**Comprehensive Industry-Oriented Case Study**

**Mining Indian Railway Twitter Complaints**

**Context**

Indian Railways’ Twitter handle (@RailMinIndia) receives thousands of real-time complaints.

Your task is to build a **production-ready mini–analytics pipeline** that cleans, enriches and analyses these tweets to surface actionable insights for operations teams.

❶ **Data Ingestion & Exploration (5 marks)**

| **Marks** | **Sub-Task** | **Expectations (what to show/submit)** |
| --- | --- | --- |
| 2 | **Load & schema check** | Read the CSV(s) into a dataframe; verify column datatypes & null counts. |
| 3 | **Exploratory snapshot** | 5–10 bullet-point observations (e.g., top sources, tweet length distribution). Use df.describe() or a simple bar/line plot. |

❷ **Cleaning & Feature Engineering (15 marks)**

| **Marks** | **Sub-Task** | **Guidelines** |
| --- | --- | --- |
| 4 | **User-mention extraction** | Create mentions column: list of all @handles present. |
| 3 | **Hashtag extraction** | Create hashtags column: list of #tags (lower-cased, duplicates removed). |
| 3 | **URL isolation** | Create urls column(move url to new column); remove URLs from clean\_text |
| 3 | **Emoji / special-char handling** | Strip or translate emojis (💧→“water” etc.) |
| 2 | **Stop-word strategy note** | Brief note on domain-specific stop words you added / kept. |

❸ **Text Pre-Processing with NLTK & spaCy (20 marks)**

| **Marks** | **Sub-Task** | **Requirements** |
| --- | --- | --- |
| 4 | **Tokenisation comparison** | Run both NLTK’s word\_tokenize and spaCy’s tokenizer on 100 random tweets; discuss at least two differences. |
| 4 | **Stemming** | Apply Porter or Snowball stemmer; store in stemmed\_text. |
| 4 | **Lemmatisation** | Apply spaCy lemmatisation; store in lemmatized\_text. |
| 4 | **POS tagging** | Add pos\_tags column (spaCy). Highlight three domain-specific POS issues you observed. |
| 4 | **Pipeline efficiency note** | Show average preprocessing time per tweet (use time module) |

❹ **Named Entity Recognition & Domain Mapping (10 marks)**

| **Marks** | **Sub-Task** | **Expectations** |
| --- | --- | --- |
| 6 | **NER extraction** | Use spaCy’s en\_core\_web\_sm (or larger) to tag entities; create entities column. |
| 4 | **Domain mapping** | Map entities into one of: TRAIN\_NO, STATION, COACH, SERVICE (water, AC, cleanliness, etc.). Submit mapping function |

❺ **Sentiment & Emotion Analysis (20 marks)**

| **Marks** | **Sub-Task** | **Requirements** |
| --- | --- | --- |
| 8 | **VADER / AFINN / NRC tri-model** | Score every tweet with all three lexicons; keep results in separate columns. |
| 4 | **Model alignment check** | Compute agreement rate between lexicons (e.g., % tweets where all three agree on polarity). |
| 4 | **Emotion profiling (NRC-8 or NRC-10)** | For each tweet, record dominant emotion; produce a bar chart of emotion counts. |
| 4 | **Aspect-level sentiment** | Using the entities from section ❹, calculate sentiment per aspect (e.g., Water=–0.43). Present top 3 negative aspects. |

❻ **Insight Extraction for Ops Teams (10 marks)**

| **Marks** | **Sub-Task** | **Deliverable** |
| --- | --- | --- |
| 6 | **Positive vs. Negative nuggets** | For 10 random positive tweets extract the clause that states what passengers *liked*; repeat for negatives (what they *disliked*). Use dependency parsing or simple chunking; highlight in table. |
| 4 | **Action recommendation** | Two concise, data-driven suggestions for IRCTC (≤100 words each). |

**7 - Image PreProcessing (20 marks)**

**Transformations to Apply (on every image):**

1. **Resize** all images to a consistent shape (e.g., 224x224) if any image is smaller then covert to 64x64 all the images
2. **Convert to grayscale all the images**
3. **Apply random brightness adjustments**
4. **Apply horizontal flip**
5. Save each augmented image to a directory named after its class.
   1. sad\_aug1.png
   2. sad\_aug2.png
   3. happy\_aug2.png
6. **Apply edge detection** using skimage.filters.sobel on happy face any 10 and show them in a subplot

**Bonus (Optional)**

❽ **Dashboard Prototype (Streamlit)**

|  | **Feature** | **Minimum Functionality** |
| --- | --- | --- |
|  | **Interactive filters** | Dropdowns for polarity (Positive/Negative) & aspect; list top 3 complaints/praises. |
|  | **Entity-level sentiment heat map** | Heat map or bar chart showing average sentiment per entity group (e.g., Coach > –0.6). |
|  | **Live demo or screen-cap** | Submit app URL (if deployed) *or* a 60-sec screen recording. |