Report: MIMIC II Surgical Intensive Care Unit Dataset

Final Project

Thâmiris Barcarolo Rodrigues Guze

Higher Diploma in Science in Data Analytics

Statistics for Data Analytics

Alexander Mutiso Mutua

City Colleges

❖ Introduction

In this project, Python programming language was employed to analyze the MIMIC II Surgical Intensive Care Unit (SICU) Dataset. The dataset is tabular, featuring diverse columns encompassing medical and demographic information about distinct individuals. Each row corresponds to a specific patient, offering a comprehensive representation of their health-related attributes.

1. The libraries employed in this project include:

In the course of this project, several libraries were instrumental in facilitating data analysis, model training, and performance evaluation. Each library played a specific role, contributing to the overall success of the project. Here is an overview of the key libraries employed:

- pandas: Pandas stands out as a versatile library designed for effective data manipulation and analysis. It introduces valuable data structures such as DataFrames, which prove powerful in handling and analyzing structured data.

- sklearn.model_selection.train_test_split: Within the scikit-learn toolkit, the `train_test_split` function plays a crucial role in partitioning datasets into training and test sets. This step is vital for evaluating the performance of machine learning models.

- sklearn.preprocessing.LabelEncoder: The LabelEncoder is an essential tool for converting categorical variables into a numeric format. This conversion is a prerequisite for numerous machine learning algorithms.

- numpy: NumPy serves as a foundational library for performing numerical operations and array manipulations in Python. It offers support for large, multi-dimensional arrays and matrices, coupled with mathematical functions designed to efficiently operate on these elements.

- sklearn.preprocessing.StandardScaler: Included in scikit-learn, the StandardScaler is employed for standardizing features. This involves removing the mean and scaling to unit variance, a critical preprocessing step for machine learning algorithms sensitive to feature scales.

- sklearn.decomposition.PCA: PCA, or Principal Component Analysis, is a valuable technique for dimensionality reduction in machine learning. The scikit-learn implementation, PCA, facilitates transforming data into a lower-

dimensional space while preserving most of its original information. This proves useful for simplifying models and enhancing computational efficiency.

- matplotlib.pyplot: Matplotlib stands out as a comprehensive library for generating static, interactive, and animated visualizations in Python. The `pyplot` module offers a MATLAB-like interface, providing functions for creating a diverse range of plots and charts.

- seaborn: Built on top of Matplotlib, Seaborn provides a high-level interface for crafting informative and visually appealing statistical graphics. It streamlines the process of generating complex visualizations and introduces aesthetic enhancements.

- sklearn.linear_model.LogisticRegression: Logistic Regression, a popular algorithm for binary classification, is readily available in scikit-learn as `LogisticRegression`. It proves valuable for training models to predict whether an instance belongs to a particular class.

- sklearn.metrics: This module within scikit-learn supplies various metrics essential for evaluating machine learning models. In your code, specific metrics like `accuracy_score`, `classification_report`, and `confusion_matrix` are imported, aiding in the assessment of model performance and behavior.

❖ Create pairwise scatterplots of variables of interest. Describe your discoveries and the relationships, if any.

For the six variables of interest a pairwise scatterplot and heatmap were generated. The analysis of these visualizations yielded the following insights:

1. **Age and SAPS (Simplified Acute Physiology Score) at ICU Admission (sapsi_first):**
    - There is a moderate positive correlation (0.36) between age and SAPS at ICU admission. This suggests that older patients tend to have higher SAPS scores, indicating potentially more severe physiological derangements.

2. **Age and Heart Rate at ICU Admission (hr_1st):**
    - There is a moderate negative correlation (-0.31) between age and heart rate at ICU admission. This implies that older patients tend to have a lower heart rate at admission.

3. **SAPS at ICU Admission and SOFA (Sequential Organ Failure Assessment) Score at ICU Admission (sofa_first):**
    - There is a moderate positive correlation (0.43) between SAPS at ICU admission and the SOFA score at ICU admission. This suggests that patients with higher SAPS scores are likely to have higher SOFA scores, indicating a greater degree of organ dysfunction.

