Kristianstad University
SE-291 88 Kristianstad
+46 44-250 30 00
www.hkr.se

Course project for **DA380A Machine Learning**
Semester: HT24**,** Year: 2024.
Faculty of Natural Sciences,
Department of Computer Science

Diabetic Patients' Re-admission Prediction Project Report

Ahmed Radwan, Sam El Saati

Course: DA380A Machine Learning

Project Title: Diabetic Patients' Re-admission Prediction

Group: Group 1

Group Members: Ahmed Radwan and Sam El Saati

Examiner: Dawit Mengistu

Abstract

This project focuses on predicting hospital readmissions for diabetic patients using a dataset of approximately 100,000 real-world hospital records. The goal is to identify high-risk patients and improve healthcare resource allocation. Data preprocessing involved handling missing values, outliers, feature engineering (e.g., age_group), and one-hot encoding. We applied statistical and model-based feature selection to retain the most informative features and used machine learning models, including Logistic Regression, Random Forest, and Gradient Boosting, with class imbalance addressed through SMOTE.

The Gradient Boosting model achieved the highest performance. Key predictive features included number_inpatient, number_diagnoses, and number_emergency. Challenges such as class imbalance and high-cardinality features were addressed through careful preprocessing. This work highlights the importance of robust data handling and feature selection in predictive modeling, offering potential for improved patient care and reduced hospital costs.

Keywords

Hospital Readmission, Diabetic Patients, Predictive Modeling, Data Preprocessing, Feature Selection, Classification, Class Imbalance, Machine Learning.

Content

1. Introduction

Hospital readmissions among diabetic patients present a significant challenge for healthcare systems, leading to increased costs and potentially worse patient outcomes. High readmission rates often indicate inadequacies in healthcare management and patient follow-up, emphasizing the need for early intervention strategies. Predicting readmissions allows healthcare providers to allocate resources more effectively, improve patient outcomes, and reduce costs.

This project leverages a dataset derived from 130 US hospitals, documenting approximately 100,000 diabetic patient admissions [1]. By analyzing key features of hospital stays and patient demographics, we aim to develop a predictive model that identifies patients at high risk of readmission. Our objectives are twofold: first, to build accurate predictive models, and second, to identify actionable factors that can reduce readmission rates, ultimately leading to more effective treatment and care strategies for diabetic patients.

Description of Dataset

1.1 Overview: The dataset used is the **Diabetes 130-US hospitals** dataset, containing records from 130 hospitals across the U.S. from 1999 to 2008. The dataset has **101,766 records** and **50 attributes** representing patient demographics, hospital admission details, diagnostic information, medications, and the outcome of readmission within 30 days.

- **Independent Variables**: Includes patient demographics (e.g., age, race, gender), hospital stay information (e.g., admission type, discharge disposition, time in hospital), and medical details such as lab procedures, medications (e.g., insulin, metformin), and comorbidities.
- **Dependent Variable**: The readmitted column, indicating whether a patient was readmitted within 30 days, after 30 days, or not readmitted.

This dataset is relevant in healthcare analytics to predict readmission rates and improve quality of care for diabetic patients, helping reduce hospital readmission rates and healthcare costs.

1.2 Real-World Application

The dataset is highly relevant for healthcare providers looking to reduce hospital readmissions, a critical metric for cost and care quality. By predicting which patients are likely to be readmitted, hospitals can take preventative measures to improve patient care and reduce overall healthcare expenditures.

2. Problem Definition and Motivation

Problem Statement

The primary objective of this project is to **predict high-risk diabetic patients likely to be readmitted within 30 days post-discharge**. By identifying these patients, healthcare providers can better allocate resources, tailor post-discharge care, and address underlying factors contributing to readmission.

Given the impact on healthcare costs and patient outcomes, this task is framed as a **classification problem**, aiming to categorize patients based on their readmission risk.

Motivation
Hospital readmissions can be indicative of systemic healthcare gaps, leading to increased costs and negative health outcomes. Predictive modeling offers a data-driven approach to identifying high-risk patients, enabling timely interventions.

3. Methodology

3.1 Exploratory Data Analysis

- **Techniques**:
    - **Clustering:**

        1. **K-Means Clustering**

From the clustering analysis, we can interpret the results as follows:

1. **Silhouette Score (0.048)**: The low silhouette score indicates poor clustering, as it suggests that the clusters are not well-separated and there is significant overlap between them. Ideally, a higher silhouette score (closer to 1) would reflect more distinct and meaningful clusters.

