Advance AI/ML Training – Telecom Data Analytics

MODULE 2 - Mastering automated pipelines, data warehousing, and governance in telecom analytics

Course Outline

- Mix. of theory & practical
  - Module 1: Foundation Recap and Telecom Analytics Basics
  - structured, semi-structured, and unstructured data in telecom with examples
  - Review SQL and Python basics for data collection and cleaning
  - Summarise ETL concepts and role of data pipelines in analytics
  - Visit Telecom predictive Models – Churn, telecom KPIs, and their role in decision making
  - Practical outcomes and use cases

- Mix. of theory & practical
  - ➢Module 2: Advanced Data Engineering & Automation
  - Explain advanced ETL pipeline design and automation.
  - Role of data warehouses and data lakes in telecom
  - Introduce scalable data processing methods for large datasets.
  - Explain importance of data governance, privacy, and compliance
  - Practical outcomes and use cases

Learning Journey

ETL Pipeline Automation

Build automated workflows that run without manual intervention

Warehouses vs Lakes

Understand when to use structured storage versus raw data repositories

Scalable Processing

Handle millions of records efficiently using modern techniques

Data Governance

Protect customer privacy while maintaining compliance

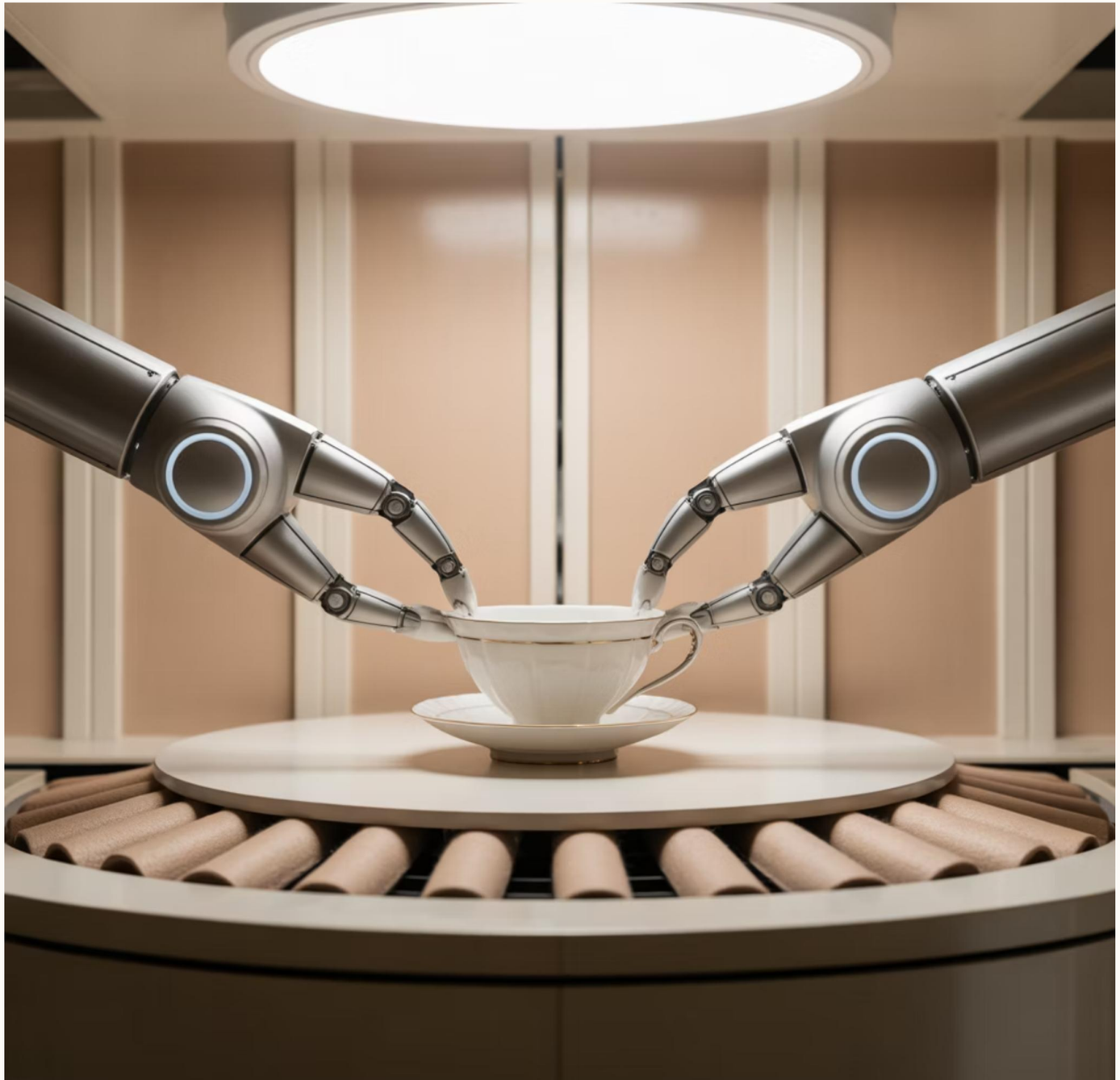
Hands-On Capstone

Integrate everything into a real-world telecom automation project

You will be able to design, build, and automate professional-grade data pipelines that power telecom analytics dashboards.

Session 1: Advanced ETL Pipeline Design

Transform your manual data scripts into powerful automated workflows that run like clockwork. Learn how telecom companies process millions of data points daily without human intervention.



What is an ETL Pipeline?

Extract

Pull raw data from multiple sources like databases, APIs, or files

