Hospital Readmission Analysis: Baseline vs. Regularised Logistics Regression

Gayatri Jadhav
MSc. in Data Analytics
National College of Ireland
23407476

***Abstract*— Hospital readmissions among diabetic patients significant challenges to healthcare systems by suggesting operating costs and suggesting potential gaps in treatment management. The present paper proposes a predictive model to detect at-risk diabetic patients using machine learning algorithms. A real-world patient dataset was used that included patient demographics, hospital contact details, diagnosis codes, and treatment history. Extensive data preparation, feature engineering, and multicollinearity reduction through statistical methods including Chi-squared testing and Variance Inflation Factor (VIF) analysis form the framework of the model. Starting with Logistic Regression as a baseline model with its L2-Regularization and Elastic Net versions, to increase generalization and avoid overfitting. The model performance was evaluated using F1-score, recall, precision, and accuracy to balance the moderate class skewness of the dataset. The results shows that L2-Regularized Logistic Regression performed with a competitive trade-off between recall and F1-score and can be a good tool for detecting high-risk diabetic patients at early stages and facilitating targeted interventions at the right time.**

Keywords—Logistic Regression, Diabetes Patients, Imbalanced Classification, Feature Engineering

I. Introduction

Hospital readmissions of people receiving treatment for diabetes are a major problem in today's healthcare systems because they result in higher costs, more stress on resources, and worse outcomes for patients. One of the most common chronic diseases in the world, diabetes often leads to complications that require frequent hospital visits, if not well controlled. Accurate prediction of which patients are most likely to be readmitted can help to greatly enhance patient care, best treatment approaches, and lower healthcare costs. Traditional clinical decision-making tools can depend on small number of criteria, thereby ignoring complicated trends in patient data.

Machine learning models have recently become known for their capacity to examine massive medical records and gain useful insights. Among these models, logistic regression remains widely applied since it can be interpreted and robustly fits structured medical data. Logistic regression is a statistical method used for binary classification, which means it predicts the probability of one of two possible outcomes. Unlike linear regression, which is used for continuous outcomes, logistic regression is used when the dependent variable is categorical. Although logistic regression and linear regression are related, logistic regression is specifically designed for binary classification problems, whereas linear regression predicts continuous outcomes. If you try to solve a logistic regression problem using linear regression, the model may give predictions outside the range [0, 1], making it unsuitable for classification tasks. In contrast, logistic regression, uses a logistic function to ensure outputs have probabilities between 0 and 1, which can then be used for classification.

This work presents a predictive modelling pipeline to classify patients depending on their chance of hospital readmission using logistic regression and its regularised variants (L2-Regularization and Elastic Nett). By incorporating feature engineering techniques, statistical feature selection, and class imbalance handling, the objective is to develop a model that can reliably assist healthcare professionals in identifying patients at elevated risk of readmission following diabetes-related treatment.

II. Dataset Despriction

The research uses hospitaldata.csv dataset with 101,763 patient encounters across multiple hospital settings. The 47 columns of the dataset record a mix of patient-level demographics, clinical contact details, diagnoses history, treatment history, and patient outcomes. Specifically, it includes variables such as age, gender, race, admission source, discharge disposition, length of stay, and medical specialty, as well as key laboratory and diagnostic codes (e.g., ICD-9 coded diagnoses, A1C results, glucose serum levels). Several columns also record medication use (e.g., insulin and oral medications) and the number of hospital visits across different types of encounter (inpatient, outpatient, emergency).