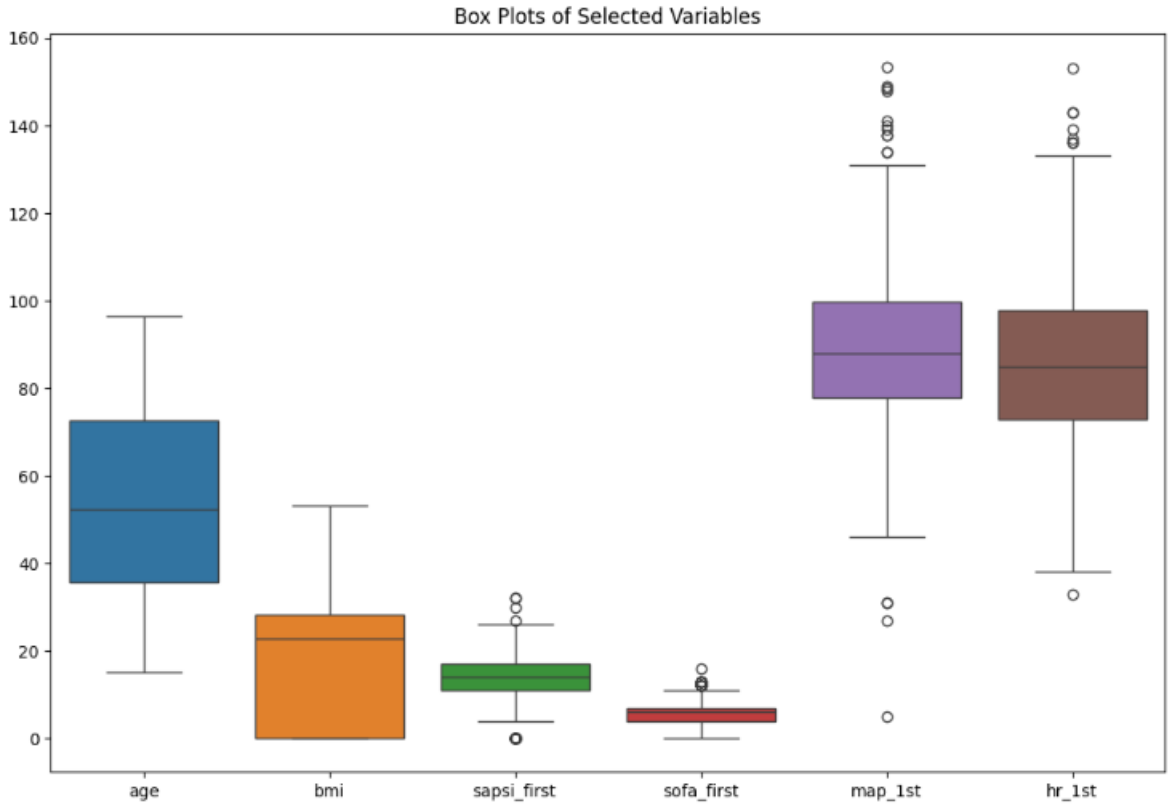
4. **Age and BMI (Body Mass Index):**
    - There is a very weak negative correlation (-0.01) between age and BMI. This correlation suggests a minimal association between age and BMI in the selected dataset.

5. **No Strong Correlations between Other Variable Pairs:**
    - The correlations between other variable pairs (BMI and SAPS, BMI and SOFA, BMI and MAP, SAPS and MAP, SAPS and HR, SOFA and MAP, SOFA and HR, MAP and HR) are relatively weak, indicating limited linear relationships between these variables.

This analysis provides valuable insights into the relationships among key variables, shedding light on potential patterns and associations within the dataset.

❖ Create box plots to identify outliers.

During this exercise, a thorough examination of the dataset revealed the presence of outliers in specific columns. Notably, the variables sapsi_first, sofa_first, map_1st, and hr_1st exhibit apparent outliers, indicating potential anomalies or extreme values in these datasets.



- Perform data cleaning and preprocessing, his might include data amputation of the extreme variables.

Upon conducting a thorough examination of the dataset, it was evident that no missing values were present. However, the exploration of box plots uncovered outliers in key columns. To address this issue, the chosen strategy involves the application of Winsorizing, a data transformation technique.

