BIS568: Homework 2

Rong Sun

2025-02-21

Contents

1 Introduction **2**

2 Exploratory Data Analysis (EDA) **2**

3 Visualizing Descriptive Statistics **7**

3.1 Age, Gender and Clarity Visulization 7

3.2 Bacteria and WBC Visulization 8

3.3 Race and Ethnicity Visulization 8

3.4 Summary 10

4 Short Answer Questions **10**

4.1 Discuss which features you selected to include in your ML models and explain why 10

4.2 Discuss the significance of separating data into training, validation, and test sets 10

4.3 Discuss the concept of data leakage 11

5 Appendix **12**

5.1 Exploratory Data Analysis (Cont'd) 12

5.2 Model Performance 14

1 Introduction

This report details the development of machine learning models to predict urinary tract infections (UTIs) in the emergency department using a de-identified dataset from the Yale system. The primary objectives are to conduct exploratory data analysis (EDA), preprocess the dataset, implement model training pipelines, and evaluate predictive performance using AUC, PR-AUC, and calibration curves.

To ensure robust model development, the dataset is first explored to identify patterns in continuous and categorical features. Summary statistics and visualizations, including histograms and bar plots, provide insights into key variables such as patient demographics and urinalysis measures. Based on EDA findings, feature selection and preprocessing steps, such as handling missing data and encoding categorical variables, are applied to improve model performance.

The dataset is then split into training, validation, and testing sets to evaluate model generalizability and prevent overfitting. Logistic regression and gradient-boosted models (e.g., XGBoost) are implemented to predict UTI diagnoses, with performance assessed through multiple metrics. Additionally, calibration curves are generated to evaluate model reliability in estimating probabilities.

This report presents the step-by-step methodology, including EDA outputs, selected features, preprocessing steps, model evaluation results, and interpretations. The final section addresses key considerations such as feature selection rationale, the importance of data partitioning, and the risk of data leakage in predictive modeling.

2 Exploratory Data Analysis (EDA)

The exploratory data analysis (EDA) provides an overview of the dataset by summarizing key statistics and visualizing relationships between variables.

Summary Statistics and Data Structure:

- The summary statistics (`summary(df)`) provide insights into the distribution of continuous and categorical variables, identifying potential missing values and extreme values.
- The structure of the dataset (`str(df)`) helps to understand variable types (numeric vs. categorical) and detect any inconsistencies in data types.

Continuous Variables vs. UTI Diagnosis:

- Urine Specific Gravity (`ua_spec_grav`): There is little spread in the data, but some extreme outliers are present in the non-UTI group. The presence of a long whisker extending upwards suggests that some values deviate significantly from the central distribution.
- Urine pH (`ua_ph`): Created a bar plot to examine the distribution of urine pH values across UTI diagnosis groups.

- Age (`age`): Used a violin plot with a boxplot overlay to compare the distribution of age between diagnosed and non-diagnosed groups.

Categorical Variables vs. UTI Diagnosis:

- Gender (`gender`): Generated a bar chart to compare the gender distribution between UTI diagnosis groups.
- Race (`race`): Created a bar chart to examine the race distribution across UTI diagnosis groups.
- Ethnicity (`ethnicity`): Produced a bar chart to explore the ethnicity distribution between UTI-diagnosed and non-diagnosed groups.
- Urine Clarity (`ua_clarity`): Analyzed the distribution of urine clarity levels across UTI diagnosis groups.
- Urine Color (`ua_color`): Investigated the distribution of urine color across UTI diagnosis groups.
- Urine Bacteria (`ua_bacteria`): Examined the presence of bacteria in urine and its association with UTI diagnosis through a bar chart.

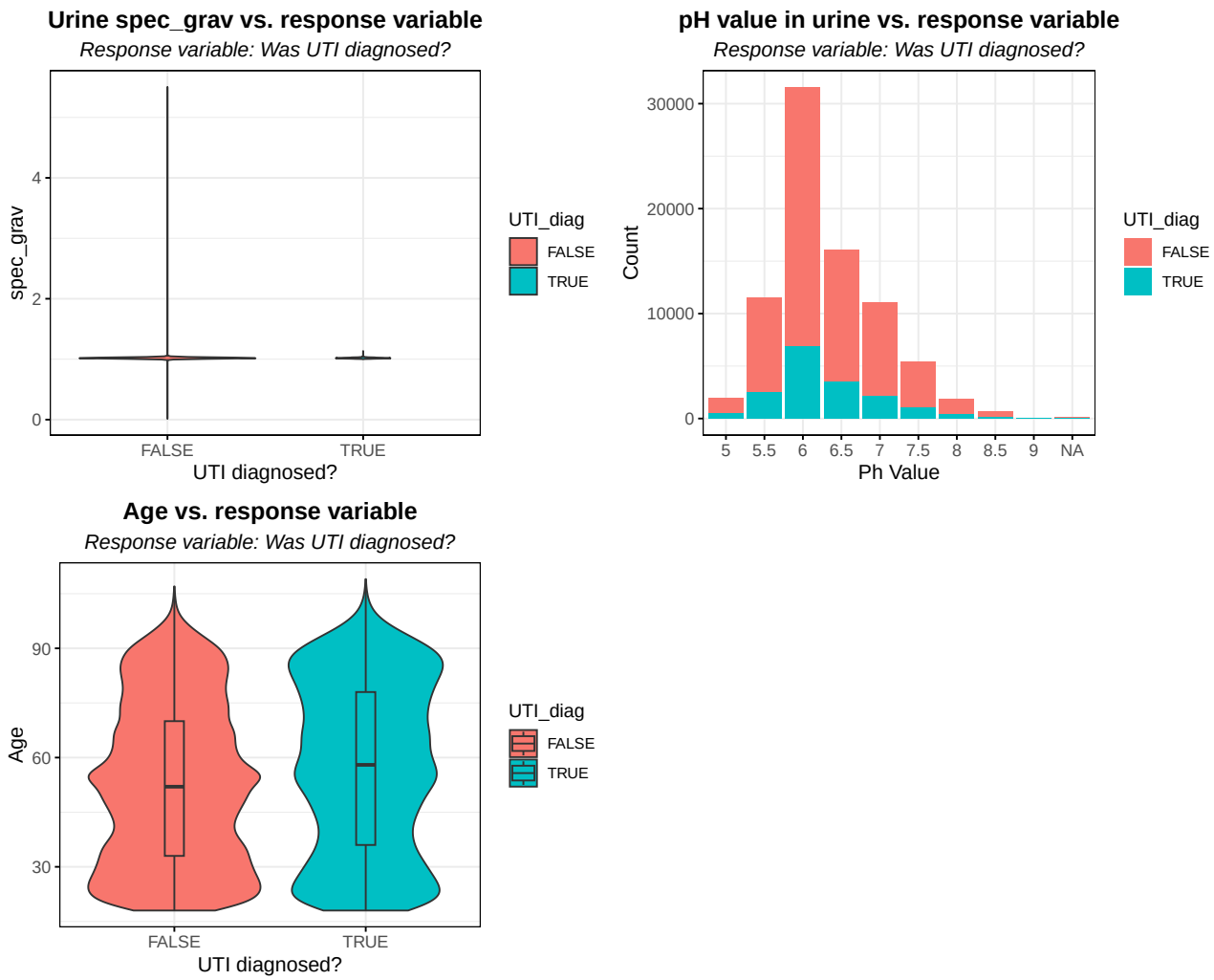

Figure 1: Exploratory Data Analysis of UTI Diagnosis Variables (Part I)

