Cardamom Planters' Association College

Re-Accredited by NAAC at 'A' Grade (3rd Cycle) by NAAC

Pankajam Nagar, Bodinayakanur – 625513

Big Data Analytics

TOPIC: Data Explosion and Big Data Analytics, Analytical Theory and Real–Time Analysis

R. SUJITHA M.Sc., M.Phil., NET.,

Assistant Professor,

Department of Computer Science

Unit I: Data Explosion and Big Data Analytics

Introduction

1) Data Explosion — the big picture

- **Data (computing):** any digital information that can be stored, moved, or processed.
- **Data explosion:** the very fast growth of data from people, devices, and organizations because storage is cheap, networks are fast, and everything is online.
- **Old model vs new model:** earlier, a few companies produced data and the rest consumed it; now **everyone** both produces **and** consumes data (posts, clicks, sensors, apps, logs).

Everyday examples that cause data growth

- Social media uploads of photos/video, comments, reactions.
- Trading systems (e.g., stock exchanges) generate ~terabytes/day.
- Aircraft engines can generate **>10 TB in 30 minutes**; thousands of flights daily.
- E-commerce clickstreams, mobile apps, CCTV, smart meters, wearable devices.

2) Units of data (know these for exams)

- **bit (b):** smallest unit (0/1).
- **nibble:** 4 bits.
- **byte (B):** 8 bits.
- **Kilobyte (KB):** 1024 bytes (10^3 approx).
- **Megabyte (MB):** 1024 KB (10^6 approx).
- **Gigabyte (GB):** 1024 MB (10^9).
- **Terabyte (TB):** 1024 GB (10^{12}).
- **Petabyte (PB):** 1024 TB (10^{15}).
- **Exabyte (EB):** 1024 PB (10^{18}).
- **Zettabyte (ZB):** 1024 EB (10^{21}).
- **Yottabyte (YB):** 1024 ZB (10^{24}).

3) "Digital Universe" – opportunity & challenge

(From IDC figures summarized in your table.)

- **Size**: from ~4.4 ZB (2013) to ~44 ZB (2020).
- **Useful/analyzable share**: only a fraction of data is tagged/usable, but this share keeps rising as more data is labeled and curated.
- **Sensor-generated data** grows fast (IoT).
- **Emerging markets** contribute a large—and growing—portion of total data.
  Implication: huge opportunity for analytics, but also major challenges in storage, quality, security, governance, and skills.

4) Mobile & real-time data

- **Why mobile data grows:** smartphones, 4G/5G, video streaming, cloud apps, and IoT devices.
- **Global Mobile Data (2016→2021):** ~7, 11, 17, 24, 35, **49 exabytes/month** (bar chart).
- **Forecast 2020→2030 (line chart):**

- **Without M2M (machine-to-machine):** ~57 → 144 → 353 → 830 → 1910 → **4394 EB/month**.
- **With M2M included:** ~62 → 158 → 390 → 938 → 2194 → **5016 EB/month**.

- **Takeaway:** M2M traffic (devices talking to devices) adds a **huge** extra load; real-time access is increasingly necessary for business.

5) Internet of Things (IoT)

Definition

A network of physical objects—vehicles, appliances, wearables, animals or people—with embedded electronics (sensors/actuators), software, and network connectivity. Each thing has a **unique identifier (UID)** and can collect and exchange data over a network without human involvement.

Examples

Healthcare: a heart-monitor implant sends continuous readings; it can auto-alert a hospital during an emergency (24×7 monitoring).

- **Automotive:** in-car sensors detect tyre pressure, obstacles, or sleepy driving and can warn or act (e.g., play a wake-up tone).
- **Smart home:** thermostats, lights, security cameras controlled from phone; data stored in the cloud.

Growth snapshots from your figures

- **Connected devices vs population:**
  - 2003: population ~6.3B; devices ~0.5B (≈0.08 per person).
  - 2010: ~6.8B people; **12.5B** devices.
  - 2015: ~7.2B people; **25B** devices.
  - 2020: ~7.6B people; **50B** devices (≈6.58 per person).
- **IoT devices (2015→2025, billions):** 15.41 → 20.35 → 26.66 → 35.82 → 51.11 → **75.44**.
- **Market value (note):** IoT became a multi-trillion-dollar market by 2020.

Why IoT matters in Big Data

- **High-volume streaming data** from sensors.
- **Variety**: logs, images, signals; structured & unstructured.
- **Velocity**: near real-time ingestion/processing needed.
- **Value**: predictive maintenance, safety, efficiency, personalization.

