## EDA 流程濃縮摘要

> 交給夥伴時，只要照「**目的 → 具體操作 → 成果 / 後續要做的事**」這個順序講解，他就能快速複製流程。

| 步驟 | 目的 & 核心概念 | 具體做法 | 之後前處理要注意 |
|---|---|---|---|
| **1. 載入並合併三張表** `train + features + patient_notes` | 把診斷詞、病歷全文、標註整合到同一筆資料 | `pd.merge()` 兩次（先用 `feature_num`，再用 `pn_num`） | **保持主鍵唯一**；後面切分資料時用 `case_num` 避免洩漏 |
| **2. 解析** `annotation` / `location` | 原欄位是「字串化 list」，需還原成真正的 Python list | `ast.literal_eval()` → `annotation_list`、`location_list` | 任何轉換失敗回傳空 list，避免後續報錯 |
| **3. 建立** `has_annotation` **標記** | 判斷該筆病歷有無標註診斷詞 | `has_annotation = len(annotation_list) > 0` | 之後可用來做**正負樣本抽樣 / stratified split** |
| **4. 病歷長度檢查** | 確認輸入長度是否超過 BERT 512 tokens | `AutoTokenizer` 算 `tok_len`，畫直方圖 | 若 >512 比例高，考慮長序列模型或滑動視窗截斷 |
| **5. 類別分佈分析** | 總樣本數 & 標註率 是否嚴重失衡 | - `value_counts()` 畫 **Zipf plot**- 排序後畫 **Lorenz curve** & Gini | Gini 越大→越不均；可決定 **loss 加權 / 重新抽樣策略** |
| **6. 單診斷詞標註率** | 哪些 `feature_text` 容易被標註? | `pos_rate = pos / total` 排序觀看 | 幫助挑重點類別做錯誤分析或數據增強 |
| **7. 一致性檢查** | 標註數 $\stackrel{?}{=}$ 位置數 | 比對 `len(annotation_list)` vs `len(location_list)` | 若不相等須人工排查（目前 0 筆） |
| **8. 重複病歷偵測** | 避免同一病歷被重複訓練多次 | 去除空白後做 `MD5` → `pn_hash`，統計出現次數 | 針對 `cnt > 1` 的病歷：- 只保留一筆或- 交叉驗證時分層抽樣 |

---

### 傳達給夥伴的重點

1. **先跑步驟 1–3**，把欄位轉乾淨，`has_annotation` 做完才能正確分群。
2. **步驟 4** 決定輸入長度策略（>512 token 就要截斷或換模型）。
3. **步驟 5–6** 看完不平衡情形，決定 **class weight / sampler**。
4. **步驟 7–8** 做資料品質把關：標註對齊、刪重複。
5. 後續 **資料前處理（tokenize→label encoding）** 時，一定以 `annotation_list` 和 `location_list` 為準，別用原始字串欄位。