2. **Davies-Bouldin Index (3.010)**: This index quantifies how similar clusters are to each other, with lower values indicating better clustering quality. A Davies-Bouldin index above 2 is generally considered suboptimal, and in this case, a score of 3.010 suggests that the clusters may not represent meaningful groupings.

3. **PCA Plot Visualization**:

    - **Overlap**: The visualization shows considerable overlap between clusters, especially between the orange and blue clusters, indicating that the clusters may not be well-separated in the feature space.
    - **Cluster Separation**: The clusters are not well-separated, and there is no clear distinction between the groups. In well-defined clusters, we would expect to see distinct groups with minimal overlap.
    - **Cluster Size**: The green cluster has very few points, suggesting it may represent outliers or noise rather than a meaningful grouping.

The clustering analysis does not appear very informative based on these metrics and the visualization. This could indicate that the data might not have a natural clustering structure.

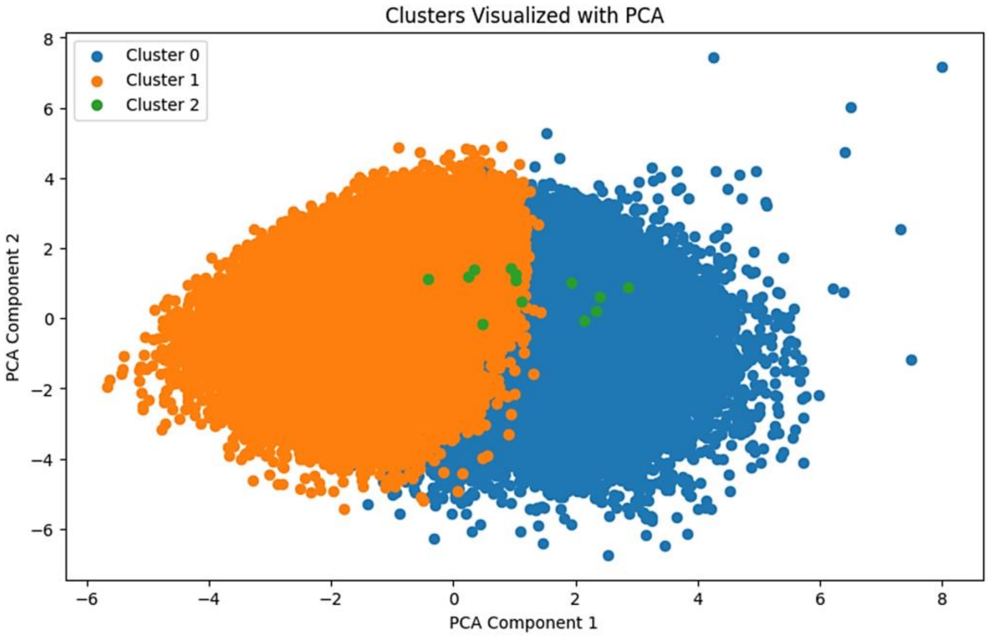


Figure 1 - Clustering to visualize classes

2. DBSCAN Clustering:

From the DBSCAN clustering analysis, the following points can be observed:

1. **Silhouette Score (-0.102)**: The negative silhouette score indicates poor clustering quality, as it suggests that many points are likely misclassified or belong more closely to other clusters than to their assigned clusters. Ideally, a positive silhouette score closer to 1 would indicate better-defined clusters.

2. **Davies-Bouldin Index (1.042)**: The Davies-Bouldin index is relatively low, which might suggest better clustering compared to previous attempts, but in this case, the low silhouette score takes precedence, indicating suboptimal clustering.

3. **Cluster Visualization (PCA Plot)**:

    - **Cluster Overlap**: The PCA plot shows that clusters are not clearly separated. There is a high degree of overlap, making it difficult to distinguish between the different clusters in the data.
    - **Cluster -1**: DBSCAN assigns -1 to points considered noise. In this plot, a significant portion of the data is labeled as -1, which could indicate that these points don’t fit well into any cluster according to DBSCAN's density-based criteria.
    - **Other Clusters**: Although some clusters (0, 1, 2, and 3) were identified, they are not distinct or well-separated. This suggests that the data may not have a clear density-based structure suitable for DBSCAN clustering.

