**Final Report: Predicting Diabetes Readmission Using Machine Learning**

**Jessica McPhaul – QTW 7333 – Case Study 2 – February 3, 2025**

**1. Introduction**

Hospital readmission, especially among diabetic patients, presents a significant challenge to healthcare systems. This study aimed to develop and evaluate machine learning models to predict diabetes readmission risk, enabling proactive interventions and improved patient outcomes. The study involved data preprocessing, model training and validation, and a comparative analysis to select the most effective model.

**2. Data Preprocessing**

**2.1 Data Loading and Cleaning**

The dataset, sourced from the Health Facts database, contained over 100,000 patient records detailing demographic information, diagnoses, lab results, medications, and hospital visits. The following preprocessing steps were performed:

* Converted categorical variables to string types.
* Replaced missing values ('?') with NA and performed imputation.
* Transformed numerical columns into appropriate data types.
* Merged admission type mappings for interpretability.
* Removed irrelevant columns such as weight, encounter\_id, and payer\_code.
* Converted 'readmitted' into numerical categories (0: No readmission, 1: Readmitted after 30 days, 2: Readmitted within 30 days).

**2.2 Feature Engineering**

* **One-hot encoding:** Applied to categorical features like race, gender, and diabetesMed.
* **Categorical encoding:** Converted diagnosis codes (diag\_1, diag\_2, diag\_3) into numerical representations.
* **Binary encoding:** Medication usage columns were converted to binary values (0 for 'No' and 1 for other values).
* **Feature scaling:** Standardized numeric features using StandardScaler.

**2.3 Train-Test Split**

The dataset was split into training (80%) and testing (20%) sets while ensuring stratified sampling to maintain class distribution.

**3. Model Training and Evaluation**

**3.1 Logistic Regression (Baseline Model)**

* **Performance:**
  + **Test Accuracy:** 42% (Readmitted), 85.7% (Non-Readmitted)
  + **Macro-F1 Score:** 0.33
  + **Strengths:** Simple and interpretable.
  + **Weaknesses:** Poor recall for readmitted patients.

**3.2 Addressing Class Imbalance**

To improve model performance, the following techniques were employed:

* **Class weighting:** Adjusted weights to penalize underrepresented classes.
* **SMOTE Oversampling:** Generated synthetic samples for minority classes.
* **Solver adjustment:** Switched to 'saga' solver for better handling of imbalanced data.

**3.3 Random Forest (Ensemble Model)**

* **Final Model Parameters:**
  + **85 Trees, log2 features, Max Depth = 7**
  + **Test Accuracy:** 94%
  + **Strengths:** Handles class imbalance, high accuracy.
  + **Weaknesses:** Slightly lower recall on readmissions.

**3.4 Stacking Classifier**

* **Best Model for Readmission Recall**
* **Performance:**
  + **Macro-F1 Score:** 0.49
  + **Weighted-F1 Score:** 0.91
  + **Strengths:** Improved recall compared to Random Forest.
  + **Weaknesses:** Slightly lower overall accuracy than Random Forest.

**4. Model Comparison & Insights**

| **Model** | **Accuracy** | **Macro-F1 Score** | **Weighted-F1 Score** |
| --- | --- | --- | --- |
| Logistic Regression | 0.92 | 0.33 | 0.50 |
| Random Forest | 0.94 | 0.33 | 0.50 |
| Stacking Classifier | 0.91 | 0.49 | 0.91 |

* **Random Forest performed best in overall accuracy.**
* **Stacking Classifier achieved the highest recall for readmitted patients.**
* **SMOTE improved recall but slightly reduced precision.**

**5. Visualizations**

A graph with different colored bars

Description automatically generated

This chart compares the weighted F1-scores of different models. The Stacking Classifier, Random Forest, and Logistic Regression models all achieved high scores near 1.0. The Custom Ensemble Model has a noticeably lower score, around 0.7. This suggests that, for this dataset and task, the Stacking Classifier, Random Forest, and Logistic Regression models performed very well in terms of weighted F1-score, considerably outperforming the Custom Ensemble Model**.**

A graph with blue bars

Description automatically generated

Macro-F1-Score Distribution: This histogram shows the distribution of Macro-F1 scores, likely across different runs or folds of a specific model. The concentration around 0.2, 0.33, and 0.48 suggests some variability in performance, possibly due to randomness in the training process or data splits.

A graph of a number of people

Description automatically generated

Readmissions vs. Age: This chart illustrates the relationship between patient age and readmission rates. Readmissions peak in the 70-80 age group, declining slightly in the oldest group (90+). This suggests that age is a significant factor in readmission risk.

A graph with numbers and lines

Description automatically generated

From this: Stacking Classifier is the best-performing model overall, while Random Forest is a reasonable alternative if accuracy is the primary concern. The Custom Ensemble Model should likely be discarded. The choice between Stacking Classifier and Random Forest depends on the specific priorities of the application: if correctly identifying the minority class (readmissions) is crucial, the Stacking Classifier is strongly preferred despite slightly lower accuracy.

A group of colorful bars

Description automatically generated with medium confidence

These bar charts compare four different machine learning models — Custom Ensemble Model, Logistic Regression, Random Forest, and Stacking Classifier — across four performance metrics: Macro-F1 Score, Micro-F1 Score, Weighted-F1 Score, and Accuracy.