6) Three fast-growing data segments

- **Embedded systems & IoT:** By mid-2020s, a significant share of global data comes from embedded/IoT devices; a portion is **hyper-critical** (life-or-safety-critical).
- **Mobile & real-time data:** on-demand, anywhere, anytime; drives digital transformation.
- **Cognitive/AI systems:** ML, NLP, and AI analyze huge datasets; both **data created** and **data analyzed by AI** are expanding rapidly.

7) Key terminology

- **Real-time data:** processed fast enough to act immediately (milliseconds–seconds).
- **Machine-to-Machine (M2M):** devices exchange data without human input.

- **Sensor/actuator:** sensor measures; actuator acts (opens valve, changes speed).
- **Firmware:** device-level software stored in hardware.
- **UID:** unique ID of a device in a network (e.g., MAC address).
- **Cloud:** on-demand servers/storage over the internet.

8) Opportunities vs Challenges

Opportunities: personalization, efficient operations, predictive maintenance, smart cities/healthcare, new services & revenue.
Challenges: data quality & labeling, privacy and security, storage/processing cost, interoperability/standards, skills gap, governance & compliance.

Big Data Analytics

- Big Data Analytics involves scrutinizing, cleansing, manipulating, and demonstrating underlying data to discover usable information.
- It supports decision-making by creating an immersive connection between data and effective decision-making within an organization.

Analysis vs Analytics

Analysis:

- Focuses on understanding the past.

- Answers the question "What happened?"

- Concerns only the past.

- Conclusion is made on past data.

Analytics:

- Focuses on "Why it happened?" and predicts "What may probably happen next."

- Concerns both past and future.

- Aims to plan the future.

- Performed after analysis.

Process of Analytics

1. Objective: Define a well-defined objective to be achieved.

2. Ground Knowledge: Acquire related knowledge to achieve the objective.

3. Outcome/Profit: Determine the final outcome/profit in terms of quality and quantity.

Data Analytics Life Cycle

- Similar to the software development life cycle (SDLC).

- Phases driven by specific activities in the life cycle.

Phase 1: Setting Objective

- Like drafting the Software requirement in SDLC, the objective of the analytics is set here.
- Derived from questions such as:
  - Why is it done?
  - What outcome is to be achieved?
  - Which supporting empirical data is available?
  - Initial Hypothesis Formulation is a key activity.

Phase 2: Data Preparation and Enhancement

- Creation of an analytics sandbox for working with data.
- Involves ETLT (extract, load, transform, extract, load, transform) for data cleaning and transformation.
- Data enhancement can include new data channels.

Phase 3: Model Planning

- Decide on methodologies and workflow for model construction.
- Investigate data to learn about variable relationships and choose critical variables and models.

Phase 4: Analytics through Model Building

- Generate datasets for testing, training, and production.
- Build and execute models based on model planning.
- Examine if existing tools suffice or if a more robust environment is needed.

Phase 5: Data Visualizations and Communicating the Results

- Identify key insights, quantify business value, and develop a chronicle to convey findings.
- Use data visualizations for result-oriented discussions.

Steps in Data Analytics (Figure 1.16)

1. Collect: Gather data from sources like social media, data sources, or open databases.
2. Store: Store the collected data in a database for further processing.
3. Process: Apply analytical techniques to find insights and derive new information.
4. Visualise: Create visualizations to reflect the analytical outcomes.

Life Cycle of Data Analytics (Figure 1.17)

- Empirical data collection and consideration of the business domain lead to:

- Setting objective

- Hypothesis

- Data preparation and enhancement (resulting in sandboxed, cleaned data)

- Planning the right model (involving methods, techniques, workflow, and model building)

- Analytics through model (yielding insights and results)

- Generating results via data visualizations and communicating the results (leading to visualizations, findings, and results)

- If findings are incompatible with the hypothesis, redefine the objective.