The results indicate that DBSCAN did not identify well-defined clusters in the data. The clustering appears to be uninformative, possibly due to a lack of natural cluster structure. Further preprocessing or trying other clustering methods (such as hierarchical clustering) might yield better insights.

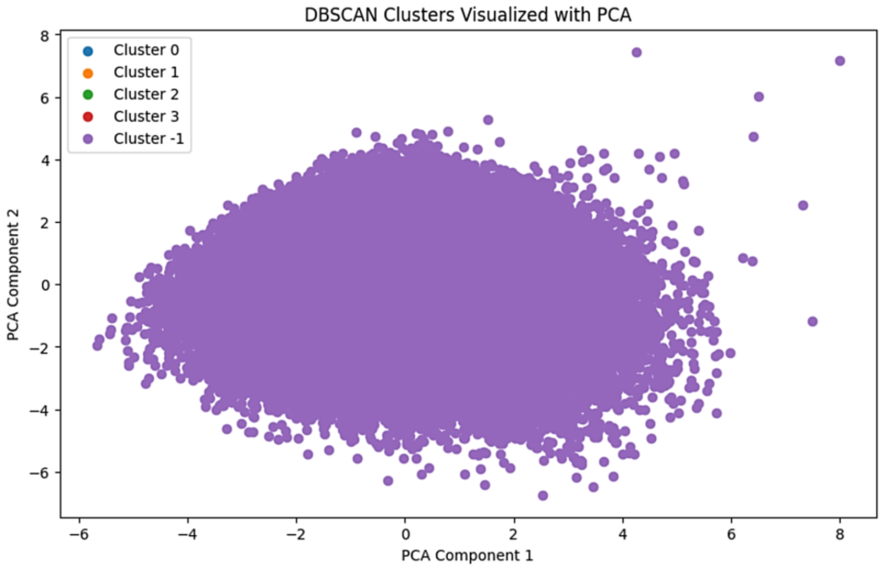


Figure 2 - DBSCAN Clustering to visualize classes

- **Univariate Analysis**: Histograms and boxplots were used to understand the distribution of key features. For instance, the `age` feature revealed a predominance of older patients (e.g., `[70-80)` was the most common age group) and race showed the dominance of the Caucasian race.

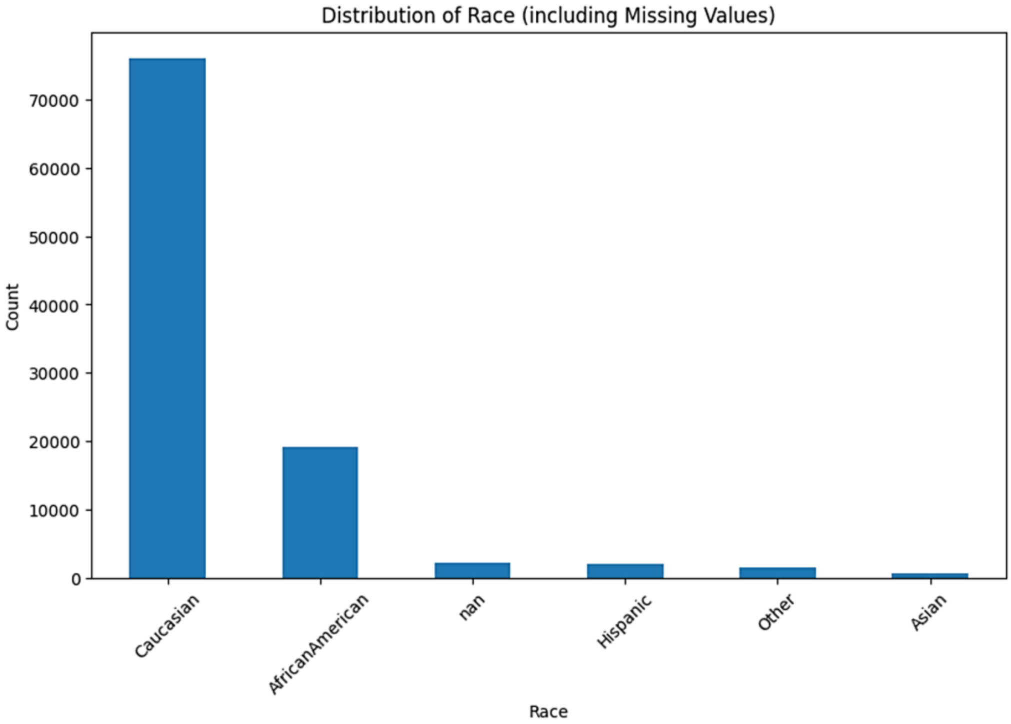


Figure 3 - Univariate analysis for Race feature

- **Bivariate Analysis**: Correlation matrices and scatter plots identified relationships between `A1Cresult` and `readmitted` status.

