20MCA201

DATA SCIENCE & MACHINE LEARNING

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA201	DATA SCIENCE & MACHINE LEARNING	CORE	3	1	0	4

COURSE OUTCOME

Course Outcomes: After the completion of the course the student will be able to:

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Discuss the fundamental concepts of data science and data visualization techniques.	Level 2: Understand
CO 2	Explain the basics of machine learning and use lazy learning and probabilistic learning algorithms to solve data science problems.	Level 3: Apply
CO 3	Describe decision trees, classification rules & regression methods and how these algorithms can be applied to solve data science problems.	Level 3: Apply
CO 4	Solve data science problems using neural networks and support vector machines.	Level 3: Apply
CO 5	Discuss clustering using k-means algorithm and evaluate & improve the performance of machine learning classification models.	Level 3: Apply

Syllabus

Module 1 (9 Hours)
Introduction to data science, Data science classification, Data science process - Prior knowledge, Data preparation, Modelling, Application, Data exploration - Data sets, Descriptive statistics for univariate and multivariate data Data visualisation – Histogram, Quartile plot, Distribution chart, Scatter plot, Bubble chart, Density chart
Module 2 (9 Hours)
Introduction to machine learning: How machines learn - Data storage, Abstraction, Generalisation, Evaluation, Machine learning in practice - Types of machine learning algorithms. **Lazy learning:** Classification using K-Nearest Neighbour algorithm - Measuring similarity with distance, Choice of k, Preparing data for use with k-NN. **Probabilistic learning:** Understanding Naive Bayes - Conditional probability and Bayes theorem, Naive Bayes algorithm for classification, The Laplace estimator, Using numeric features with Naive Bayes.
Module 3 (9 Hours)
Decision tree learning: Concept of decision tree, Divide and conquer approach, C5.0 Decision tree algorithm, Choosing the best split, Pruning the decision tree. **Classification rules learning:** Concept of classification rules, Separate and conquer approach, The 1R algorithm, Rules from decision trees. **Regression methods:** Concept of regression, Simple linear regression, Ordinary least squares estimation, Correlations, Multiple linear regression.
Module 4 (9 Hours)
Neural network learning: Artificial neurons, Activation functions, Network topology, Training neural networks with backpropagation. **Support vector machines:** Hyperplanes, Classification using hyperplanes, Maximum margin hyperplanes in linearly separable data, Using kernels for non-linear spaces.
Module 5 (9 Hours)
Clustering: The k-means clustering algorithm, Using distance to assign and update clusters, Choosing number of clusters. **Evaluating model performance:** Confusion matrices, Precision and recall, Sensitivity and specificity, Precision and recall, F-measure, ROC curves, Cross validation - K-fold cross validation, Bootstrap sampling. Improving model performance - Bagging, Boosting, Random forests.

20MCA203

DESIGN & ANALYSIS OF ALGORITHMS

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA203	DESIGN & ANALYSIS OF ALGORITHMS	CORE	3	1	0	4

COURSE OUTCOME

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Discuss the basic concepts in computer algorithms and their analysis & design using Divide and Conquer.	Level 2: Understand
CO 2	Explain the concepts of Greedy Strategy and Dynamic Programming to use it in solving real world problems.	Level 3: Apply
CO 3	Explain the Branch & Bound technique, Backtracking technique and Lower bounds.	Level 2: Understand
CO 4	Describe the fundamental concepts of Computational Complexity and Network Flows.	Level 2: Understand
CO 5	Discuss the concepts of Approximation and Randomised Algorithms.	Level 2: Understand

Syllabus

Module 1: (8 Hours)
Review of Algorithm Analysis: Time and Space Complexity, Asymptotic Notations, Recurrence Equations, Solving Recurrence Equations- Substitution method and Iteration method. **Divide and Conquer**: Control Abstraction, Merge Sort, Quick Sort, Matrix Multiplication.
Module 2: (9 Hours)
Greedy Strategy: Control Abstraction, Knapsack Problem, Minimal Spanning Tree Algorithms- Prim's and Kruskal's Algorithm, Job Scheduling with deadlines **Dynamic Programming**: Control Abstraction, Principle of Optimal Substructure, All Pairs shortest path problem, Travelling Salesman Problem, Bellman-Ford Algorithm
Module 3:(7 Hours)
Backtracking: Control Abstraction, N-Queens problem, Sum of Subsets Problem **Branch and Bound**: Control Abstraction, 8- Puzzle problem **Lower Bounds:** The Decision Tree method, Lower Bounds for Comparison based Sort and Searching *(Analysis not required)*
Module 4: (11 Hours)
Complexity Theory: Class P and NP, Polynomial time reductions, Class NP Hard and NP-Complete, Example Problems- Vertex Cover problem, Clique Problem. **Network Flows**: Flow Networks and Network Flow, Max- Flow Min Cut Theorem, Ford Fulkerson method, Bipartite matching (*Analysis not required)*
Module 5: (10 Hours)
Introduction to Approximation Algorithms: Approximation Ratio, 2-approximation algorithm for Vertex Cover problem, Vertex Cover Approximation using Linear Programming and LP Rounding Algorithm. **Introduction to Randomised Algorithms**: Review of Basic Probability, Schwartz-Zippel Lemma and Polynomial Identity Testing, Randomized Quick Sort (*Proof of Expected Worst Case Analysis not required)*