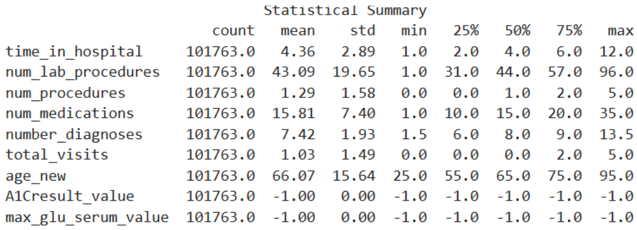
The target variable originally had three classes: 'No', 'Within30Days', and 'After30Days' to indicate if a patient was readmitted within 30 days or after30 days, or not readmitted at all. There was a relatively moderate level of class inequality in the original dataset (approximately 54% No Readmitted and 46% Readmitted), and this was taken into accounts during training and testing.

III. Methodology

Methodology involves data pre-processing, feature engineering, Exploratory Data Analysis (pre and post cleaning), statistical modelling. By resolving missing values, class imbalance, and multicollinearity problems, the approach emphasises preparing real-world healthcare data to be used with machine learning

A. Summary of Variables

A statistical summary was performed to explore the nature of the dataset and to identify potential patterns in hospital readmission. Descriptive statistical measures such as central tendency (mean, median, mode) were calculated. The statistical summary provided important insights into distribution of the dataset. The average stay in the hospital was approximately 4 days, with majority of the patients being in the hospital between 2 and 6 days based on the interquartile range. The variable num_lab_procedures showed that patients were given an average of 43 tests and some were given up to 96 tests, indicating a high diagnostic procedure intensity since the median was also high with 44 tests



```
                              Statistical Summary
                     count   mean    std   min   25%   50%   75%   max
time_in_hospital   101763.0   4.36   2.89   1.0   2.0   4.0   6.0  12.0
num_lab_procedures 101763.0  43.09  19.65   1.0  31.0  44.0  57.0  96.0
num_procedures     101763.0   1.29   1.58   0.0   0.0   1.0   2.0   5.0
num_medications    101763.0  15.81   7.40   1.0  10.0  15.0  20.0  35.0
number_diagnoses   101763.0   7.42   1.93   1.5   6.0   8.0   9.0  13.5
total_visits       101763.0   1.03   1.49   0.0   0.0   0.0   2.0   5.0
age_new            101763.0  66.07  15.64  25.0  55.0  65.0  75.0  95.0
A1Cresult_value    101763.0  -1.00   0.00  -1.0  -1.0  -1.0  -1.0  -1.0
max_glu_serum_value 101763.0 -1.00   0.00  -1.0  -1.0  -1.0  -1.0  -1.0
```

Table 1: Statistical Summary

The variable num_medications showed that patients were prescribed an average of 15.8 medications, with some receiving up to 35 medications, indicating cases of polypharmacy. Additionally, the total_visits variable that includes outpatient, emergency, and inpatient visits was found to have a mean of 1.03 and median of 0, which means that a large number of patients were found to have only one recorded visit to the hospital. Lastly, age_new variable showed that on average patient age is around 66 years with interquartile range of 55 to 75 years and this indicates that the dataset is mainly made up of older adults as is typical for diabetic patients

B. EDA

The exploratory data analysis was done first to gain insight into the distribution and structure of the raw data before any transformation. The target variable “readmitted” originally had three categories: showing a moderate bias toward "No", which represented the majority of cases. To simplify the classification task, this variable was later converted into a binary format (Readmitted vs. Not Readmitted), to help train the logistics regression model, the original distribution helped in identifying the imbalance problem in the raw data.

