**Mahamadou Mahi Diallo**

**Predictive Modelling for ICU Admission**

**QuickStart**

**August 8, 2024**

**Research Report: Predictive Modelling for ICU Admission**

**Introduction**

The COVID-19 pandemic has had a profound impact on healthcare systems worldwide, exposing vulnerabilities and unpreparedness for managing such a large-scale health crisis. Brazil, which reported its first COVID-19 case on February 26, 2020, and reached community transmission by March 20, 2020, faced significant challenges in meeting the increased demand for ICU (Intensive Care Unit) beds, healthcare professionals, personal protective equipment (PPE), and other critical resources.

The urgency to predict ICU admissions accurately became paramount to ensure healthcare systems could prepare adequately and avoid collapse due to overwhelming patient loads. Traditional approaches relying on epidemiological and population data proved insufficient in addressing the real-time needs of healthcare facilities. Thus, there emerged a critical need to utilize individual clinical data to develop predictive models that could better forecast ICU admissions.

This study aims to leverage detailed patient visit information and various medical metrics to build robust predictive models for ICU admissions. By analysing individual clinical data, we seek to identify key factors influencing ICU needs and develop models that provide actionable insights for healthcare providers. These models can assist in proactive resource allocation, timely interventions, and ultimately, in saving lives during ongoing and future health crises.

**Data Description**

The dataset used in this study was sourced from Kaggle and contains detailed information on patient visits to the hospital. It includes demographic details, disease groupings, and a variety of medical metrics. The dataset consists of 1,925 rows and 231 columns. Key attributes of the dataset include:

**Patient Demographics:**

**PATIENT\_VISIT\_IDENTIFIER:** Unique identifier for each patient visit.

**AGE\_ABOVE65:** Indicator of whether the patient is above 65 years old.

**AGE\_PERCENTIL:** Percentile of the patient's age.

**GENDER:** Gender of the patient.

**Disease Groupings:**

**DISEASE\_GROUPING\_1 to DISEASE\_GROUPING\_6:** Indicators of the presence of specific diseases.

**HTN:** Hypertension status.

**IMMUNOCOMPROMISED:** Immunocompromised status.

**OTHER:** Indicator of other unspecified conditions.

**Medical Metrics:**

Various measures of blood components, including albumin, arterial and venous blood gases.

Variables such as ALBUMIN\_MEDIAN, ALBUMIN\_MEAN, ALBUMIN\_MIN, ALBUMIN\_MAX, and ALBUMIN\_DIFF.

Metrics related to blood pressure, respiratory rate, and lactate levels.

ICU: The target variable indicating whether the patient was admitted to the ICU (1) or not (0).

**Summary Statistics**

The dataset's complexity is highlighted by the vast number of variables, many of which are highly specific medical metrics. The majority of patients are under the age of 65, with a roughly equal distribution across different disease groupings. The summary statistics suggest that many medical measurements have missing or uniform values, indicating the need for careful data preprocessing to handle these issues.

**Methods**

**Data Exploration and Preprocessing**

**Missing Values**

Missing values are common in medical datasets due to various reasons such as measurement errors, incomplete records, or data entry omissions. Identifying and handling these missing values is crucial to ensure the integrity and accuracy of the predictive models.

1. **Identification:**

* We identified the missing values in each column using the sapply() function.

1. **Imputation:**

* For numeric variables, we imputed missing values using the median. The median is robust to outliers and provides a central tendency measure that is less affected by extreme values compared to the mean.
* For categorical variables, we imputed missing values using the mode (the most frequent value).

1. **Visualization:**

* We visualized the distribution of missing values using the plot\_missing() function from the DataExplorer package.

**Categorical Variables:**

Categorical variables need to be encoded as factors to be utilized in machine learning models. This encoding allows the models to interpret these variables correctly.

1. **Encoding:**

* We converted categorical variables into factors using the mutate\_if() function from the dplyr package.

**Feature Selection**

Given the high dimensionality of the dataset, feature selection is essential to identify the most relevant features for predicting ICU admissions. We used Recursive Feature Elimination (RFE) to systematically remove less important features and select the most significant ones

1. **RFE with Random Forest:**

* We employed RFE with a Random Forest model, which iteratively builds models and ranks features based on their importance.
* This process helped us narrow down the features to those most predictive of ICU admissions.

**Data Visualization**

Visualizing the data helps in understanding the distribution and relationships between variables, which can provide insights for model building.

1. **Histogram of Numeric Variables:**

* We plotted histograms for numeric variables to visualize their distributions and identify any skewness or outliers.

1. **Bar Charts for Categorical Variables:**

* We created bar charts to visualize the distribution of categorical variables, providing insights into the frequency of different categories.

