NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

Sessional Examination, September 2025
M.Tech. in Data Science, Semester – I
6CS302CC25 – Data Science System Design

Roll / Exam No. □ Supervisor's Initial with Date □

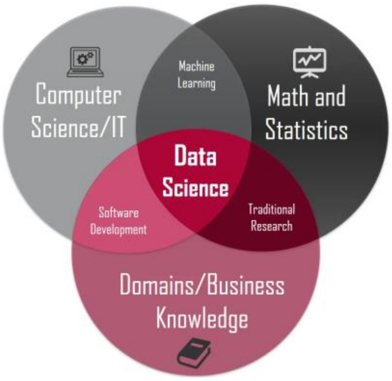
Time: 2 Hours Max Marks :50

Instructions: 1. Attempt all questions.
2. Figure to right indicate full marks
3. Draw neat sketches wherever necessary.
4. Assume necessary data wherever required, and indicate clearly.

Q.1 CO1 **Answer the following** **[14]**

(A) BL1 Explain Data Science, and Data Science System . Briefly explain key components of Data science systems. (8)

Data Science : Data Science is the discipline of extracting meaningful insights and knowledge from data using techniques from statistics, machine learning, data engineering, and domain expertise.
It is intersection of Mathematics & Statistics, Programming & Data Engineering, and Domain knowledge, communication & Business Decision-making.



Data Science System : defines the high-level structure of how data science applications, infrastructure, and components interact to process data, train models, and serve predictions reliably and efficiently
Key Components of Data Science System :

- **Data Sources :** Raw data from databases, sensors, APIs, external systems
- **Data Ingestion Pipeline :** Tools and processes for importing streaming data and batch jobs to storage
- **Data Storage :** Repository for storing data such as Data Lake, Data Warehouse, Database
- **Data Processing :** Frameworks for transforming and analyzing data (e.g., Apache Spark, Apache Flink).

- **Model Training:** Infrastructure for developing machine learning models (e.g., TensorFlow, PyTorch).
- **Model Serving:** Deploying models for inference (e.g., TensorFlow Serving)
- **Monitoring & Logging:** Tracking system performance and logging events
- **Security & Compliance:** Ensuring data privacy and regulatory compliance (e.g., GDPR, HIPAA).
- **User Interface/API Layer:** Providing access to data and models through dashboards or API

(B) BL6 A hospital wants to reduce patient readmissions. Someone suggests using ML to predict whether a patient will be readmitted within 30 days after discharge. They think they can train a model on past patient records (age, diagnoses, treatments, length of stay, etc.). After training the model, they want to flag high-risk patients at discharge so doctors can schedule extra follow-ups. How should they frame their problem in ML terms? (6)

Ideal Outcome : Reduce patient readmissions within 30 days of discharge.
Model's goal: Predict whether a patient will be readmitted within 30 days.
Model output: Binary classification → *readmitted / not readmitted.*
Success metrics: Reduction in 30-day readmission rate (e.g., lower than current baseline or below regulatory threshold)./ improved patient's satisfaction level

Q.2 CO3 **Do as Directed** **[18]**

(A) BL3 A batch data pipeline performs heavy feature engineering and feeds a feature store for model training. Which of following two system designs is better in terms of availability? Justify you answer by comparing availability of single worker as well as availability of total pipeline. (6)

System A: Three worker nodes working in parallel with same capability, where each worker node fails once every 5000 hours of operation approximately. When a failure occurs, it takes about 20 hours to detect, repair, and restore.

System B: Five worker nodes working in parallel with same capability, where each worker node fails once every 10000 hours of operation approximately. When a failure occurs, it takes about 100 hours to detect, repair, and restore.

Availability (A)= $$\frac{\text{MTBF}}{\text{MTTR+MTBF}}$$

System A:
Availability of individual worker node = 5000 / (5000+20) = 99.6%
Availability of worker pool = (1- (1-0.996)^3) =0.999999936

System B:
Availability of individual worker node = 10000 / (10000+100) = 99%
Availability of worker pool = (1- (1-0.990)^5) =0.9999999999

System A is better in terms of single worker availability (faster recovery)
System B is better in terms of total pipeline availability.

Streaming pipeline generally requires low latency, faster recovery (lower MTTR = high availability) at worker level. Failover helps, but doesn't eliminate risk of multiple overlapping failures. If a worker goes down, the pipeline may stall, so fast recovery is more important than adding more parallel workers. ➔ So, System A is preferred for streaming

For Batch pipelines, throughput is more important, and system can tolerate some worker downtime. So, it prefer more workers / higher redundancy➔ System B is preferred for batch pipeline as total pipeline is more important in batch pipeline.