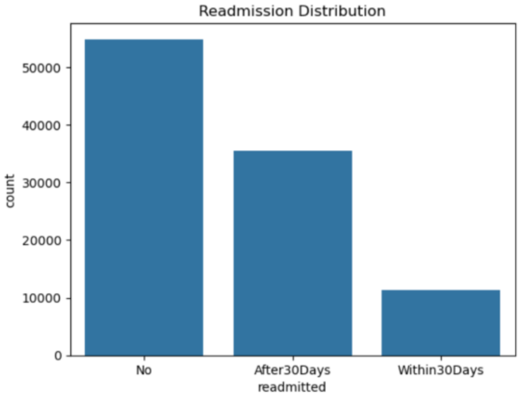


Figure 1: Distribution of Target Variable

The study used count plots for a Univariate analysis, revealing high "Unknown" and "Other" categories in various variables, causing initial concerns about data quality and missingness. The dataset exhibits missing or placeholder values across several key columns. Specifically, "Unknown" values are present in race (2.23%) and significantly in weight (96.86%), indicating that nearly all records lack valid weight data. The "None" placeholder is prevalent in clinical test results, with max_glu_serum missing in 94.75% and A1Cresult missing in 83.28% of cases, suggesting that these tests were not performed for the majority of patients. Additionally, the "Other" category dominates categorical fields like medical_specialty (65.6%) and appears in admission_source_id and race, which may indicate a need for further consolidation or binning to reduce noise. This highlights the importance of careful imputation and feature engineering during preprocessing.

The study conducted a bivariate analysis to examine the relationship between independent variables and the target variable, identifying distinct patterns in race against readmission rates.

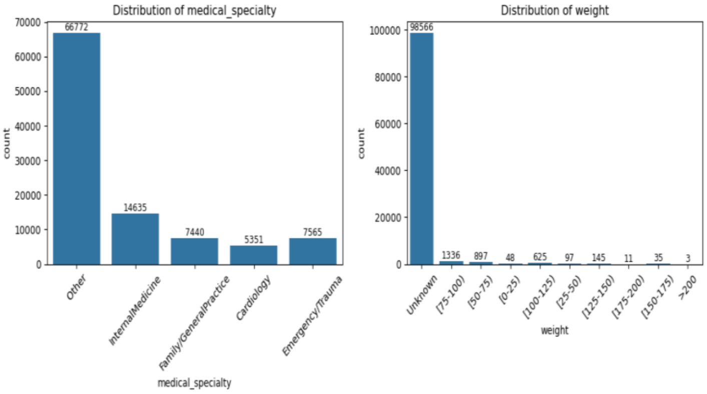


Figure 2: Individual Feature Analysis

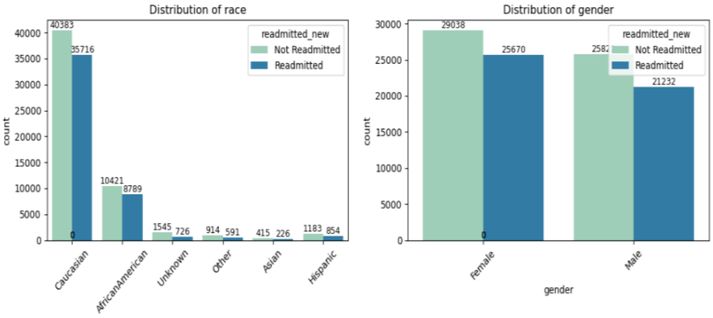


Figure 3: Bivariate Analysis

Finally, a correlation heatmap of numerical variables was produced to assess relationships between continuous features such as total_visits, time_in_hospital, and num_procedures. Several moderate correlations were found, justifying the need for further multicollinearity checks and variable selection steps in the preprocessing phase.

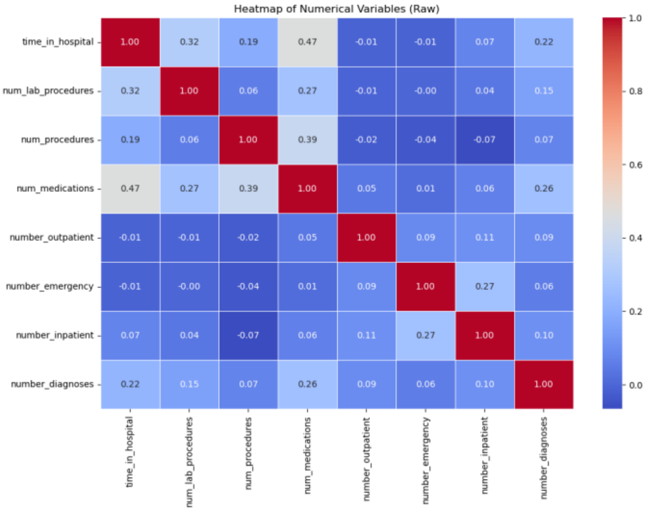


Figure 4: Heatmap of numeric variables (Raw)