The purpose of the code is to effectively handle outliers in the selected numeric columns using Winsorizing. This technique, which involves capping extreme values, has been selected for its ability to preserve the overall distribution of the data while mitigating the impact of outliers. The resulting DataFrame, now named project_clean_selected, reflects the Winsorized values for the identified numeric columns, with extreme values adjusted based on specified percentiles.

The incorporation of Winsorizing aims to fortify the dataset for subsequent statistical analyses and machine learning model training by reducing the influence of outliers.

Following the outlier management process, the next step involves transforming categorical data into a format compatible with machine learning algorithms. Post-execution of this code, the categorical columns in the project_clean_selected DataFrame are replaced with numerical labels. This

transformation streamlines the integration of categorical information into machine learning workflows.

The dual application of Winsorizing and categorical data transformation contributes to a more refined dataset.

❖ Conduct data analysis: calculate the median, mean, and standard deviation of variables of interest. Discuss the implications of the results.

- **Age:** The distribution of ages is not perfectly symmetric, with a right-skewed tendency. The standard deviation indicates a notable spread in age values.

- **BMI (Body Mass Index):** There seems to be a significant spread in BMI values. The difference between the median and mean BMI could indicate potential skewness or outliers in the distribution.

- **SAPSI (Simplified Acute Physiology Score) at First Measurement:** The distribution of SAPSI scores is relatively centered around the median, but there is some variability in severity scores.

- **SOFA (Sequential Organ Failure Assessment) at First Measurement:** The distribution of SOFA scores is centered around the median, indicating a certain level of organ dysfunction variability.

- **MAP (Mean Arterial Pressure) at First Measurement:** The distribution of MAP values has variability, and the difference between the median and mean could suggest potential skewness.

- **HR (Heart Rate) at First Measurement:** The distribution of heart rates has notable variability, and the difference between the median and mean may indicate potential skewness or outliers.

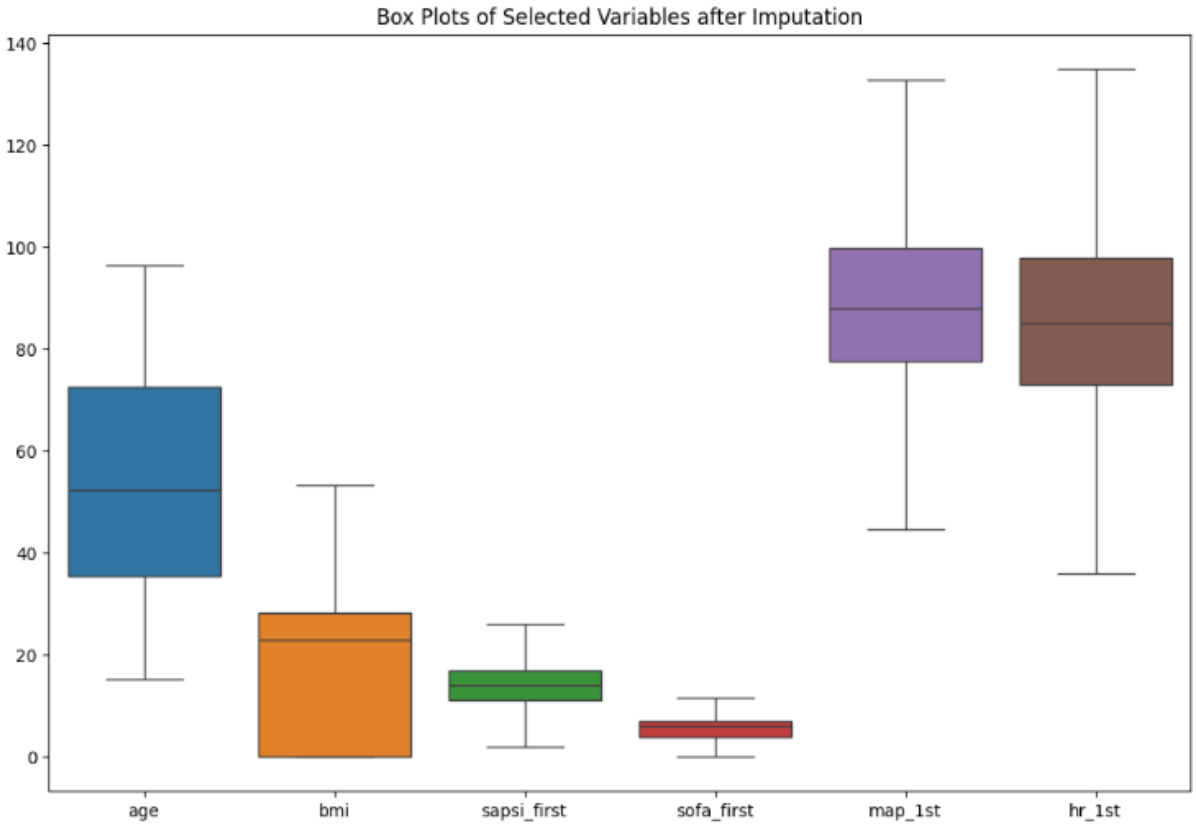
Overall Implications:

- Variability in BMI, SAPSI, SOFA, MAP, and HR suggests that patients in the dataset exhibit diverse physiological characteristics.
- The difference between median and mean values in some variables suggests potential skewness or outliers in the distributions.

	Variable	Median	Mean	Standard Deviation
age	age	52.277470	53.144155	21.407871
bmi	bmi	22.898715	18.620344	13.557910
sapsi_first	sapsi_first	14.000000	13.700611	4.667975
sofa_first	sofa_first	6.000000	5.806517	2.116567
map_1st	map_1st	88.000000	88.774611	16.580339
hr_1st	hr_1st	85.000000	85.718941	18.116857

- Create new box plot after data imputation and analyse the results.

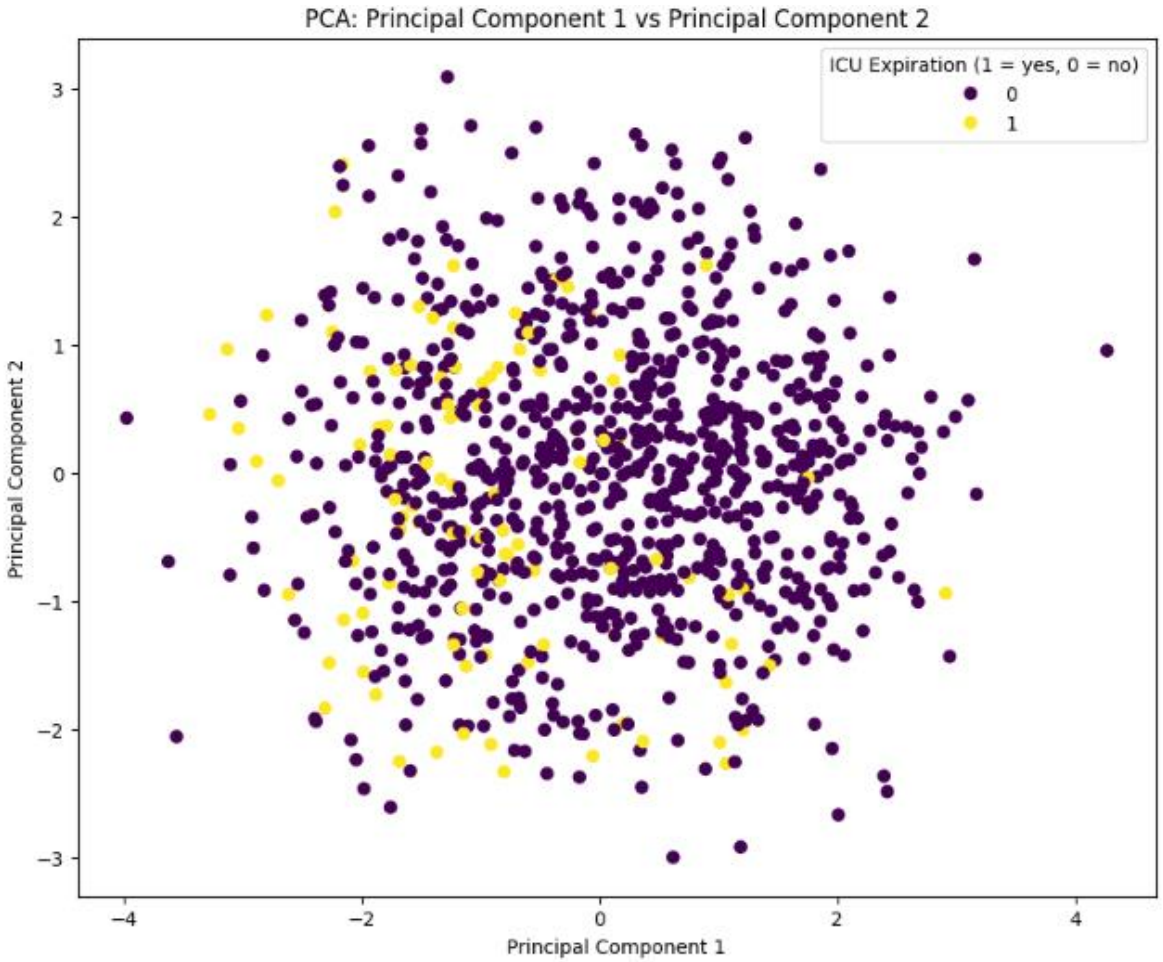
The updated box plot illustrate the distribution of the variables after Winsorizing, allowing for an immediate comparison with the previous box plot. This visual assessment aims to determine whether the extreme values have been appropriately capped, signifying effective handling of outliers without significantly altering the overall characteristics of the dataset.



