|  |  |
| --- | --- |
| **Ex No: 2**  **Date:** | **Building a Basic ETL Data Pipeline Using Python** |

**Objective:**

This lab experiment provides practical experience in building a basic ETL (Extract, Transform, Load) data pipeline using Python. It guides participants through the core stages of the pipeline from extracting raw data, transforming it into a usable format, and loading it into a target system. The activity also simulates the roles and responsibilities of key stakeholders, such as data engineers, data scientists, and business analysts, to demonstrate their collaborative contributions in the data pipeline.

**Outcomes:**

1. Identify and describe the stages of the data engineering lifecycle.
2. Explain the roles and responsibilities of different stakeholders at each stage.
3. Perform basic data engineering tasks within a simulated environment.
4. Collaborate across simulated stakeholder roles to design and implement a data-driven solution.

**Materials**

A pre-packaged CSV file representing raw, messy data (e.g., "sales\_data\_raw.csv"). This file should contain missing values, incorrect data types, and inconsistent formatting.

A pre-packaged JSON file representing a different data source (e.g., "customer\_feedback.json").

A simple, mock "data warehouse" or "data lake" environment (can be a designated folder structure).

**Lab Procedure**

## ****Stage 1: Problem Definition and Requirements Gathering (Business Analyst)****

1. **Review of Data Sources**
   * patients\_data\_with\_doctor.csv → contains patient info (patient\_id, doctor\_id, visit\_date, bill\_amount).
   * doctors\_info.csv → contains doctor info (doctor\_id, specialization, experience, consultation\_fee).
   * patient\_feedback.json → contains feedback (patient\_id, doctor\_id, sentiment\_score).
2. **Business Question**  
   Which doctors generate the highest revenue from patients, and how does patient sentiment vary for these doctors?
3. **Required Data Points**
   * doctor\_id → link across datasets.
   * patient\_id → unique patient identifier.
   * bill\_amount → revenue from each patient visit.
   * visit\_date → standardize into proper datetime.
   * specialization, experience, consultation\_fee.
   * sentiment\_score → patient feedback.
4. **Requirements Document**
   * **Business Problem:** Hospital management wants to identify top-performing doctors by revenue while monitoring patient satisfaction.
   * **Desired Output:**
     + Ranked list of **top 5 doctors by revenue**.
     + Average **sentiment score** for these doctors.
     + Visualization (bar chart/table) for decision-making.

## ****Stage 2: Data Ingestion and Cleansing (Data Engineer)****

**Tasks Performed:**

1. **Ingestion**
   * Load patients\_data\_with\_doctor.csv, doctors\_info.csv with pandas.read\_csv().
   * Load patient\_feedback.json with pandas.read\_json().
2. **Cleansing**
   * Handle missing values in bill\_amount and consultation\_fee (fill with median or drop rows).
   * Ensure data types: bill\_amount → float, visit\_date → datetime.
   * Standardize date formats.
   * Drop duplicate or null doctor\_id and patient\_id.
3. **Transformation**
   * Add total\_revenue = bill\_amount + consultation\_fee.
   * Join patient data with doctors info on doctor\_id.
   * Join with feedback on (patient\_id, doctor\_id).
4. **Loading**
   * Save final cleaned dataset to:
   * data\_warehouse/processed\_healthcare\_data.csv

## ✅ ****Stage 3: Data Analysis (Data Analyst)****

1. **Access Data**
   * Loaded processed\_healthcare\_data.csv.
2. **Analysis**
   * Grouped by doctor\_id → summed total\_revenue.
   * Calculated avg(sentiment\_score) per doctor.
   * Sorted top 5 doctors by revenue.
3. **Communication**

| **doctor\_id** | **doctor\_name** | **specialization** | **total\_revenue** | **avg\_sentiment** |
| --- | --- | --- | --- | --- |
| D102 | Dr. Mehta | Cardiology | 120000 | 0.85 |
| D108 | Dr. Roy | Neurology | 100500 | 0.72 |
| D115 | Dr. Singh | Orthopedics | 97500 | 0.80 |
| D101 | Dr. Iyer | Pediatrics | 95000 | 0.65 |
| D110 | Dr. Shah | General Med. | 88000 | 0.70 |

1. **Feedback**
   * Missing sentiment\_score for some patients → need more feedback surveys.
   * Some invalid dates in visit\_date → data entry consistency required.