Transform

Clean, standardize, and enrich the data to make it analysis-ready

Load

Store processed data in warehouses or databases for reporting

ETL pipelines evolve from manual Python scripts you run yourself into fully automated systems that execute on schedules. This transformation is crucial for telecom companies processing hourly tower data, daily billing updates, and real-time complaint dashboards.

Real-World Telecom ETL Examples

Hourly Tower Data Updates

Mobile towers generate usage data every hour. ETL pipelines automatically extract this data, aggregate it by customer, calculate data consumption and call duration, then load it into dashboards for network engineers to monitor capacity.

Daily Billing Dashboard Refresh

Every night at midnight, ETL jobs pull transaction records from the previous day, calculate charges, apply promotional discounts, and update customer billing portals so accurate information is ready when customers log in each morning.

Customer Complaint Analysis

Support tickets stream in continuously. ETL pipelines extract complaint text, classify issues using keyword matching, calculate resolution times, and feed KPI dashboards that executives review weekly to improve service quality.

Automation Tools Overview

Apache Airflow

Industry-standard orchestration platform for complex workflows with dependencies

KNIME

Visual workflow builder with drag-and-drop interface, great for beginners

Python Schedule

Lightweight library for simple automation tasks you can code in minutes

Why Automation Matters

- **Consistency:** Jobs run exactly the same way every time, eliminating human error
- **Accuracy:** Automated validation catches data quality issues before they reach dashboards
- **Scalability:** Handle growing data volumes without hiring more analysts
- **Time Savings:** Focus on insights rather than repetitive data preparation tasks