1.2 EVOLUTION OF DATABASE TECHNOLOGY AND BIG DATA

Evolution of Database Technology:

- **1960s:** Data collection began; IMS (Information Management System) and network DBMS were introduced.
- **1970s:** Relational data models and relational DBMS came into use.
- **1980s:** Advanced models like object-oriented, spatial, and scientific DBMS emerged.
- **1990s:** Rise of data mining, warehousing, multimedia and web databases.
- **2000s:** Focus on stream data management, mining, and applications.
- **2012-Present:** Emergence of Big Data Technology with tools like:
    - XML
    - Social Networks
    - Data integration
    - Cloud Computing
    - Global Information Systems

Evolution of Big Data:

- **1990s:** Digital storage overtook paper storage. Problems with data overload became visible.
- **2000s onwards:** Researchers addressed challenges in managing large and diverse data.
- **2005:** O'Reilly Media coined the term *Big Data*. Yahoo! introduced Hadoop.
- **2009:** India launched a biometric database with iris scans, fingerprints, and photos of 1.2 billion citizens.

WHO IS GENERATING BIG DATA? (Fig 1.8)

Big Data is produced by various sources:

- **Social Media & Networks:** Facebook, Twitter, YouTube, etc.
- **Scientific Instruments:** Tools used in labs and research.
- **Mobile Devices:** Data from calls, GPS, and apps used 24x7.
- **Sensor Technology & IoT:** Smart farming, satellites, automated machines.

TYPES OF DATA:

- **Social Data:** Collected from social networking platforms.
- **Machine Data:** Comes from sensors, barcodes, RFID chips.
- **Transactional Data:** Comes from online purchases, business transactions, etc.

DATA GROWTH INSIGHTS

- In 2010, Eric Schmidt said: *Every 2 days, the world creates as much data as it did from the dawn of civilization till 2003*.
- By 2018, the U.S. had a shortage of 1.5 million data managers.
- McKinsey predicted data analytics could generate **$9.5 to $15.4 trillion** in value by 2021.

OPPORTUNITIES & CHALLENGES (Table 1.4)

Feature	2013	2020
Size of digital universe	4.4 trillion GB	44 trillion GB
Useful data	22%	37%
Sensor-generated data	2%	10%
Data from emerging markets	40%	60%

CHALLENGES IN HANDLING BIG DATA:

- Traditional databases (RDBMS) can't handle huge volumes.
- Historical data is stored in **Data Warehouses**.
- Useful techniques: **Association Rules**, **Clustering**, **Classification**, **Outlier Detection**, etc.
- Data is stored in **distributed file systems** to manage complexity.
- Need for **cloud computing** to store and process large-scale data.

1.3 Elements of Big Data

Gartner defines 4 major characteristics of Big Data, known as the **4 Vs**:

1. **Volume**
2. **Velocity**
3. **Variety**
4. **Veracity**

Later, a 5th V was added:
5. **Value**

1.3.1 Volume (Amount of Data)

- **Definition**: Refers to the vast amounts of data generated by organizations and individuals.
- **Units**: Terabytes → Petabytes → Exabytes → Zettabytes → Yottabytes
- **Example**: Oracle reported data growth from 2009 to 2020 reached 45 Zettabytes per month.
- **Key Point**: Data is growing exponentially. By 2030, it may reach **78 Yottabytes**.

1.3.2 Velocity (Speed of Data)

- **Definition**: Refers to the speed at which data is generated, captured, and processed.
- **Types**:
  - **Batch Data**: Collected and processed in chunks.
  - **Streaming Data**: Processed in real-time.
- **Data-in-Motion**: Generated from mobile devices, sensors, etc., at any time/place.
- **Examples**: eBay processes ~5 million transactions/day.

3 Aspects of Velocity:

- **Data-in-Motion** (e.g., sensor data)
- **Lifetime of Utility** (data value decreases with time)
- **Real-Time Analytics** (used for fraud detection, instant decision-making)

1.3.3 Variety (Types of Data)

- **Definition**: Refers to different types of data formats and sources.
- **Data Forms**:
  - **Structured** (e.g., relational tables)
  - **Unstructured** (e.g., videos, images, social media)
  - **Semi-structured** (e.g., XML, JSON)

Sources:

- Internal (e.g., product records)
- External (e.g., market trends, tweets)

Examples:

- Social media
- IoT sensors
- Medical devices
- Security cameras
- RFID systems

1.3.4 Veracity (Truthfulness/Trust of Data)

- **Definition**: Relates to the quality and trustworthiness of data.
- **Concerns**: Incomplete, inconsistent, or unverified data.
- **Survey**:
  - 1 in 3 leaders make decisions using untrusted data.
  - 60% of CEOs have more data than they can use.

Goal: Ensure only correct and reliable data is used in analysis.

1.3.5 Value (Business Use of Data)

- **Definition**: Transforming large volumes of data into meaningful business insights.
- **Includes**:
  - Identifying useful data
  - Extracting and analyzing insights
- **Note**: Data without value is **waste**.

