BCA (HONS) 6th SEMESTER
DISCIPLINE SPECIFIC ELECTIVE COURSE (DSE)
OPTION - I

BCA621D1A: MACHINE LEARNING

CREDITS: THEORY: 4; PRACTICAL: 2
MAX. MARKS: THEORY: 60; PRACTICAL: 30
MIN. MARKS: THEORY: 24; PRACTICAL: 12

UNIT I

Introduction: **(8 Lectures)**

Concept of Machine Learning, Applications of Machine Learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning, Statistical Learning: Bayesian Method, The Naive Bayes Classifier

Softwares for Machine Learning and Linear Algebra Overview: **(7 Lectures)**

Plotting of Data, Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using available tool such as MATLAB.

UNIT II

Linear Regression: **(15 Lectures)**

Prediction using Linear Regression, Gradient Descent, Linear Regression with one variable, Linear Regression with multiple variables, Polynomial Regression, Feature Scaling/Selection.

UNIT III

Logistic Regression: **(8 Lectures)**

Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one variable and with multiple variables.

Regularization: **(7 Lectures)**

Regularization and its utility: The problem of Overfitting Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance.

UNIT IV

Neural Networks: **(15 Lectures)**

Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Multilayer Perceptrons, Multiclass Representation, Backpropagation Algorithm.

Suggested Books:

1. Ethem Alpaydin, "Introduction to Machine Learning" 2nd Edition, The MIT Press, 2009.
2. Tom M. Mitchell, "Machine Learning", First Edition by Tata McGraw-Hill Education, 2013.
3. Christopher M. Bishop, "Pattern Recognition and Machine Learning" by Springer, 2007.
4. Mevin P. Murphy, "Machine Learning: A Probabilistic Perspective" by The MIT Press, 2012.

LAB: MACHINE LEARNING (CREDITS: 4; THEORY: 60 LECTURES)

For practical Labs for Machine Learning, students may use softwares like MABLAB/Octave or Python. For later exercises, students can create/use their own datasets or utilize datasets from online repositories like UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/).

1. Perform elementary mathematical operations in Octave/MATLAB like addition, multiplication, division and exponentiation.

2. Perform elementary logical operations in Octave/MATLAB (like OR, AND, Checking for Equality, NOT, XOR).

3. Create, initialize and display simple variables and simple strings and use simple formatting for variable.

4. Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.

5. Use command to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current scope.

6. Perform basic operations on matrices (like addition, subtraction, multiplication) and display specific rows or columns of the matrix.

7. Perform other matrix operations like converting matrix data to absolute values, taking the negative of matrix values, adding/removing rows/columns from a matrix, finding the maximum or minimum values in a matrix or in a row/column, and finding the sum of some/all elements in a matrix.

8. Create various type of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further label different axes in a plot and data in a plot.

9. Generate different subplots from a given plot and color plot data.

10. Use conditional statements and different type of loops based on simple example/s.

11. Perform vectorized implementation of simple matrix operation like finding the transpose of a matrix, adding, subtracting or multiplying two matrices.

12. Implement Linear Regression problem. For example, based on a dataset comprising of existing set of prices and area/size of the houses, predict the estimated price of a given house.

13. Based on multiple features/variables perform Linear Regression. For example, based on a number of additional features like number of bedrooms, servant room, number of balconies, number of houses of years a house has been built - predict the price of a house.

14. Implement a classification/ logistic regression problem. For example, based on different features of student's data, classify, whether a student is suitable for a particular activity. Based on the available dataset, a student can also implement another classification problem like checking whether an email is spam or not.

15. Use some function for regularization of dataset based on problem 14.

16. Use some function for neural networks, like Stochastic Gradient Descent or backpropagation - algorithm to predict the value of a variable based on the dataset of problem 14.

BCA 6th SEMESTER
DISCIPLINE SPECIFIC ELECTIVE COURSE (DSE)
OPTION - II

BCA620D1B: PYTHON PROGRAMMING

CREDITS: THEORY: 4; PRACTICAL: 2
MAX. MARKS: THEORY: 60; PRACTICAL: 30
MIN. MARKS: THEORY: 24; PRACTICAL: 12

UNIT I: Introduction to Python: (3 Lectures)

Structure of a Python Program, Atoms, Identifiers and keywords, Elements of Python; Python Interpreter, Using Python as calculator, Python shell, Indentation.

Builtin Data Types: (10 Lectures)

Numbers: Literal representations of numbers, Operators for numbers, Methods on numbers; Lists: Literal representation of lists, Operators on lists, Methods on lists, List comprehensions; Strings: Characters, Operators on strings, Methods on strings, Raw strings, Unicode strings; Dictionaries: Literal representation of dictionaries, Operators on dictionaries, Methods on dictionaries; Files; None; The booleans True and False.

Operators: (2 Lectures)

Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator).

UNIT II: Statements: (15 Lectures)

Assignment statement; Input and Output Statements, Print statement; Control statements: if: statement; for: statement; while: statement (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.). Exceptions and the try: except: and raise statements.

UNIT III: Functions (15 Lectures)

Optional arguments and default values; Passing functions as arguments; Extra args and keyword args; Order of arguments (positional, extra, and keyword args; Functions and duck typing and polymorphism; Recursive functions; Generators and iterators.

UNIT IV: Object Oriented Programming and classes (15 Lectures)

Constructor; Inheritance Implementing a subclass; Classes and polymorphism; Recursive calls to methods; Class variables, class methods, and static methods; Decorators for class method and static method.

Reference Books:

1. T. Budd, Exploring Python, TMH, 1 st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers, How to think like a computer scientist: learning with Python, Freely available online.2012
4. http://docs.python.Org/3/tutorial/index.html
5. http://interactivepython.org/courselib/static/pythonds
6. http://www.ibiblio.org/g2swap/byteofpython/read/

LAB: PYTHON PROGRAMMING (CREDITS: 2; LECTURES: 60)

Section: A (Simple programs):

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:

   Grade A: Percentage >=80 Grade B:
   Percentage>=70 and <80 Grade C: Percentage>=60
   and

   <70 Grade D:

   Percentage>=40 and <60 Grade E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms: 1 - 2/2! + 3/3!.....n/n!
7. WAP to calculate the sum and product of two compatible matrices.

Section: B (Visual Python):

All the programs should be written using user defined functions, wherever possible.

1. Write a menu-driven program to create mathematical 3D objects
   I. curve
   II. sphere
   III. Cone
   IV. arrow
   V. ring
   VI. cylinder
2. WAP to read n integers and display them as a histogram.
3. WAP to display sine, cosine, polynomial and exponential curves.
4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.
5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula m=60/(t+2), where t is the time in hours. Sketch a graph for t vs. m, where t>=0.
6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:

   P(t) = (15000(l+t))/(15+e)

   Where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.

7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:
   I. velocity wrt time (v=u+at)
   II. distance wrt time (s=u*t+0.5*a*t*t)
   III. distance wrt velocity (s=(v*v-u*u)/2*a)

   WAP to show a ball bouncing between 2 walls. (Optional)