C. *Data Transformation and Post EDA*

The data were carefully pre-processed to ensure consistency and improve feature quality. Missing values and non-standard placeholders such as 'Unknown', 'None', and 'Other' were calculated and handled feature-by-feature basis. Categorical features such as ICD-9 diagnosis codes were grouped into clinically meaningful classes. The age dimension, which was initially in associated with numerical hierarchies, was binned into ordinal numerical classes. In addition, laboratory test values such as A1Cresult and max_glu_serum were converted to ordinal scales to reflect their severity in a clinical setting. The high-missing weight feature was dropped before model training because it had 96% of Unknown count. These conversions made all features machine learning algorithm-friendly without losing their interpretability in a clinical setting

Feature	Original	Transformed
diag_1, diag_2, diag_3	ICD-9 codes	Grouped into 20 disease categories
age	Binned ranges (0,10), (10,20)	Mapped to ordinal integers (5, 15, 25, etc.)
A1Cresult	Categories: None, Norm, >7, >8	Mapped to ordinal values (-1, 0, 1, 2)
max_glu_serum	Categories: None, Norm, >200, >300	Mapped to ordinal values (-1, 0, 1, 2)

Table 1: Data Transformation

ICD-9 groupings rendered diagnostic columns (diag_1, diag_2, diag_3) clinically meaningful after data processing. Diag_1_new, for instance, was derived from initial diag_1 column and grouped into twenty different disease categories, including groupings like Circulatory System Disorders, Respiratory Diseases, Diabetes, Mental Disorders, and Injury and Poisoning, among many others. In an effort to promote model generalizability and decrease data sparsity, this grouping consolidated more than 100 ICD-9 codes into clinically more applicable groupings. diag_2 and diag_3 also underwent a similar change giving rise to diag_2_new and diag_3_new rows.

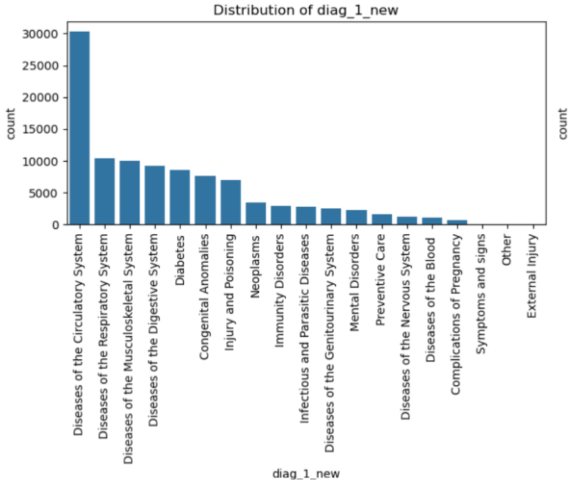


Figure 5: Transformation of Diagnosis Columns

A correlation heatmap after numerical variable cleaning was generated, which captured the improved structure of data. Most prominently, engineered feature total_visits and polypharmacy joined as a part of the new heatmap, providing a further detailed feature space. The heatmap also indicated moderate correlations among encounter features total_visits, time_in_hospital, and num_medications, which made additional multicollinearity checks at subsequent modeling stages appropriate.