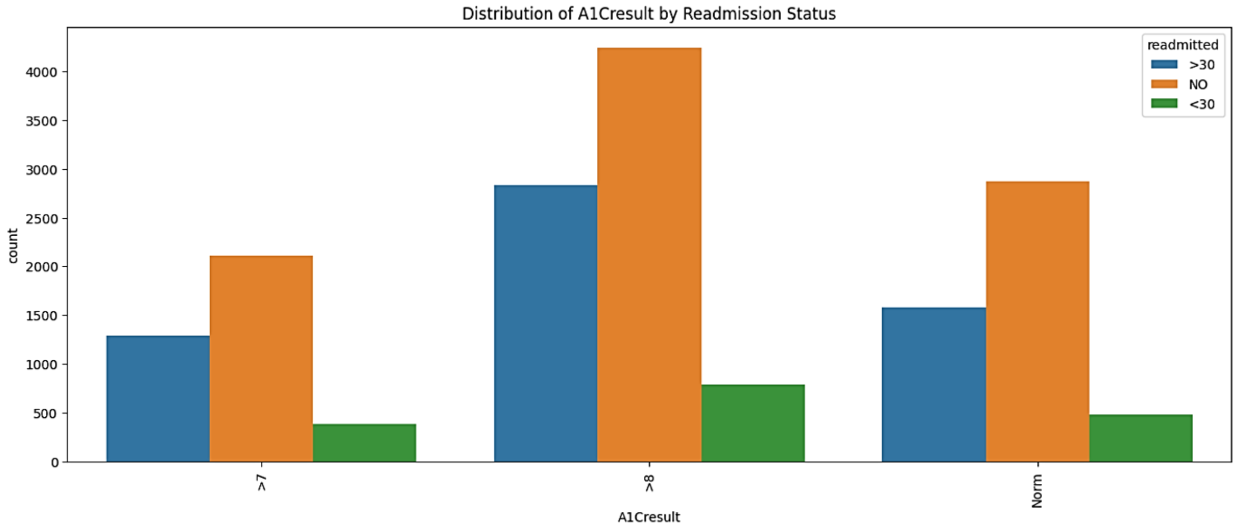


Figure 4 - Bivariate analysis (A1Cresult vs. readmitted)

- **Class Imbalance**: Only about 11% of records were labeled as `readmitted`, highlighting the need for strategies to address class imbalance.

3.2 Preprocessing Techniques

- **Handling Missing Values**:
  - For `payer_code`, missing values were imputed with "Unknown" to preserve data while minimizing bias.
  - Mode imputation was employed for race and other categorical features, ensuring data consistency without dropping records.
  - Features with a high percentage of missing values (like weight) were dropped.
- **Outlier Management**:
  - Extreme values were capped using standardization methods for numerical features like `num_procedures` and `num_medications` to reduce the skewness of the data and enhance model stability.