1. Structured Data

Definition:
Structured data is **highly organized** and stored in a **fixed format**, usually in rows and columns like in a **relational database (RDBMS)**.

Features:

- Easy to **search, sort, and analyze**
- Follows a **fixed schema** (structure)
- Uses **tables** (with fields like name, age, etc.)

Example:

Student_ID	Name	Age
101	Anu	20
102	Karthik	21

Sources of Structured Data:

- Databases (MySQL, Oracle)
- Spreadsheets (Excel)
- Online forms

2. Unstructured Data

Definition:
Unstructured data has **no fixed format or structure**, and is difficult to store in tables.

Features:

- Cannot be stored in rows/columns
- May be **text-heavy**, or include **images, audio, video**
- Harder to process and analyze

Example:

- Social media posts (Instagram, Facebook)
- Videos on YouTube
- Emails
- Scanned documents
- News articles

Note:

80% of enterprise data today is **unstructured**.

3. Semi-Structured Data

Definition:
Semi-structured data **does not follow a strict structure** like structured data, but it is **not completely unstructured**. It contains **tags or markers** to separate data items.

Features:

- No fixed schema, but uses tags (like XML, JSON)
- Easier to analyze than unstructured data
- Does not follow table format, but still organized

Examples:

- XML data:
  `<student><name>Ram</name><age>20</age></student>`
- JSON data:
  `{ "name": "Ram", "age": 20 }`
- Emails (have structure like subject, to, from, body)

Common Tools:

- Web logs
- NoSQL databases (like MongoDB)
- APIs that return JSON/XML data

Summary Table

Type of Data	Structure	Examples	Ease of Processing
Structured	Fixed (tables)	Databases, Excel	Easy
Unstructured	No structure	Videos, Social media, PDFs	Hard
Semi-Structured	Partial structure	XML, JSON, Emails, Web logs	Moderate

Big Data System Components

- **Layers of Abstraction:** Big data initiatives have several layers for delegated functionality.
- **Components of Analytical Aspect:** Typical components shown in Figure 1.15 (not visible in the image).
- **Higher-Level Components:** Act as a platform to make big data projects more productive.
- **Macro-Level Architecture:** Accommodates further layers as per requirement and functionality.

The architecture is divided into three layers for efficient big data processing.

Layers:

- **Infrastructure Layer:** Provides low-level storage of data. It supports various storage options like RDBMS, NoSQL, and stores log files and real-time data streams. It can create connectors for incoming streaming data in a cloud-based distributed system.
- **Computing Layer:** Sits on top of the infrastructure layer. It performs analytical activities by querying data, integrating operations, and managing data from multiple sources. It facilitates mapper and reducer jobs for parallel distributed processing.
- **Application Layer:** The topmost layer for user interaction. It performs business logic and functionality, transforming data into desired results. It supports analytics operations like clustering, classification, and communicates with the computing layer for results.

Key Points:- The infrastructure layer acts as an entry point to the architecture.

- The computing layer executes data operations and manages data processing.

- The application layer is for user interaction and business logic implementation.

Importance in Big Data Analytics:

- This layered architecture helps in organizing big data processing efficiently.
- Each layer has distinct responsibilities, ensuring structured data handling and analysis.

Why Model?

- To study and understand the behaviour of a system.
- To predict outcomes based on system behaviour.

Types of Models

1) **Linear vs Nonlinear**

   - Linear: Relationship between variables is straight-line (linear equation).
   - Nonlinear: Relationship is curved or more complex.

2) **Static vs Dynamic**

   - Static: Variables do not change with time.
   - Dynamic: Variables change over time.

3) **Explicit vs Implicit**

   - Explicit: All inputs & outputs are clearly known.
   - Implicit: Some unknowns; solved through iterations.

4) **Discrete vs Continuous**

   - Discrete: System is broken into fixed steps or states.
   - Continuous: Changes smoothly over time.

5) **Deterministic vs Stochastic**

   - Deterministic: Output is always the same for the same input.
   - Stochastic: Includes randomness and probability.

6) **Deductive, Inductive, Floating**

   - Deductive: Based on theory.
   - Inductive: Based on data & observation.
   - Floating: No fixed theory or structure.

Types of Big Data Analytics

1) **Descriptive Analytics**

   - Explains **what happened in the past**.
   - Uses past data to find trends and reasons.
   - Helps plan future actions.