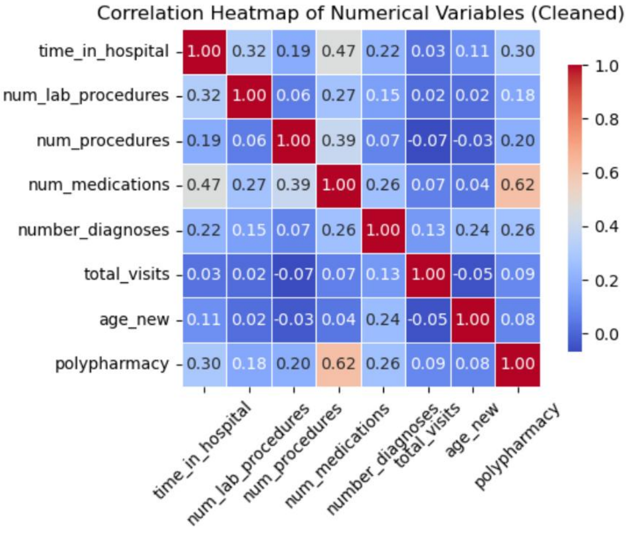


Figure 6: Heatmap of numeric variables (Cleaned)

D. *Feature Selection*

Feature selection was performed to improve model performance, enhance interpretability, and reduce redundancy in the dataset. Two statistical approaches were performed: the chi-square test for categorical variables and the variance inflation factor (VIF) for numerical variables. A chi-square test was performed to assess statistical independence between each categorical feature and the target variable readmitted_new.

- **Null Hypothesis (H_0):** The categorical feature is independent of the target variable (no association).
- **Alternative Hypothesis (H_1):** The categorical feature is dependent on the target variable (there is an association).

By comparing p-values at a significance level of **0.05**, variables with strong associations ($p < 0.05$) were retained and highlighted weaker features ($p > 0.05$) so that they could be removed. Features such as diag_1_new, diag_2_new, diag_3_new, and admission_source_id were found to be statistically significant and were retained. In contrast, several features related to medication showed weak associations and were dropped.

The Variance Inflation Factor (VIF) quantifies the degree of multicollinearity among numerical features by measuring how much the variance of a regression coefficient is inflated due to correlations with other predictors. VIF was applied exclusively to numerical features, as multicollinearity is concern primarily in continuous variables with linear relationships, whereas categorical variables are typically handled by encoding and assessed using statistical association tests such as the Chi-squared test

VIF values above 10 points are typically indicative of strong multicollinearity. Variables related to visits such as number_inpatient, number_outpatient, number_emergency, and the constructed feature total_visits showed significant multicollinearity. Since total_visits is a derived aggregate of other three, it was retained and the others were excluded. Combining the two tests ensured that the dataset was reduced to a set of statistically significant independent features, making it ready for model training.

E. *Handling Outliers*

Features classified as either continuous or ordinal numerical variables where extreme values potentially skew model training and statistical interpretation were selectively treated using the interquartile range (IQR) technique. Features include time_in_hospital, num_lab_procedures, num_medications, and total_visits reflect count-based or scaled data—that is, data that is naturally sensitive to anomalies. Extreme values were limited depending on their respective IQR thresholds to help to offset this.

Categorical or binary variables, such as gender, race, and engineered flags like polypharmacy, were excluded from outlier treatment, as these features are non-continuous and their values do not follow a distribution where capping would be meaningful. Using IQR on such variables could produce skewed or erroneous feature representations.

Although age_new is an ordinal variable derived from binned age intervals, it was included in the IQR capping to ensure that exceptionally high age bins (e.g., patients over 85 years) do not disproportionately influence the model. This guarantees consistent processing of numerical traits by the model while preserving the ordinal links among age groups.

Key variables such as num_medications were capped between -5 and 35, and total_visits was capped between -3 and 5, where the lower bounds (e.g., -5, -3) were theoretical outlier cutoffs but had no practical effect as the data contained no negative values. The time_in_hospital variable was capped at 12 days, controlling for unusually long stays that could bias model coefficients.

