MACHINE LEARNING

SYLLABUS:

Unit-I: Introduction- Artificial Intelligence, Machine Learning, Deep learning, Types of Machine Learning Systems, Main Challenges of Machine Learning.
Statistical Learning: Introduction, Supervised and Unsupervised Learning, Training and Test Loss, Tradeoffs in Statistical Learning, Estimating Risk Statistics, Sampling distribution of an estimator, Empirical Risk Minimization.

Unit-II: Supervised Learning(Regression/Classification):Basic Methods: Distance based Methods, Nearest Neighbours, Decision Trees, Naive Bayes, **Linear Models:** Linear Regression, Logistic Regression, Generalized Linear Models, Support Vector Machines, **Binary Classification:** Multiclass/Structured outputs, MNIST, Ranking.

Unit-III: Ensemble Learning and Random Forests: Introduction, Voting Classifiers, Bagging and Pasting, Random Forests, Boosting, Stacking.
Support Vector Machine: Linear SVM Classification, Nonlinear SVM Classification SVM Regression, Naïve Bayes Classifiers.
.
Unit-IV: Unsupervised Learning Techniques: Clustering, K-Means, Limits of K-Means, Using Clustering for Image Segmentation, Using Clustering for Preprocessing, Using Clustering for Semi-Supervised Learning, DBSCAN, Gaussian Mixtures.
Dimensionality Reduction: The Curse of Dimensionality, Main Approaches for Dimensionality Reduction, PCA, Using Scikit-Learn, Randomized PCA ,Kernel PCA.

Unit-V: Neural Networks and Deep Learning: Introduction to Artificial Neural Networks with Keras, Implementing MLPs with Keras, Installing TensorFlow 2, Loading and Preprocessing Data with TensorFlow.

BIG DATA ANALYTICS

SYLLABUS:

UNIT-I: Data structures in Java: Linked List, Stacks, Queues, Sets, Maps; Generics: Generic classes and Type parameters, Implementing Generic Types, Generic Methods, Wrapper Classes, Concept of Serialization

UNIT-II: Working with Big Data: Google File System, Hadoop Distributed File System (HDFS) Building blocks of Hadoop (Namenode, Datanode, Secondary Namenode, Job Tracker, Task Tracker), Introducing and Configuring Hadoop cluster (Local, Pseudo-distributed mode, Fully Distributed mode), Configuring XML files.

UNIT-III: Writing MapReduce Programs: A Weather Dataset, Understanding Hadoop API for Map Reduce Framework (Old and New), Basic programs of Hadoop Map Reduce: Driver code, Mapper code, Reducer code, Record Reader, Combiner, Practitioner

UNIT-IV: StreamMemory and Spark: Introduction to Streams Concepts– Stream Data Model and Architecture , Stream computing, Sampling Data in a Stream , Filtering Streams ,Counting Distinct Elements in a Stream , Introduction to Spark Concept , Spark Architecture and components , Spark installation , Spark RDD(Resilient Distributed Dataset) – Spark RDD operations.

UNIT-V: **Pig:** Hadoop Programming Made Easier Admiring the Pig Architecture, Going with the Pig Latin Application Flow, Working through the ABCs of Pig Latin, Evaluating Local and Distributed Modes of Running Pig Scripts, Checking out the Pig Script Interfaces, Scripting with Pig Latin.

Applying Structure to Hadoop Data with Hive: Saying Hello to Hive, Seeing How the Hive is Put Together, Getting Started with Apache Hive, Examining the Hive Clients, Working with Hive Data Types, Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works, Querying and Analysing data

CRYPTOGRAPHY & NETWORK SECURITY

SYLLABUS:

UNIT–I: Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks, Security services, Security Mechanisms, A model for Network Security Cryptography. Classical Encryption Techniques-symmetric cipher model, Substitution techniques, Transposition techniques, Rotor Machines, Stegnography.
Introduction to Mathematics of Cryptography: Integer Arithmetic, Euclidean Algorithm, The Extended Euclidean Algorithm, Linear Diophantine Equations, Modular Arithmetic, Inverses, Matrices, Linear Congruence.

