Diabetic Patients Readmission Project

Overview

This project utilizes a dataset from a US hospital containing over 100,000 records. While the dataset provides ample data points for training a classical machine learning (ML) model, it suffers from two key challenges:

1. **Heavy class imbalance:** which requires careful handling.
2. **Missing values:** across several columns, necessitating preprocessing.

Additionally, model explainability is a critical requirement for medical applications, which must be addressed.

Data

Missing Values

First, I identified seven columns with missing values. To handle these:

- Columns with missing values exceeding **35% of all rows** were dropped. This threshold is a tunable hyperparameter and can be adjusted for better results.
- For the remaining missing values:
- **Numerical features:** were filled with the median.
- **Categorical features:** were filled with a new category labeled **"Unknown."**

While these are simple imputation strategies, more sophisticated methods could be explored for improved performance.

Imbalanced Dataset

As noted earlier, the dataset is highly imbalanced, which biases ML models toward the majority class. To address this, two balancing techniques were considered:

1. **Undersampling** (reducing the majority class).
2. **Upsampling** (e.g., SMOTE for synthetic minority class generation).

Given the project's demo nature and computational constraints, only **undersampling** was implemented, while upsampling (SMOTE) was left for future exploration as an option implemented in code.

Normalization

Normalization is essential in ML to prevent models from overemphasizing features with larger scales. Here, **standard scaling** was applied to all numerical features.

Models

Five ML models were evaluated for their readmission prediction performance:

1. **Random Forest**
2. **XGBoost**
3. **LightGBM**
4. **SVM**
5. **MLP**

All models were implemented using either **scikit-learn** or their respective specialized libraries (e.g., **xgboost**, **lightgbm**). They were trained and tested on the same dataset, with results documented in the notebook.

Evaluation Metric

Standard classification metrics (e.g., accuracy, F1-score) were reported, but **recall for the <30 days readmission class** was prioritized. This is because:

- **False negatives** (missed early readmissions) pose a greater risk in healthcare.
- Misclassifying patients who either **will not be readmitted** or will be readmitted **after 30 days** is less critical and carries less risk.

Among all models, **SVM achieved the highest recall** for the "<30 days" class. However, its performance may still be insufficient for real-world medical deployment, necessitating further tuning.

Explainability

Explainability is crucial in medical AI applications. While deep learning models often lack interpretability, classical ML models (like those used here) offer better transparency.

For this project, **SHAP (SHapley Additive exPlanations)** was employed to interpret model decisions. One key visualization—the **feature importance plot** for LightGBM—is included in the **images** directory.

Future Considerations

To further enhance model performance, the following strategies could be explored:

1. **Advanced Modeling Techniques**

•Experimenting with more sophisticated models (e.g., ensemble methods, deep learning architectures) may yield better predictive accuracy.

2. **Hyperparameter Optimization**

•Conducting **grid search or brute force** for key hyperparameters (e.g., learning rate, tree depth, regularization) could fine-tune existing models.

3. **Improved Class Imbalance Handling**

•Implementing **upsampling techniques (e.g., SMOTE)** instead of undersampling may better address data imbalance and improve recall for minority classes.

4. **Enhanced Explainability**

•While SHAP's **feature importance plot** was used here, deeper interpretability analysis (e.g., **dependency plots, interaction effects, or decision plots**) could provide richer insights into model behavior.

Note: See test_run.ipyn for a sample running of code

Ali Shendabadi

15 August 2025