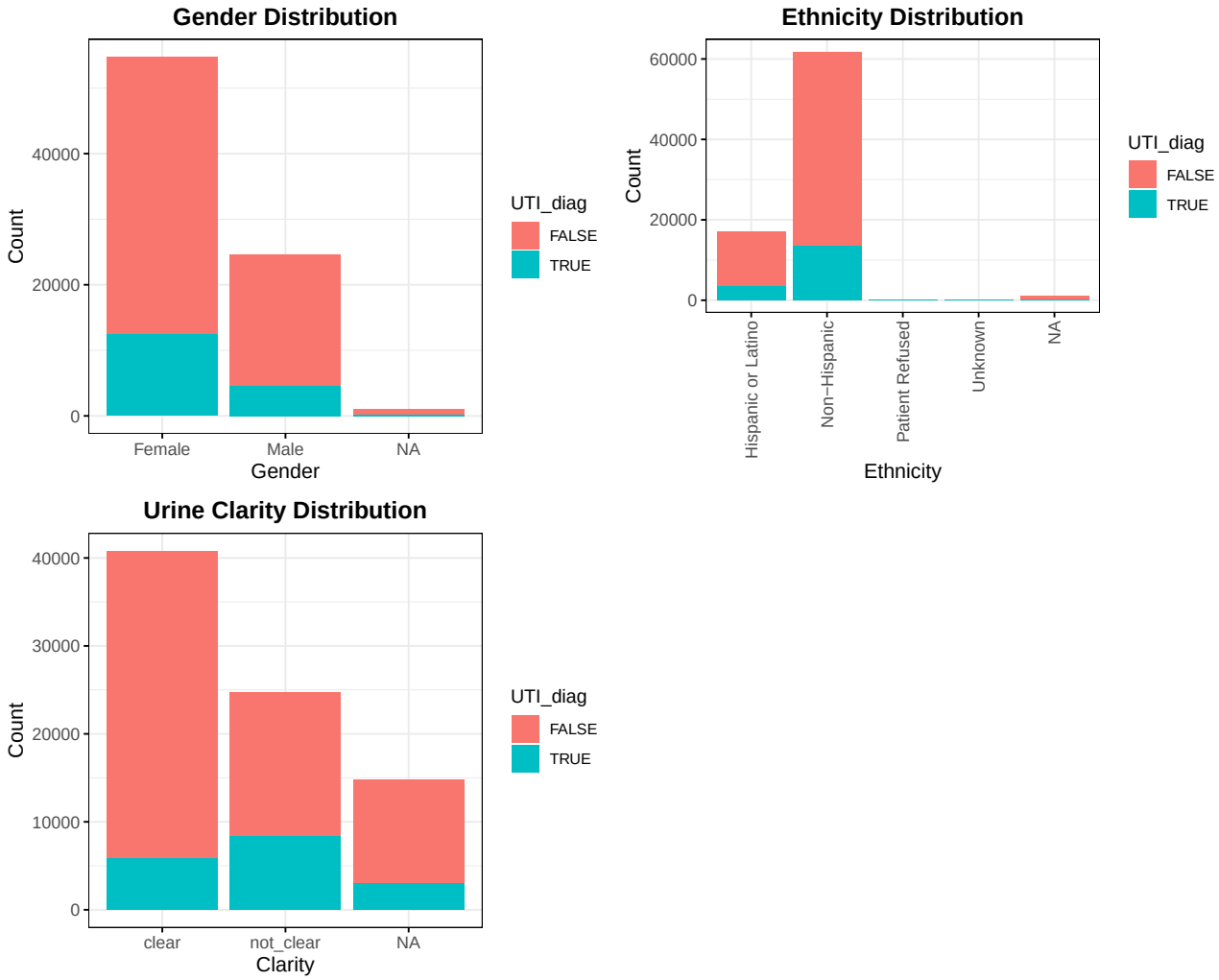

Figure 2: Exploratory Data Analysis of UTI Diagnosis Variables (Part II)

