# Hospital Readmission Prediction for Diabetic Patients

## 1. Dataset Description

The main full dataset consists of 50 columns and 101766 samples, where `?` is replaced with `NaN` for missing values. Table 1 shows the 13 numerical columns along with missing and unique counts. Notice the number of missing values. The other 37 categorical columns, including the readmission label, are shown in Table 2. Notice that 3 admission and discharge IDs are considered more categorical than numerical in the following.

Readmission label was converted to binary $0, 1$ from three classes. Thus, the 49 features left results a samples-to-features ratio of 2035.32 for the full dataset, which is plenty of samples to work with. We chose to visualise the distributions rather than showing exact descriptive statistics which only small subset of features have.

Slightly unbalanced label distribution (56:46 for 0:1) as shown in Figure 1 appears manageable. Figure 2 shows the distribution of 8 numerical features except for id and codes, which are mostly right skewed, with some sparse features like outpatient, emergency, and inpatient. There are 4 high cardinality features, with disparate values. Their top 5 frequent values are shown in Tables 3, 4, 5, 6. Notice that heart disease is most common in diag_1 while diabetes is prevalent in the others. Medical specialty has high missing values ratio.

There are 11 regular categorical features, with a variety of distributions shown in Figure 3. There are 24 medication features characterised by high sparsity, shown in Figure 4. To obtain an aggregated view, Figure 5 shows the number of patients per medication, with insulin being the most common. Figure 6 shows the number of medications per patient, with almost 23% not taking any medication, 31% taking 1, and the rest distributed among 2-7 medications.

Figure 7 shows correlation among numerical features, and Figure 8 does the same for regular categorical features. No clear indication of class separation between the two labels. Marginal effects likely originate from class imbalance.

Notice the existence of expired discharge disposition or hospice, which hinder readmission from human reasoning. However, we leave these entries on purpose to see if the model can learn this rule. See Task 4 for more details. So current challenges include missing data, high cardinality features, sparse features, scales for numerical and encoding need for categorical features.

## 2. Data assembling and initial preprocessing

Before everything else, we clean the data by removing encounter_id and patient_nbr columns because they are not informative, and removing the only 3 entries in gender with "Invalid/Unknown" to simplify. The data is separated between label and 47 features. We also conduct a train-test split (with a random seed fixed along the whole coursework), with 50% for training and cross-validation, and 50% for multiple test folds, which remain untouched for this part. A train-validate set of 50881 samples is sufficient.

As preprocessing, the pipeline is created but only fitted by training sets in cross-validation. We impute missing values in columns in Table 7 with "missing" (suggested in [2]) and drop columns with high missing values (Table 9). We chose this strategy dut to the high cardinality of these features. Imputing "missing" to race is a decision made after viewing Figure 9, categories with fewer samples than missing exist. Moreover, we binary encode medication features to alleviate sparsity, with "No" as 0 and the rest as 1, and, except for insulin, which has considerable number of all 4 values according to Figure 10.

High cardinality (about 700) of the 3 diagnoses columns could be problematic, so we choose to convert them to major categories as in Table 8 following the ICD-9 codes [1] and finally one-hot encode 20 values. Other possible strategies exist such as target encode it for the high cardinality.

Finally, all the transformations, categorical encoding, numerical standard scaling, and feature-dropping steps are summarised in Table 9. The encoding handles unseen values too. In the end,

the model produces 148 features, 1 label column, 71234 samples for training-validation.

## 3. Design and Build a Machine Learning Pipeline

Chosen metrics are: accuracy, precision and recall, especially recall as the aim is to identify readmissions from all patients. F1-score as a balance metric and is sensitive to class imbalance. ROC (Receiver-Operator Curve) AUC (Area Under Curve) captures the trade-off across dynamic threshold, and PR (Precision-Recall) AUC is additionally robust to class imbalance.

Apart from SVM as baseline, we choose Random Forest and Gradient Boosting Classifier as additional models because of their well-known track records on tabular data. There are sufficient samples to train these non-linear models, which are well-suited to this problem. Random forest serves as a robust baseline, while gradient boosting usually performs the best in the family. For SVM, we use Linear SVC, as it scales better.

We conduct Hyper-Parameter Optimisation (HPO) with `GridSearchCV` from `sklearn`. Hyperparameter grids are introduced inside search for each model along with the pipeline's preprocessing and modelling steps, avoiding subtle validation leakage. We use 5-folds for cross-validation, and a custom scorer for PR AUC, to measure dynamic thresholds and maintain robustness against imbalance. Both scores and estimators are recorded.

The set of hyperparameters and their effects are visualised. For linear SVC, we compare L1 and L2 penalty with different C values. As Figure 11 shows, L1 with $C = 0.01$ is the best choice, but the two penalties exhibit similar performance. Despite standard deviation (std) not being high in absolute sense (0.0015 at 0.64 scale), cross-fold variations are higher than hyperparameter variations, suggesting the model is more sensitive to data than to hyperparameters.