2) **Diagnostic Analytics**

- Explains **why something happened**.
- Finds root causes using data mining, drill-down, regressions.

3) **Predictive Analytics**

- Predicts **what is likely to happen**.
- Uses trends & patterns to forecast future events.

4) **Prescriptive Analytics**

- Suggests **what should be done**.
- Uses data & rules to recommend actions.

Applications of Big Data

✓ **Banking**

- Risk detection, fraud prevention, better services.

✓ **Government**

- Efficiency in services, track crime, reduce costs.

✓ **Healthcare**

- Improve patient care, predict diseases, manage records.

✓ **Education**

- Monitor student progress, improve teaching.

✓ **Retail**

- Understand customer behaviour, personalize offers.

✓ **Manufacturing**

- Improve output quality, solve problems faster.

Challenges in Big Data

❖ Large data volume
❖ High complexity
❖ Different formats (heterogeneity)
❖ Linking and relating data
❖ High speed needed (throughput)

Skills for Big Data Professionals

For Analyst:

- Understand big data tools
- Knowledge of AI & machine learning
- Predictive modelling

For Developer:

- Programming (Java, SQL, etc.)
- Big data technology tools

Soft Skills:

- Problem-solving
- Communication
- Business understanding

Unit II: Analytical Theory

Classification Algorithms

Classification is a **supervised learning** technique where the model learns from class-labeled data and predicts class labels for new data. Test data is used to estimate accuracy, and if acceptable, the model is deployed.

1) Decision Tree Induction

- Builds a tree structure to make decisions or predictions.
- Works by splitting data into smaller sets based on the most significant attributes.
- Nodes:
  - **Root Node**: Top node, starting point.
  - **Decision Nodes**: Internal nodes where tests happen.
  - **Leaf Nodes**: Final outcome or class label.
- Example: Fit or Not based on age, diet, and exercise.
- Common algorithms: ID3, C4.5, CART.

2) Support Vector Machine (SVM)

- Maps data into a higher-dimensional space using a **kernel function** (kernel trick).
- Finds the optimal **hyperplane** that separates classes with maximum margin.
- Works well for high-dimensional data.

3) Classification Using Frequent Patterns

- Uses frequent patterns or association rules to classify data.
- Example: If many customers buy bread & butter together, use the pattern for predictions.
- Useful in data mining & market basket analysis.

4) K-Nearest Neighbors (KNN)

- A **lazy learner**: Doesn't build a model but memorizes data.
- Classifies new instances based on similarity with k-nearest training examples using distance metrics:
  - Euclidean, Manhattan, Minkowski (continuous variables)
  - Hamming (categorical variables)
- Requires normalization of variables.

5) Fuzzy Set Classification

- Removes sharp cut-offs by using fuzzy (gradual) membership.
- Example: Loan approval for salary around ₹49,900:
  - Medium salary membership: 0.15
  - High salary membership: 0.96
- Allows overlap of categories (low, medium, high) with degrees of membership.

6) Genetic Algorithms

- Finds optimal classification rules using evolution-inspired steps:
  - **Initial Population**: Set of rules represented as chromosomes.
  - **Selection**: Pick the best individuals.
  - **Crossover & Mutation**: Create new solutions.
  - Repeat until a stable and good model is created.
- Can handle both classification and optimization problems.

7) Logistic Regression

- Used for binary and multi-class classification.
- Uses **sigmoid function** to map predictions between 0 and 1.
- Types:
  - **Binary Logistic Regression**: Two classes (True/False).
  - **Multinomial Logistic Regression**: More than two nominal categories.
  - **Ordinal Logistic Regression**: Ordered categories (e.g., ratings: 1–5).
- The sigmoid curve ('S' shaped) outputs probability.

8) Naïve Bayes Classifier

- Based on **Bayes' theorem**.
- Calculates the probability of each class given the input and picks the highest.
- Assumes strong independence between features.
- Example: Predicts purchase of a laptop based on age, income, and status.
- Disadvantage: Assumes features are independent.

Regression Analysis Overview

- Fitting regression lines to show relationships between dependent and independent variables.
- Regression equation: `Y = 120 + 5X + error`.

Types of Regression

1. Linear Regression

- **What it does:** Finds a straight line that best fits the data points.
- **Formula:**

  Y=mX+CY = mX + CY=mX+C

  where:

  - **X** = independent variable (input),
  - **Y** = dependent variable (output),
  - **m** = slope,
  - **C** = intercept.
- **Example from the book:** Relationship between **Celsius and Fahrenheit** is linear.