(B) BL6 A data science team deploys a machine learning model inference service on a single server, which handles up to 500 inference requests per second. During peak hours, the incoming request rate can reach 1500 requests per second. Answer following questions with respect to this system design with necessary diagrams (6)

i) What is the scalability limitation of the current system?
ii) Suggest a horizontal scaling solution to handle the peak load.
iii) Propose system design with additional components to improve scalability further.

i)
Maximum capacity = 500 req / sec
Capacity requirement at Peak scenario = 1500 req / sec
So, Limitation : other requests get queue / dropped at peak usage

ii) LB-> S1 / S2 / S3 …can add/remove
iii) LB, Auto scaling, caching server, Msg

© BL2 A company has deployed a recommendation system for an e-commerce platform. The system collects user interactions in real-time and updates product recommendations. If the system is distributed across multiple data centers, how would the CAP theorem (Consistency, Availability, Partition tolerance) influence your design choices for the recommendation system? Which trade-offs would you make and why? (6)

Context :

- The system is **distributed across multiple data centers**.
- It **collects user interactions in real-time** (clicks, views, purchases).x
- It **updates recommendations** that influence personalization and product ranking.

So, both **freshness of data (consistency)** and **system responsiveness (availability)** are important — but their criticality depends on business needs.

CAP theorem :

The CAP theorem states a distributed system can only guarantee two of three properties: Consistency, Availability, and Partition Tolerance.

- Consistency (all reads see the most recent write or an error),
- Availability (all requests receive a response, even if it's stale data),
- Partition Tolerance (the system continues to operate despite network failures).

A distributed system can have properties : CA, CP, or AP

Since the **system is distributed across multiple data centers, network partitions are inevitable**, making Partition Tolerance (P) a requirement. This forces a choice between **C** and **A**.

PA :
For e-commerce application, It is better to show slightly stale recommendations than to show an error or have a non-loading component. A user is unlikely to notice if a recommendation is a few seconds out of date, but they will definitely notice if the recommendation engine is down.
For example:
Case A: Prioritize Availability over Strong Consistency (AP system)
During a partition, users can still browse, interact, and receive recommendations. This recommendation may be stale (e.g. user just bought a phone but still sees phone suggestions for a few minutes)
It is acceptable, because, recommendations are probabilistic by nature.
Here, recommendation model across data centers can be updated eventually as eventual consistency in place of than strict consistency. Local copy also can give recommendation in place of partition to ensure availability of recommendation service.

Case B: Prioritize Consistency over Availability (CP system)
To provide strict consistency, All users (with similar properties)across globe should see same recommendation. But in case of partition, some requests cannot reach to server. This will give strict consistency . But, for that rejecting request reduces availability.

Q.3 CO4 **Do as Directed** **[18]**

(A) BL4 A fashion retail company has a machine learning model that predicts whether a clothing item will be "in fashion" next season . (6)
Multiple clients need to use this service:

- Mobile app: quick predictions, minimal data transfer
- Web dashboard: flexible queries with detailed analytics
- Internal system: batch predictions for inventory

Which API style would you choose for exposing the prediction service: REST or GraphQL? Justify your choice based on the client requirements.

Client	**Needs**	**Implications for API**
Mobile App	Fast predictions, minimal data transfer	API should allow fetching only necessary fields, lightweight response
Web Dashboard	Flexible queries, detailed analytics	API should allow fetching specific fields and multiple related objects in a single request
Internal Inventory System	Batch predictions, possibly large datasets	API should support bulk fetching efficiently

For this case study, **GraphQL is preferred** because:

1. The **mobile app** can request only the fields it needs (fast, low bandwidth).
2. The **web dashboard** can perform complex queries for analytics in a single request.
3. Future changes to the model schema (new features, new metadata) can be added without breaking existing clients.
4. It reduces multiple network calls compared to REST when fetching related data.

Alternate Approach :

- A hybrid approach is also possible: GraphQL for flexible client-facing APIs and REST for internal batch processing

(B) BL2 A hospital system has built a machine learning model to predict patient risk for complications based on medical records, lab results, and real-time monitoring devices (e.g., heart rate, blood pressure). Data sources include: (6)

- Electronic Health Records (EHRs) – updated daily
- Lab test results – updated every few hours
- Real-time monitoring devices – continuous data streams

The hospital wants to update patient risk scores for doctors and dashboards. Should the hospital use batch processing or stream processing or hybrid approach to update patient risk scores? Justify your choice.

Data Source	**Update Frequency**	**Latency Requirement**	**Implication**	
EHRs	Daily	Low urgency	Can be processed in batch	
Lab results	Every few hours	Medium urgency	Batch or micro-batch processing possible	
Real-time monitoring	Continuous (seconds /minutes)	High urgency	Requires near real-time updates, stream processing	

o Batch processing for historical data, EHR updates, and lab results.

o Stream processing for real-time monitoring devices and urgent

alerts.
•This ensures efficient processing of large datasets while still providing low-latency updates for critical patient care.

(C) BL1 Explain Cross Industry standard process for Data Mining with help of suitable diagram (6)