Capstone Project: Hospital Readmission Prediction

Name: Myo Myint Aung Jimmy

Program: General Assembly Singapore – Data Analytics Bootcamp
Deliverables: Two Jupyter notebooks, a Power BI dashboard, Word report, and PowerPoint presentation slides.
Date: 06/11/2025

1) Introduction & Project Setup

This capstone simulates a real engagement where I act as a remote data analyst supporting a hospital's quality team. The objective is to predict 30-day readmissions (binary outcome) for diabetic inpatients, surface top risk drivers, and hand off dashboard-ready outputs for clinical and admin stakeholders.

Dataset : Public, de-identified hospital readmissions data (10 years of inpatient visits; clinical, demographic, and outcome variables). This project was chosen based on my background in biopharmaceutical manufacturing and clinical trial industry experience and it also balances healthcare relevance with manageable scope for an end-to-end pipeline in the bootcamp timeline.

Workflow of my project is as follows:

- **Part 1 (Notebook):** Data acquisition, cleaning, feature engineering, EDA.
- **Part 2 (Notebook):** Model development, evaluation, threshold selection, export of dashboard-ready outputs.
- **Power BI:** One consolidated performance & interpretation dashboard for the selected model.
- **Microsoft Word report:** Business-friendly narrative + visual evidence (this word report).
- **PowerPoint Presentation (next):** 10–12 visual storytelling slides for a 10–15-minute final talk

---

2. Problem Statement & Success Criteria

Problem: Hospital readmissions are expensive and reflect care-continuity gaps. The objective is to identify patients at higher risk of 30-day readmission using admission-time features, allowing early, targeted interventions.

Primary Metric: ROC-AUC for ranking performance across thresholds.

Operational Metric: Recall at the chosen threshold (sensitivity) to minimize missed readmissions, with F1 to balance precision/recall.

I rank with ROC-AUC and **operate** at a tuned threshold prioritizing recall/F1.

Capstone Project: Hospital Readmission Prediction

3. Data Description & Acquisition

Source: Kaggle — Hospital Readmissions (diabetic inpatients). I verified parity between a ZIP-extracted CSV and a Kaggle-API download and proceeded with the ZIP version for consistency.

Raw Shape: 25000 rows × 17 columns (before one-hot encoding and drops).

Target variable: readmitted (True/False).

Key Fields (subset): age band, gender, race, time_in_hospital, counts of lab procedures, medications, prior visits (outpatient, emergency, inpatient), primary/secondary diagnoses (diag_1/diag_2/diag_3), medical_speciality, medication change flags.

Please refer to ***Jupyter Notebook Part 1 HTML – Data Acquisition section*** for more detail.

---

4. Part 1 — Data Cleaning & EDA (Notebook 1)

4.1 Cleaning Strategy

- **Duplicate check:** none detected; no rows dropped.
- **Placeholder text:** standardized all Missing/blank-like entries to 'Unknown' (medical_speciality, diag_1, diag_2 and diag_3). This preserves data while keeping semantics: 'Unknown' = information not recorded.
- **Type normalization:** ensured Booleans for yes/no flags; consistent casing and also ensured categorical/numeric fields are consistent and ready for encoding.

Please refer to ***Jupyter Notebook Part 1 – Steps 1C and 1D*** for more detail.

4.2 Outlier Review & Pruning

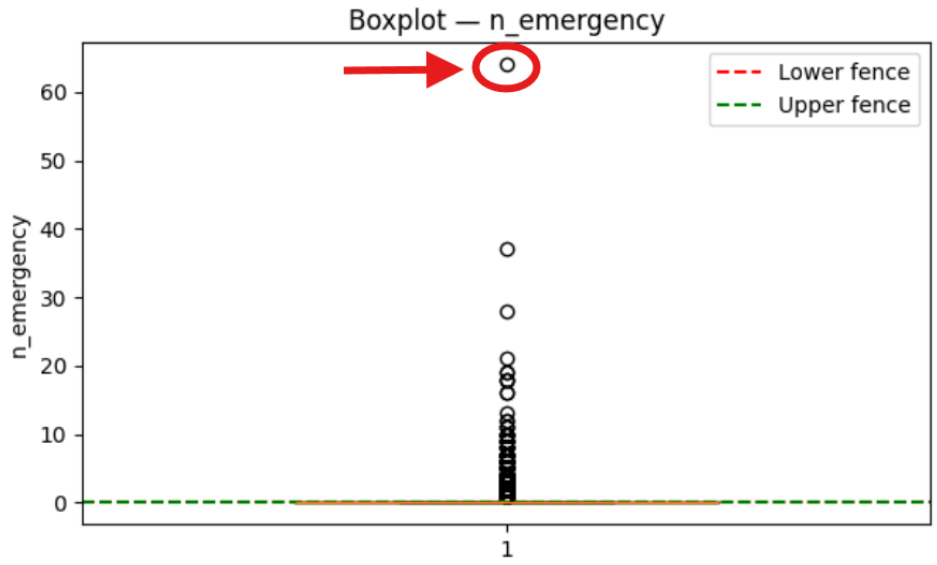
Method: Applied an IQR-style approach with domain checks to trim only extreme, implausible values (e.g., unusually large encounter counts) while keeping clinical variability. IQR rule was applied to flag candidate numeric outliers; then visual verification was done via box/hist charts; finally, conservative trimming of only extreme and very rare values to avoid distorting model training.

Decisions based on Analysis of Boxplot charts:
• n_outpatient > 25 (rare) → removed
• n_emergency > 30 (rare) → removed
• n_inpatient > 12 (rare) → removed
• time_in_hospital > 13 (upper tail) → removed
• n_lab_procedures ≥ 96 (extreme) → removed
• n_procedures left intact to avoid excessive row loss
• n_medications: extreme highs capped by removing only the most extreme rows

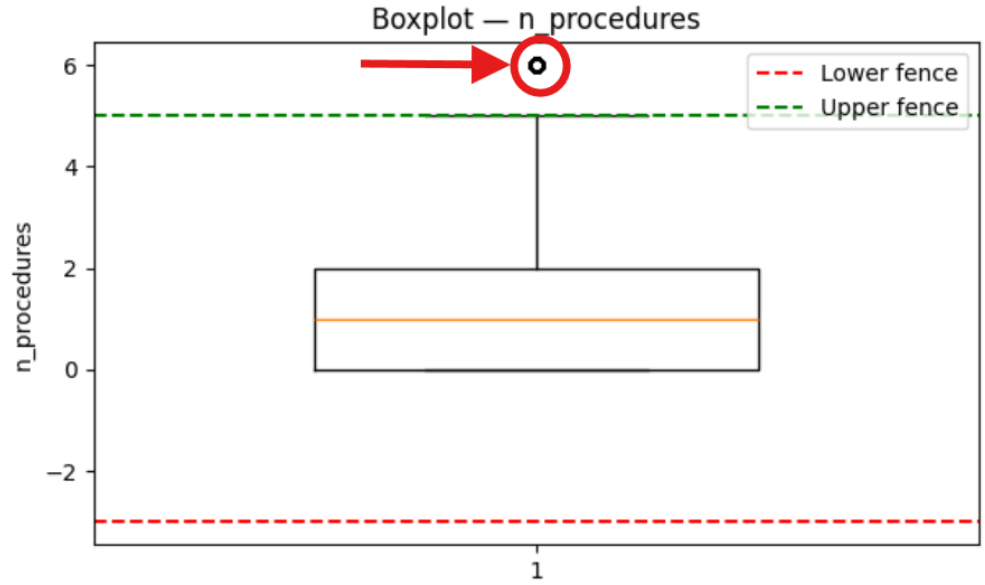
Rationale: Conservative removal reduces noise without discarding meaningful clinical variation.

Capstone Project: Hospital Readmission Prediction

I flagged numeric columns with outlier supports that were tiny (e.g., very rare extreme n_emergency, n_outpatient, etc.) and trimmed only the most extreme values to avoid distorting model fit. This preserved >98% of data while removing implausible tails.



