Core Logic Implementation I: Develop project-specific backend models. P1: Adherence detection rules. P2: NLP sentiment analysis. P3: First/Last-touch attribution models.

**Main Idea**

We are developing **the brain (logic)** behind three parts of our data analytics project:

| **Project** | **Logic Type** | **Purpose** |
|---|---|---|
| **P1** | Adherence Detection Rules | Check if patients are following the trial schedule |
| **P2** | NLP Sentiment Analysis | Understand emotions from patient feedback text |
| **P3** | First/Last-Touch Attribution | Find which campaign helped users convert (like enroll or buy) |

**P1 — Adherence Detection Rules**

**Goal**

To detect if a patient **misses doses or visits**.

**Steps**

1. **Collect data**
   Patient events like:
   - When they took medicine (dose_taken)
   - When they visited the clinic
   - Device activity (health band / app check-in)
2. **Set rules**
   Example rules:
   - If patient misses medicine for 2 days → "Missed Dose"
   - If patient doesn't visit for 7 days → "Missed Visit"
3. **Check automatically**
   Use Excel, SQL, or Python to compare **expected date vs actual date**.
4. SELECT patient_id, expected_date, actual_date,
5. CASE

6. WHEN actual_date IS NULL THEN 'Missed Dose'
7. WHEN actual_date > expected_date + INTERVAL '2 days' THEN 'Late Dose'
8. ELSE 'On Time'
9. END AS status
10. FROM schedule;
11. **Generate alerts**
    Show alerts in dashboard — red color for missed, yellow for late.

**Example**

| Patient | Expected Date | Actual Date | Status |
|---|---|---|---|
| P001 | 10 Oct | 11 Oct | On Time |
| P002 | 10 Oct | NULL | Missed Dose |
| P003 | 10 Oct | 13 Oct | Late Dose |

**P2 — NLP Sentiment Analysis**

**Goal**

Understand **how patients feel** from their written feedback.

**Steps**

1. **Collect feedback**
   Example: "I feel tired after the new dose."
2. **Clean text**
   - Convert to lowercase
   - Remove punctuation and stopwords
3. **Analyze sentiment**
   - Use pre-trained AI models (like BERT or TextBlob)
   - Model tells whether text is **Positive, Negative, or Neutral**
4. from textblob import TextBlob
5. txt = "I feel tired after the new dose."
6. TextBlob(txt).sentiment.polarity

7. # negative value → negative sentiment
8. **Store results**
   Save output like:

   | **Patient** | **Text** | **Sentiment** |
   |---|---|---|
   | P001 | “I feel good” | Positive |
   | P002 | “I feel tired” | Negative |

9. **Use insight**
   If many patients are negative → alert researchers to review side effects.

### P3 — First / Last-Touch Attribution

**Goal**

Find **which marketing channel** (email, ad, SMS, etc.) led a user to join the trial or buy a product.

**Steps**

1. **Collect user journey**
   Example:

   | **User** | **Event** | **Date** |
   |---|---|---|
   | U001 | Email | 1 Oct |
   | U001 | Ad Click | 3 Oct |
   | U001 | Signup | 5 Oct |

2. **Apply logic**
   - **First-touch** → first event before signup = Email
   - **Last-touch** → last event before signup = Ad Click
3. SELECT user_id, MIN(event_time) AS first_touch, MAX(event_time) AS last_touch
4. FROM events WHERE event_time < signup_time
5. GROUP BY user_id;
6. **Use for reporting**
   - Helps marketing know which channel works best.

| Channel | Conversions |
|---|---|
| Email | 40 |
| Ad Click | 60 |

## Simple Summary

| Part | What It Does | Example |
|---|---|---|
| P1 | Finds patients not following schedule | Missed dose alert |
| P2 | Reads and classifies patient emotions | "I feel sick" → Negative |
| P3 | Tracks which marketing source caused a conversion | Email → Signup |

## Tools You Can Use

| Task | Easy Tool Option |
|---|---|
| Data Rules | Excel formulas / SQL |
| Sentiment | Python (TextBlob / Hugging Face) |
| Attribution | SQL / Power BI |