  F=95C+32F = \frac{9}{5}C + 32F=59C+32

2. Multiple Linear Regression

- **What it does:** Uses **two or more input variables** to predict the output.
- **Formula:**

  Y=5X1+8X2−6X3Y = 5X_1 + 8X_2 - 6X_3Y=5X1+8X2−6X3

  where X1,X2,X3X_1, X_2, X_3X1,X2,X3 are multiple independent variables.

- **Example from the book:** Predicting **building price** using factors like number of rooms, locality, and building age.

3. Polynomial Regression

- **What it does:** Fits a **curved line** instead of a straight line, useful when data shows **non-linear trends**.
- **Formula:**

  Y=a1X+a2X2+a3X3+⋯+bY = a_1 X + a_2 X^2 + a_3 X^3 + \dots + bY=a1X+a2X2+a3X3+⋯+b

  where powers of XXX create a curve.

- **Example from the book:** Predicting **customer buying capacity** where data points form a curve.

4. Ridge Regression

- **What it does:** A type of linear regression with a **penalty term** added to reduce overfitting.
- **Why used:** When data has **many features** or **high correlation** among features.
- **Key Point:** Introduces a parameter λ\lambdaλ that controls the penalty.
- **Example from the book:** Adjusts regression line to fit both training and testing data well

5. Lasso Regression

- **What it does:** Similar to ridge regression but uses **absolute values** of coefficients for penalty.
- **Why special:** Can shrink some coefficients to **zero**, effectively performing **feature selection**.

Key Difference (Ridge vs Lasso):

- **Ridge:** Reduces coefficients but **doesn't make them zero**.
- **Lasso:** Can make some coefficients **exactly zero**, removing irrelevant variables.

IN-DATABASE ANALYTICS & TEXT ANALYTICS

[1] Traditional Database Analytics vs. In-Database Analytics

✓ **Traditional Database Approach**

- **Process Flow**:
  - Data is stored in the **database**.
  - Queries (SQL) are processed by **middleware** or **front-end software**.
  - Logic/filters are applied by the **server**.
  - Processed results are sent to **end users** for **visualization**.
- **Drawbacks**:
  - Time-consuming – because data must move **back and forth** between the **database** and the **analytics software**.
  - Increased **data transfer** and **processing delays**.

✓ **In-Database Analytics Approach**

- **New technology** where:
  - **Analytics happens inside the database itself.**
  - Eliminates the need for separate middleware for logic.
  - Logic/filters are incorporated **within the database**.
- **Advantages:**
  - Saves **time** – no need to export/import large datasets.
  - Handles **huge volumes of data** efficiently.
  - Enables **parallel processing, scalability, and optimization.**

[2] History of In-Database Analytics

- **Mid-1990s:** First commercially offered by **IBM, Informix, Oracle, Illustra** as "Object-Related Database Systems."
- **2005:** Thomas Tileston introduced the concept of **embedding analytics in databases** (Teradata Conference).
- **2006–2008:** The concept was presented globally and gained popularity.
- **Data Explosion Era:** The boom of **mobile technology** and "big data" (petabytes of information) increased demand for **in-database processing**.

[3] Need for In-Database

- Traditional analytics methods move data to a **separate environment** for processing.
- **In-database** avoids this – it allows:
  - **Blending** of huge amounts of data directly.
  - **Improved performance** and **efficiency**.

[4] In-Database Tools & Purpose

These tools are part of platforms like **Alteryx, SAP Hana, Teradata, Oracle SQL Server, Apache Spark, Hive**.

- ✓ **Key Tools**:

  - **Browse In-DB tool** – Review data at any point inside database workflow.
  - **Connect In-DB tool** – Connect to a database for processing.
  - **Data Stream In/Out** – Stream data between standard workflow and In-DB workflow.
  - **Join In-DB tool** – Combine datasets within DB.
  - **Formula In-DB** – Create/update fields inside DB using SQL.
  - **Dynamic Input/Output** – Input/output data streams dynamically for predictive modeling.

- ✓ **Predictive Tools**:

  - **Decision Tree tool** – Creates if–then rules to predict target variables.
  - **Linear Regression tool** – Builds models to predict based on one/more variables.
  - **Logistic Regression tool** – Predicts **binary outcomes** (e.g., yes/no, pass/fail).