UNIT–II: Mathematics of Symmetric Cryptography: Algebraic Structures-Groups, Rings, Fields, GF(2^n) fields, Polynomials. **Mathematics of Asymmetric cryptography:** Primes, Checking For Primness, Eulers phi-functions, Fermat's Little Theorem, Euler's Theorem, Generating Primes, Primality Testing, Factorization, Chinese Remainder Theorem, Quadratic Congruence, Exponentiation And Logarithm.

UNIT–III: Symmetric key Ciphers: Block Cipher principles, DES, AES, Blowfish, IDEA, Block cipher operation, Stream ciphers: RC4, RC5
Asymmetric key Ciphers: Principles of public key cryptosystems, RSA algorithm, Diffie-Hellman Key Exchange, Elgamal Cryptographic system, Elliptic Curve Arithmetic, Elliptic Curve Cryptography.

UNIT–IV: Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Requirements and Security, Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithms (SHA)
Message Authentication Codes: Message Authentication Requirements, Message Authentication Functions, Requirements for Message Authentication Codes, Security of MAC'S,MAC'S Based On Hash Functions: HMAC, MAC'S Based On Block Ciphers: DAA And CMAC
Digital Signatures: Digital Signatures, Elgamal Digital Signature Scheme, Elliptic Curve Digital Signature Algorithm, RSA-PSS Digital Signature Algorithm.

UNIT–V: Network and Internet Security: Transport-Level Security: Web Security Considerations, Transport Level Security, HTTPS, SSH.
IP Security: IP Security Overview, IP Security Policy, Encapsulating Security Payload, Authentication Header Protocol.
Electronic-Mail Security: Internet-mail Security, Email Format, Email Threats and Comprehensive Email Security, S/MIME, PGP.

NETWORK PROGRAMMING

SYLLABUS:

UNIT-1: Introduction to Network Programming: Introduction to Network Programming: OSI model, UNIX standards, TCP and UDP & TCP connection establishment and Format, Buffer sizes and limitation, standard internet services, Protocol usage by common internet application Elementary Sockets: Sockets introduction, Elementary TCP sockets.

UNIT-2: TCP client server: Introduction, TCP Echo server functions, Normal startup, terminate and signal handling server process termination, Crashing and Rebooting of server host shutdown of server host.I/O Multiplexing: I/O Models, the select and poll functions, Batch input and buffering, shutdown function.

UNIT-3: UDP and Socket options: Elementary UDP sockets: Introduction UDP Echo server functions, lost datagram, summary of UDP example, Lack of flow control with UDP. **Socket options:** getsockopt and setsockopt functions. Socket states, Generic socket options IPV4 socket options, IPV6 socket options, ICMPV6 socket options and TCP socket options, SCTP socket options, fcntl function.

UNIT-4: Advanced Sockets and Daemon Processes: IPV4 and IPV6 interoperability, introduction, IPV4 client: IPV6 server, IPV6 client: IPV4 Server, IPV6 Address-testing macros. Daemon Processes and inetdSuperserver – Introduction, syslogd Daemon, syslog Function, daemon_init Function, inetd Daemon, daemon_inetd. Advanced I/O functions: Socket timeouts, recv and send functions, ready and writev functions, recvmsg and send msg functions, Ancillary data.

UNIT-5: Broadcasting and Multicasting: Broadcasting introduction, broadcast addresses, unicast versus Broadcast, dg_cli function using broadcasting, race conditions, Multicasting addresses, multicasting versus broadcasting on a LAN, multicasting on a WAN, source-specific multicast, multcast socket options. **Raw Sockets**: Introduction, Raw Socket Creation, Raw Socket Output, Raw Socket Input, Ping Program, Traceroute Program

NATURAL LANGUAGE PROCESSING

SYLLABUS:

UNIT I: INTRODUCTION: Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance.