Macro-F1 Score: Measures the average F1-score of each class, giving equal weight to all classes. The Stacking Classifier performs best (around 0.5), significantly outperforming the others, particularly the Custom Ensemble Model (around 0.2). Logistic Regression and Random Forest have similar, lower scores (around 0.33).

Micro-F1 Score: Calculates the F1-score globally by considering the total true positives, false negatives, and false positives. The Stacking Classifier again leads with a score close to 0.9. The other three models fall considerably behind, with Logistic Regression achieving about 0.6 and Random Forest and the Custom Ensemble Model scoring even lower.

Weighted-F1 Score: Similar to Macro-F1 but weights each class's contribution to the average based on its prevalence in the dataset. The Stacking Classifier demonstrates clear superiority here, achieving nearly 1.0, indicating excellent performance across all classes considering the class imbalance. Logistic Regression and Random Forest again perform similarly (around 0.5), significantly trailing the Stacking Classifier but exceeding the Custom Ensemble Model (around 0.7).

Accuracy: Measures the overall correctness of the model's predictions. Random Forest and the Stacking Classifier perform similarly well, both close to 0.95, followed by Logistic Regression (about 0.92). The Custom Ensemble Model trails considerably behind, at around 0.6.

The Stacking Classifier consistently outperforms the other models across all four metrics, demonstrating its effectiveness, particularly in handling class imbalance and achieving high recall and precision across different classes.

Random Forest shows good overall accuracy but falls behind the Stacking Classifier on Macro-F1, Micro-F1, and Weighted F1-score.

Logistic Regression shows reasonably good accuracy and Micro-F1 but performs poorly on Macro-F1 score.

The Custom Ensemble Model performs poorly across all metrics, suggesting it's not well-suited for this problem.

A graph of different types of data

Description automatically generated

Performance of Methods for Choosing `max\_features`: This plot demonstrates the impact of the `max\_features` hyperparameter in the Random Forest model. Using the square root (`sqrt`) or logarithm (`log2`) of the number of features consistently outperforms using all features (`None`). This highlights the importance of feature selection in preventing overfitting and improving generalization.

A graph with blue and orange bars

Description automatically generated

Readmissions vs. Number of Visits: This chart shows a strong negative correlation between the number of hospital visits and readmission rates. Patients with fewer visits are much more likely to be readmitted. This suggests that initial visits might not provide sufficient care or follow-up to prevent readmissions for a significant portion of patients.

**6. Recommendations**

**Random Forest for general accuracy** **Stacking Classifier for improving recall** **SMOTE for balancing dataset** **Further Improvements:**

* **Feature Engineering:** Use SHAP or permutation importance for better feature selection.
* **Alternative Models:** Test Gradient Boosting, XGBoost, or deep learning models.
* **Explainability:** Utilize SHAP values for clinical interpretation.

**7. Future Work**

* **Refining Feature Selection:** Using domain expertise and advanced selection techniques.
* **Deploying the Model:** Integrating into a hospital setting for real-time predictions.
* **Enhancing Interpretability:** Applying model explainability techniques to gain clinical insights.

**8. Conclusion**

This study successfully developed predictive models for diabetes readmission, highlighting the value of machine learning in addressing this healthcare challenge. While Random Forest offered excellent overall accuracy, the Stacking Classifier provided superior recall for readmissions, emphasizing the importance of choosing a model aligned with specific clinical objectives. Future work will focus on refining model performance, improving interpretability, and ultimately deploying these tools to enhance patient care and reduce readmission rates.

**9. References/Citations:**

**General Diabetes Readmission & Healthcare Costs:**

1. Jiang, H. J., Russo, J. E., & Barrett, M. L. (2014). *Risk factors for 30-day hospital readmission among patients with diabetes*. Diabetes Care, 37(6), 1609–1615.
2. Donahue, R. P., Coffman, C. J., Turchin, M. C., & Herman, W. H. (2005). *Hospital readmissions for patients with diabetes*. Diabetes Care, 28(12), 2971–2976.

**Data Preprocessing and Handling Imbalance:**

1. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: synthetic minority over-sampling technique*. Journal of artificial intelligence research, 16, 321-357.
2. He, H., Garcia, E. A., & Li, S. (2009). *Learning from imbalanced data*. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

**Predictive Modeling and Machine Learning:**

1. Daghistani, T. H., Ibrahim, S., & Aljarah, I. (2021). *A comparative study of machine learning algorithms for predicting diabetes type II*. ICT Express, 7(4), 500-504. [For: <https://www.sciencedirect.com/science/article/pii/S2666990021000318>]
2. Kavakiotis, I., Tsave, O., Salifis, N., Maglaveras, N., & Chouvarda, I. (2021). *Machine learning and data mining methods in diabetes research*. Computational and structural biotechnology journal, 19, 1049-1066. [For: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8306487/>]
3. Hasan, M. M., Alam, M. M., Das, D., Hossain, E., Hasan, M. R., & Lee, J. J. (2021). *Predicting type 2 diabetes using logistic regression and machine learning approaches*. [For: <https://www.researchgate.net/publication/353150592_Predicting_Type_2_Diabetes_Using_Logistic_Regression_and_Machine_Learning_Approaches>]

[Link to notebook](https://github.com/texaschikkita/7333-CaseStudy-2/blob/main/McPhaul_J_CaseStudy2_FINAL_diabetes.ipynb)