1. **Correlation Analysis:**

* We performed correlation analysis and visualized the correlation matrix to identify relationships between numeric variables. This helped in understanding multicollinearity and potential feature interactions.

**Machine Learning Model Development**

**Model Selection and Hyperparameter Tuning**

The choice of machine learning models and their hyperparameter tuning is crucial for achieving optimal performance

**Random Forest**

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification. It is robust to overfitting and can handle high-dimensional data effectively.

1. **Hyperparameter Tuning:**

* We conducted hyperparameter tuning for the mtry parameter (number of variables randomly sampled as candidates at each split) using grid search and cross-validation.

**Support Vector Machine (SVM)**

SVM is a powerful classification algorithm that finds the hyperplane that best separates the classes in the feature space. We used the radial basis function (RBF) kernel for its effectiveness in handling non-linear relationships.

1. **Hyperparameter Tuning:**

* We performed hyperparameter tuning for the sigma (kernel width) and C (regularization parameter) using grid search and cross-validation.

**Training and Testing**

1. **Data Splitting:**

* We split the dataset into training (80%) and testing (20%) sets to evaluate model performance. The training set is used to train the models, and the test set is used to assess their generalization ability.

1. **Ensuring Consistent Factor Levels:**

* We ensured that the factor levels for the target variable (ICU) were consistent across both training and testing sets to avoid issues during model evaluation.

**Model Evaluation**

**Random Forest**

1. **Performance on Training Data:**

* The Random Forest model achieved a high accuracy of 97.14% and a Kappa of 0.9245 on the training data, indicating excellent agreement.

1. **Performance on Test Data:**

* The model's accuracy dropped to 85.97% with a Kappa of 0.603 on the test data, suggesting moderate agreement and potential overfitting.

**Support Vector Machine (SVM)**

1. **Performance on Training Data:**

* The SVM model showed balanced performance with an accuracy of 86.23% and a Kappa of 0.622 on the test data, indicating better generalization compared to the Random Forest model.

**Confusion Matrices**

1. **Random Forest:**

* **Test Data:**
  + Accuracy: 85.97%
  + Sensitivity: 96.10%
  + Specificity: 58.25%
  + Kappa: 0.603

1. **SVM:**

* **Test Data:**
  + Accuracy: 86.23%
  + Sensitivity: 94.68%
  + Specificity: 63.11%
  + Kappa: 0.622

**Overfitting Analysis**

Overfitting occurs when a model performs exceptionally well on training data but poorly on test data. This discrepancy indicates that the model has learned the noise in the training data rather than the underlying patterns.

1. **Random Forest:**

* The significant difference in accuracy between the training (97.14%) and test (85.97%) sets indicates overfitting.
* High sensitivity (1.000) on the training set compared to lower sensitivity (0.961) and specificity (0.5825) on the test set further highlights this issue.

1. **SVM:**

* The SVM model showed more balanced performance, suggesting better generalization and less overfitting compared to the Random Forest model.

**Model Comparison**

1. **Resampling Summary:**

* We summarized the resampling results to compare the mean accuracy and Kappa values for both models.
* The Random Forest and SVM models exhibited comparable performance, with the SVM slightly outperforming the Random Forest in terms of balanced accuracy.

1. **Visualization:**

* We used boxplots to visualize the performance metrics for both models across different resampling iterations, highlighting the consistency and variability in their performance.

**Conclusion**

In this study, we explored and pre-processed a comprehensive dataset to predict ICU admissions using machine learning models. We developed and tuned two candidate models: Random Forest and Support Vector Machine (SVM). The Random Forest model demonstrated high accuracy on training data but faced overfitting issues, resulting in moderate performance on test data. The SVM model showed more balanced performance and slightly outperformed the Random Forest model in terms of generalization.

The significant performance drop of the Random Forest model from training to test data highlights the importance of addressing overfitting in predictive modelling. Techniques such as cross-validation, regularization, and ensemble methods can be employed to mitigate overfitting and enhance model robustness.

Future work could involve exploring additional preprocessing techniques, alternative feature selection methods, and other machine learning models to further improve prediction accuracy and robustness. Incorporating domain knowledge from medical experts can also enhance feature engineering and model interpretability, leading to more actionable insights.

Additionally, evaluating the models in a real-world clinical setting and integrating them into healthcare systems can provide practical insights and further validate their effectiveness. Continuous monitoring and updating of the models with new data will ensure their relevance and accuracy over time.

In conclusion, predictive modelling for ICU admissions has the potential to significantly improve patient care and resource allocation in hospitals. By leveraging advanced machine learning techniques and addressing challenges such as overfitting, we can develop robust models that assist healthcare professionals in making timely and informed decisions, ultimately enhancing patient outcomes.