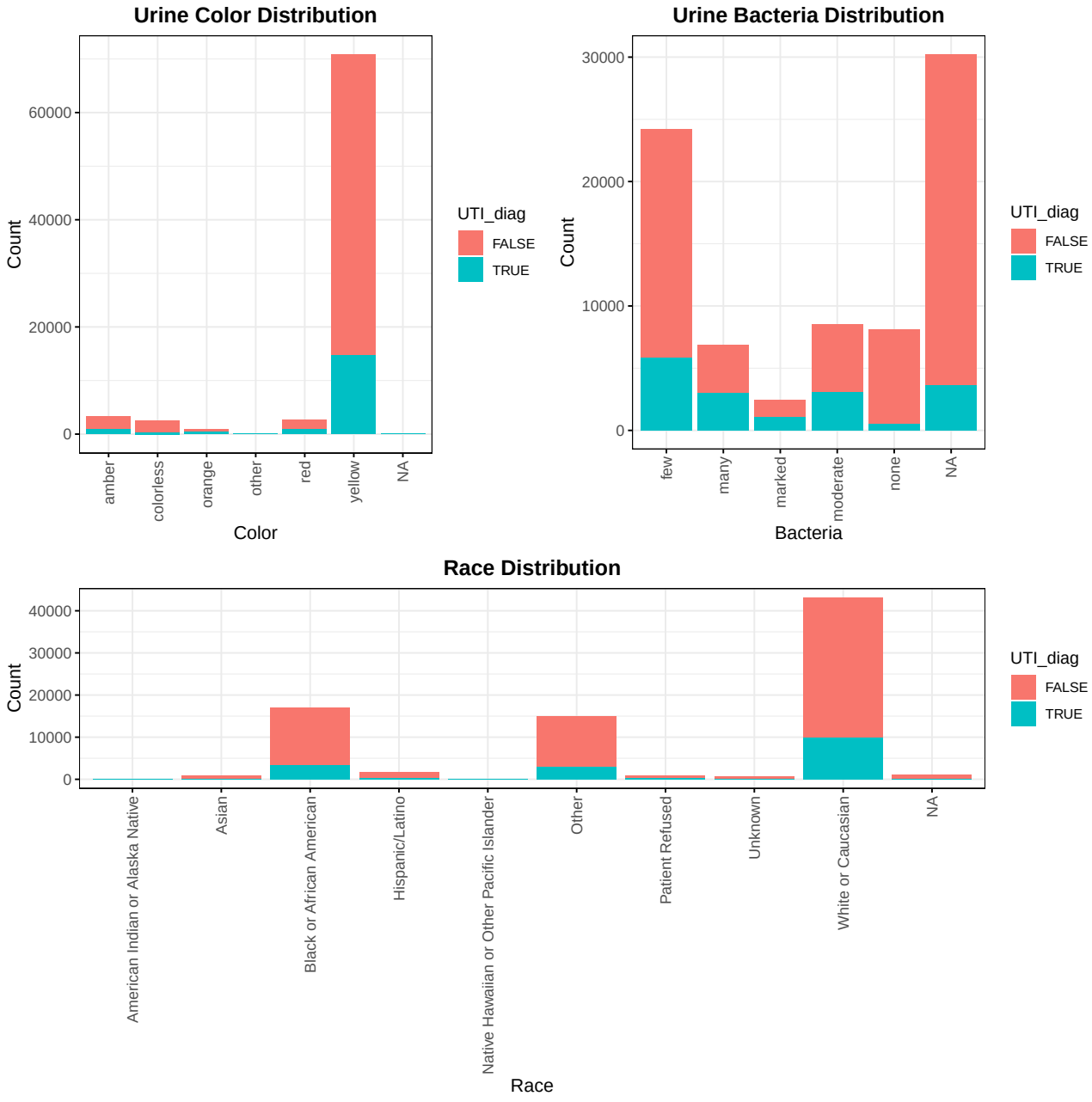

Figure 3: Exploratory Data Analysis of UTI Diagnosis Variables (Part III)

3 Visualizing Descriptive Statistics

We create bar plots and histograms to visualize the distribution of the following variables: age at visit, gender, race, ethnicity, and three measures from the urinalysis section of the dataset.

3.1 Age, Gender and Clarity Visulization

From the bar plots and histograms, we can observe that the age distribution by gender peaks at 55 years, indicating that this age group has the highest number of individuals. The presence of NA values at this age suggests some missing data. The gender distribution shows a higher number of females compared to males overall. In the urine clarity distribution, there are many NA values, but among the recorded data, the number of clear samples is greater than the number of not_clear samples. This indicates that, where data is available, clarity is more commonly observed.

See Figure 7 in appendix for more visulizations on age analysis.

Figure 4: Age, gender and clarity visulization

3.2 Bacteria and WBC Visulization

From the bar plots, we can observe that in the bacteria distribution, most samples have few bacteria, with moderate counts being the third highest. The distribution is fairly consistent across other categories, except for a notable number of NA values, which suggests a substantial amount of missing data in this category. In the White Blood Cell (WBC) distribution, the counts are predominantly on the smaller end, with a significant number of NA values indicating missing data. This indicates that while few bacteria and smaller WBC counts are common, the presence of missing data should be considered when interpreting these results.

See Figure 8 and Figure 9 in appendix for more visulizations on WBC and bacteria analyses.

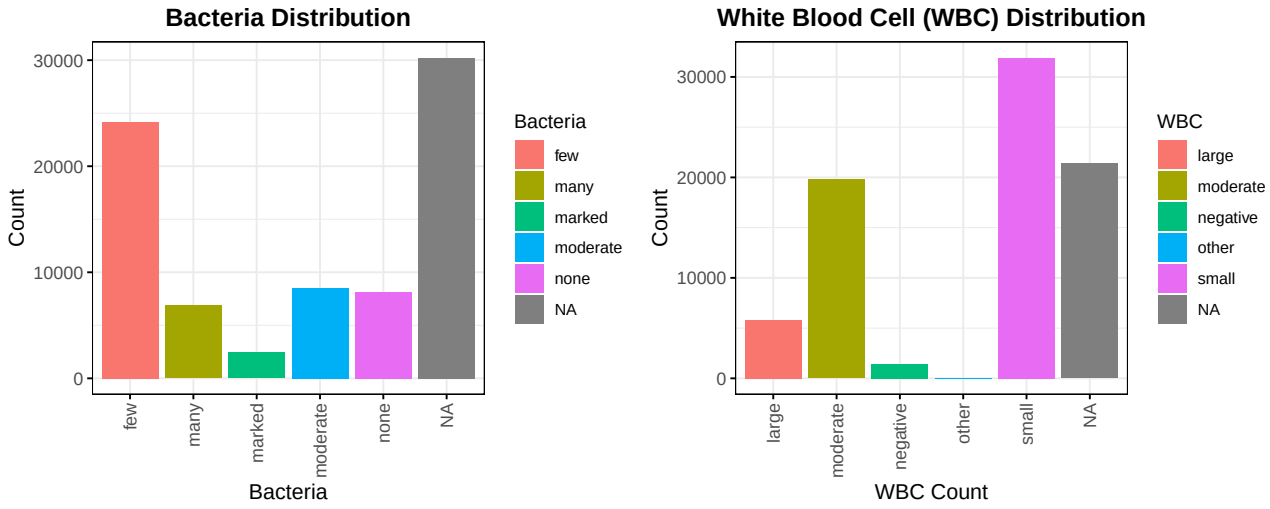

Figure 5: Bacteria and WBC Visulization

3.3 Race and Ethnicity Visulization

From the bar plots, we can observe that the race distribution is predominantly white, with white individuals being the most dominantly represented group in the dataset. Black individuals are the second most represented, followed by other races, which are very few in comparison. In the ethnicity visualization, there are few NA values, unknown or patient refused, with the majority of individuals identified as Non-Hispanic, followed by Hispanic or Latino. This indicates a significant racial and ethnic homogeneity in the dataset, with white and Non-Hispanic individuals being the most prevalent.