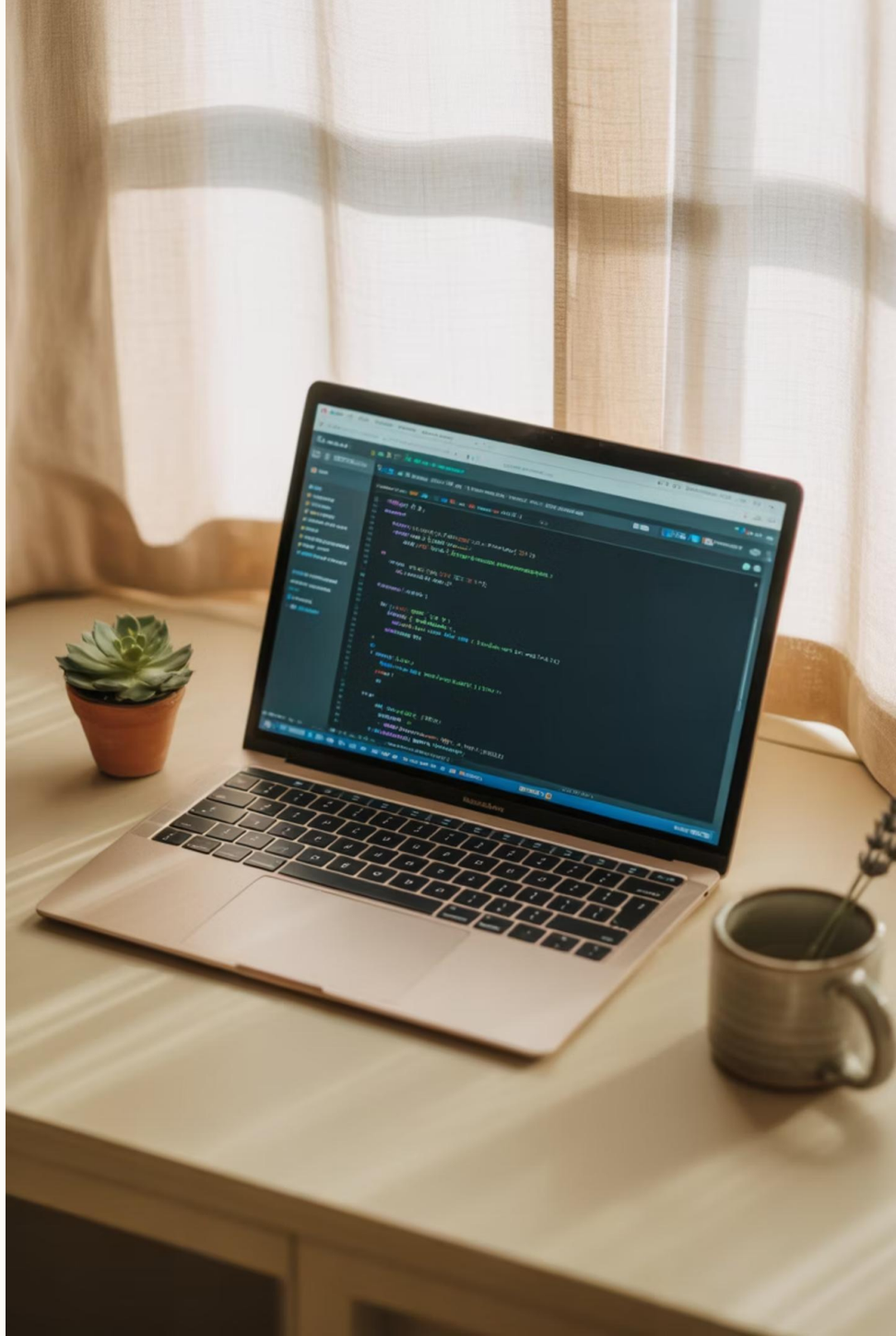
Hands-On: Build Your First Automated ETL

Scenario: You're a telecom analyst who needs to clean customer usage data every day. Instead of running your script manually, let's automate it!

```
import pandas as pdimport scheduleimport timedef etl_job():    print("Starting ETL...")    # Extract: Read raw data    df = pd.read_csv("telecom_raw.csv")        # Transform: Clean and standardize    df.dropna(inplace=True)    df['date'] = pd.to_datetime(df['date'])        # Load: Save cleaned data    df.to_csv("telecom_cleaned.csv", index=False)    print("ETL completed successfully.")# Schedule to run every 10 seconds (simulated)schedule.every(10).seconds.do(etl_job)while True:    schedule.run_pending()    time.sleep(1)
```

REFER MOP: MOP 3 - Exercise 6 - Automated ETL Pipeline

In production environments, tools like **Airflow DAGs** (Directed Acyclic Graphs) replace this simple scheduler with sophisticated workflows that handle failures, send alerts, and manage dependencies between multiple jobs.



Session 2: Data Warehouses vs vs Data Lakes

What is a Data Warehouse?

Structured & Organized

Data is cleaned, validated, and organized into tables with clear schemas before storage

Optimized for Speed

Queries run incredibly fast because data is pre-processed and indexed for analytics

Ready for Reporting

Powers dashboards, reports, and business intelligence tools that executives rely on daily

Popular Tools: Google BigQuery, Snowflake etc.

What is a Data Lake?

Stores Everything

Raw data in any format: CSV, JSON, XML, images, audio recordings, video files, logs, and more

Future-Ready Storage

Keep data now, decide how to use it later. Perfect for machine learning projects that need historical training data

Cost-Effective

Storage is cheap, so you can keep everything without worrying about selecting only certain fields

Popular Tools: AWS S3, Azure Data Lake Storage, Hadoop HDFS

Telecom Use Case: Store raw call detail records, customer support call recordings, and unstructured network logs for future analysis or regulatory compliance

Warehouse vs Lake: Quick Comparison

Feature	Data Warehouse	Data Lake
Data Type	Structured (tables, columns)	Any type (structured, semi-structured, unstructured)
Query Speed	Very fast	Slower, depends on processing
Schema	Schema-on-write (defined upfront)	Schema-on-read (defined when querying)
Use Case	Business intelligence, dashboards, reporting	Raw storage, data science, machine learning
Cost	Higher (optimized storage)	Lower (basic storage)
Examples	BigQuery, Snowflake	AWS S3, Azure Data Lake, HDFS
Users	Business analysts, executives	Data scientists, engineers

Pro Tip: Modern architectures often use both! Raw data lands in a lake, gets processed, then moves to a warehouse for fast querying.