Outlier Illustration — Boxplot for n_emergency feature



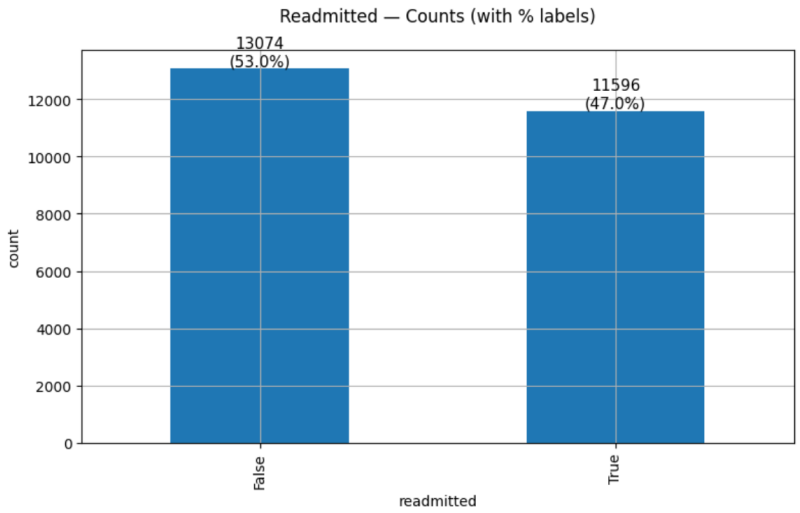
Outlier Illustration — Box/Hist for n_lab_procedures feature

Please refer to ***Jupyter Notebook Part 1 – Steps 2A to 2C*** for more detail.

Capstone Project: Hospital Readmission Prediction

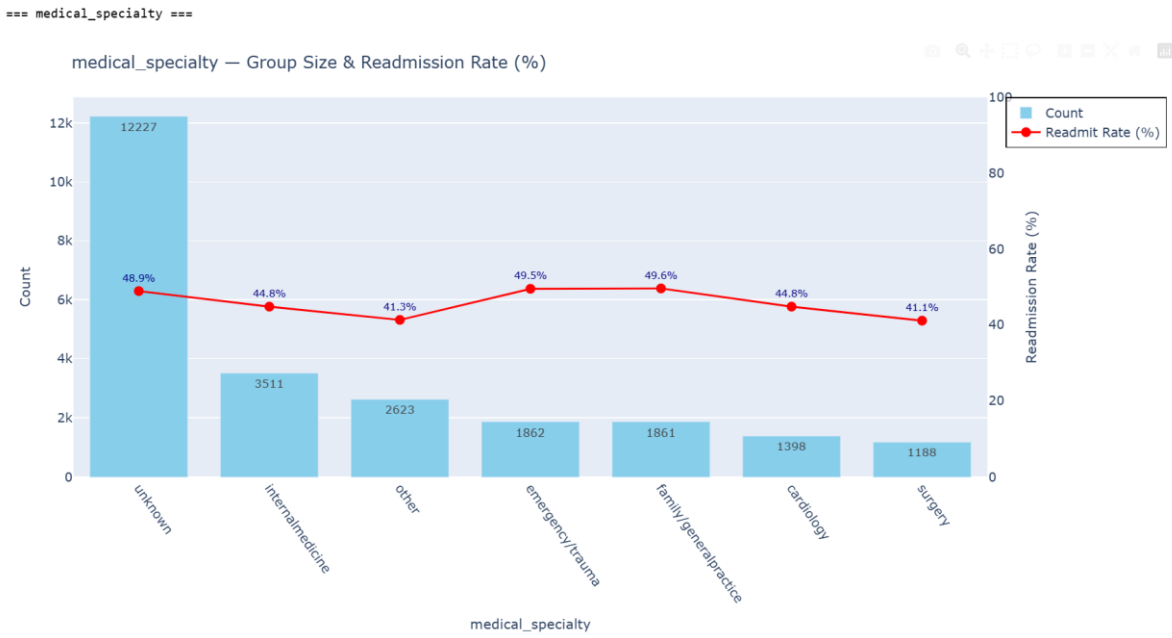
4.3 Exploratory Data Analysis

Target balance: roughly 47% True vs 53% False — moderately imbalanced but not severe; still worth checking class-weighted baselines. Because positives are only ~47%, I later emphasize recall to avoid missed readmissions



Target Overview Illustration — Bar Chart for Readmitted vs Not Readmitted count

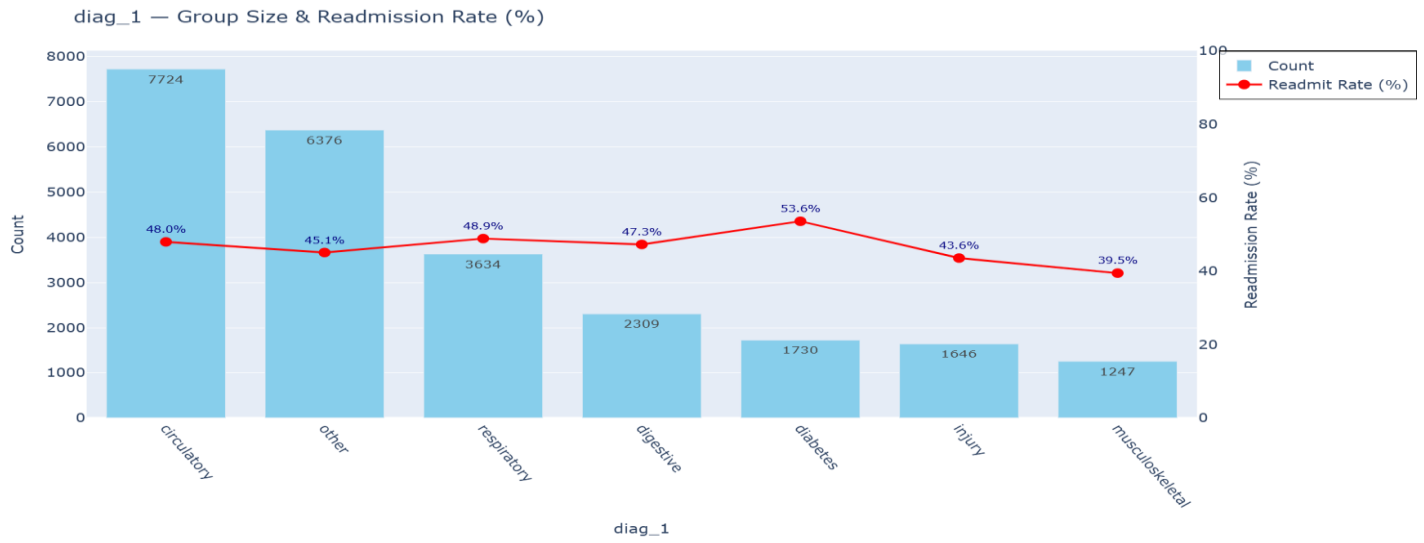
Categorical Features vs Target Feature: Categorical comparisons (**medical_speciality, diag_1/2/3, age**) show pattern differences but with overlap—consistent with multi-factor clinical risk.