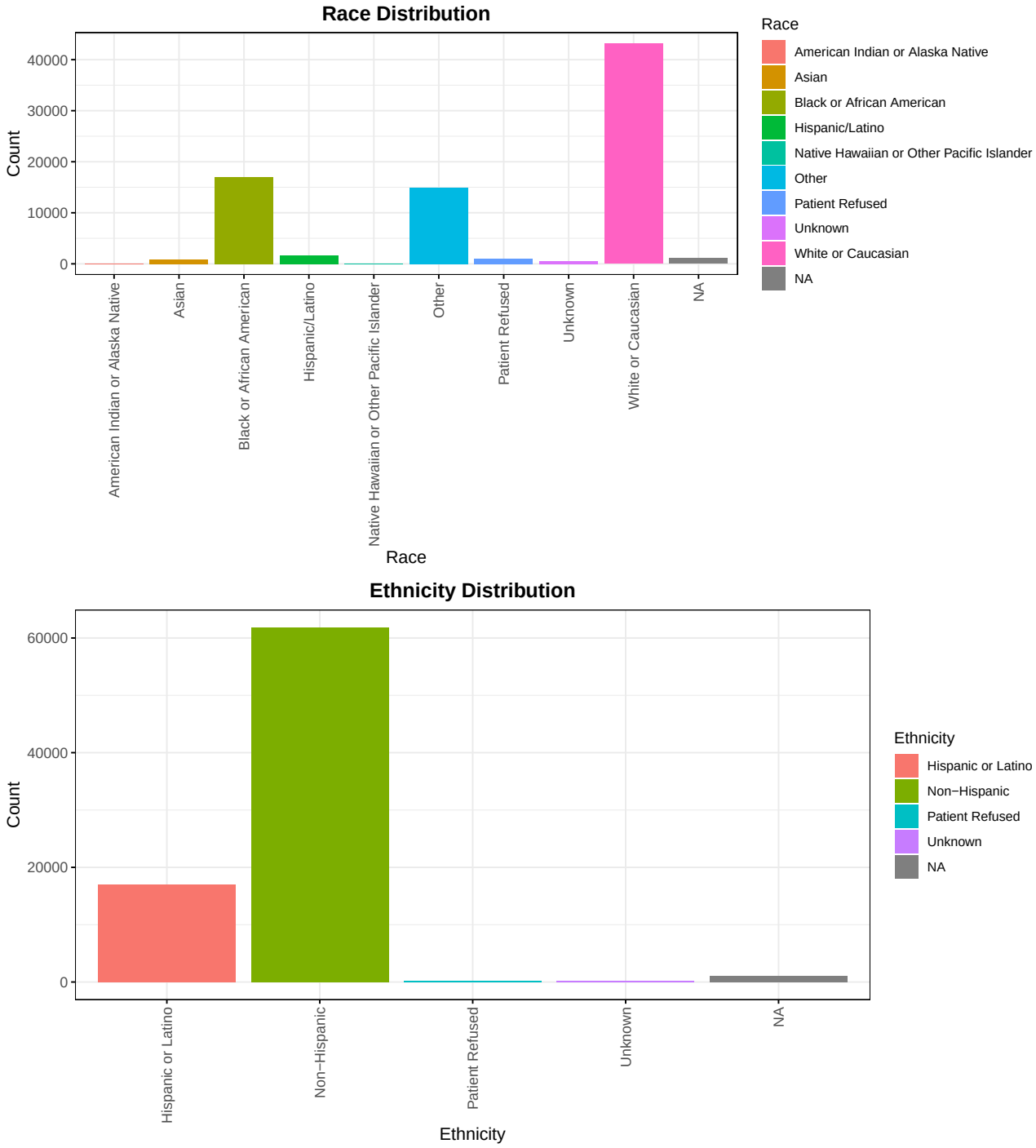

Figure 6: Race and Ethnicity Visulization

3.4 Summary

The dataset primarily consists of white and female individuals, with a notable presence of Non-Hispanic ethnicity. Urine samples generally show few or no reported bacteria, and most samples exhibit clear urine, though there is a substantial amount of missing clarity data. A significant proportion of data is missing in bacteria and White Blood Cell (WBC) counts. Interestingly, there are inconsistencies between bacteria and clarity distributions, with the presence of few bacteria contrasting with more moderate and many bacteria observations. The data highlights a racial and ethnic homogeneity, with clear urine being more common where available.

4 Short Answer Questions

4.1 Discuss which features you selected to include in your ML models and explain why

I employed multiple approaches to feature selection, including logistic regression (GLM) with p-values, backward selection, and domain-driven variable engineering. Initially, I used p-values to refine features, then leveraged backward selection to further streamline the model while maintaining interpretability. Additionally, I analyzed feature importance from Random Forest and XGBoost, both of which offered robust ranking methods. Interestingly, XGBoost and Random Forest provided similar rankings and achieved better AUC than GLM. To ensure a balanced feature set across all models, I selected common variables identified by multiple approaches, including `patid`, `ua_bacteria`, `ua_wbc`, `age`, `ua_clarity`, `dispo`, `antibiotics`, `ua_leuk`, `abx`, and `ua_nitrite`.

4.2 Discuss the significance of separating data into training, validation, and test sets

Dividing data into training, validation, and test sets is crucial for building a reliable machine learning model. The training set is used to train the model, allowing it to learn patterns from the data. The validation set helps fine-tune hyperparameters and prevent overfitting by providing an unbiased evaluation of model performance during development. The test set serves as the final checkpoint, assessing the model's generalizability to unseen data. I split the dataset into 70% training, 15% validation, and 15% testing to ensure a balanced approach. This setup helps prevent bias, improves model performance, and ensures that results are not overly optimistic due to data leakage. Specifically:

- Training set: The training set (70% of the data) is used to fit the model, allowing it to learn patterns and relationships between features and the target variable (`UTI_diag`). A larger training set helps the model generalize better to unseen data.
- Validation set: The validation set (15% of the data) is mainly used to fine-tune hyperparameters. It provides an unbiased evaluation of the model during training, helping to prevent overfitting before final testing.

- Test set: The test set (15% of the data) is reserved for the final evaluation of the trained model. Since it is not used in training or validation, it provides an objective measure of model performance on completely unseen data, ensuring reliability before deployment.

- Splitting decision: The choice of a 70-15-15 split balances training efficiency and evaluation reliability. A larger training set ensures the model learns effectively, while separate validation and test sets help optimize and fairly assess model performance. Alternative splits could be considered based on dataset size and complexity. For example, 80-10-10 would prioritize training but reduce validation and testing reliability, while 60-20-20 would provide a more thorough evaluation at the cost of less training data. The selected 70-15-15 split provides a good trade-off between model training and unbiased assessment.

- Comparison of data splitting methods: In R, there are different ways to implement this split. The traditional method uses the `sample()` function to randomly assign data to training, validation, and test sets. While this method is flexible, it requires manual handling of each split step. It also requires explicit verification of the data sizes to ensure the proper proportions (e.g., 70%, 15%, 15%). On the other hand, the `caTools` package provides a more streamlined approach to splitting datasets. The `sample.split()` function from this package simplifies the process by directly handling the random sampling and splitting. It can be more efficient and less error-prone, particularly when dealing with larger datasets. The `caTools` package simplifies the random splitting process, handling both the training and testing splits in fewer lines of code and ensuring a consistent sampling strategy. This can be particularly advantageous when scaling the approach to larger datasets or automating the process for repeated experiments.