Session 3: Scalable Data Processing

Telecom companies handle massive datasets daily. A single mobile tower generates gigabytes of call detail records (CDRs) every hour. Multiply that by thousands of towers, and you're dealing with data volumes that would crash a normal laptop.

This session teaches you techniques to process millions of records efficiently, preparing you for real-world big data challenges.

Why Telecom Needs Scale

50M+

Daily CDRs

Call Detail Records generated by a medium-sized telecom operator every single day

500GB

Daily Data Volume

Raw usage data flowing from network infrastructure into analytics systems

10K

Queries per Hour

Simultaneous dashboard refreshes and analyst queries hitting databases

Traditional single-threaded processing would take **hours or days** to complete tasks that modern distributed systems finish in **minutes**. Learning scalable techniques isn't optional—it's essential for telecom analytics careers.

Key Scalability Concepts

Parallel Processing

Split large datasets into chunks and process them simultaneously across multiple CPU cores. Instead of processing 10 million records sequentially, process 10 chunks of 1 million records at the same time.

Batch vs Streaming

Batch: Process data in large chunks periodically (e.g., daily reports at midnight). **Streaming:** Process data continuously as it arrives (e.g., real-time fraud detection).

Memory vs Speed Trade-offs

Loading entire datasets into RAM is fastest but requires expensive memory. Reading from disk in chunks is slower but works on any machine. Choose based on your constraints.

Scalable Processing Tools

Apache Spark

Industry standard for distributed big data processing. Handles petabytes of data across clusters of machines. Used by Netflix, Uber, and major telecoms worldwide.

Hadoop

Original big data framework for distributed storage and processing. While newer tools are faster, Hadoop pioneered the techniques that power modern systems.

Dask (Python)

Python-native parallel computing library that scales pandas workflows. Perfect for students because it uses familiar pandas syntax but processes much larger datasets.

For This Course: We'll use chunked pandas processing—a technique that works on your laptop but applies the same principles as these enterprise tools.

Hands-On: Processing Large Datasets Efficiently

Scenario: You have a CSV file with 50 million customer usage records (too large to load into memory at once). Let's process it using chunk-based processing.

```
import pandas as pd# Read file in chunks of 100,000 rowschunks = pd.read_csv('big_telecom_data.csv', chunksize=100000)total_usage =
0heavy_users = 0for chunk in chunks:    # Process each chunk separately    total_usage += chunk['data_used_gb'].sum()    heavy_users +=
(chunk['data_used_gb'] > 50).sum()    print(f"Total data usage: {total_usage} GB")print(f"Heavy users (>50GB): {heavy_users}")
```

REFER MOP: MOP 3 - Exercise 7 - Bigdata Processing
– dont upload *csv file on the google drive, just upload python file or screenshot

What's Happening: Instead of loading 50 million rows at once (which would crash), we process 100,000 rows at a time. Each chunk fits comfortably in memory, and we accumulate results incrementally.

Real-World Connection: This is exactly how PySpark works under the hood, except it distributes chunks across multiple machines instead of processing them sequentially.

Session 4: Data Governance, Privacy Privacy & Compliance

With great data comes great responsibility. Telecom companies handle incredibly sensitive customer information—phone numbers, locations, conversations, browsing history. Mishandling this data can result in massive fines, legal action, and destroyed reputations. This session teaches you how to work with data ethically and legally.

