

#### Aim

#### Perform other matrix operations like converting a matrix's data to absolute values, taking the negative of matrix's values, adding/removing rows/columns, finding maximum and minimum values of a matrix in a row/column, finding the sum of all the elements in a matrix, and concatenating two matrices.

#### Software Used

OS (Windows 11), IDE (Visual Studio Code (1.69.2)), and Programming Language (Python (3.10.5))

#### Code

|  |
| --- |
| import numpy as np#Convertig matrix data to absolute valuesa1 = np.random.randint(-100,100,(3, 3))print("The given array is:\n",a1)print("The absolute value of the above array is:\n",np.absolute(a1))a2 = np.random.randint(-100,100,(7, 6))print("The given array is:\n",a2)print("The absolute value of the above array is:\n",np.absolute(a2))#Taking the negative of a matrix's valuesa3 = np.random.randint(-100,100,(3,3))print("The given array is:\n",a3)print("The negative of the values of above mtrix are:\n",np.negative(a3))a4 = np.random.randint(-100,100,(7, 6))print("The given array is:\n",a4)print("The absolute value of the above array is:\n",np.negative(a4))#Adding Rows to a matrixa5 = np.random.randint(-100,100,(3,3))r1 = np.random.randint(-100,100,(1,3))add\_r1 = np.vstack([a5,r1])print("The given array is:\n",a5)print("The above matrix after an addition of a row:\n",add\_r1)a6 = np.random.randint(-100,100,(7,6))r2 = np.random.randint(-100,100,(1,6))add\_r2 = np.vstack([a6,r2])print("The given array is:\n",a6)print("The above matrix after an addition of a row:\n",add\_r2)#Removing a row from a matrixa7 = np.random.randint(-100,100,(3,3))a7\_r\_del = np.delete(a7,2,0)print("The given matrix is:\n",a7)print("The given matrix after deletion of row number 2:\n",a7\_r\_del)a8 = np.random.randint(-100,100,(7,6))a8\_r\_del = np.delete(a8,2,0)print("The given matrix is:\n",a8)print("The given matrix after deletion of row number 2:\n",a8\_r\_del)#Removing a column from a matrixa9 = np.random.randint(-100,100,(3,3))a9\_c\_del = np.delete(a9,2,1)print("The given matrix is:\n",a9)print("The given matrix after deletion of column number 2:\n",a9\_c\_del)a10 = np.random.randint(-100,100,(7,6))a10\_c\_del = np.delete(a10,2,1)print("The given matrix is:\n",a10)print("The given matrix after deletion of column number 2:\n",a10\_c\_del)#Finding maximum value in a row of a matrixa11 = np.random.randint(-100,100,(3,3))r\_max = np.max(a11,1)[1]print("The given array is:\n",a11)print("The maximum element from row number 1 of the above matrix:\n",r\_max)a12 = np.random.randint(-100,100,(7,6))r\_max = np.max(a12,1)[1]print("The given array is:\n",a12)print("The maximum element from row number 1 of the above matrix:\n",r\_max)#Finding minimum value in a row of a matrixa13 = np.random.randint(-100,100,(3,3))r\_min = np.min(a13,1)[1]print("The given array is:\n",a13)print("The minimum element from row number 1 of the above matrix:\n",r\_min)a14 = np.random.randint(-100,100,(7,6))r\_min = np.min(a14,1)[1]print("The given array is:\n",a14)print("The minimum element from row number 1 of the above matrix is:\n",r\_min)#Finding maximum value in a column of a matrixa15 = np.random.randint(-100,100,(3,3))c\_max = np.max(a15,0)[1]print("The given array is:\n",a15)print("The maximum element from column number 1 of the above matrix:\n",c\_max)a16 = np.random.randint(-100,100,(7,6))c\_max = np.max(a16,0)[1]print("The given array is:\n",a16)print("The maximum element from column number 1 of the above matrix:\n",c\_max)#Finding minimum value in a column of a matrixa17 = np.random.randint(-100,100,(3,3))c\_min = np.min(a17,0)[1]print("The given array is:\n",a17)print("The minimum element from column number 1 of the above matrix:\n",c\_min)a18 = np.random.randint(-100,100,(7,6))c\_min = np.min(a18,0)[1]print("The given array is:\n",a18)print("The minimum element from column number 1 of the above matrix:\n",c\_min)#Finding sum of all elements of a matrixa19 = np.random.randint(-100,100,(3,3))sum = np.sum(a19)print("The given matrix is:\n",a19)print("Sum of all elements from above matrix is:\n",sum)a20 = np.random.randint(-100,100,(7,6))sum = np.sum(a20)print("The given matrix is:\n",a20)print("Sum of all elements from above matrix is:\n",sum)#Concatenation of two matricesa21 = np.random.randint(-100,100,(3,3))a22 = np.random.randint(-100,100,(3,3))concat\_mat = np.concatenate((a21,a22))print("The given matrix is:\n",a21,"\n\n",a22)print("Concatenation of above two matrices is:\n",concat\_mat) |

**Result**

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, text, application

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated

**Conclusion**

The objective of this experiment is efficiently achieved by using the python library “NumPy”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| **Total** | 6 |  |  |

# Experiment-3

#### Aim

#### Create various types of plots/charts like histograms, pie chart, scatter plot, plot based on sine/cosine function(s) based on data from a matrix. Further, label different axes in a plot and data in a plot.