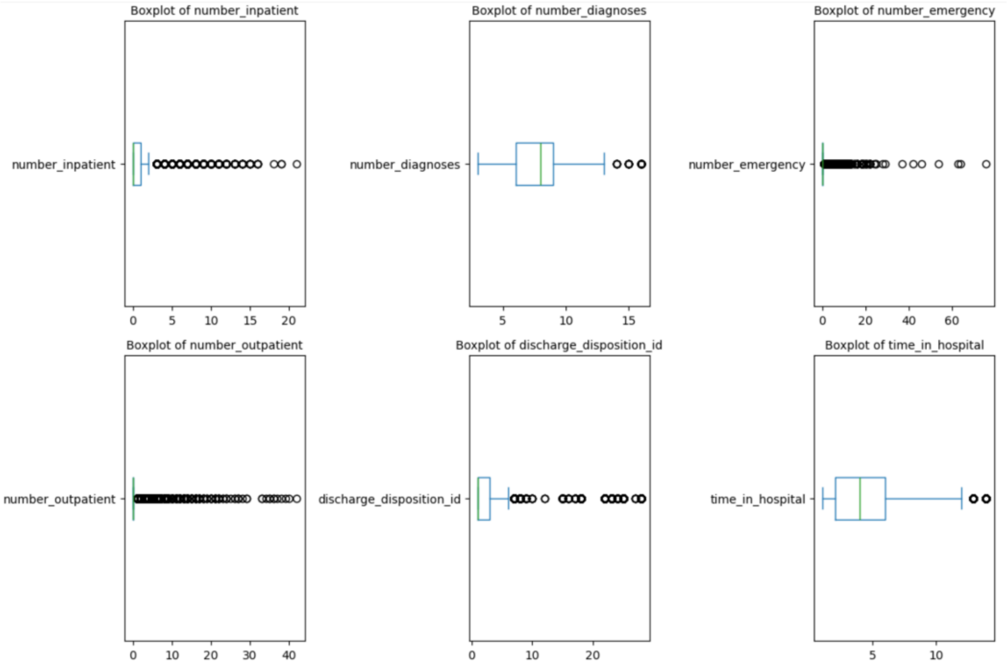


Figure 5 - Outliers visualization using Box plot

For outliers in the 'Time in Hospital' feature, we retained them in the dataset due to their high importance in predicting readmission risk.

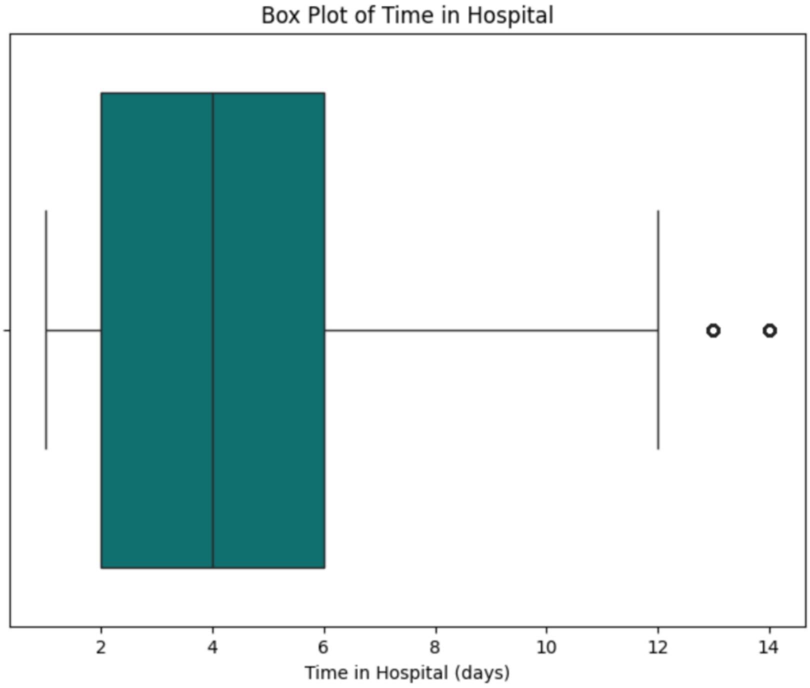


Figure 6 - Box plot for Time in Hospital feature

- **Feature Engineering**:
    - Created a new feature `age_group` to simplify the analysis of age ranges.
    - Feature selection was performed using SelectKBest (ANOVA F-test) and Random Forest importance metrics. Top predictors included `number_inpatient`, `number_diagnoses`, and `number_emergency`.

- **Feature Reduction**:
  - Dropped features with high cardinality and low predictive value (e.g., `payer_code`) after assessing their impact on the target variable.
- **Encoding**: Categorical variables (e.g., medications, diagnostic codes [4]) were encoded using techniques like OneHotEncoding using `pandas.get_dummies()`, enabling the models to interpret these variables correctly. Features with only two values were encoded using Binary encoding.
- **Balancing the Dataset**: To address class imbalance in the readmitted target, Synthetic Minority Over-sampling Technique (SMOTE) was applied to ensure the model is not biased toward the majority class [5].

- **Scaling**: Numerical features were standardized to ensure consistency, especially given varying scales across features.