20MCA265

Cloud Computing

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA265	Cloud Computing	ELECTIVE	3	1	0	4

Course Outcome

Course Outcomes: After completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Understand the basic concepts in cloud computing and OpenStack logical architecture	Level 2: Understand
CO 2	Discuss OpenStack cloud controller and common services	Level 3: Apply
CO 3	Compare different OpenStack compute service components and storage types	Level 2: Understand
CO 4	Describe the OpenStack Networking- Connection types and networking services	Level 2: Understand
CO 5	Discuss orchestration, HA and failover in OpenStack	Level 2: Understand

Syllabus

Module 1: Overview of OpenStack (7 Hours)
Introduction to cloud computing, private cloud, public cloud, hybrid cloud architecture. Cloud Services - Infrastructure as a Service, Platform as a Service, Storage as a Service. Designing OpenStack Cloud Architectural Consideration - OpenStack - The new data centre paradigm - OpenStack logical architecture - Nova - Compute Service-Neutron - Networking services - Gathering the pieces and building a picture - A sample architecture setup.
Module 2: OpenStack cluster - Controller and common services (6 Hours)
OpenStack Cluster – The Cloud Controller and Common Services- Asymmetric clustering, Symmetric clustering, The cloud controller - The keystone service. The nova-conductor service, The nova-scheduler service, The API services, Image management, The network service, The horizon dashboard, The telemetry services.
Module 3: OpenStack compute and Storage (12 Hours)
OpenStack Compute -The compute service components - Deciding on the hypervisor - OpenStack Magnum Project - Segregating the compute cloud - Overcommitment considerations - Storing instances' alternatives - Understanding instance booting - Planning for service recovery. OpenStack Storage - Block, Object, and File Share - Understanding the storage types - Ephemeral Storage - Persistent storage - A spotlight on Swift - Deploying Swift service - Using block storage service: Cinder.
Module 4: OpenStack Networking (10 Hours)
The architecture of Neutron - Implementing virtual networks - Connecting virtual networks with routers - Implementing network security in OpenStack. OpenStack Networking - The architecture of Neutron - Implementing virtual networks - VLAN, Tunnel based, Virtual Switches, The ML2 Plugin. Neutron Subnets - Connecting virtual networks with routers - Configuring the routing service - connecting networks using a virtual router, connecting to the external world, connectivity from the external world, associating a floating IP - Implementing network security in OpenStack
Module 5: OpenStack Orchestration, HA and failover (10 Hours)
Orchestration in OpenStack - Heat and its Components, stacking in OpenStack, OpenStack Orchestration with Terraform. OpenStack HA and failover: Scope of HA in OpenStack, HA in the database, HA in the Queue, Implementing HA on RabbitMQ.

20MCA281

INTERNET OF THINGS

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA281	INTERNET OF THINGS	ELECTIVE	3	1	0	4

COURSE OUTCOMES

Course Outcomes: After completion of the course the student will be able to

CO No:	Course Outcome (CO)	Blooms Category Level
CO 1	Describe the main concepts and features of the IOT paradigm.	Level 2: Understand
CO 2	Discuss Fog computing, TinyOS - nesC and programming frameworks for IOT	Level 2: Understand
CO 3	Describe the data management techniques applied to the IOT environment.	Level 2 Understand
CO 4	Explain security, and privacy in IOT environments	Level 2 Understand
CO 5	Discuss key enablers and solutions to enable practical IoT systems	Level 2 Understand

Syllabus

Module 1 (9 Hours)
Overview of Internet of Things: Open-source semantic web infrastructure for managing IOT resources in the Cloud - Device/Cloud Collaboration framework for intelligence applications.
Module 2 (11 Hours)
Introduction to Fog Computing: principles, architectures, and applications. TinyOS – NesC, Programming frameworks for Internet of Things
Module 3 (8 Hours)
Stream processing in IoT: foundations, state-of-the-art, and future directions - A framework for distributed data analysis for IoT
Module 4 (9 Hours)
Security and privacy in the Internet of Things- Internet of Things - robustness and reliability. TinyTO: two-way authentication for constrained devices in the Internet of Things - Obfuscation and diversification for securing the Internet of Things

Module 5 (8 Hours)
Creating a simple IoT project - Preparing Raspberry Pi – Interfacing the hardware - Internal representation of sensor values- Persisting data - Creating the actuator project - Creating a controller.