Bar + Line Chart for medical_speciality vs Readmission Rate

Capstone Project: Hospital Readmission Prediction

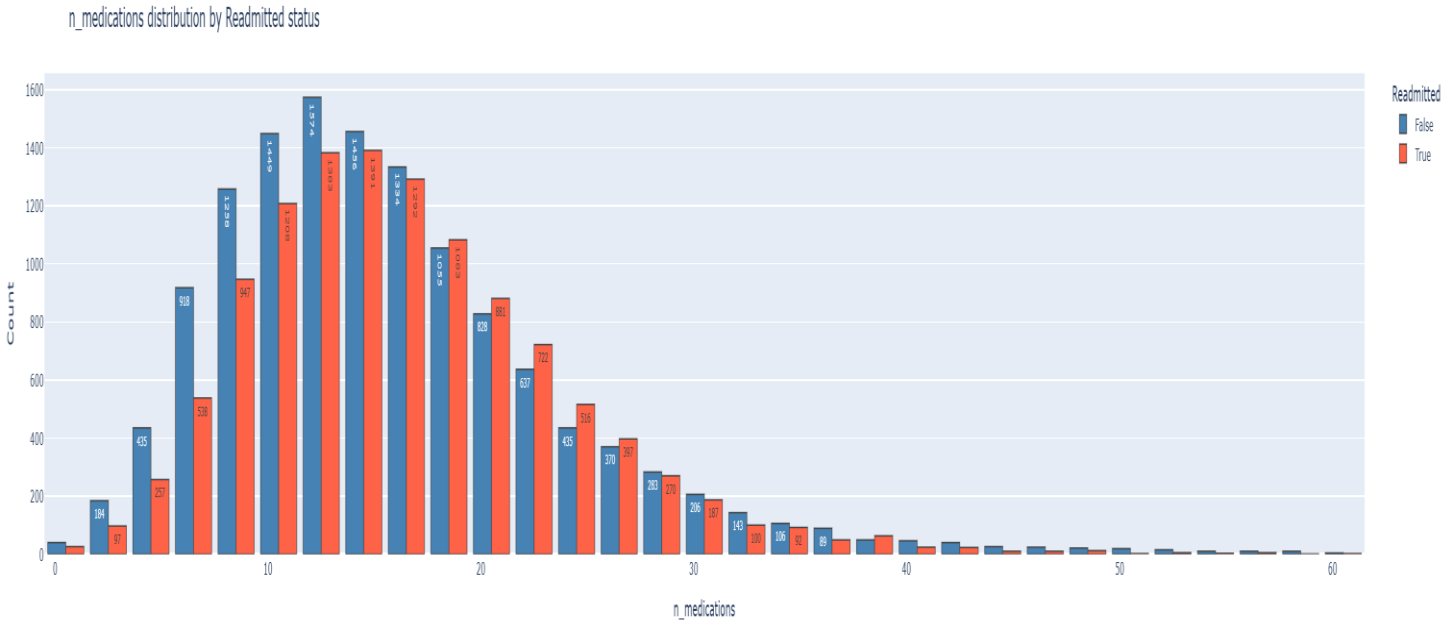
```
=== diag_1 ===
```



Bar + Line Chart for diag_1 vs Readmission Rate

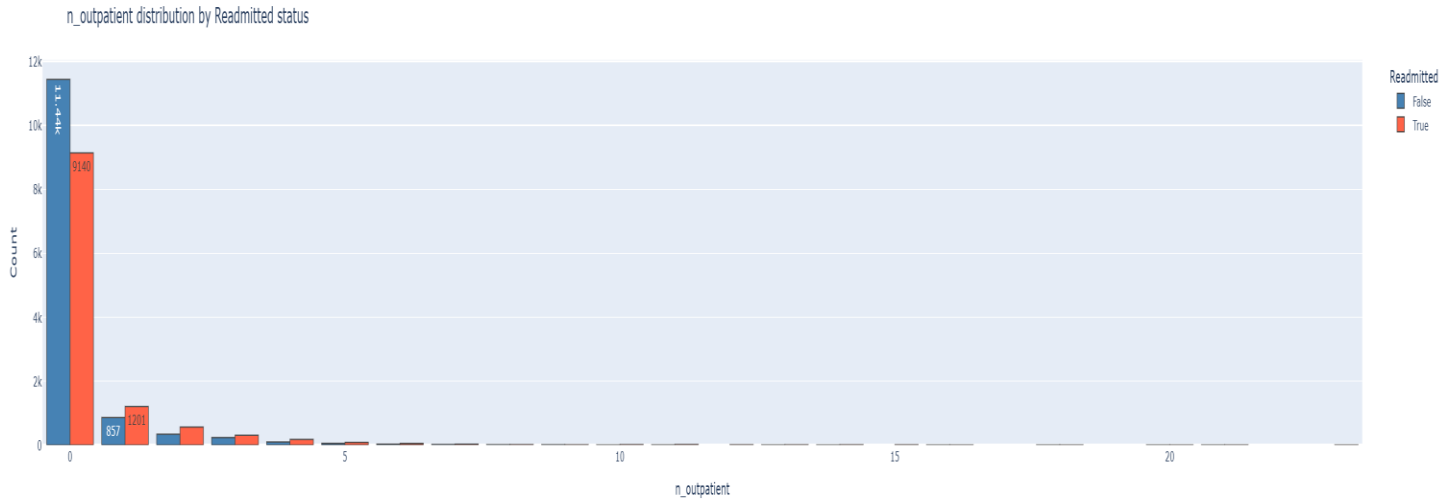
Numerical Features vs Target Feature: higher utilization and resource intensity relate to higher readmission probability (e.g., n_inpatient, n_emergency, time_in_hospital, n_medications).

```
=== n_medications ===
```



Histogram Chart for n_medication vs Readmission Rate

```
=== n_outpatient ===
```



Histogram Chart for n_outpatient vs Readmission Rate

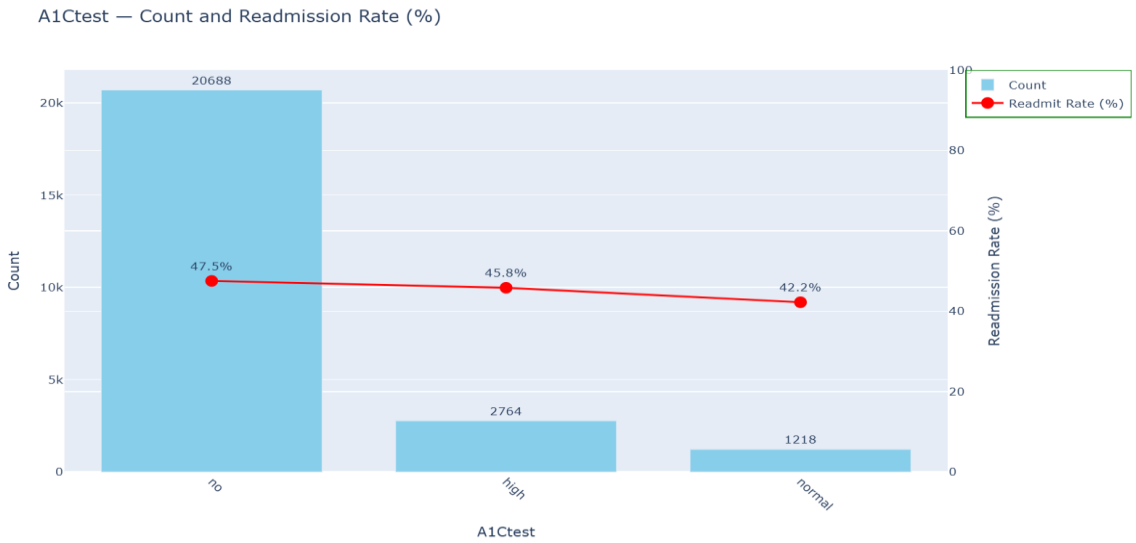
4.4 Simple Clinical Flags

From glucose/A1C test status strings: glucose_done, glucose_high, A1C_done, A1C_high

Purpose: to capture “test performed” and “abnormal” signals in a clear, binary way for modelling and interpretation. This preserves 'no test' vs 'done/normal/high' signaling without high-cardinality categories. Please refer to ***Jupyter Notebook Part 1 – Steps 2E-6*** for more detail.

