**Big Data Technologies and Cloud Computing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course Code** | **Title of the course** | **Credits** | **Level** | **Credit Split**  **Lecture-Lab- Seminar-Project** |
| M5221365 | Big Data Technologies and Cloud Computing | 3 | 500 | 1-1-0-1 |

|  |  |
| --- | --- |
| **Course Outcomes** | |
| CO1 | Introducing Apache Spark |
| CO2 | Text Mining in Big data |
| CO3 | Link analysis and recommendation systems |
| CO4 | Introduction to cloud computing |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mapping of course outcomes with programme outcomes** | | | | | | |
|  | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 |
| CO1 | 3 | 3 | 2 | 1 | 2 | 1 |
| CO2 | 3 | 2 | 1 | 1 | 1 | 1 |
| CO3 | 3 | 3 | 1 | 1 | 1 | 1 |
| CO4 | 3 | 3 | 2 | 1 | 2 | 1 |

|  |  |
| --- | --- |
| **Module** | **Content** |
| 1 | Topics: Overview of modern big data architecture: Lambda, Kappa, and Delta Architecture, Spark Core, Spark Context, RDD, DAG Execution Model, Lazy Evaluation and Caching Strategies, DataFrame and Spark SQL, Machine Learning with MLlib and PySpark: feature engineering, classification, clustering, Introduction to Spark Struc-tured Streaming,  Lab Work: Setup and execute a Spark job on a cluster (local or cloud), Implement a classification task using PySpark MLlib, Create real-time analytics dashboards using Spark Structured Streaming |
| 2 | Topics: Advanced similarity detection techniques in bigdata: Shingling of documents, Min-hashing and signature matrix construction, Locality Sensitive Hashing (LSH), In-cremental computation over data streams, Real-time semantic matching and indexing, Integration with vector databases for similarity search (e.g., FAISS, Pinecone)  Lab Work: Implement LSH-based text similarity in PySpark, Analyze stream data using Spark Streaming and sliding windows |
| 3 | Topics: Graph-based algorithms: PageRank, computation and convergence issues, Topic-sensitive PageRank and Link Spam Detection, HITS algorithm and hub-authority dynamics, Mining frequent itemset: A-Priori, FP-Growth, Introduction to modern graph engines, Building Recommendation Systems: Collaborative filtering, content-based filtering, Real-time recommendation via batch + stream pipelines  Lab Work: Compute PageRank using PySpark and GraphFrames, Build a collabo-rative filtering model using ALS on a sample user-product dataset |
| 4 | Topics: Cloud-native architecture for big data processing, Containers and orchestration: Docker, Kubernetes, Helm, Introduction to Serverless computing (e.g., AWS Lambda, Azure Functions), Cloud components and service models: IaaS, PaaS (e.g., GCP App Engine, AWS Elastic Beanstalk), and SaaS, Cloud benefits and limitations: scalability, elasticity, multi-tenancy, security, Cloud storage systems: S3, HDFS on cloud, GCS, Azure Blob, Data pipeline orchestration with Airflow or Prefect, Introduction to cloud-based MLOps and data versioning tools (e.g., DVC, MLflow)  Lab Work: Deploy a Spark job on a cloud platform (Databricks, EMR, or GCP DataProc), Create a simple CI/CD pipeline for a PySpark job using GitHub Actions and Docker, Build a cloud-hosted REST API to serve a recommendation model |
| **Text Books:**   1. Data Analytics with Spark Using Python, By Jeffrey Aven, Addison Weley Data and Analytics series, 2018 2. Big Data Analytics with Spark, Mohammed Guller, APress, 2015   **References:**   1. Anand Rajaraman, Jeffrey D Ullman. Mining of Massive Datasets, Cambridge University Press 2010 | |