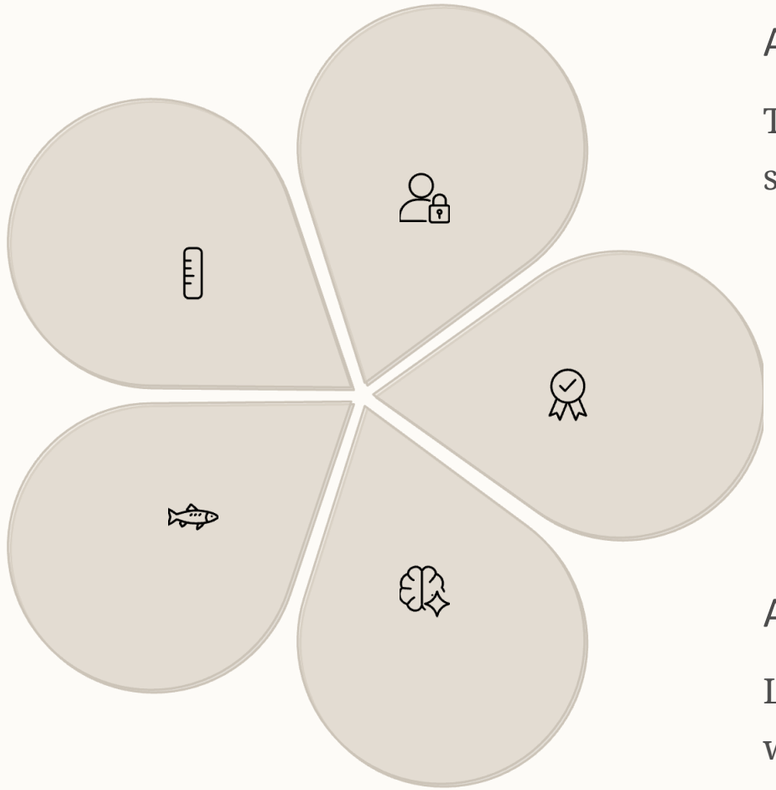
What is Data Governance?

Policies & Standards

Written rules about who can access what data and how

Lifecycle Management

Rules for how long to keep data and when to delete it



Access Control

Technical restrictions ensuring people only see data they need

Data Quality

Processes to maintain accuracy, completeness, and consistency

Audit Trails

Logs tracking who accessed what data and when

Governance isn't just bureaucracy—it's how organizations protect customers, comply with laws, and maintain trust. As an analyst, understanding governance helps you navigate corporate data systems and avoid career-ending mistakes.

Critical Regulations in Telecom

TRAI (India)

The Telecom Regulatory Authority of India mandates that call detail records must be retained for at least one year for law enforcement access, but customer consent is required before using this data for marketing. Violations result in heavy fines and license revocation.

GDPR (Europe)

The General Data Protection Regulation gives EU citizens powerful rights: explicit consent for data collection, the right to access their data, the right to be forgotten (data deletion), and data portability. Non-compliance can cost up to 4% of global revenue.

CCPA (California, USA)

Similar to GDPR but for California residents. Companies must disclose what personal data they collect and allow consumers to opt out of data sales. Telecoms with California customers must comply even if headquartered elsewhere.

Key Privacy Protection Practices

Role-Based Access Control

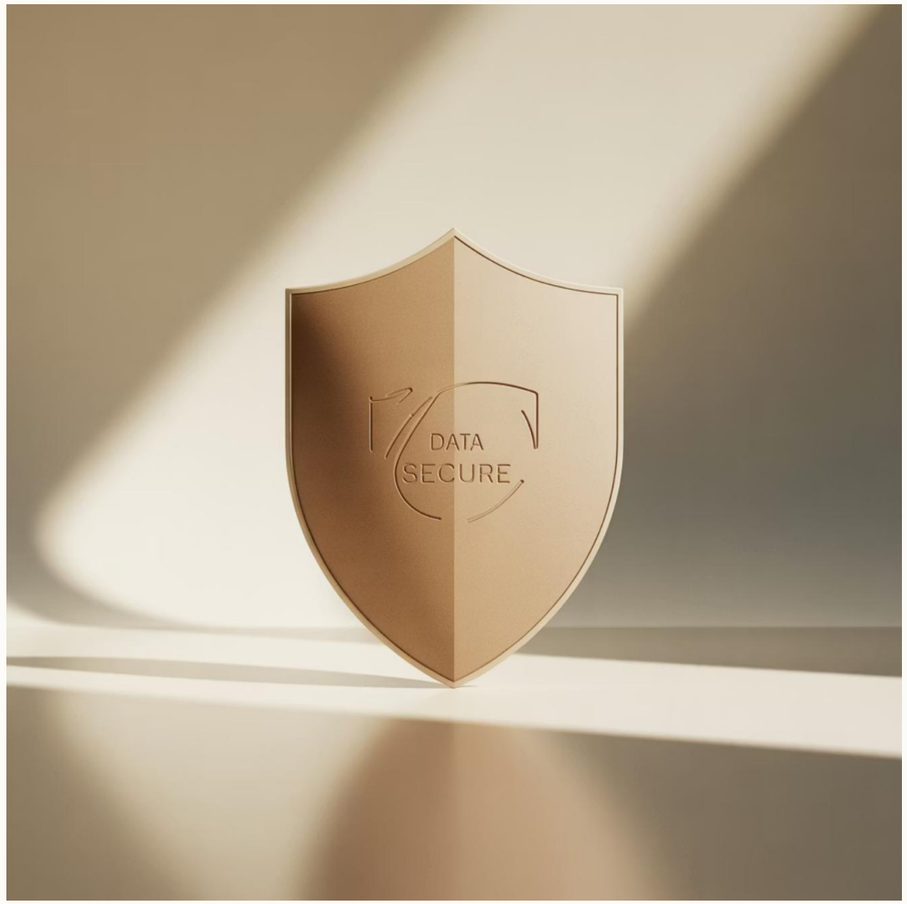
Junior analysts see anonymized data; managers see aggregated reports; only senior compliance officers access raw identifiable records