glucose_test	count	readmit_rate_pct
no	23303	46.8
normal	686	48.3
high	681	52.0

Bar + Line Chart for glucose_test vs Readmission Rate



A1Ctest	count	readmit_rate_pct
no	20688	47.5
high	2764	45.8
normal	1218	42.2

Bar + Line Chart for A1C test vs Readmission Rate

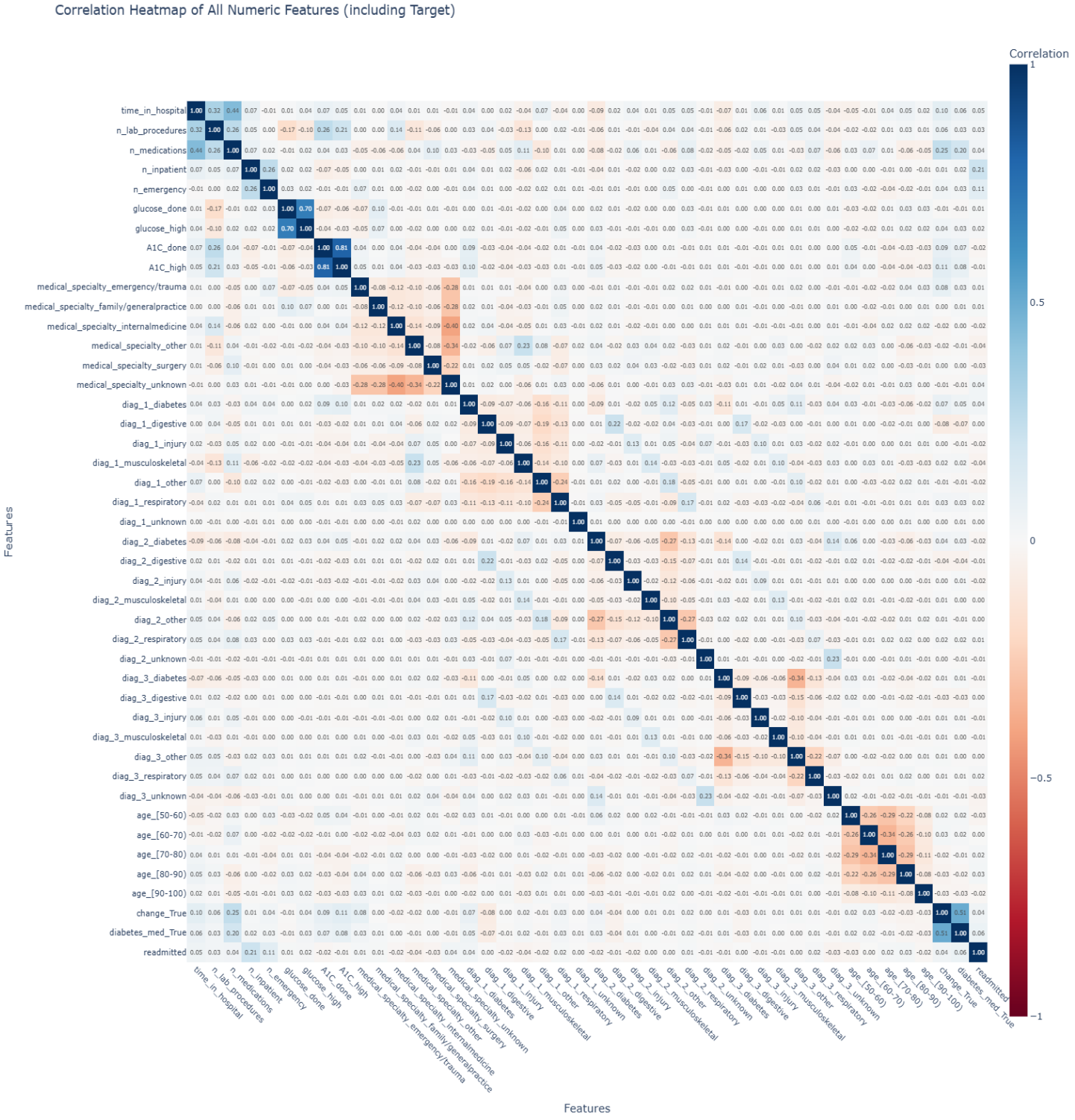
4.5 Encoding & Correlation

One-hot encoding applied to categorical fields; correlation heatmap computed on encoded matrix. No problematic multicollinearity (no $|r| \geq 0.8$ across features). Target correlations are modest (e.g., n_inpatient ~ 0.21), which is realistic for healthcare data.

- Decision: **keep all features** for a broad non-linear model (RF); prune later only if needed.

Please refer to ***Jupyter Notebook Part 1 – Steps 3A to 3B*** for more detail.

Capstone Project: Hospital Readmission Prediction



Correlation Heatmap of all Numeric Features — Encoded features vs. target

Part 1 Conclusion: After performing data cleaning and EDA, data are clean, encoded, and ready for modeling in Part 2 which will be done in Jupyter Notebook 2. These cleaning and EDA steps ensured that the dataset was reliable and interpretable before modeling.

5. Part 2 – Modeling (Setup & Baselines)

5.1 Data Split & Metrics

- **Data:** Used the encoded frame (df_enc) from **Notebook 1**.
- Stratified train/validation split at 80/20 with a fixed random seed to maintain reproducibility.
- **Metrics: ROC-AUC** (threshold-independent), **F1** (balance of precision/recall), **precision**, **recall**, and **accuracy**. In this domain, **recall** and **F1** are emphasized to avoid missing true readmissions.

Please refer to ***Jupyter Notebook Part 2 – Steps 0*** for more detail.

5.2 Baseline Models

Dummy Classifier (Stratified): Establishes the true NAIVE benchmark approximating class proportions. As expected, low ROC-AUC and poor F1/recall.

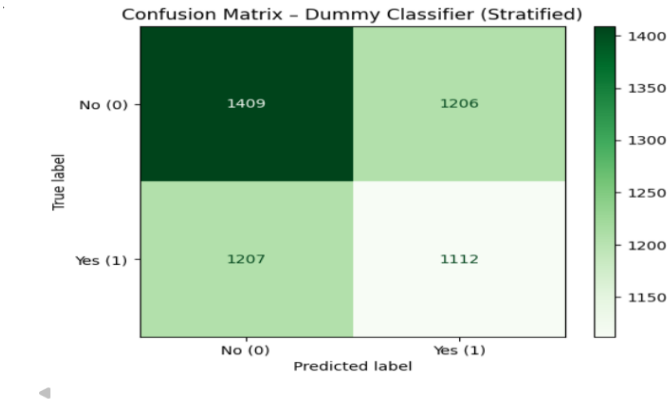
```
=== Step 1A — Dummy Baseline (Stratified) ===
accuracy: 0.5109
precision: 0.4797
recall: 0.4795
f1: 0.4796
roc_auc: 0.5092

Confusion Matrix:
[[1409 1206]
 [1207 1112]]

Classification Report:
              precision    recall  f1-score   support

       False       0.54      0.54      0.54      2615
        True       0.48      0.48      0.48      2319

    accuracy                           0.51      4934
   macro avg       0.51      0.51      0.51      4934
weighted avg       0.51      0.51      0.51      4934
```




```
Confusion Matrix (with labels):
TN: 1409   FP: 1206
FN: 1207   TP: 1112
```

Confusion Matrix for Dummy Classifier (Stratified)

Please refer to ***Jupyter Notebook Part 2 – Steps 1A*** for more detail.

5.3 Logistic Regression

• **Baseline LR**: Penalized (L2) logistic model; interpretable and strong linear baseline.
• **Class-weighted LR**: 'balanced' weights to counter any residual imbalance or hard-to-separate positives.
Observation: Class-weighted LR has similar ROC-AUC to baseline but generally improved recall/F1 for the balanced version—useful when sensitivity matters. This sets a credible linear baseline.

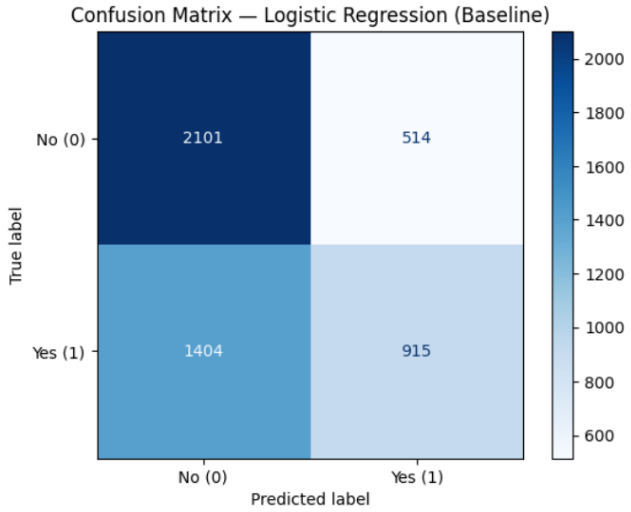
```
=== Step 1B — Logistic Regression (Baseline) ===
accuracy: 0.6113
precision: 0.6403
recall: 0.3946
f1: 0.4883
roc_auc: 0.641

Confusion Matrix:
[[2101  514]
 [1404  915]]

Classification Report:
              precision    recall  f1-score   support

       False       0.60      0.80      0.69      2615
        True       0.64      0.39      0.49      2319

    accuracy                           0.61      4934
   macro avg       0.62      0.60      0.59      4934
weighted avg       0.62      0.61      0.59      4934
```




```
Confusion Matrix (with labels):
TN: 2101   FP: 514
FN: 1404   TP: 915
```

	accuracy	precision	recall	f1	roc_auc	model
0	0.5109	0.4797	0.4795	0.4796	0.5092	Dummy (Stratified)
1	0.6113	0.6403	0.3946	0.4883	0.6410	LogReg (baseline)

Confusion Matrix for Logistic Regression (Baseline)

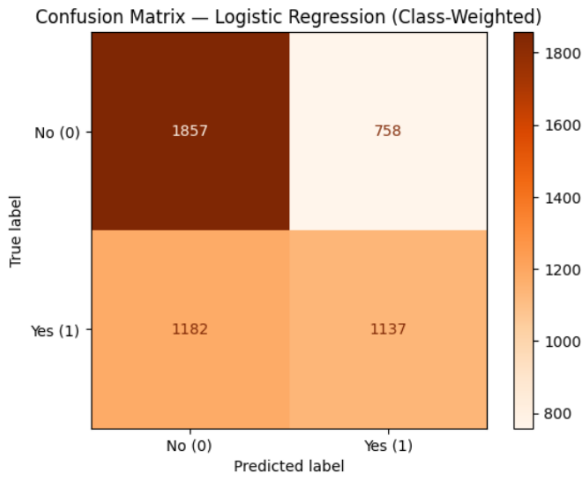
```
=== Step 1B — Logistic Regression (Class-Weighted) ===
accuracy: 0.6068
precision: 0.6
recall: 0.4903
f1: 0.5396
roc_auc: 0.641

Confusion Matrix:
[[1857  758]
 [1182 1137]]

Classification Report:
              precision    recall  f1-score   support

       False       0.61      0.71      0.66      2615
        True       0.60      0.49      0.54      2319

    accuracy                           0.61      4934
   macro avg       0.61      0.60      0.60      4934
weighted avg       0.61      0.61      0.60      4934
```




```
Confusion Matrix (with labels):
TN: 1857   FP: 758
FN: 1182   TP: 1137
```