UNIT II: WORD LEVEL ANALYSIS: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part-of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

UNIT III: SYNTACTIC ANALYSIS: Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing – Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures

UNIT IV: SEMANTICS AND PRAGMATICS: Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

UNIT V: DISCOURSE ANALYSIS AND LEXICAL RESOURCES: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC).

EMPLOYABILITY SKILLS-II

SYLLABUS:

UNIT I:

Numerical ability l: Number system, HCF & LCM, Average, Simplification, Problems on numbers

Numerical ability II: Ratio & Proportion, Partnership, Percentages, Profit & Loss

UNIT II:

Arithmetical ability l: Problems on ages, Time & Work, Pipes & Cistern, Chain Rule.

Arithmetical ability ll: Time & Distance, Problems on boats &Steams, Problems on Trains

UNIT III:

Arithmetical ability lll: Allegation, Simple interest and compound interest, Races & Games of skills, Calendar and Clock,

Logical ability: Permutations and Combination and Probability.

UNIT IV:

Mensuration: Geometry, Areas, Volumes,

Data interpretation: Tabulation, Bar graphs, Pie charts, line graphs

BIG DATA ANALYTICS LAB

Software Requirements:

1. **Hadoop** : https://hadoop.apache.org/release/2.7.6.html
2. **Java** : https://www.oracle.com/java/technologies/javase/javase8u211-later-archive-downloads.html
3. **Eclipse** : https://www.eclipse.org/downloads/

List of Experiments:

Experiment 1: Week 1, 2:

1. Implement the following Data structures in Java
   a) Linked Lists b) Stacks c) Queues d) Set e) Map

Experiment 2: Week 3:

2. (i)Perform setting up and Installing Hadoop in its three operating modes: Standalone, Pseudo distributed, Fully distributed
   (ii)Use web based tools to monitor your Hadoop setup.

Experiment 3: Week 4:

3. Implement the following file management tasks in Hadoop:
   - Adding files and directories
   - Retrieving files
   - Deleting files

   Hint: A typical Hadoop workflow creates data files (such as log files) elsewhere and copies them into HDFS using one of the above command line utilities.

Experiment 4: Week 5:

4. Run a basic Word Count MapReduce program to understand MapReduce Paradigm**.**

Experiment 5: Week 6:

5. Write a map reduce program that mines weather data.
   Weather sensors collecting data every hour at many locations across the globe gather a large volume of log data, which is a good candidate for analysis with Map Reduce, since it is semi structured and record-oriented.

Experiment 6: Week 7:

6. Use MapReduce to find the shortest path between two people in a social graph.
   Hint: Use an adjacency list to model a graph, and for each node store the distance from the original node, as well as a back pointer to the original

node. Use the mappers to propagate the distance to the original node, and the reducer to restore the state of the graph. Iterate until the target node has been reached.

Experiment 7: Week 8:

1. Implement Friends-of-friends algorithm in MapReduce.
   Hint: Two MapReduce jobs are required to calculate the FoFs for each user in a social network .The first job calculates the common friends for each user, and the second job sorts the common friends by the number of connections to your friends.

Experiment 8: Week 9:

2. Implement an iterative PageRank graph algorithm in MapReduce.

Hint: PageRank can be implemented by iterating a MapReduce job until the graph has converged. The mappers are responsible for propagating node PageRank values to their adjacent nodes, and the reducers are responsible for calculating new PageRank values for each node, and for re-creating the original graph with the updated PageRank values.

Experiment 9: Week 10:

3. Perform an efficient semi-join in MapReduce.

Hint: Perform a semi-join by having the mappers load a Bloom filter from the Distributed Cache, and then filter results from the actual MapReduce data source by performing membership queries against the Bloom filter to determine which data source records should be emitted to the reducers.