4.3 Discuss the concept of data leakage

Data leakage refers to a situation where information from outside the training dataset, or knowledge about the target variable, unintentionally influences the model during training. This results in overly optimistic performance estimates because the model gains access to information that would not be available in a real-world scenario where it must make predictions on truly unseen data. As a result, the model may perform well during training and validation, but fail to generalize to new data.

One common form of data leakage occurs when the validation set is used during feature selection. The validation set is intended to provide an unbiased estimate of the model's performance during hyperparameter tuning and model evaluation. However, if the validation data is involved in feature selection or any part of the training process, it indirectly influences model training. This can cause the model to appear more accurate on the validation set, even though it may fail to generalize to the test set or new data in practice, as the validation set is no longer independent.

To avoid data leakage, it is critical to maintain a strict separation between training, validation, and test datasets. The validation set should only be used for hyperparameter tuning and final model evaluation, while feature selection and training should only rely on the training data. Additionally, other strategies, such as avoiding the inclusion of variables that may be derived from the target, using a temporal cutoff, adding random noise, or even re-sampling the data

appropriately, can help prevent leakage and ensure the model's performance reflects its true predictive power on unseen data.

5 Appendix

5.1 Exploratory Data Analysis (Cont'd)

The plots below are used to analyze relationships between age, WBC count, bacteria presence, and demographic factors. Boxplots are used to compare age distributions across gender, urine clarity, ethnicity, and WBC count, highlighting any age differences within these groups. Additionally, bar plots are used to examine the distribution of WBC count and bacteria presence by gender, urine clarity, and ethnicity, showing the counts within each category.