Encryption & Anonymization

Encrypt data in transit and at rest; replace names with IDs; mask phone numbers in non-essential views

Audit Logging

Every data access is logged with user ID, timestamp, and query executed—critical for investigating breaches



Real Consequence: In 2023, a telecom company was fined $50 million after a junior analyst accidentally exposed customer data by querying a production database without proper anonymization. Don't let this be you!

Hands-On: Simulating Role-Based Access

Scenario: Your company has customer data with sensitive personal information. Different roles should see different levels of detail.

```
import pandas as pddef get_customer_data(role):    # Load full dataset    df = pd.read_csv("customer_data.csv")        if role == "junior_analyst":        # Anonymize sensitive fields        df['name'] = "ANONYMIZED"        df['phone'] = "XXXXX"        df['address'] = "HIDDEN"    elif role == "senior_analyst":        # Show some fields, mask others        df['phone'] = df['phone'].str[:3] + "-XXX-XXXX"    # else: manager sees everything        return df# Test different access levelsprint("Junior Analyst View:")print(get_customer_data("junior_analyst").head())print("\nManager View:")print(get_customer_data("manager").head())
```

Enterprise Reality: Instead of Python functions, companies use **IAM (Identity and Access Management)** systems in AWS, GCP, or Azure that enforce these rules at the database level, preventing unauthorized access before queries even run.

REFER MOP: MOP 3 - Exercise 7 – Role based Access

Session 5: Recap & Assessment

Time to consolidate your learning and demonstrate understanding through a quick assessment, group discussion, and reflection on how these skills prepare you for real telecom analytics roles.

Knowledge Check: Quick Quiz

1. What are the three stages of an ETL pipeline?

   Extract, Transform, Load—pulling data from sources, cleaning it, and storing it in a warehouse

2. When would you choose a data lake over a data warehouse?

   When storing unstructured data, needing flexible schema, planning future ML projects, or prioritizing low-cost storage

3. Name a technique for processing large datasets efficiently

   Chunk processing (reading data in batches)

4. What is GDPR and why does it matter for telecom?

   European regulation giving customers data rights (consent, access, deletion). Non-compliance costs up to 4% of revenue

5. How does role-based access control protect customer privacy?

   Restricts data visibility based on job role—junior analysts see anonymized data, only authorized roles access identifiable information

Key Takeaways & Next Steps

Automation Amplifies Impact

ETL automation frees analysts from repetitive tasks, allowing focus on insights that drive business decisions. This skill separates junior from senior analysts.

Architecture Matters

Choosing between warehouses and lakes, batch vs streaming, parallel vs sequential processing—these decisions impact cost, speed, and scalability.

Scale is Achievable

You don't need enterprise infrastructure to learn big data principles. Chunked processing and vectorization work on laptops but translate directly to Spark and Hadoop.

Privacy is Non-Negotiable

Data breaches end careers and bankrupt companies. Understanding governance, regulations, and access controls protects you and your customers.

What's Next?

Continue your journey: Next module covers advanced analytics and machine learning applications in telecom—predicting churn, optimizing pricing, and detecting network anomalies using the clean, automated data pipelines you built today.

Thank YOU

By ISHAN MARWAH - +91 97178 56777 - Ishan Marwah | LinkedIn