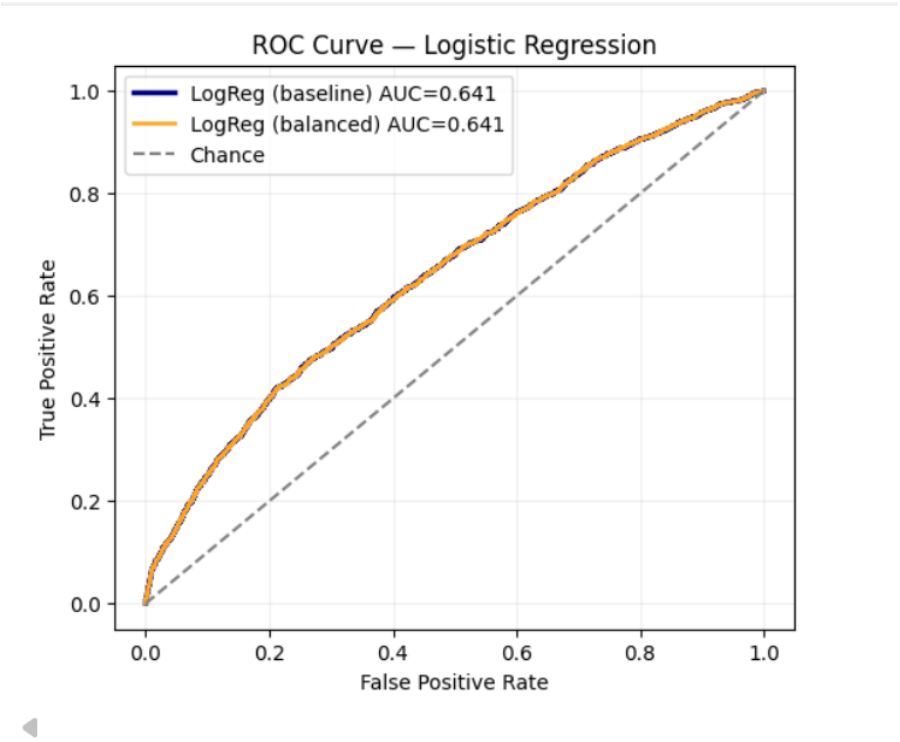
	accuracy	precision	recall	f1	roc_auc	model
1	0.6113	0.6403	0.3946	0.4883	0.6410	LogReg (baseline)
2	0.6068	0.6000	0.4903	0.5396	0.6410	LogReg (balanced)
0	0.5109	0.4797	0.4795	0.4796	0.5092	Dummy (Stratified)

Confusion Matrix for Logistic Regression (Class-Weighted)

ROC Curve Analysis: Logistic Regression Models

ROC Curve Analysis is performed to visualize the ranking performance of both Logistic Regression variants (baseline and class-weighted) using ROC curves. This illustrates the trade-off between sensitivity (true positive rate) and 1 – specificity (false positive rate) across all thresholds, providing insight into each model's ability to distinguish readmitted from non-

readmitted patients. Based on analysis of the chart shown below, bboth logistic regression models (baseline and class-weighted) achieve similar ROC-AUC scores (~0.64), confirming comparable ability to rank patients by readmission risk. Class weighting does not improve ROC-AUC, but shifts the balance between precision and recall at the default threshold. For clinical use, threshold tuning is important to optimize recall or precision based on application needs.




```
ROC-AUC  baseline: 0.641 | balanced: 0.641
```

ROC Curve Chart for Logistic Regression Analysis of both LR variants

5.4Random Forest & Threshold Tuning

RF (baseline) and RF (class_weight='balanced') provided better F1/recall than LR, with comparable ROC-AUC.

Capstone Project: Hospital Readmission Prediction

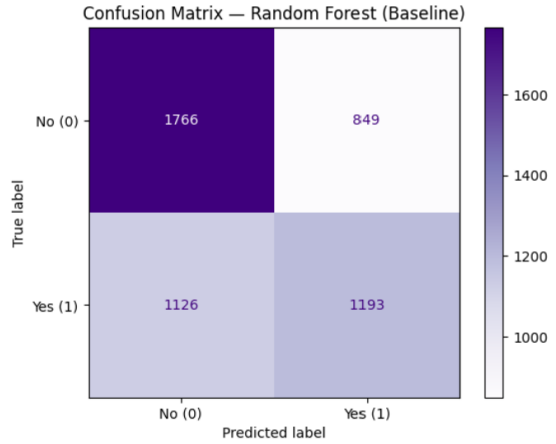
```
=== Step 1C — Random Forest (Baseline) ===
accuracy: 0.5997
precision: 0.5842
recall: 0.5144
f1: 0.5471
roc_auc: 0.6371

Confusion Matrix:
[[1766  849]
 [1126 1193]]

Classification Report:
              precision    recall  f1-score   support

       False       0.61      0.68      0.64      2615
        True       0.58      0.51      0.55      2319

    accuracy                           0.60      4934
   macro avg       0.60      0.59      0.59      4934
weighted avg       0.60      0.60      0.60      4934
```




```
Confusion Matrix (with labels):
TN: 1766   FP: 849
FN: 1126   TP: 1193
```

	accuracy	precision	recall	f1	roc_auc	model
0	0.6113	0.6403	0.3946	0.4883	0.6410	LogReg (baseline)
1	0.6068	0.6000	0.4903	0.5396	0.6410	LogReg (balanced)
2	0.5997	0.5842	0.5144	0.5471	0.6371	RF (baseline)
3	0.5109	0.4797	0.4795	0.4796	0.5092	Dummy (Stratified)

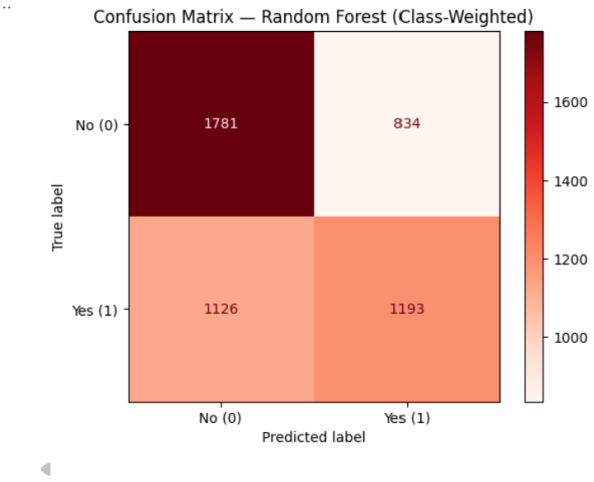
```
=== Step 1C — Random Forest (Class-Weighted) ===
accuracy: 0.6028
precision: 0.5886
recall: 0.5144
f1: 0.549
roc_auc: 0.6354

Confusion Matrix:
[[1781  834]
 [1126 1193]]

Classification Report:
              precision    recall  f1-score   support

       False       0.61      0.68      0.65      2615
        True       0.59      0.51      0.55      2319

    accuracy                           0.60      4934
   macro avg       0.60      0.60      0.60      4934
weighted avg       0.60      0.60      0.60      4934
```




```
Confusion Matrix (with labels):
TN: 1781   FP: 834
FN: 1126   TP: 1193
```

	accuracy	precision	recall	f1	roc_auc	model
0	0.6113	0.6403	0.3946	0.4883	0.6410	LogReg (baseline)
1	0.6068	0.6000	0.4903	0.5396	0.6410	LogReg (balanced)
2	0.5997	0.5842	0.5144	0.5471	0.6371	RF (baseline)
3	0.6028	0.5886	0.5144	0.5490	0.6354	RF (balanced)
4	0.5109	0.4797	0.4795	0.4796	0.5092	Dummy (Stratified)

Confusion Matrices for Random Forest (Baseline) and Random Forest (Class-Weighted)

Please refer to ***Jupyter Notebook Part 2 – Steps 1C-1 and Step1C-2*** for more detail.

I then performed **randomized hyperparameter search** (trees, depth, splits, features) and **selected the best RF** on validation F1/ROC-AUC. Grid Search **(Step 1C-3 in Jupyter notebook 2)** tuned n_estimators, max_depth, min_samples_split/leaf to improve ranking performance.