3.3 Machine Learning Algorithms

- **Logistic Regression**: Employed as a baseline model for its simplicity and ease of interpretation.
- **Random Forest**: Chosen for its ability to capture non-linear interactions and robustness against overfitting.
- **Gradient Boosting (XGBoost)**: Utilized for its high accuracy and capability to optimize complex decision boundaries through iterative boosting.
- **Other models**: AdaBoost, Neural Networks, MLP Classifier, etc…
- **Packages Used**: scikit-learn [2], numpy, pandas, XGboost [3], etc… were utilized for model implementation and evaluation.

3.4 Model Improvement Strategies

- **Hyperparameter Tuning**: GridSearchCV was employed to optimize hyperparameters, including `max_depth` and `n_estimators` for the Random Forest and XGBoost models.
- **Class Imbalance Handling**: The Synthetic Minority Oversampling Technique (SMOTE) was used to create synthetic samples for the minority class, effectively balancing the training dataset.

3.5 Challenges and Solutions

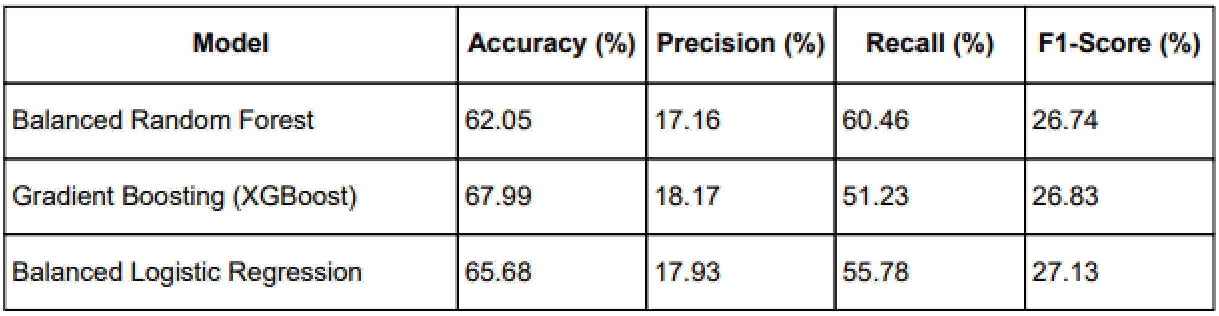
- **Class Imbalance**: The skewed distribution of the target variable (`readmitted`) posed a challenge. SMOTE was applied to generate synthetic samples, improving model performance.

4. Results and Evaluation

4.1 Quantitative Results

The evaluation of our models was based on multiple performance metrics, including accuracy, precision, recall, and F1-score. Here are the results along with an evaluation for each model:

Model Performance Table



Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Balanced Random Forest	62.05	17.16	60.46	26.74
Gradient Boosting (XGBoost)	67.99	18.17	51.23	26.83
Balanced Logistic Regression	65.68	17.93	55.78	27.13

Table 1 - Model performance parameters

Models Evaluation:

- **Balanced Random Forest**: The Balanced Random Forest model displayed moderate recall, indicating it correctly identified a relatively high proportion of true positive cases. However, the low precision suggests that many of these positive predictions were incorrect, leading to a high number of false positives. This makes it effective for cases where capturing all potential readmissions is critical, but with the trade-off of a higher false alarm rate.
- **Gradient Boosting (XGBoost)**: The Gradient Boosting model achieved the highest accuracy, and a balanced precision-recall performance compared to the other models. Its recall indicates a reasonable capability to detect true positive cases, and it strikes a better balance between precision and recall compared to the Balanced Random Forest. This makes it a strong choice for predictive performance while maintaining an acceptable level of false positives.

- **Balanced Logistic Regression**: Balanced Logistic Regression showed relatively strong performance, achieving a recall close to that of the Balanced Random Forest and a slightly higher precision. The balance between precision and recall was reflected in its F1-score, suggesting that this model is reasonably well-suited for identifying readmissions while managing both false positives and false negatives to some extent.

Model Ranking by Accuracy:
When considering accuracy alone, the ranking of additional models provided is:

1. Random Forest (88.32%)
2. Gradient Boosting (88.15%)
3. AdaBoost Model (86.34%)
4. MLP (85.54%)
5. Decision Tree (80.41%)
6. Neural Networks (80.08%)

4.2 Qualitative Results

- **Key Features**:
  - `number_inpatient`: Higher inpatient visits were a strong predictor of readmission.
  - `number_diagnoses`: A higher count of diagnoses correlated with increased readmission likelihood.

  - `number_emergency`: Frequent emergency visits were indicative of high-risk patients.