[5] In-Database Process Flow

- ✓ **Steps**:
  1. Connect to the Database (e.g., via Connect In-DB tool).
  2. Use other In-DB tools (Join, Formula, Browse) to process.
  3. Write or Stream Data back using "Data Stream Out" or "Write Data In-DB tool."
  4. Run Workflow – Results stay inside the database but are reported for visualization.

- ✓ **Benefits**:
  - **Faster processing** – no constant transfer of data.
  - **Handles large datasets** smoothly.

Text Analytics

- **Definition**: The process of analyzing **unstructured text** (customer reviews, social media posts, documents) to extract useful information.
- Uses **Natural Language Processing (NLP)** and **Machine Learning (ML)** techniques.

✓ Steps in Text Analytics (Figure 2.36)

1. **Language Identification** – Detect which language the text is in (e.g., Tamil, English).
2. **Tokenization** – Break text into **tokens** (words/phrases).
3. **Sentence Breaking Phase** – Identify where sentences **start** and **end**.
4. **Normalization** – Standardize text (remove stop words, whitespace, apply stemming/lemmatization).
5. **POS Tagging (Part-of-Speech)** – Assign **noun/verb/adjective** tags to tokens.
6. **Chunking** – Group tagged words into **phrases**. Ex: *"The short lady" = Noun Phrase.*
7. **Syntax Parsing** – Analyze **sentence structure** using syntax trees.
8. **Sentence Chaining** – Link sentences to form **relationships** (contextual meaning).

✓ Difference Between Text Mining vs. Text Analytics

Text Mining	Text Analytics
Cleans up and prepares text data for use.	Uses statistics & ML to derive insights.
Similar to ETL (Extract–Transform–Load).	Adds **business value** through summaries, charts, predictions.
Uses Python, R.	Uses Python, R + BI tools (Power BI, Azure, KNIME).
Example: text categorization, clustering.	Example: sentiment analysis, predictive analytics.

Real-time Analysis

3.1 Introduction

- Real-time systems complete all tasks and generate output within a fixed **time span**.
- The output time is **crucial**, and consistency is required across all tasks.

3.1.1 Real-time System

A real-time system:

- Responds to external input **within a specified time**.
- The result must be both **logically correct** and **timely**.
- Failure to respond is as bad as giving a wrong result.
- Real-time systems are **predictable** and **deadline dependent**.

Examples:

- **Aircraft control system** – must function correctly to prevent crashes.
- **Airbag deployment system** – must deploy within 10ms of collision detection.

Figure 3.1 – Basic Real-time System:

- Shows flow of **input and output events** in real-time systems.

Figure 3.2 – Detailed Real-time System:

Includes components: Sensors, A/D Conversion, Input Interface, Real-Time Computer, Output Interface, Actuators.

What is a Real-time Program?

- A **real-time program** gives output based on:
  - The **correct logic**
  - The **correct timing**
- It uses a **real-time clock** to make sure tasks run **at the right time**.

What is a Real-time System made of?

1. **Controlling system** – Monitors the environment (example: sensors, computers)
2. **Controlled system** – The part being controlled (example: airbag, aircraft)

🛰️ ISRO Example – BHUVAN Portals

- BHUVAN helps monitor **weather conditions** using **live satellite images**.
- It is used to:
  - Track natural calamities
  - Help in advance planning

Real-life Examples:

1. Auto-pilot in Aircraft

- Aircraft is the **controlled system**
- Sensors, computers, etc. are the **controlling system**

2. Airbag System in Cars

- **Airbag** is the **controlled system**
- Sensors and devices that detect collisions are the **controlling system**

3.1.2 Types of Real-time Systems

1. **Clock-based Systems**
   - Actions performed within specific time intervals (real-time clock).
   - Example: Water tank level control.

   Steps:

   - Read tank level (Input)
   - Process using control logic
   - Actuate valve (Output)

   Figure 3.3: Diagram of clock-based tank control system.

2. **Event-based Systems**
   - Triggered by specific **events**, not time intervals.
   - Example: Closing valve when tank is full.
3. **Interactive Systems**
   - Combination of **clock and event-based systems**.
   - Executes tasks within an **average response time**.
   - Examples: ATM, Online booking.

Polling vs. Interrupts

- **Interrupts**: Signal the system to take action.
- **Polling**: System checks periodically if action is needed.

3.2 Characteristics of Real-time Systems

Based on deadline consequence:

1. **Hard Real-time System**
   - Missing deadline = **System failure**

- Example: Airbag system.
- Deadline result: ❌ Not accepted.