For Random Forest, we tune n_estimators, max_depth, and min_samples_split. As visualising 3 altogether is challenging, we opt instead to visualise them in 3 subplots to show marginal effects separately. Figure 12 shows that $max\_depth = 15$ is the best choice, with less variability and highest mean PR AUC score. We fix this condition and observe other subsets of hyperparameters in Figure 13, where $n\_estimators = 300$ and $min\_samples\_split = 10$ could be the best spot. This is the best choice in HPO results. Hyperparameter variations are comparable to cross-fold variations, suggesting importance of hyperparameter tuning.

For Gradient Boosting, as Figure 14 shows, $learning\_rate = 0.1$ is the best choice, with significantly higher scores and much lower deviation. Fixing this to 0.1, we observe the complex interaction between hyperparameters in Figure 15. Despite $n\_estimators = 300$ and $max\_depth = 3$ having larger and lower std bound, this is the best choice in CV results. Even larger hyperparameter variation suggests how gradient boosting is sensitive to HPO, particularly learning rate. It also highlights need for further exploration of complex hyperparameter interactions.

Validation scores are consistently derived from the 5 validation folds. Test performance is computed on 5 test folds of 10177 samples each, split using `RepeatedStratifiedKFold` to maintain class distribution and robustness. Best estimators from HPO for each model are used to predict and metrics are calculated. The results are shown in Table 10 for validation and Table 11 for test.

We notice that std is not large overall, so we focus on the mean. Random Forest has a good performance on validation sets, but not as good on test sets, indicating overfitting and poor generalisability. Gradient Boosting has the best performance on test sets, with accuracy of 64.29%, recall of 52.71%, and F1 score of 57.69%. Dynamic threshold metrics like ROC AUC and PR AUC show average performance, with scores of 69.78% and 65.92% respectively.

To better illustrate, we plot confusion matrices and curves for both validation and test folds for all three models in Figures 16, 17, and 18 respectively. Models generally fail to capture the positive instances, yielding high false negative counts and low recall. This is a challenge we predicted at the beginning. Although the curves are above

chance-level, they reveal overfitting in Random Forest, as we have observed in the metrics, being SVM the most robust one.

## 4. Model Interpretation

We obtain the best estimator's coefficient for SVM as well as feature importances for Random Forest and Gradient Boosting. The feature names are extracted and sorted according to absolute coefficient or importance. We visualise the top 10 features in Figures 19, 20, and 21 for SVM, Random Forest, and Gradient Boosting respectively. Since some features are one-hot encoded ids, a translation table is provided in Table 12.

SVM gives more importance to certain discharge and admission details, especially expiry discharge, which is ranked as the second important in other two models. This makes sense since patients who passed away or went to hospice are unlikely to be readmitted. This was anticipated in Task 1 and all models effectively learned this rule. The importance given to diagnosis related to pregnancy is somehow unexpected, but positive features like swing bed condition and discharge to psychiatric site are reasonable.

On the other hand Random Forest and Gradient Boosting place greater emphasis on numerical features such as number of inpatient, emergencies, and diagnosis. This is especially true for number of inpatient visits, which is consistently ranked as the highest in these two models. These features are expected to be important in predicting readmission as one one would expect, for example, that an increase in inpatient and emergency visits are worsen signs and will increase the chances of readmission. Age also appears as a high risk factor which aligns expectations.

Thus, these two ensemble methods share similar importance in features, whereas SVM emphasises categorical features. We still observe some of these discharge and admission details in Gradient Boosting.

## 5. Alternative Machine learning Pipeline

We select Gradient Boosting model to be used in the alternative pipeline. For feature selection, we opt for the wrapper method Recursive Feature Elimination (RFE), from `sklearn`, to assess the combined effects of features. It is embedded in the pipeline after the preprocessing step and before the model step.

The algorithm to perform RFE is logistic regression instead of Gradient Boosting due to lower computational cost and simplicity. We choose 30 features to be selected, which is about 20% of the total features - enough to retain meaningful feature space. We fix this number to avoid grid searching on it, which would be time-consuming.

We perform HPO as before with the alternative pipeline, resulting in the following best hyperparameters: $learning\_rate = 0.1$, $max\_depth = 3$, and $n\_estimators = 100$. Notice that the number of estimators is reduced - likely due to fewer features to learn from, so we fewer trees needed. The mean and std results on validation and test folds are shown in Tables 13 and 14 respectively.

We observe that the performance dropped across all metrics, particularly recall and F1 score, by about 10% and 8% respectively. AUC metrics also dropped by about 5%. This suggests that the feature selection removed some crucial features or combined features important for predicting readmissions. As a result, the model has lost some of its predictive power.

Despite this drop in performance, it is less pronounced in test set. One observation is that model has generalised better to test data, gaining robustness to combat overfitting. This would have been especially beneficial for Random Forest which was heavily overfitting. To further validate this, refer to Figure 22. The gap between validation and test curves are indeed reduced, barely noticeable.

Finally, the model now requires less computing resources at inference time, due to using only 20% of the features. By naively measuring runtime, validation time reduced from $0.50 \pm 0.05$ to $0.40 \pm 0.05$ and test time reduced from $0.47 \pm 0.01$ to $0.36 \pm 0.02$ seconds. This is about 20% reduction, which could be significant in large scale applications if the decrease in performance is acceptable.

## References


[1] CMS.gov. ICD-9-CM Diagnosis and Procedure Codes: Abbreviated and Full Code Titles. 1

[2] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014(1):781670, 2014. 1