Hospital Readmission Prediction and Clinical Entity Extraction Report

This report summarizes my work on predicting hospital readmissions within 30 days and extracting clinical insights from patient discharge notes. The goal was to build a predictive model to identify patients at risk of readmission while also pulling out key clinical details, like diagnoses and medications, from unstructured text.

Approach

The dataset from `Assignment_Data.csv`, which included patient details like age, length of stay, diagnosis codes, and discharge notes, with the target being whether a patient was readmitted within 30 days (`readmitted_30_days`). The dataset was clean—no missing values or duplicates, but the target variable was imbalanced, with far fewer readmissions (1) than non-readmissions (0). This imbalance was a big challenge, as it could skew the models toward predicting non-readmissions.

Data Prep and Feature Engineering

To set up the data for modeling, I have split it into training (70%), validation (15%), and test (15%) sets, using stratified sampling to keep the class imbalance consistent across splits and then engineered features to make the data model-ready:

- Numerical Features: Scaled `age`, `num_previous_admissions`, and `length_of_stay` using `StandardScaler` to normalize them.

- Categorical Features: One-hot encoded `gender`, `medication_type`, and `diagnosis_code` to turn them into usable binary columns.

- Derived Features: Created `has_followup` and `has_surgery` by searching for specific patterns in the `discharge_note` text (e.g., mentions of “follow-up” or “surgery”).

- Text Features: Applied TF-IDF to the discharge notes to capture important terms, creating a sparse matrix of text-based features.

To tackle the class imbalance, I used SMOTE on the training set, which generated synthetic samples to balance the readmission and non-readmission classes. This helped ensure the models wouldn’t just predict “no readmission” every time.

Model Building and Evaluation

I have trained two models—XGBoost and Logistic Regression—using both the full feature set and a reduced set of the top 30 features (selected based on XGBoost’s feature importance). For XGBoost, used GridSearchCV to tune hyperparameters, while Logistic Regression was set up with balanced class weights to handle the imbalance. I have evaluated both models on the validation set using AUC-ROC and F1-Score, which are good metrics for imbalanced data since they balance precision and recall.For the clinical entity extraction task, I have used the Flan-T5 pretrained model for Named Entity

Recognition (NER). I have crafted a detailed prompt with examples to guide Flan-T5 in extracting entities like diagnoses and medications from discharge notes. This was a separate task but tied into the broader goal of making sense of clinical data.

Challenges Along the Way

The limited dataset size made it hard for models to generalize, and the class imbalance was a constant hurdle, even with SMOTE. For the NER task, incomplete outputs (e.g., missing test set results) forced me to rely on validation set metrics and manual checks to piece things together. While applying TF-IDF with n-grams and bi-grams accuracy were dropped off because of sparsity and cant generalise well. Then tried with unigram and it could improve the generalisation.

Readmission Prediction

- Tuned XGBoost (Full Features): On the validation set, it scored an AUC-ROC of 0.595 and an F1-Score of 0.444. The confusion matrix showed it was picking up some readmissions (true positives) but still had false positives and negatives, indicating room for improvement.

- Logistic Regression (Full Features): This model struggled, with an AUC-ROC of 0.35 and an F1-Score of 0.333, suggesting it wasn't capturing the patterns as well as XGBoost.

- Tuned XGBoost (Top 30 Features): With selected features, the AUC-ROC was 0.59 and F1-Score was 0.333, showing a slight dip in performance but still better than Logistic Regression.

- Logistic Regression (Top 30 Features): AUC-ROC improved slightly to 0.375, but the F1-Score stayed at 0.333, confirming XGBoost's edge.

The ROC curves visually confirmed that XGBoost outperformed Logistic Regression, especially with the full feature set. Feature importance from XGBoost highlighted `age` and `length_of_stay` as key predictors, which makes sense—older patients or those with longer hospital stays are often at higher risk of readmission.

Clinical Entity Extraction

The Flan-T5 model did a decent job extracting entities from discharge notes. For example, it correctly identified "prescribed antibiotics" as a medication and flagged diagnoses like "pneumonia." But it sometimes missed subtler mentions, like "follow-up" without clear context, showing it needs more fine-tuning to catch nuances in clinical text.

Practical Implications:

Readmission Prediction: The XGBoost model, with an AUC-ROC of ~0.59, could help hospitals flag high-risk patients (e.g., older patients with extended stays) for targeted interventions, like closer follow-up care or discharge planning. This could cut

readmission rates and save costs, though the model's accuracy isn't high enough yet for fully automated use—doctors would still need to review its predictions.

Clinical Entity Extraction: The NER tool could automate parts of clinical documentation, pulling out key details like medications or diagnoses from discharge notes. This could save time for healthcare workers and integrate into electronic health record (EHR) systems, making it easier to spot trends or ensure medication adherence.

If I had more time or a bigger dataset, here's how I'd level up this work:

1. Better Feature Engineering: With more data, I'd dive deeper into the discharge notes using pre-trained language models (e.g., BERT or ClinicalBERT) to extract richer text features. I'd also create interaction features, like combining `age` and `length_of_stay`, to capture more complex patterns.

2. Advanced Models: I'd try models like LightGBM or neural networks, which could handle the data's complexity better, especially with more samples. Deep learning could be particularly useful for unstructured text or time-series data (e.g., lab results over time).

3. Robust Evaluation: A larger dataset would let me use k-fold cross-validation instead of a single train/validation/test split, giving a more reliable estimate of performance. I'd also expand hyperparameter tuning for XGBoost and test other feature selection methods, like SHAP values, to understand feature impacts better.

4. Improved NER: I'd fine-tune Flan-T5 on a labeled clinical corpus to boost its accuracy, adding more entity types like procedures or allergies. I'd also validate it on the test set and analyze where it's missing the mark. I could also enable a layer using Retrieval Augmented Generation to enhance the accuracy of discharge notes labelling.

5. Handling Concept Drift: If the data spans multiple years, I'd check for concept drift—changes in how features relate to readmissions over time—and build models to adapt to those shifts.

6. Interpretability for Clinicians: I'd focus on making the models more interpretable using tools like SHAP or LIME, so doctors can trust and act on the predictions. Collaborating with healthcare experts would help ensure the features and outputs align with clinical needs.

Final Thoughts

This project shows promise but also highlights the challenges of working with limited, imbalanced data in healthcare. The XGBoost model offers a decent starting point for predicting readmissions, and the NER tool demonstrates how AI can extract actionable insights from clinical text. With more data and time, I could build a more accurate and trustworthy system.