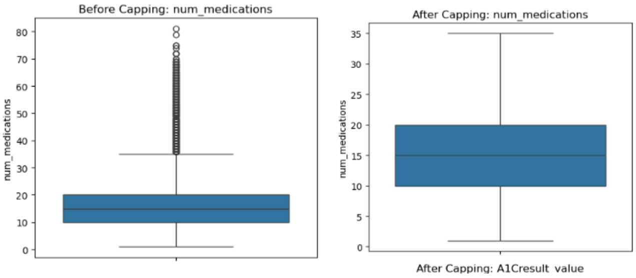


Figure 7: Outlier handling (num_medication)

F. *Preprocessing Pipeline*

A data transformation pipeline was constructed for effective data transformation prior to training models. The pipeline comprised three distinct phases based on variable type. All numerical columns were standardized with StandardScaler, which scales features with a mean value of zero and unit standard deviation, avoiding variables with different scales from dominating a model with a disproportionate influence.

Second, aside from the initial category for avoiding the dummy variable trap, OneHotEncoder encoded categorical columns into binary representation vectors. It ensured categorical variables were encoded correctly for use within logistic regression models.

In the end, as binary columns such as custom flags (i.e., polypharmacy) and additional yes/no features were already in a form conducive to machine learning, they were passed through unchanged. The pipeline for this preprocessing was created with a column transformer for automating and consistently applying the transformations on both training and test data

IV. Model Training

A. Baseline Logistic Regression

The initial model utilized was the Basic Logistic Regression without any regularization. class_weight = ‘balanced' is used for the class imbalance between the "Readmitted" and "Not Readmitted" classes. class_weight = 'balanced' adjusts the class weights in inverse proportion to class frequencies so that the minority class "Readmitted" contributes the same amount towards the loss function when training the model. This baseline was the point of comparison for the performance-based and bias-handling model.

B. Regularized Logistic Regression Models

Model 2 utilized the L2-Regularized Logistic Regression with the saga solver. L2 (Ridge) regularization was chosen because it would prevent overfitting by penalizing large coefficients without being computationally intensive on large datasets. The saga solver was chosen because it accommodates Elastic Net and L1/L2 penalties

Model 3 employed ElasticNet-regularized Logistic Regression that combines both L1 (Lasso) and L2 (Ridge) penalties for sparsity and shrinkage of the coefficients. ElasticNet finds the balance between automatic feature selection (by L1) and regularization for preventing overfitting (by L2). In this model, the parameter l1_ratio=0.5 has been selected, i.e., the model applies half L1 and half L2 penalties (50% L1 and 50% L2). Having this balanced ratio means the model has both sparsity (setting certain coefficients towards zero) and stability (shrinking non-zero coefficients) and generalizes better while maintaining important predictors

C. Final Model

Along with further maximizing the ElasticNet-regularized Logistic Regression performance, Model 4 also consisted of a GridSearchCV pipeline whereby the most significant hyperparameters would be optimized in a systematic fashion. Specifically, both the regularization strength parameter C and the l1_ratio were optimized using a 5-fold cross-validation grid search such that the model would have the ideal balance between feature selection and regularization. The optimized final model used an l1_ratio = 0.5 and used an even balance between L1 (Lasso) and L2 (Ridge) penalties so that the model would both encourage sparsity by forcing certain coefficients towards zero and stabilize others by performing shrinkage

In line with the earlier models, class_weight='balanced' was used to address class imbalance, and the saga solver was chosen for compatibility with ElasticNet regularization and large dataset performance. Before model training, the entire preprocessing pipeline was executed in which the numeric features were standardized using StandardScaler and the categorical features were one-hot encoded using the ColumnTransformer. All of this helped the ElasticNet model in generalizing well on new instances as well as decreasing multicollinearity and class bias towards the most frequent class.