- **Visualizations**:
  - Confusion matrices illustrated true positive and negative rates, while ROC curves compared model performance visually [6].

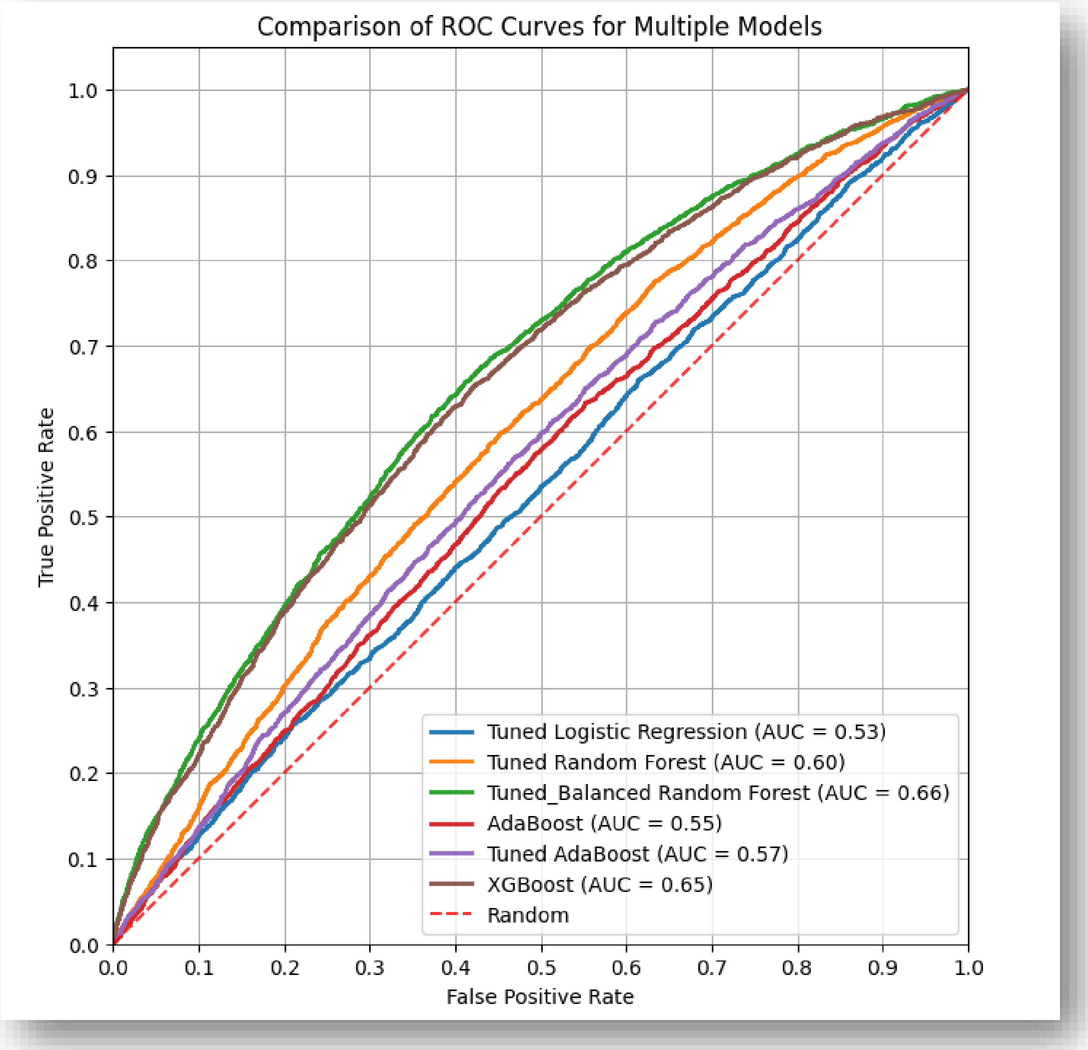


Figure 7 - Comparison of ROC curves for different models

5. Discussion

5.1 Challenges Encountered

- **Data Quality**: Managing missing values, outliers, and high-cardinality features required extensive preprocessing and feature engineering. A1C results and Max Glucose Serum were dropped due to high percentage of missing values (94% and 83% respectively) even though for diabetic patients, they are key factors for diagnosis.
- **Class Imbalance**: Handling the imbalance in the target variable was crucial, as naive models performed poorly without addressing this imbalance.