2. **Firm Real-time System**
   - Late result = Useless, but **not system failure**.
   - Example: Satellite tracking.
   - Deadline result: ❌ Not accepted.
3. **Soft Real-time System**
   - Late results are still useful.
   - Example: Online ticket booking.
   - Deadline result: ✅ Accepted.

Table 3.3: Hard vs. Soft Real-time System

Hard Real-time	Soft Real-time
Must meet deadlines	Desirable but not mandatory
Delay causes critical failure	Delay is tolerable
Safety/mission-critical	Non-critical
Cost function involved	Often linked to QoS

Feedback Structure (Figure 3.5)

- **Real-time system = Hardware + RTOS + Application**
- Involves sensors, actuators, and processing environment.
- Distributed nature allows flexibility in actions.

Batch Processing vs Stream Processing

Common features:

- Handle non-ending data
- Perform aggregation
- Support real-time decision making

Batch Processing vs Stream Processing – Explained Simply

Point	Batch Processing	Stream Processing
1. How it works	Collects data, stores it, and processes it all **at once** later.	Processes data **immediately** as it arrives (live).
2. Example	Like collecting exam papers and correcting them later.	Like checking answers while the student is writing.
3. Speed	**Slow**, because it waits to collect enough data.	**Fast**, as it works in real-time.
4. Usage	Used in **old systems** or when speed is not important.	Used in **modern apps** like live traffic updates, sensor alerts.
5. Storage	Needs **more storage** to save data before processing.	Needs **less storage**, because it doesn't pile up data.
6. Result Timing	Gives results **after some time**.	Gives results **immediately**.
7. Suitability	Not good for situations needing quick action.	Best for systems needing **instant decisions** (e.g., airbag system).
8. Insight	Cannot detect patterns or changes immediately.	Helps find **patterns or alerts quickly** (e.g., fraud detection).

Hard vs. Soft Real-time System Characteristics

Characteristic	Hard Real-time	Soft Real-time
Response Time	Predictable	Degrades
Load Performance	Required	Optional
Environment	Critical	Non-critical
Redundancy/Recovery	Checkpoint recovery	Long-term recovery
Detection	Error detection	User assisted

3.3 Real-Time Processing Systems for Big Data

3.3.1 Introduction

- **Business Intelligence (BI) systems** today require:
  - Integration of large-scale data.
  - Support for real-time analytics.
  - Ability to handle **fast-growing, diverse, and huge volumes of data** (Big Data).
- **Traditional BI technology** is **not sufficient** for big data because of limitations in handling speed, scalability, and diversity.
- **Cloud computing technologies** (IaaS, PaaS, SaaS) provide **virtually unlimited computing resources and storage**.
- **MapReduce Framework**:
  - A **software framework** for processing large amounts of data in a **distributed and parallel** manner.
  - Applications are written using MapReduce to analyze massive datasets.
  - Works across a **large number of pluggable nodes in clusters**.
  - Provides **fault-tolerance** (system continues working even if some nodes fail).
  - Supports multiple programming languages (Java, Ruby, Python, C++).
  - **Hadoop** is an open-source platform capable of running MapReduce applications.
- **Advantages of MapReduce/Hadoop**:
  - Scalable – handles growing data easily.
  - Cost-effective – open source and efficient.
  - Reliable – provides fault-tolerance.
  - Widely adopted since its introduction in 2004.

3.3.2 Data Integration and Analytics

- Big Data analytics has **two stages**:
  1. **Data Integration**
  2. **Data Analysis**

1. Data Integration

- Data comes from **different sources** → must be integrated into a **single, consistent view**.
- Performed using **ETL (Extract, Transform, Load)**:
  - **Extract** → Data taken from multiple sources.
  - **Transform** → Data converted into a usable format.
  - **Load** → Data stored into a central repository (like a Data Warehouse).
- Tools used: **Informatica, Talend, DataStage, etc.**
- **Enterprise Data Replication (EDR)**: Synchronizes real-time data from different sources.
- **Enterprise Application Integration (EAI)**: Middleware that integrates applications and systems.

2. Data Analysis

- After integration, analysis is done using:
  - **OLAP (Online Analytical Processing)** → Multi-dimensional analysis of data.
  - **In-memory Analytics** → Data stored in RAM for **faster processing**.
  - **Operational Analytics** → Analyzes **live/real-time data**.
  - **Business Process Management (BPM)** → Improves efficiency using analytics results.
  - **Complex Event Processing (CEP)** → Analyzes data streams in **real time** to detect events and patterns.