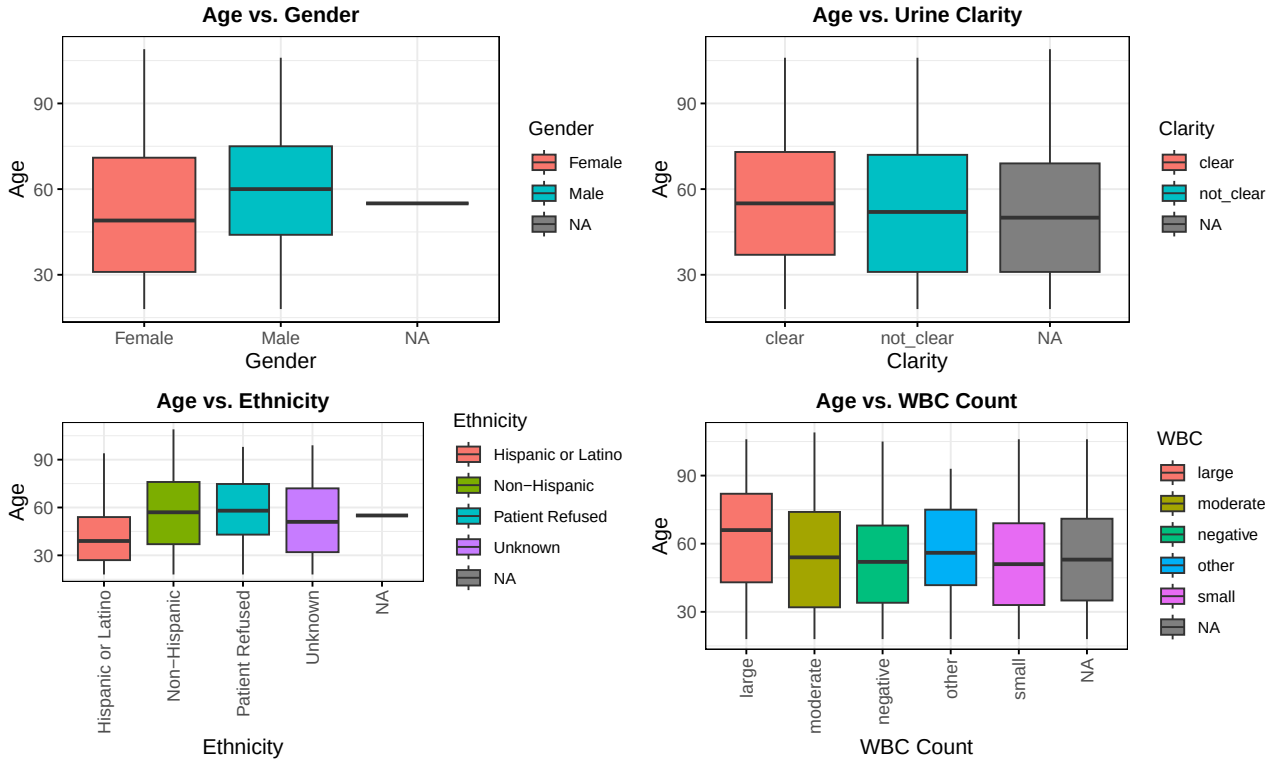

Figure 7: Appendix - Age analysis

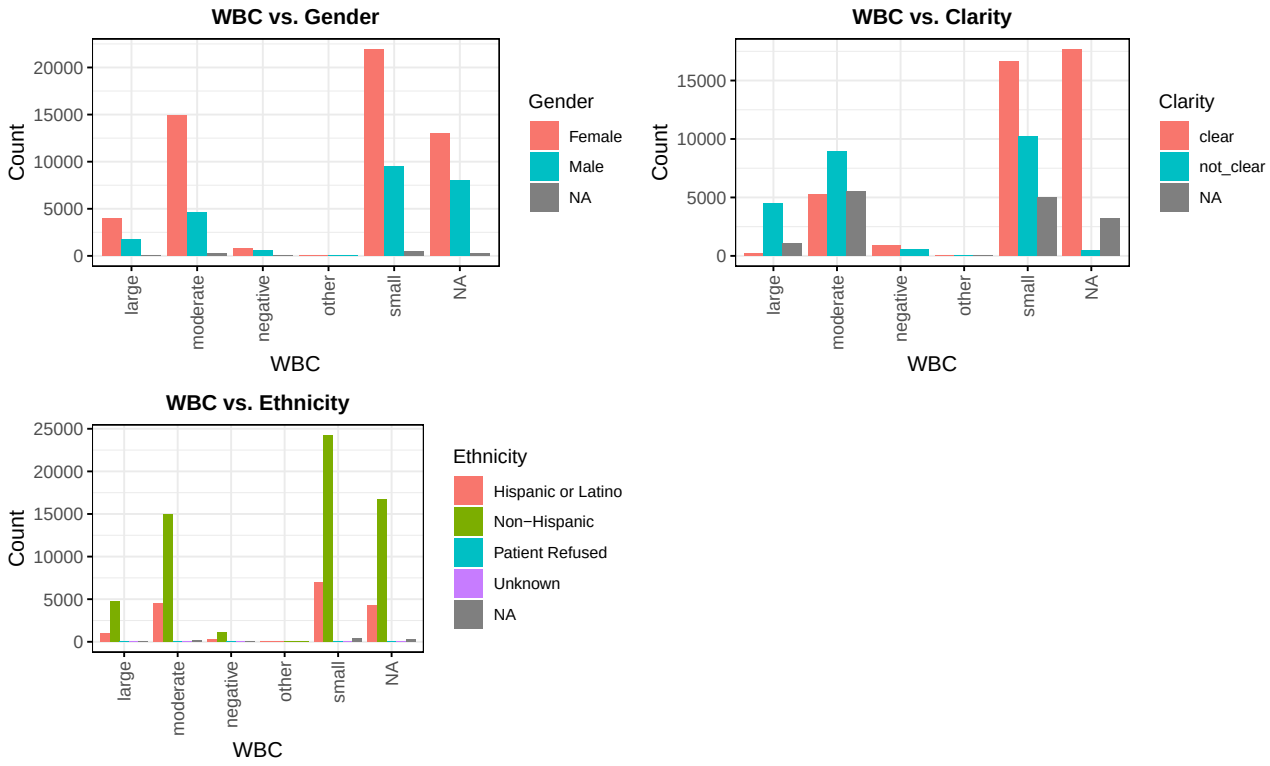

Figure 8: Appendix - WBC analysis

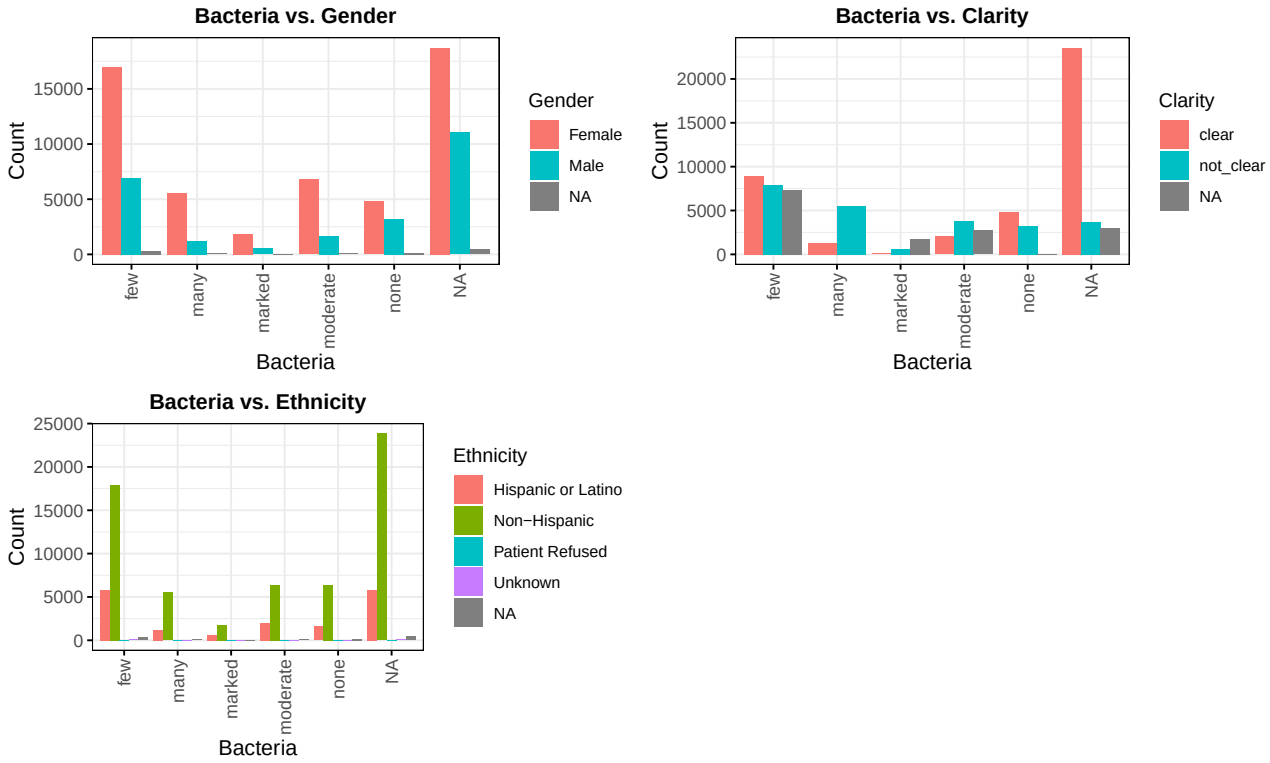

Figure 9: Appendix - Bacteria analysis

5.2 Model Performance

5.2.1 Preprocessing

The preprocessing steps involve splitting the dataset into training (70%), validation (15%), and testing (15%) sets while ensuring reproducibility. Outliers in key numeric variables (e.g., `ua_spec_grav` and `age`) are removed based on the 1st and 99th percentiles. Categorical variables are converted to character format, and missing data patterns are visualized. Measurement variables (e.g., temperature, heart rate, and oxygen saturation) are examined, with redundant columns dropped to reduce multicollinearity. Missing categorical values are replaced with `not_reported` for interpretability. Finally, the cleaned datasets are saved for model training.