Experiment 10: Week 11:

4. Install and Run Pig then write Pig Latin scripts to sort, group, join, project, and filter your data.

Experiment 12: Week 12:

5. Install and Run Hive then use Hive to create, alter, and drop databases, tables, views, functions, and indexes

MACHINE LEARNING USING PYTHON LAB

Requirements: Develop the following program using Anaconda/Jupiter/Spider and evaluate ML models.

Experiment-1:
Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.

Experiment-2:
For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.

Experiment-3:
Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Experiment-4:
Exercises to solve the real-world problems using the following machine learning methods: a) Linear Regression b) Logistic Regression c) Binary Classifier

Experiment-5: Develop a program for Bias, Variance, Remove duplicates , Cross Validation

Experiment-6: Write a program to implement Categorical Encoding, One-hot Encoding

Experiment-7:
Build an Artificial Neural Network by implementing the Back propagation algorithm and test the same using appropriate data sets.

Experiment-8:
Write a program to implement k-Nearest Neighbor algorithm to classify the iris data set. Print both correct and wrong predictions.

Experiment-9: Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs

Experiment-10:
Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your dataset.

Experiment-11: Apply EM algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.

Experiment-12: Exploratory Data Analysis for Classification using Pandas or Matplotlib.

Experiment-13:
Write a Python program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set

Experiment-14:
Write a program to Implement Support Vector Machines and Principle Component Analysis

Experiment-15:
Write a program to Implement Principle Component Analysis

CRYPTOGRAPHY AND NETWORK SECURITY LAB

Experiment 1:

1. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should XOR each character in this string with 0 and displays the result.

Experiment 2:

2. Write a C program that contains a string (char pointer) with a value 'Hello world'. The program should AND or and XOR each character in this string with 127 and display the result.

Experiment 3:

3. Write a Java program to perform encryption and decryption using the following algorithms
   a. Ceaser cipher b. Substitution cipher c. Hill Cipher

Experiment 4:

4. Write a C/JAVA program to implement the DES algorithm logic.

Experiment 5:

5. Write a C/JAVA program to implement the Blowfish algorithm logic.

Experiment 6:

6. Write a C/JAVA program to implement the Rijndael algorithm logic.

Experiment 7:

7. Write the RC4 logic in java Using Java cryptography; encrypt the text "H ello world" using Blowfish. Create your own key using Java key tool.

Experiment 8:

8. Write a Java program to implement RSA algorithm.

Experiment 9:

9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.

Experiment 10:

10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

Experiment 11:

11. Calculate the message digest of a text using the MD5 algorithm in JAVA.

Experiment 12:

12. Implementation of few programs using Mathematica

Experiment 13:

13. Implementation of few programs using MatLab

Experiment 14:

14. Implementation of few programs using Maple

MEAN STACK TECHNOLOGIES Module-I

List of Experiments:

Experiment 1: Download and Install Node.js and NPM
Experiment 2: Write a Program to implement the Hello World Server with HTTP Node.js Module
Experiment 3: Write a program to create Calculator Node.js Module with functions adds, subtract & multiply and use the Calculator module in another Node.js file.
Experiment 4: Write a Node.js for File System to perform the following operations

i. Create a File
ii. Read a File
iii. Write to a File
iv. Delete a File

Experiment 5: Create and manage an Employee Database Using Node.js MySQL
Experiment 6: Implement the following in Angular JS

i. Angular Js data binding.
ii. Angular JS directives and Events.
iii. Using angular Js fetching data from MySQL.

Experiment 7: Write a program to implement AngularJS Scope
Experiment 8: Write a program to implement the following using Angular JS

i. Input Validation
ii. Backend Building

Experiment 9: Write a program to include the framework's Stylesheet in the Angular CLI's Configuration

Experiment 10: Write a program to Create an application for Students Record using AngularJS

Experiment 11: Create database, Create collection, insert data, find, find one, sort, limit, skip,distinct, projection

Experiment 12: Develop and demonstrate Invoking data using Mongo DB.

Experiment 13: Create an Online fee payment form using MangoDB