Optimal Threshold Selection

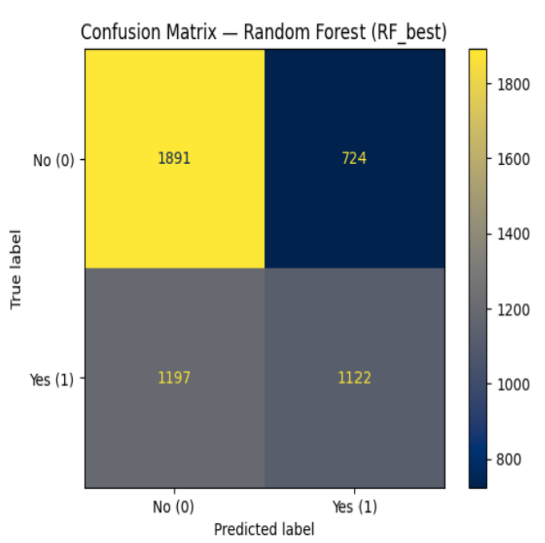
Default 0.50 probability is often not optimal when recall matters. I scanned thresholds and chose 0.373 that maximized F1 on the validation set. Re-evaluating on held-out test delivered the best recall-F1 balance among all tested models.

Capstone Project: Hospital Readmission Prediction

```
Classification Report:
              precision    recall  f1-score   support

       False       0.61      0.72      0.66      2615
        True       0.61      0.48      0.54      2319

    accuracy                           0.61      4934
   macro avg       0.61      0.60      0.60      4934
weighted avg       0.61      0.61      0.60      4934
```



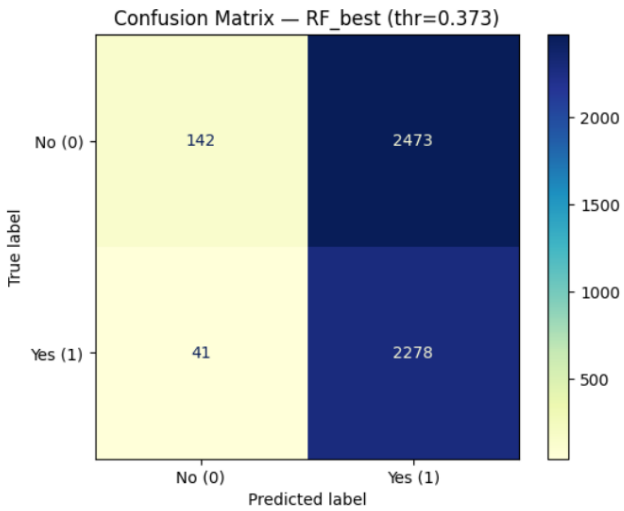

```
Confusion Matrix (with labels):
TN: 1891   FP: 724
FN: 1197   TP: 1122
```

	accuracy	precision	recall	f1	roc_auc	model
0	0.6107	0.6078	0.4838	0.5388	0.6527	RF_best
1	0.6113	0.6403	0.3946	0.4883	0.6410	LogReg (baseline)
2	0.6068	0.6000	0.4903	0.5396	0.6410	LogReg (balanced)
3	0.5997	0.5842	0.5144	0.5471	0.6371	RF (baseline)
4	0.6028	0.5886	0.5144	0.5490	0.6354	RF (balanced)
5	0.5109	0.4797	0.4795	0.4796	0.5092	Dummy (Stratified)

```
Classification Report:
              precision    recall  f1-score   support

       False       0.78      0.05      0.10      2615
        True       0.48      0.98      0.64      2319

    accuracy                           0.49      4934
   macro avg       0.63      0.52      0.37      4934
weighted avg       0.64      0.49      0.36      4934
```




```
Confusion Matrix (with labels):
TN: 142   FP: 2473
FN: 41   TP: 2278
```

	accuracy	precision	recall	f1	roc_auc	model
0	0.4905	0.4795	0.9823	0.6444	0.6527	RF_best (thr=0.37)
1	0.6107	0.6078	0.4838	0.5388	0.6527	RF_best
2	0.6113	0.6403	0.3946	0.4883	0.6410	LogReg (baseline)
3	0.6068	0.6000	0.4903	0.5396	0.6410	LogReg (balanced)
4	0.5997	0.5842	0.5144	0.5471	0.6371	RF (baseline)
5	0.6028	0.5886	0.5144	0.5490	0.6354	RF (balanced)
6	0.5109	0.4797	0.4795	0.4796	0.5092	Dummy (Stratified)

Confusion Matrixes for Random Forest (RF_best) and (RF_best(thr=0.373)

Interpretation: 0.373 means predict True (readmission) whenever the RF's probability ≥ 0.373. This catches more true readmissions (higher recall) at the acceptable cost of some extra false positives.

Final Choice: RF_best (thr = 0.373) for downstream evaluation, calibration check, and deployment artifacts.

Please refer to ***Jupyter Notebook Part 2 – Steps 1C-4 to Step 1C-5 for*** more detail.

Capstone Project: Hospital Readmission Prediction

6. Model Explainability & Communication

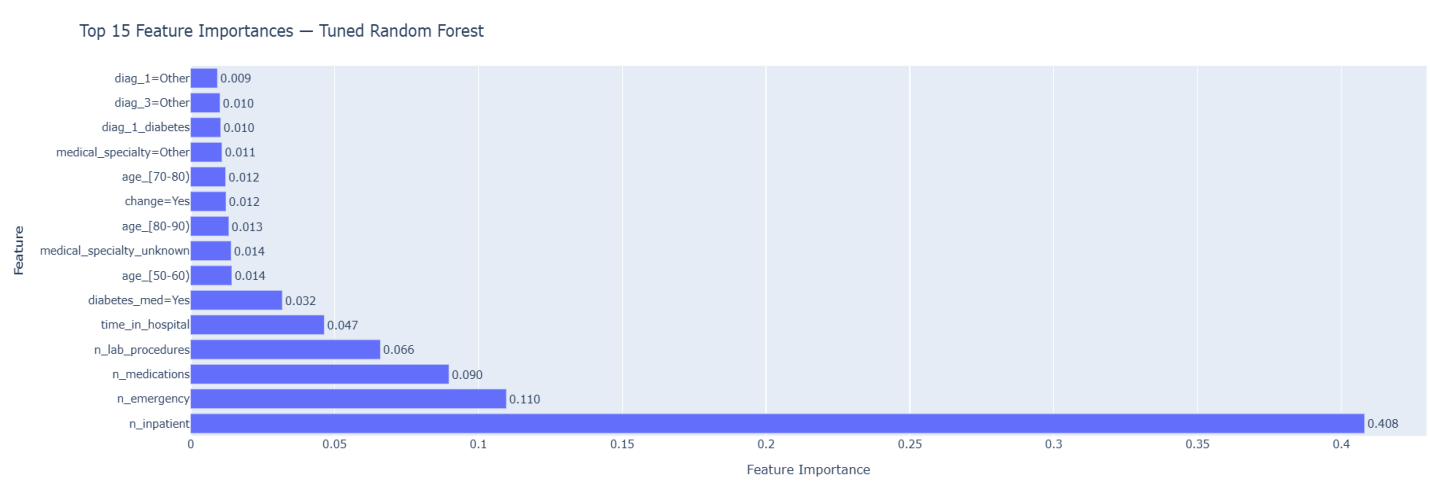
6.1 Feature Importances (RF)

RF global importances show the top drivers:

- **Utilization counts** (e.g., n_inpatient, n_emergency, n_outpatient)
- **Resource intensity** (n_medications, time_in_hospital, n_lab_procedures)
- **Diabetes factors** (diabetes_med, A1C_done, A1C_high, glucose_done, glucose_high)
- Select **diagnosis** and **medical_specialty** one-hot levels also contribute.

These align with clinical intuition: sicker and more complex patients (longer stays, more meds, abnormal labs, frequent prior visits) are more likely to bounce back.

Please refer to ***Jupyter Notebook Part 2 – Step 2A-2*** *for* more detail.



Feature Importance Bar Chart: Top 15 predictors of hospital readmission risk from the tuned Random Forest model: higher bars contribute more to readmission risk in this model

6.2 Calibration & Error Analysis

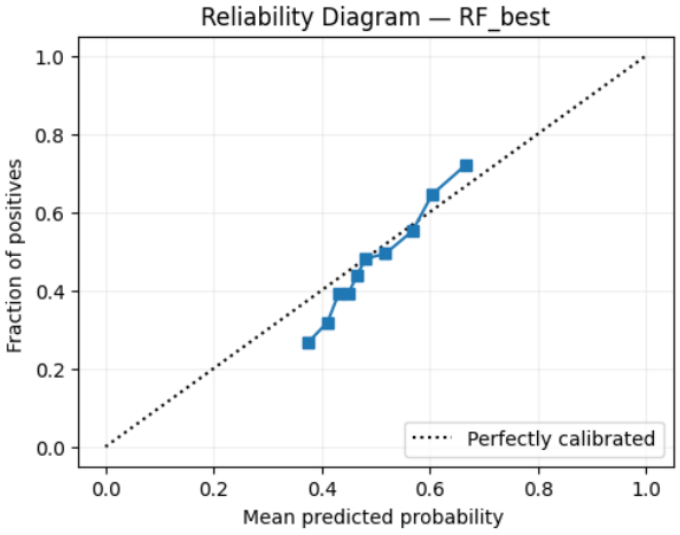
To assess how well the model's predicted probabilities reflect true readmission risk and identify patterns in prediction errors, I plot a reliability (calibration) curve to assess how well the predicted probabilities from the tuned Random Forest model match actual readmission outcomes.