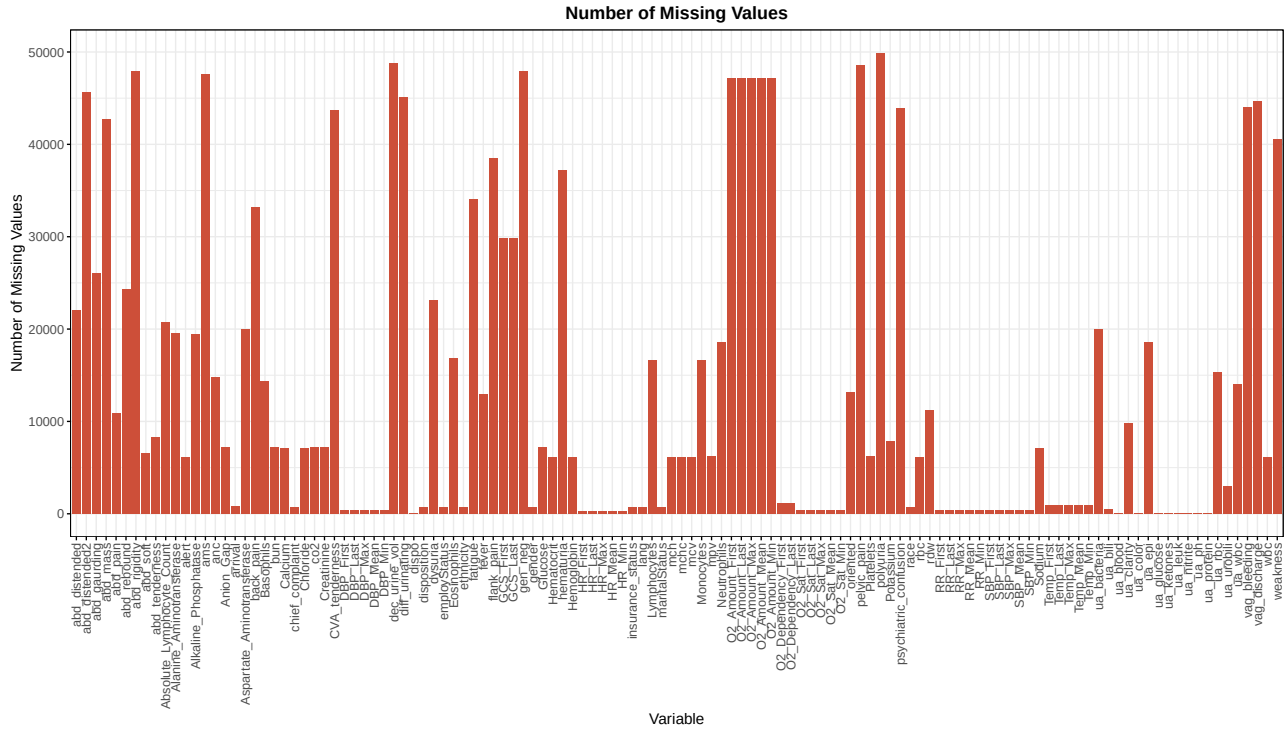

Figure 10: Missing Data Distribution

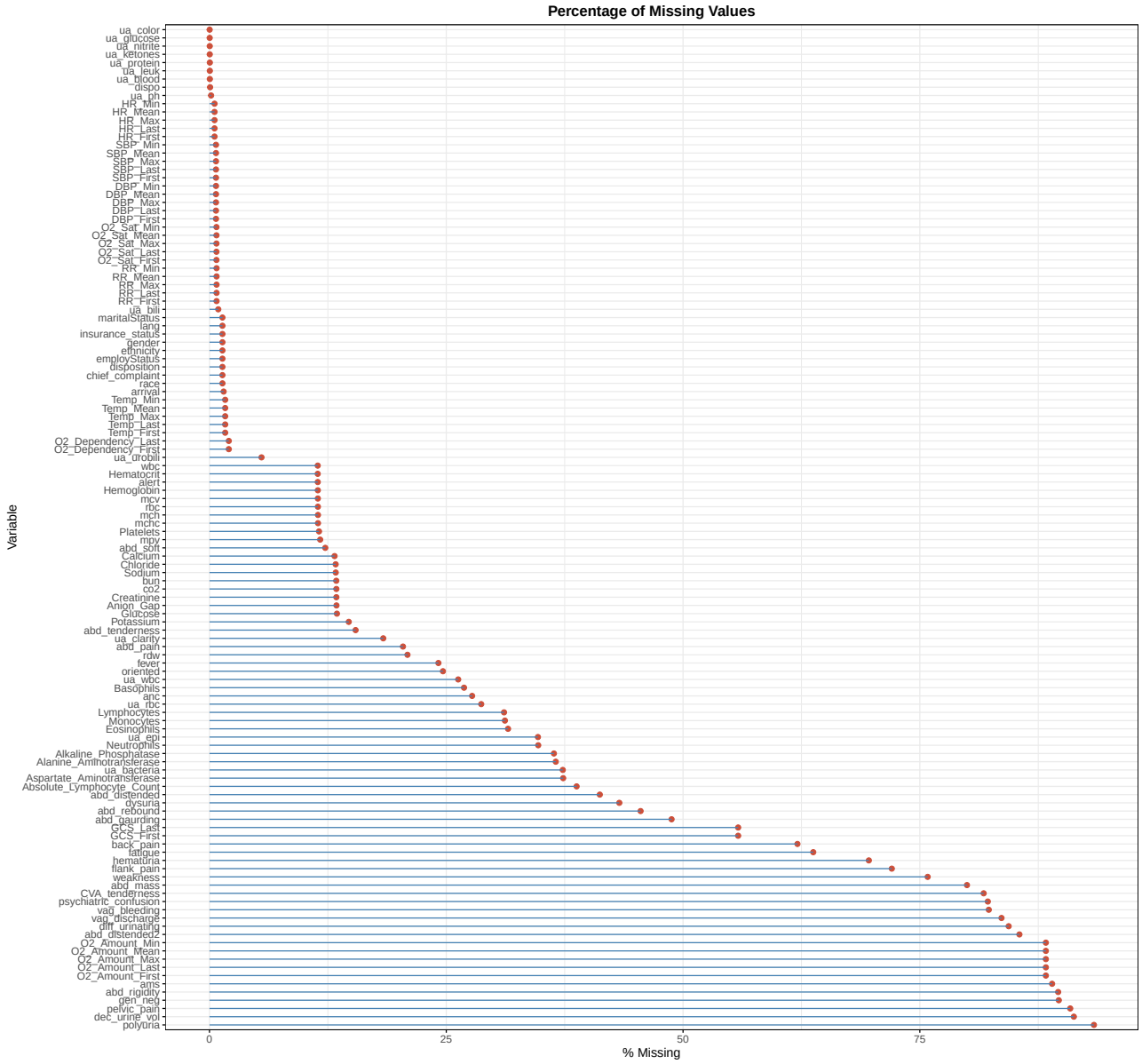

Figure 11: Percentage of Missing Data (%)

5.2.2 Feature Selection for Analysis

This analysis explores feature selection and modeling for predicting UTI diagnosis using logistic regression, XGBoost, and random forest. Logistic regression was used for manual feature selection, while XGBoost performed automatic feature selection. Random forest served as an additional method primarily to evaluate model performance.

The feature selection process began with an initial logistic regression model including all available predictors. Variables with p-values < 0.1 were retained, followed by further refinement to remove less significant features. Categorical variables, such as insurance status and ethnicity, were transformed into binary indicators. Stepwise backward selection was applied to optimize the model based on AIC and BIC criteria. Through iterative refinement, the number of predictors was progressively reduced, leading to a final streamlined model with 11 key variables. The final model demonstrated strong predictive performance (AUC = 0.8117) and retained clinically relevant features, including urinalysis results, CVA tenderness, psychiatric confusion, flank pain, age, gender, and ethnicity.