❖ Explain the concept of multicollinearity in the context of the dataset. Appling PCA and visualize Principal Component 1 and Principal Component

Multicollinearity is a phenomenon inherent in regression analysis, where two or more predictor variables within a model exhibit high correlation. This correlation between predictors can introduce challenges in interpreting the individual effects of each variable and may lead to unstable estimates of regression coefficients.

In the specific context of our dataset, multicollinearity becomes a concern if there are substantial correlations between certain variables. The presence of strong correlations can complicate the identification of each variable's unique contribution to the model, as the effects of correlated predictors become entangled.



❖ Using Logistic Regression

Logistic Regression is indeed a statistical model commonly used for binary classification tasks. It predicts the probability of an instance belonging to a particular class by transforming the output into a range between 0 and 1 using the logistic function. The model then classifies instances based on a chosen threshold. Logistic Regression is favored for its simplicity, interpretability, and efficiency in predicting outcomes with two possible categories.

- **Normalize the data:**

  Normalization is a preprocessing step that brings all features to a similar scale, facilitating the training and performance of machine learning models. It is particularly beneficial for algorithms that rely on numerical stability and feature magnitudes.

- **Split your dataset into train and test data:**

  Dividing the dataset into training and test sets is indeed essential in machine learning. This process is crucial for assessing a model's generalization capability, detecting overfitting, and ensuring that the model's performance metrics provide reliable indicators of real-world performance. The training set is used to train the model, while the test set serves as an independent dataset to evaluate the model's performance on unseen data. This practice helps ensure the model's effectiveness and generalizability beyond the data it was trained on.

- **My results:**

  Training Set:

  - **Training Accuracy:** 97.07%
    - The logistic regression model achieved a high accuracy of 97.07% on the training set. This indicates that the model performed well in predicting outcomes on the data it was trained on.

  Test Set:

  - **Test Accuracy:** 79.19%
    - The accuracy on the test set is 79.19%. This metric represents the percentage of correctly predicted outcomes in the test set.

  - **Precision and Recall:**

- **Precision (Positive Predictive Value):** 32%
    - Precision is the proportion of true positives among all predicted positives. In this case, it means that when the model predicts a positive outcome (ICU Expired), it is correct 32% of the time.
- **Recall (Sensitivity or True Positive Rate):** 100%
    - Recall is the proportion of true positives among all actual positives. A recall of 100% indicates that the model captures all instances of actual positive outcomes.