5.2 Lessons Learned

- **Importance of Preprocessing**: Effective handling of missing data, feature engineering, and selection greatly influences predictive model accuracy.
- **Iterative Model Building**: Performance improved with iterative feature refinement, model tuning, and handling of data imbalance.

5.3 Future Directions

- **Deep Learning**: Explore deep learning models (e.g., LSTM) for temporal data analysis, potentially capturing sequential patterns in patient visits.
- **Data Enrichment**: Integrating external data sources could provide richer patient profiles and improve model accuracy.

6. Documentation

6.1 Notebook Documentation

Each step of data preprocessing, feature engineering, and model building was documented in detail, with markdown explanations for reproducibility.

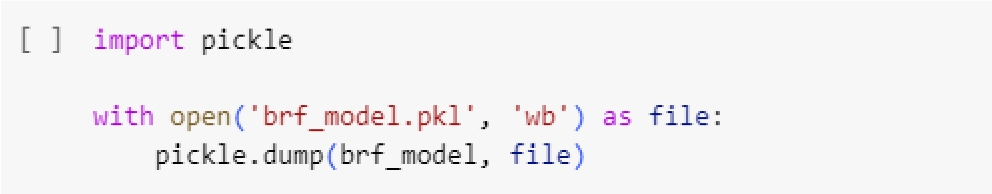
6.2 Project Components

The project comprises data preprocessing scripts, modeling scripts, and evaluation methods, all documented for seamless reproducibility and analysis.

6.3 Web Interface: To pickle and deploy multiple models using a web interface, we followed these steps:

Pickling Models:

We began by training three machine learning models: a Balanced Random Forest, Gradient Boosting (XGBoost), and a Balanced Logistic Regression model. Each model was saved as a `.pkl` file using Python's `pickle` library for efficient serialization. The code for saving models looked like this:



```
[ ]  import pickle

     with open('brf_model.pkl', 'wb') as file:
         pickle.dump(brf_model, file)
```

Figure 8 - The code we used for pickling models

This allowed us to store and retrieve trained models for later use without retraining them, significantly reducing the loading time during web integration.

Web Integration:

Our application was built using FastAPI for the backend. It provided endpoints for receiving CSV files, loading the requested model, and making predictions. Key elements of the integration included:

- **Model Selection**: Users could select from available models (`xgb_model.pkl`, `BRF_model.pkl`, `best_ada_model.pkl`) through an HTML dropdown menu.
- **Dynamic Loading**: The `load_model` function dynamically loaded the chosen model using `pickle` during API calls.
- **Prediction Endpoint**: The `/predict_csv` endpoint handled file uploads, validated data structure, and used the loaded model to generate predictions on provided input data.

Frontend Interaction:

The frontend, developed with basic HTML and JavaScript, allowed users to upload CSV files and choose a model for predictions. A JavaScript function asynchronously sent form data to the FastAPI server and displayed prediction results, enabling a smooth user experience.

By integrating this approach, we created a modular, efficient, and user-friendly system for predicting diabetic patient readmissions using pre-trained models.

7. Conclusion

The results of this project indicate that predictive models, particularly Gradient Boosting (XGBoost) and Balanced Random Forest, can provide meaningful insights into the likelihood of diabetic patient readmissions. By using a comprehensive preprocessing pipeline and robust model selection, our approach demonstrated the potential for improving healthcare outcomes through proactive resource allocation and patient management. Moving forward, refining these models with additional data and advanced techniques could further enhance their predictive power and applicability in real-world healthcare settings.

8. References

1. University of California Irvine. UCI Machine Learning Repository - Diabetes 130-US Hospitals dataset [online]. Available from: https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008 [Accessed 10 November 2024].

2. Scikit-learn. scikit-learn Documentation [online]. Available from: https://scikit-learn.org/stable[Accessed 10 November 2024].

3. XGBoost. XGBoost Documentation [online]. Available from: https://xgboost.readthedocs.io/ [Accessed 10 November 2024].

4. Centers for Disease Control and Prevention. International Classification of Diseases, Ninth Revision (ICD-9) [online]. Available from: https://archive.cdc.gov/www_cdc_gov/nchs/icd/icd9cm.htm [Accessed 10 November 2024].

5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research. 2002; 16: 321–357.

6. Fawcett, T. An introduction to ROC analysis. Pattern Recognition Letters. 2006; 27(8): 861-874.