20MCA241

DATA SCIENCE LAB

20MCA241	DATA SCIENCE LAB	CATEGORY	L	T	P	CREDIT
		LAB	0	1	3	2

COURSE OUTCOMES

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Use different python packages to perform numerical calculations, statistical computations and data visualization	Level 3: Apply
CO 2	Use different packages and frameworks to implement regression and classification algorithms.	Level 3: Apply
CO 3	Use different packages and frameworks to implement text classification using SVM and clustering using k-means	Level 3: Apply
CO 4	Implement convolutional neural network algorithm using Keras framework.	Level 3: Apply
CO 5	Implement programs for web data mining and natural language processing using NLTK	Level 3: Apply

Syllabus

Review of python programming, Matrix operations, Data Visualisation using matplotlib / plotly / bokeh / seaborn, Data handling using pandas, Classification k-NN algorithm, Naïve Bayes algorithm, Implementation of linear and multiple regression techniques, Text classification using Support vector machine, Implementation of Decision Trees, Clustering using k-means algorithm, Convolutional Neural Network to classify images using Keras framework, Web Crawler and Scrapping web pages, Implementation of NLP - Part of Speech tagging, N-gram & smoothening and Chunking using NLTK.

Course Level Assessment Questions

Course Outcome 1 (CO1):

• Review of python programming – Programs review the fundamentals of python (simple python programs ice breaker) – (at most one lab session)

• Matrix operations (using vectorization) and transformation using python and SVD using Python.

• Programs using matplotlib / plotly / bokeh / seaborn for data visualisation. • Programs to handle data using pandas.

Course Outcome 2 (CO2)

• Program to implement k-NN classification using any standard dataset available in the public domain and find the accuracy of the algorithm.

• Program to implement Naïve Bayes Algorithm using any standard dataset available in the public domain and find the accuracy of the algorithm

• Program to implement linear and multiple regression techniques using any standard dataset available in the public domain and evaluate its performance.

Course Outcome 3(CO3):

• Program to implement text classification using Support vector machine.

• Program to implement decision trees using any standard dataset available in the public domain and find the accuracy of the algorithm.

• Program to implement k-means clustering technique using any standard dataset available in the public domain

Course Outcome 4 (CO4):

• Programs on feedforward network to classify any standard dataset available in the public domain. • Programs on convolutional neural network to classify images from any standard dataset in the public domain. *[Note] : Encourage students to refer standard neural network architectures such as LeNet5, ResNet, GoogLeNet etc. and use these as starting points for their models.

Course Outcome 5 (CO5):

Web Data Mining

• Implement a simple web crawler (ensure ethical conduct).

• Implement a program to scrap the web page of any popular website – suggested python package is scrappy (ensure ethical conduct).

Natural Language Processing

Problems may be designed for the following topics so that students can get hands on experience in using python for natural language processing:

• Part of Speech tagging

• N-gram and smoothening

• Chunking

20MCA243

MOBILE APPLICATION DEVELOPMENT LAB

20MCA243	MOBILE APPLICATION DEVELOPMENT LAB	CATEGORY	L	T	P	CREDIT
		LAB	0	1	3	2

COURSE OUTCOMES

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Design and develop user interfaces for mobile apps using basic building blocks, UI components and application structure using Emulator	Level 3: Apply
CO 2	Write simple programs and develop small applications using the concepts of UI design, layouts and preferences	Level 3: Apply
CO 3	Develop applications with multiple activities using intents, array adapter, exceptions and options menu.	Level 3: Apply
CO 4	Implement activities with dialogs, spinner, fragments and navigation drawer by applying themes	Level 3: Apply
CO 5	Develop mobile applications using SQLite.	Level 3: Apply

Syllabus

Fundamentals: Basic Building blocks – Activities, Services, Broadcast Receivers and Content providers, UI Components – Views and notifications Components for communication -Intents and Intent Filters

Application Structure: AndroidManifest.xml, user-permission – sdk, Resources and R.java, Assets, Layouts and Drawable Resources, Activities and Activity lifecycle.