This compute the Brier Score to quantify the accuracy of probabilistic predictions (lower is better). Then I visually inspect whether the model tends to overestimate or underestimate risk, informing potential calibration adjustments. On this dataset, the calibrated variant modestly improved precision at similar recall, but my chosen path kept the simpler threshold-tuned raw RF for clarity and speed.

Capstone Project: Hospital Readmission Prediction

	model	roc_auc	brier	precision@0.373	recall@0.373	f1@0.373
0	RF_best (raw)	0.652746	0.234057	0.479588	0.982751	0.644605
1	RF_best + sigmoid calibration	0.652247	0.231411	0.523035	0.832255	0.642370

```
Brier Score: 0.2341
```



Reliability diagram: Calibration plot: points close to the dashed line mean well-calibrated probabilities; RF_best is slightly under-calibrated in mid-probability ranges (0.4–0.6)

Calibration Comparison: I applied sigmoid (Platt) calibration to further refine probability estimates. This slightly improved the Brier score (to 0.2314), but reduced recall and did not increase F1. Since high recall is my priority for hospital readmission prediction, I retained the original RF_best model for deployment.

Decile Risk Analysis: I grouped patients into ten deciles based on predicted risk. The highest-risk decile (D1) has a readmission rate 1.5× the overall average, while the lowest decile (D10) is only 0.6×. This clear risk separation means the model can help prioritize patients most likely to be readmitted.

Interpretation: My tuned Random Forest model (RF_best @ threshold 0.373) provides strong ranking ability and nearly perfect recall, correctly flagging almost all true readmissions. Precision is moderate, which is acceptable in clinical contexts where missing high-risk cases is more costly than investigating false positives.

Please refer to ***Jupyter Notebook Part 2 – Steps 2B-1 to Step 2B-3 for*** more detail.

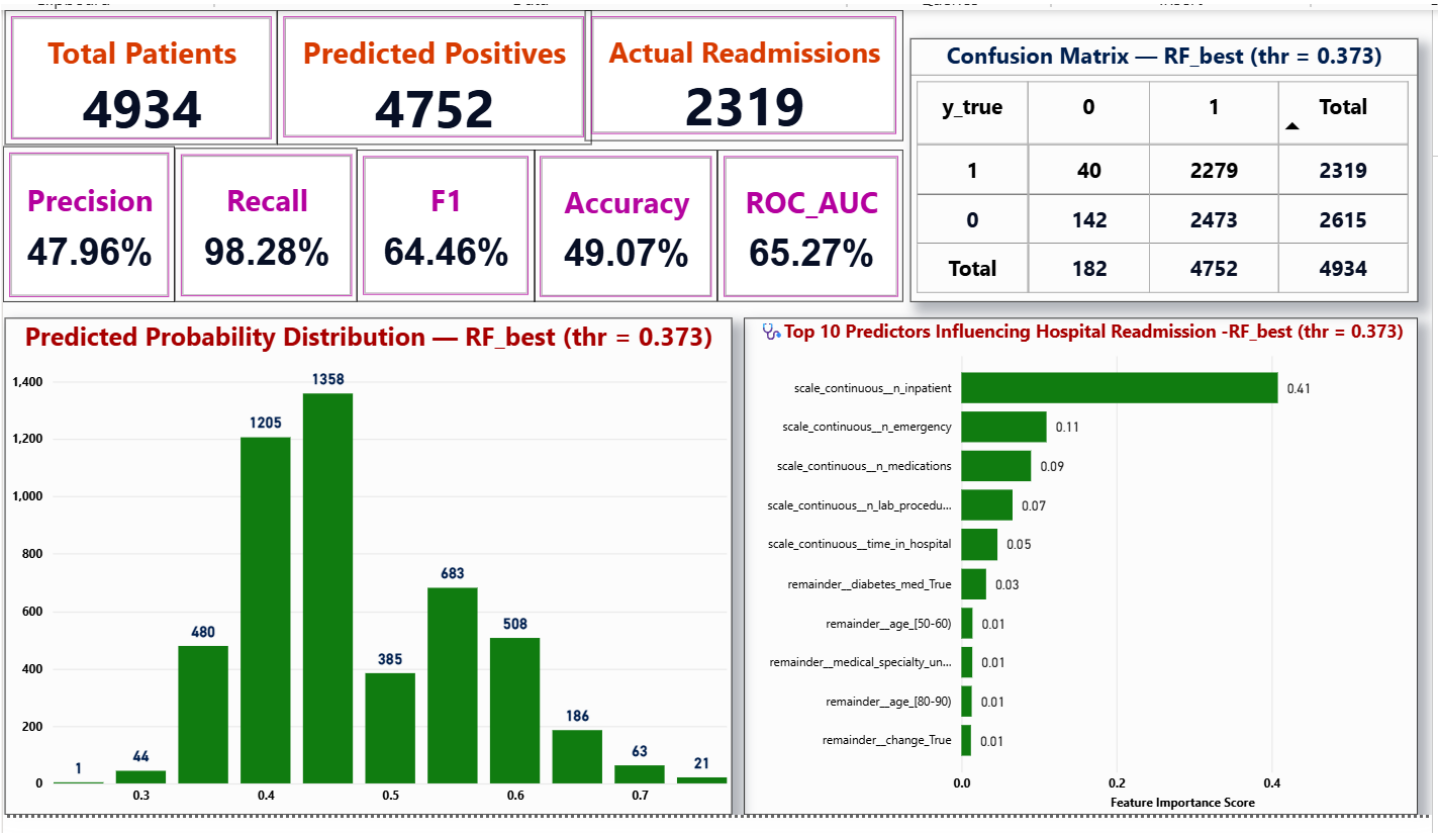
Capstone Project: Hospital Readmission Prediction

7. Power BI Dashboard

Two CSVs were exported for reporting: (1) hosp_dffinal.csv (scored records with predicted_label_037 and probability_037), (2) rf_feature_importances.csv (global feature importance).

Dashboard contents:

1. **KPI Cards:** Accuracy, Precision, Recall, F1, ROC-AUC (all % with 2 decimals).
2. **Probability Histogram:** distribution of predicted readmission probabilities.
3. **Confusion Matrix:** counts of TN, FP, FN, TP at **threshold = 0.373**.
4. **Top 10 Predictors:** from rf_feature_importances.csv (descending importance, no aggregation).



Power BI Dashboard created with two CSV files: hosp_dffinal.csv and rf_feature_importances.csv

Notes:

• hosp_dffinal.csv is a representative sample (rows and columns reduced for presentation). Full inference can be exported similarly if needed.

• Feature importance chart uses the separate CSV and does not join to the scored sample.

Capstone Project: Hospital Readmission Prediction

Dashboard Interpretation & Key Insights

I created the Power BI dashboard using two exports — **hosp_dffinal.csv**, which contains the model's predicted probabilities and threshold-based labels, and **rf_feature_importances.csv**, which records the Random Forest model's global feature importance. The dashboard consolidates model performance and interpretability into a single view. At the selected threshold of **0.373**, the model demonstrates strong recall (**98.28 %**) while maintaining a workable F1 score (**64.46 %**) and acceptable ROC-AUC (**65.27 %**), indicating a useful ranking ability for risk stratification. The trade-off is visible in the moderate precision (**47.96 %**) and overall accuracy (**49.07 %**), both expected outcomes when recall is prioritized to minimize missed readmissions.

The **probability histogram** shows that most predicted scores fall between 0.4 and 0.6, confirming the model's tendency to classify conservatively and capture nearly all true positives. The **confusion matrix** highlights a small number of false negatives but many false positives—an acceptable operational compromise when the aim is to err on the side of patient safety. The **top 10 predictors** plot emphasizes that prior inpatient and emergency visits, the number of medications, lab procedures, and hospital stay length are the most influential drivers of readmission risk. Additional signals such as diabetes medication usage, age bands (50–90 years), and medical specialty categories further contextualize patient complexity and chronic-disease burden.

