|  |  |
| --- | --- |
| MONO | **DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING****NATIONAL INSTITUTE OF TECHNOLOGY PATNA** Ashok Raj Path, PATNA 800 005 (Bihar), India |
|  Phone No.: 0612 – 2372715, 2370419, 2370843, 2371929, 2371930, 2371715 Fax – 0612- 2670631 Website: [www.nitp.ac.in](http://www.nitp.ac.in/) |

***CSL5402 Artificial Intelligence Lab***

**L-T-P-Cr: 0-0-3-1**

**Pre-requisites:** Enrollment in “Machine Learning” course and prior programming experience in python.

**Objectives/Overview:**

* To implement common pre-processing steps in machine learning on real-world datasets.
* To implement/use common unsupervised, unsupervised, reinforcement learning methods on real-world datasets.
* To implement/use common dimensionality reduction techniques and ensemble learning methods on real-world datasets.

**Course Outcomes:**

At the end of the course, a student should:

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Outcome** | **Mapping to POs** |
|  | Be able to interface a database with Python | PO3, PO4 |
|  | Be able to preprocess real life datasets using Python | PO1, PO3, PO4 |
|  | Be able to implement/use Naïve Bayes and Logistic regression classifier on simple real world datasets.  | PO2, PO3, PO4 |
|  | Be able to implement/use linear regression and K-means algorithm on simple real world datasets.  | PO1, PO2, PO3, PO4 |
|  | Be able to implement/use LDA and PCA dimensionality reduction techniques for classifying simple real world dataset.  | PO1, PO2, PO3, PO4 |
|  | Be able to implement/use ensemble learning techniques for classifying simple real world datasets programmatically.  | PO1, PO2, PO3, PO4 |
|  | Be able to programmatically extract features in case of image processing and text mining for classifying simple real world datasets.  | PO1, PO2, PO3, PO4, PO12 |
|  | Be able to programmatically implement/use multi-label classification problem on simple real world datasets.  | PO1, PO2, PO3, PO4 |
|  | Prepare a report in prescribed format of the lab experiments carried out.  | PO8 |

**List of Experiments:**

1. WAP to implement a simple calculator. The output should be something like:

Select operation.

1.Add

2.Subtract

3.Multiply

4.Divide

Enter choice(1/2/3/4): 3

Enter first number: 15

Enter second number: 14

15 \* 14 = 210

1. A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400. For example,

2017 is not a leap year

1900 is a not leap year

2012 is a leap year

2000 is a leap year

WAP to check if a year is leap. Use % operator

1. WAP to find factorial of a number. Use range() function.
2. WAP to shuffle a deck of cards. The output should be like:

You got:

5 of Heart

1 of Heart

8 of Spade

12 of Spade

4 of Spade

1. Accept a string from user consisting of multiple English language words. Now split the words and print them in alphabetical order. You can use string split() method (look up online)
2. Generate 1000 random numbers in the range 10-100. Write them to a csv file with 10 comma separated numbers in each line resulting in 100 such lines. This can be treated as values of 10 random variables each sampled 100 times. Close the file and open it again in read mode. Read the values. Find maximum and minimum value of each random variable. Find the variance of each random variable. Print the co-variance matrix of all 10 random variables.
3. Create a table in MySQL representing bank account of a person. Write a menu driven program that gives user an option to deposit, withdraw and check account balance.
4. Create a class representing statistics of a cricket player. Enter data for a few players. Now print the names of players in order of (a) decreasing strike rate, (b) increasing batting average.
5. Accept 9 values from user and store it in a 3x3 matrix in row major order. Check if the resulting matrix is orthogonal. You can find the definition of an orthogonal matrix from: <https://en.wikipedia.org/wiki/Orthogonal_matrix>
6. Mini project allocation
7. Write a program to solve 8 queens problem
8. Solve any problem using depth first search.
9. Solve any problem using best first search.
10. Solve 8-puzzle problem using best first search
11. Solve Robot (traversal) problem using means End Analysis
12. Solve traveling salesman problem.

Course instructor can add experiments to the above list and/or modify some of the experiments in the above list depending upon course contents covered and examples used in the corresponding theoretical course.