Emulator-Android Virtual Device: Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS

Basic UI design: Form widgets, Text Fields, Validation of EditText, Layouts, [dip, dp, sip, sp] versus px

Preferences: Shared Preferences, Preferences from xml

Menu: Option menu, Context menu, menu from xml, menu via code

Intents: Explicit Intents, Implicit intents

UI design: Time and Date, Images and media, Android Adapter and ListView, Composite, Alert Dialogs and Toast, Popup, Fragments, Navigation drawer

Tabs, Tab Activity Styles & Themes: styles.xml, drawable resources for shapes, gradients (selectors), style attribute in layout file, Applying themes via code and manifest file

Content Providers: SQLite Programming, SQLite Open Helper, SQLite Database, Cursor, Reading and updating Contacts, Reading bookmarks

Course Level Assessment Questions

Course Outcome 1 (CO1):

1. Design a Login Form with username and password using LinearLayout and toast valid credentials

2. Write a program that demonstrates Activity Lifecycle.

3. Implementing basic arithmetic operations of a simple calculator

4. Implement validations on various UI controls

Course Outcome 2 (CO2)

1. Design a registration activity and store registration details in local memory of phone using Intents and SharedPreferences

2. Design a simple Calculator using GridLayout and Cascaded LinearLayout

3. Create a Facebook page using RelativeLayout; set properties using .xml file

4. Develop an application that toggles image using FrameLayout

Course Outcome 3(CO3):

1. Implement Adapters and perform exception handling

2. Implement Intent to navigate between multiple activities

3. Develop application that works with explicit intents

4. Implement Options Menu to navigate to activities

5. Develop an application that uses ArrayAdapter with ListView.

Course Outcome 4 (CO4):

1. Develop an application that use GridView with images and display Alert box on selection

2. Develop an application that implements Spinner component and perform event handling

3. Apply themes via code and manifest file

4. Develop application using Fragments

5. Implement Navigation drawer

Course Outcome 5 (CO5):

1. Create database using SQLite and perform INSERT and SELECT

2. Perform UPDATE and DELETE on SQLite database

3. Develop an application as a micro project which uses SQLite database as an assignment

20MCA245

MINI PROJECT

CODE	COURSE NAME	CATEGORY	L	T	P	CREDIT
20MCA245	MINI PROJECT	PROJECT	-	-	4	2

COURSE OUTCOMES

Course Outcomes: After the completion of the course the student will be able to

CO No.	Course Outcome (CO)	Bloom's Category Level
CO 1	Identify a real-life project which is useful to society / industry	Level 2: Understand
CO 2	Interact with people to identify the project requirements	Level 3: Apply
CO 3	Apply suitable development methodology for the development of the product / project	Level 3: Apply
CO 4	Analyse and design a software product / project	Level 4: Analyse
CO 5	Test the modules at various stages of project development	Level 5: Evaluate
CO 6	Build and integrate different software modules	Level 6: Create
CO 7	Document and deploy the product / project	Level 3: Apply

Week Schedule

Week 1. Familiarisation with build tools (editor/IDE, compiler such as gcc with commonly used options/switches, debugger like dgb). Familiarisation with an IDE (Eclipse, NetBeans…), that supports build tools and common version control operations using Git . Familiarisation with Docker Selection of Topic, Formation of Development Team, Feasibility analysis.

Week 2. Topic Approval, Meeting of Development Team including Scrum Master with Product Owner. Informal, preliminary discussions of requirements. Creating user stories in the rough record. Commencement of the Project.

Week 3.Identifying modules, Initial Design of Database & UI. Creating a Docker container for the environment Creating an empty git repository by Scrum Master / one member of the Development team and setting permission to other members. Pushing the first version of the Project along with a Readme file containing contact details of team members. Creating pull requests for sample update of Readme by each member and merging the pull requests of one by another.

Week 4-5. Setting up systems for development, testing and production. Design of the basic model of a simple deployment pipeline Creating a suitable folder structure (Maven's folder structure is desirable). Creating Unit tests using an XUnit framework, Writing the build and code analysis script, Writing acceptance test scripts and test cases, Setting up a Continuous Integration System like Jenkins. Automating acceptance tests with Selenium, Karate or an equivalent tool, writing a simple deployment script that uses scp/rsync or Ansible for copying the Dockerfile and running Docker with ssh. First Scrum Review. (Here onwards, the Scrum reviews are conducted on every other week)

Week 7. Project Presentation - Interim Evaluation to be based on Git History

Week 14. Submission of Project Report, with Scrum Book Project Presentation – Final Evaluation to be based on Git History, Scrum Book, Project Report and Presentation