Overall, this dashboard demonstrates that the model is reliable as a **recall-first triage tool**, suitable for identifying patients who warrant post-discharge follow-up. Hospitals could apply the predictions to trigger low-cost preventive measures (e.g., phone check-ins or care-coordination referrals) while monitoring the alert volume over time. Thresholds can later be refined by unit or risk tier to improve precision as workflow capacity stabilizes. The Power BI visuals ensure technical findings are translated into intuitive insights for stakeholders.

Executive Takeaway from Analysis of Power BI Dashboard

This dashboard confirms that the Random Forest model effectively identifies patients at high risk of 30-day readmission, achieving nearly complete sensitivity (Recall ~ 98 %) while maintaining a reasonable balance of precision and F1. It provides a data-driven early-warning system that hospital teams can use to prioritize discharge follow-ups and reduce costly readmissions. With ongoing monitoring of precision and threshold adjustments by patient segment, the model can evolve into a reliable clinical decision-support tool that enhances both care continuity and operational efficiency.

---

8. Results and Discussion

The final Random Forest model (**RF_best @ threshold = 0.373**) demonstrated a **recall-first performance profile**, consistent with the project's objective to minimize missed readmissions. Across the test set, the key metrics were as follows:

- **ROC-AUC ~ 0.65** – indicating moderate discriminative power and a useful ability to rank patients by readmission risk.

- **F1 ~ 0.64**, **Recall ~ 0.98**, **Precision ~ 0.48**, **Accuracy ~ 0.49** – showing strong sensitivity but modest precision, which is expected for healthcare use cases where patient safety is prioritized over alert volume.

Compared with the baseline logistic regression and weighted/tuned variants, the Random Forest model achieved the highest recall while maintaining a balanced F1-score, confirming its suitability for recall-oriented applications such as early discharge intervention.

The confusion-matrix analysis in the Power BI dashboard confirms that most true readmissions were successfully captured (very few false negatives). Although many false positives remain, this outcome is operationally acceptable if follow-up actions are low-cost (e.g., automated phone calls, tele-check-ins, or medication adherence reminders).

From an interpretability perspective, the top-ranked features — inpatient visits, emergency encounters, medication count, lab procedures, and length of stay — align closely with clinical intuition. These indicators reflect patient complexity and care intensity, which are well-known risk factors for readmission. Additional predictors such as diabetes medication usage and older-age bands (50–90 years) provide further evidence of chronic-condition influence on post-discharge outcomes.

Overall, the results demonstrate that even with a moderate ROC-AUC, a properly tuned Random Forest model can serve as an effective early-warning mechanism for identifying high-risk patients and optimizing limited care-coordination resources.

---

9. Business Recommendations

Based on the model findings and operational priorities, the following data-driven recommendations are proposed for hospital administrators and care-coordination teams:

1. **Adopt a Recall-First Operating Strategy**
    - Continue using a conservative threshold (~ 0.37) to ensure almost all at-risk patients are identified.
    - Deploy low-cost interventions such as discharge reminder calls, medication reviews, or nurse tele-visits for flagged patients.
2. **Pilot Segment-Specific Thresholds**
    - Apply slightly **higher thresholds** for lower-risk wards (e.g., short-stay surgery) to reduce unnecessary alerts.
    - Use lower thresholds for high-acuity groups (e.g., ICU step-downs, chronic diabetes or heart-failure units).
3. **Human-in-the-Loop Validation**
    - Implement a brief **nurse or clinician triage step** to confirm alerts before escalation.

- Use a structured checklist (e.g., comorbidities, medication changes, social factors) to prioritize intervention urgency.

4. **Ongoing Model Monitoring**
    - Track precision, recall, and alert count weekly.
    - If alert fatigue increases, consider incrementally raising the threshold (e.g., from 0.37 → 0.40) and reviewing trade-offs between recall and F1.
    - Log weekly alert volume per ward to tune thresholds by capacity.
5. **Data Quality and Enrichment**
    - Standardize laboratory records for A1C and glucose test results to improve chronic-disease signal accuracy.
    - Clean and standardize medical_specialty fields to reduce "Unknown" values, thereby improving feature reliability.
    - Encourage collection of additional temporal and social-determinant features (follow-up appointments, living situation, caregiver support).

These actions will help the hospital integrate the model safely into daily workflows while progressively improving precision and trust.

---

10. Limitations

Several constraints were recognized during this project:

- The dataset is observational and derived from public hospital records; potential unmeasured confounders may affect prediction outcomes.
- A moderate ROC-AUC (~0.65) suggests that while useful for triage, the model cannot fully capture the multifactorial nature of hospital readmissions.
- The analysis relied on global feature importance rather than patient-level explanations. Techniques such as SHAP values could be applied later for individualized interpretability.
- Temporal linkage across admissions was unavailable since no patient identifiers were included, limiting the ability to model longitudinal trends.
- Model performance might differ in real-world hospital systems with varying patient demographics or documentation quality.

These limitations highlight the importance of continuous validation and re-training before full deployment. Predictions should support, not replace, clinician judgment; monitor for bias across age, race, and specialty groups before deployment.

---

11. Conclusion

This capstone project successfully developed a recall-oriented hospital readmission risk model using a Random Forest classifier tuned at a decision threshold of 0.373. The model identifies nearly all patients likely to be readmitted within 30 days, making it suitable as a triage and early-warning system for care-coordination teams.

The accompanying Power BI dashboard consolidates performance metrics (Recall, Precision, F1, ROC-AUC), a confusion matrix, probability distribution, and feature-importance visualization — providing an accessible and data-driven interface for stakeholders to interpret and monitor results.

From a business perspective, this project demonstrates the practical value of predictive analytics in reducing avoidable readmissions, improving patient outcomes, and supporting resource allocation. Future work should focus on improving probability calibration, integrating SHAP-based explainability, refining thresholds for each care unit, and enriching the dataset with longitudinal and behavioural health data to enhance both precision and clinical trust.

This project marks the completion of my data analytics learning journey at General Assembly, integrating statistical, programming, and visualization skills into a full analytical pipeline.

---

12. Reproducibility & Assets

This project was designed to be fully reproducible and transparent. All datasets, notebooks, and exports are clearly documented and consistently named.

Notebooks

- **00_CapstoneProject_HospitalReadmission_Part1_DataCleaning&EDA.ipynb / HTML**
  Covers data import, cleaning, feature engineering, exploratory analysis, and one-hot encoding.
- **01_CapstoneProject_HospitalReadmission_Part2_MLModels.ipynb / HTML**
  Includes model development, parameter tuning, evaluation, and export of final CSVs for visualization.

Capstone Project: Hospital Readmission Prediction

Key Datasets and Exports

File Name	Description
hosp_df1.csv	**Dataset after outlier removal**
hosp_df3.csv	**Encoded dataset used for model training**
hosp_dffinal.csv	**Final scored file with prediction probability and label at threshold 0.373**
rf_feature_importances.csv	**Feature importance output from the RF_best model**

Model & Environment

- **Algorithm:** Random Forest Classifier (sklearn.ensemble.RandomForestClassifier)
- **Threshold:** 0.373 (chosen for balanced F1 and recall)
- **Language & Libraries:** Python 3.13.7 with scikit-learn 1.5+, pandas, plotly express
- **Visualization:** Plotly (for EDA) and Power BI (for final dashboard)
- **Environment:** Jupyter Notebook (Anaconda distribution)

All files and code were executed under the same environment to ensure full reproducibility and traceability.

---

13. References

1. Kaggle Dataset – *Hospital Readmissions (Diabetic Inpatients)*. https://www.kaggle.com/datasets/dubradave/hospital-readmissions
2. Scikit-learn Documentation – *Dummy Classifier*. https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html
3. Scikit-learn Documentation – *Logistic Regression*. https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
4. Scikit-learn Documentation – *Random Forest Classifier/Regressor*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

5. Data Camp Tutorial – *Random Forests Classifier in Python*.
   https://www.datacamp.com/tutorial/random-forests-classifier-python

6. Kaggle Notebook – *Predicting Hospital Re-Admissions* by Jeyapal.
   https://www.kaggle.com/code/jeyapal/predicting-hospital-re-admissions

7. Kaggle Notebook – *Hospital Readmission EDA and ML (61-49)* by Raphael Marconato.
   https://www.kaggle.com/code/raphaelmarconato/hospital-readmission-eda-and-ml-61-49

8. *BMJ Open Diabetes Research & Care Journal* (2020). "Predictors of Hospital Readmission Among Patients with Diabetes."
   https://drc.bmj.com/content/8/1/e001227