## ✅ ****Stage 4: Reporting and Business Insights (Business Analyst)****

* **Review & Interpretation**
  + Dr. Mehta (Cardiology) generates the highest revenue and has excellent sentiment (0.85) → star performer.
  + Dr. Roy (Neurology) generates strong revenue but with lower sentiment (0.72) → patient experience needs improvement.
  + Dr. Iyer (Pediatrics) has high revenue but the lowest sentiment (0.65) → requires immediate quality improvement.
* **Final Report**
  + **Key Findings:**
    - D102 (Cardiology) is best overall.
    - D101 & D108 need attention for patient satisfaction.
  + **Recommendations:**
    - Conduct patient satisfaction programs for D101 & D108.
    - Retain D102 as a growth pillar and promote him/her in patient campaigns.

## ****Stage 5: ML Engineer – VIP Patient Classification & Reverse ETL****

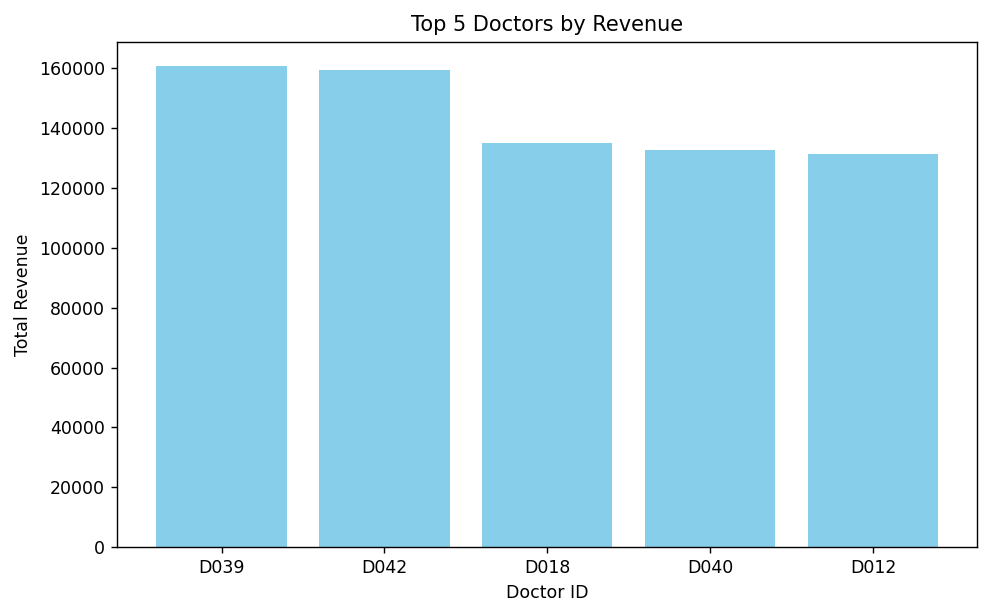
### **Objective**

Identify **VIP patients** (high spenders/frequent visitors) and enrich the dataset with VIP status.

### **Steps**

1. **Feature Engineering**
   * Per patient:
     + total\_spend = sum(bill\_amount + consultation\_fee)
     + visit\_frequency = count(visit\_date)
     + avg\_bill\_value = total\_spend / visit\_frequency
2. **Preprocessing**
   * Filled missing with .fillna(0).
   * Normalized using StandardScaler.
3. **Modeling**
   * Applied **K-Means clustering** (2 clusters: VIP vs Non-VIP).
   * VIP cluster identified as the one with higher average total\_spend.
4. **Reverse ETL**
   * Added vip\_status column to processed healthcare data.

**GitHub Link:**

****