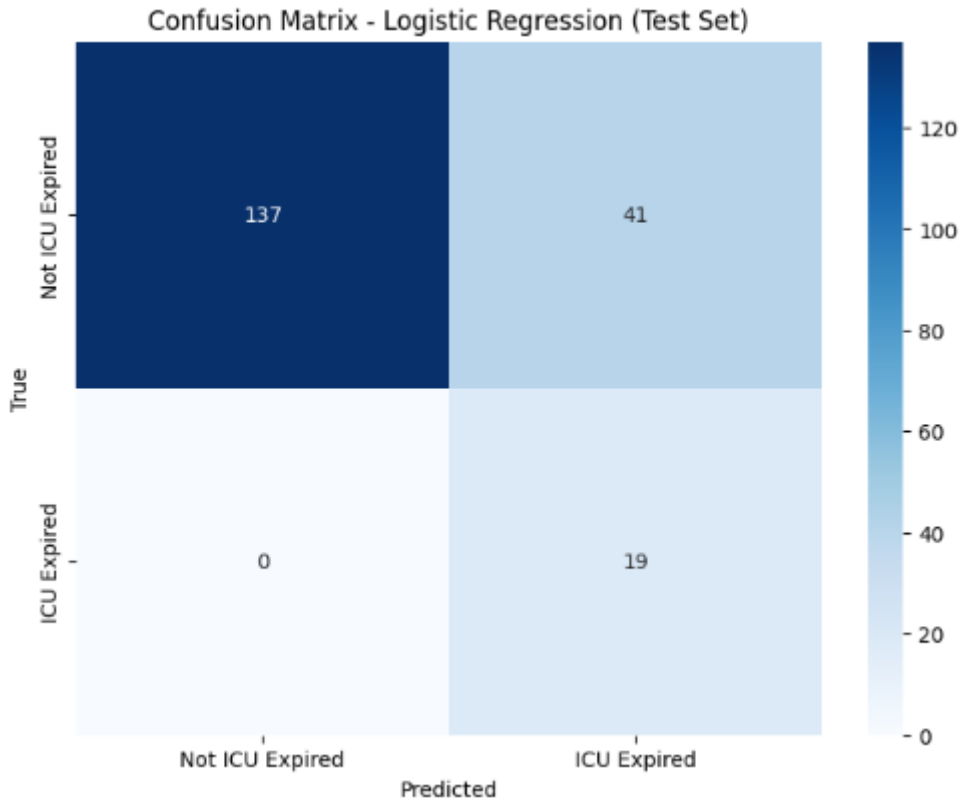
- **F1-Score:**
    - The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. The weighted average F1-score is 0.83, which is relatively good.

- **Support:**
    - Support represents the number of actual occurrences of each class in the specified dataset. For class 0 (Not ICU Expired), the support is 178, and for class 1 (ICU Expired), the support is 19.

❖ Explain the results in your confusion matrix.

The confusion matrix is a fundamental tool in evaluating the performance of a classification model. It provides a tabular representation of the model's predictions against the actual outcomes.

It shows the model's performance in terms of true positives, true negatives, false positives, and false negatives.



Four components of the confusion matrix:

1. **True Positive (TP):** 19
    - These are cases where the model correctly predicted positive outcomes (ICU Expired).
2. **True Negative (TN):** 137
    - These are cases where the model correctly predicted negative outcomes (Not ICU Expired).
3. **False Positive (FP):** 41
    - These are cases where the model incorrectly predicted positive outcomes (ICU Expired) when the actual outcome was negative (Not ICU Expired).
4. **False Negative (FN):** 0
    - There are no cases where the model incorrectly predicted negative outcomes (Not ICU Expired) when the actual outcome was positive (ICU Expired).

Interpretation:

- The model appears to perform well in identifying cases where patients did not expire in the ICU (Not ICU Expired), as evidenced by a high number of true negatives (137).
- The model correctly identified 19 cases where patients did expire in the ICU (ICU Expired).
- The number of false positives (41) indicates cases where the model incorrectly predicted ICU expiration when the patients did not expire.
- The absence of false negatives (0) indicates that the model did not miss any cases of actual ICU expiration.