V. RESULTS & DISCUSSION

The performance of all four logistic regression models was evaluated using several classification metrics, including Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

The baseline model, Model 1 - Basic Logistic Regression, achieved an accuracy of 62.7%, an F1-score of 0.584, and a ROC-AUC of 0.670, indicating reasonable predictive capability given the moderate imbalance in the dataset. The addition of L2 regularization in Model 2 (saga solver) marginally improved recall (0.565) compared to the baseline (0.564) while maintaining similar precision and F1-score values

Model 3, employing ElasticNet regularization, produced nearly identical results to the L2 model, confirming that feature sparsity introduced by the L1 component did not significantly affect performance in this context. Finally, Model 4, which utilized ElasticNet with GridSearchCV, slightly underperformed relative to the other models, achieving an F1-score of 0.583 and a ROC-AUC of 0.669. This outcome suggests that hyperparameter tuning did not yield significant gains, likely due to the modest feature space and the stability provided by standard L2 regularization.

Overall, all models performed similarly, with Model 2 (L2 - saga) marginally outperforming others on recall and F1-score. The results indicate that logistic regression, when combined with appropriate preprocessing and regularization, provides a stable and interpretable framework for predicting hospital readmissions

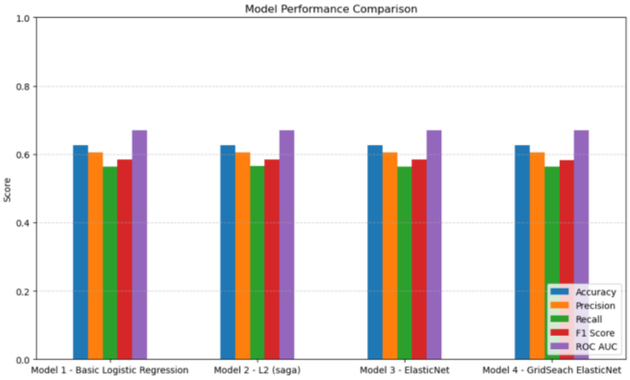


Figure 8: Model Performance Comparison

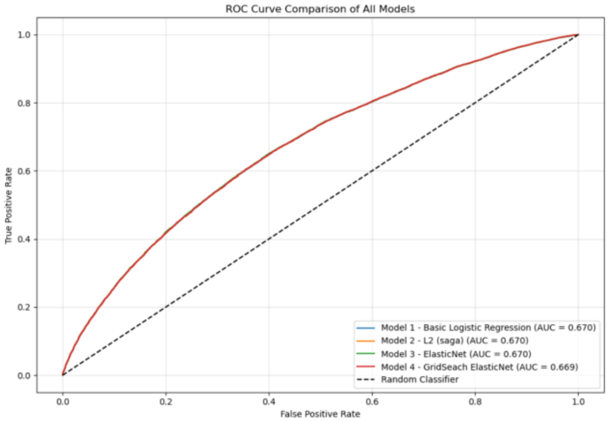


Figure 9: ROC Curve of all models

VI. References

[1] Wikipedia contributors, “List of ICD-9 codes,” *Wikipedia*, May 23, 2023. https://en.wikipedia.org/wiki/List_of_ICD-9_codes

[2] GeeksforGeeks, “Logistic regression in machine learning,” *GeeksforGeeks*, Feb. 03, 2025. https://www.geeksforgeeks.org/understanding-logistic-regression/

[3] “A hybrid prediction model for type 2 diabetes using K-means and decision tree,” *IEEE Conference Publication | IEEE Xplore*, Nov. 01, 2017. https://ieeexplore.ieee.org/document/8342938

[4] P. Schober and T. R. Vetter, “Logistic regression in medical research,” *Anesthesia & Analgesia*, vol. 132, no. 2, pp. 365–366, Jan. 2021, doi: 10.1213/ane.0000000000005247.

[5] Malamahadevan, “Step-by-Step Exploratory Data Analysis (EDA) using Python,” *Analytics Vidhya*, Jan. 07, 2025. https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python