#### Software Used

OS (Windows 11), IDE (Visual Studio Code (1.69.2)), and Programming Language (Python (3.10.5))

#### Code

|  |
| --- |
| # Importing required librariesimport numpy as npimport matplotlib.pyplot as plt# Generating Sine Function Graph# Generating x valuesAngle = np.arange(-2\*np.pi,2\*np.pi,np.pi/1000)# Generating y values at each x valueAmplitude = np.sin(Angle)# Plotting angle vs amplitudeplt.plot(Angle,Amplitude)# Setting title for the plot in blue colorplt.title('Sine Wave',color='b')# Setting x-axis label for the plotplt.xlabel('Angle in Radians'+r'$\rightarrow$')# Setting y-axis label for the plotplt.ylabel('Amplitude'+r'$\rightarrow$')# Showing gridplt.grid()# Highlighting axis at x=0 and y=0plt.axhline(y=0,color='k')plt.axvline(x=0,color='k')# Generating Cosine Function Graph# Generating x valuesAngle = np.arange(-2\*np.pi,2\*np.pi,np.pi/1000)# Generating y values at each x valueAmplitude = np.cos(Angle)# Plotting angle vs amplitudeplt.plot(Angle,Amplitude)# Setting title for the plot in blue colorplt.title('Cosine Wave',color='b')# Setting x-axis label for the plotplt.xlabel('Angle in Radians'+r'$\rightarrow$')# Setting y-axis label for the plotplt.ylabel('Amplitude'+r'$\rightarrow$')# Showing gridplt.grid()# Highlighting axis at x=0 and y=0plt.axhline(y=0,color='k')plt.axvline(x=0,color='k')# Creating Bar Graph# creating the datasetdata = {'C':20, 'C++':15, 'Java':30,        'Python':35}courses = list(data.keys())values = list(data.values())fig = plt.figure(figsize = (10, 5))# creating the bar plotplt.bar(courses, values, color ='blue',        width = 0.4)plt.xlabel("Courses offered"+r'$\rightarrow$')plt.ylabel("No. of students enrolled"+r'$\rightarrow$')plt.title("Students enrolled in different courses")plt.show()# Creating Histogramx = np.random.normal(170, 10, 250)plt.hist(x,color='pink')plt.xlabel("Heigh in centimeters"+r'$\rightarrow$')plt.ylabel("Number of students with corresponding height"+r'$\rightarrow$')plt.title("Student and their Heights")plt.show()# Creating Scatter Plot# Generating x valuesAngle = np.arange(-2\*np.pi,2\*np.pi,np.pi/20)#Generating y values at each x valueAmplitude\_Cosine = np.cos(Angle)Amplitude\_Sine = np.sin(Angle)# Scatter plotting angle vs amplitudeplt.scatter(Angle,Amplitude\_Cosine,label='cos(x)')plt.scatter(Angle,Amplitude\_Sine,label='sin(x)')# Setting title for the plot in blue colorplt.title('Trigonometric Functions',color='b')# Setting x-axis label for the plotplt.xlabel('Angle in Radians'+r'$\rightarrow$')# Setting y-axis label for the plotplt.ylabel('Amplitude'+r'$\rightarrow$')# Showing gridplt.grid()#Highlighting axis at x=0 and y=0plt.axhline(y=0,color='k')plt.axvline(x=0,color='k')# Legendplt.legend()# Creating Pie Charty = np.random.randint(10,90,(5))mylabels = ["A","B","C","D","E"]plt.pie(y,labels=mylabels,autopct='%1.1f%%')plt.show() |

#### Result

#### Graphical user interface, text, application, email Description automatically generated

#### Graphical user interface, application, Word Description automatically generated

#### Graphical user interface, application, Word Description automatically generated

#### Graphical user interface, application, Word Description automatically generated

#### Graphical user interface, application, Word Description automatically generated

#### A picture containing graphical user interface Description automatically generated

#### Graphical user interface, application, Word Description automatically generated

#### 

#### Conclusion

#### All the objectives of mentioned under the aim are achieved by using the following python libraries:

#### NumPy

#### Matplotlib

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| **Total** | 6 |  |  |