5.2.3 Logistic Regression Model

Performance Evaluation for Logistic Regression Model:

The logistic regression model demonstrates strong discriminative ability in classifying UTI-positive and UTI-negative cases, as indicated by the AUC-ROC values. With a validation AUC-ROC of approximately 0.8106 and a testing AUC-ROC of 0.8117, the model effectively balances sensitivity and specificity. An AUC above 0.80 suggests excellent performance, confirming that the model can reliably differentiate between the two classes. The consistency between validation and testing results further indicates good generalization and minimal overfitting.

In addition to AUC-ROC, the Precision-Recall (PR) Curve provides valuable insights, particularly in the presence of class imbalance. The validation and testing AUC-PR values, approximately 0.6349 and 0.6352, respectively, highlight the model's ability to correctly classify positive cases. Although these values are lower than AUC-ROC, they remain reasonable given the challenges posed by imbalanced datasets in healthcare applications. The model effectively captures relevant patterns, ensuring meaningful identification of true positives, which is essential for minimizing false negatives in medical diagnosis.

While the model performs well, there is potential for further optimization. Addressing class imbalance through resampling techniques, adjusting decision thresholds, or incorporating additional predictive features could enhance its performance. Overall, the logistic regression model provides a solid foundation for UTI diagnosis, demonstrating both robustness and reliability in real-world applications.

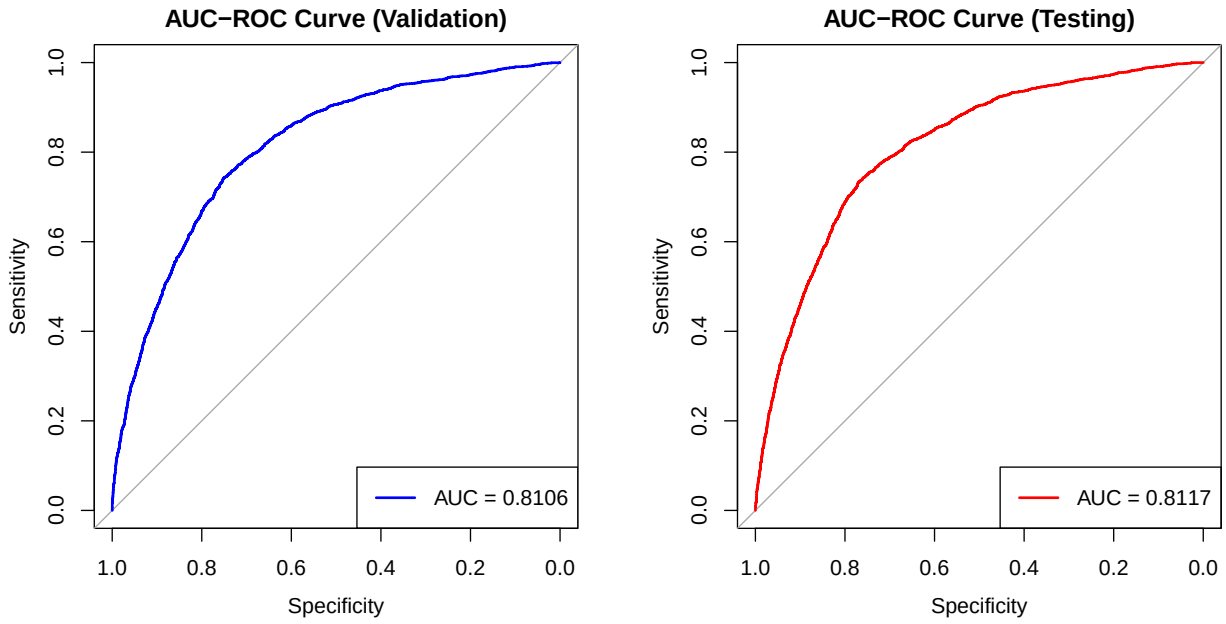

Figure 12: AUC-ROC Curve (Logistic Regression)

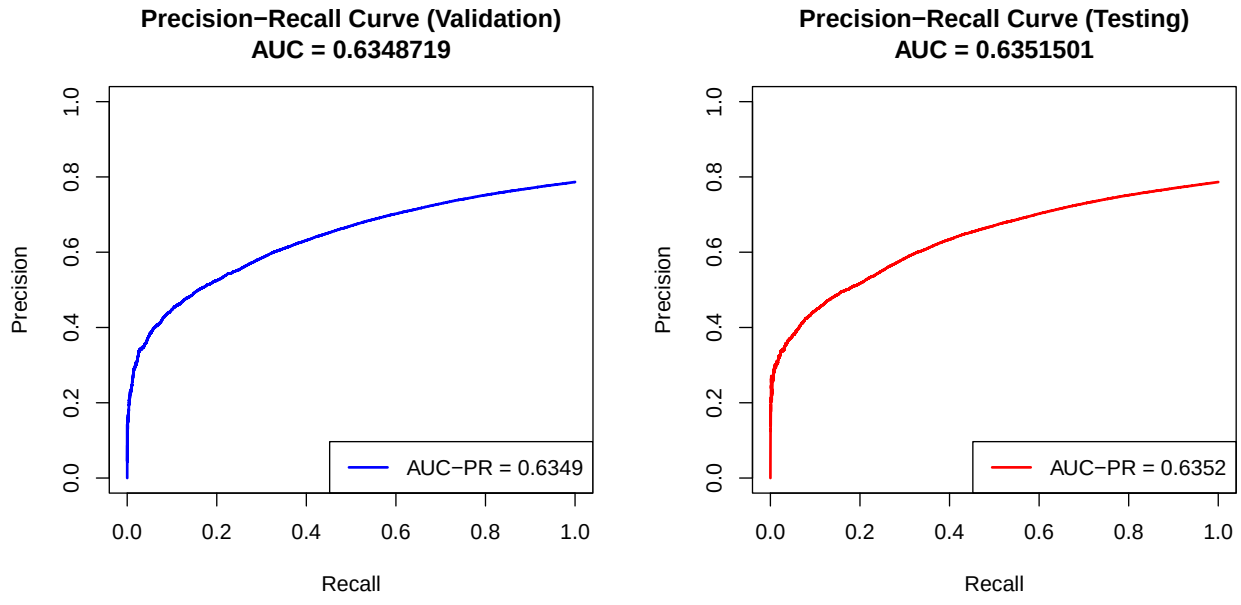

Figure 13: Precision-Recall AUC Curve (Logistic Regression)

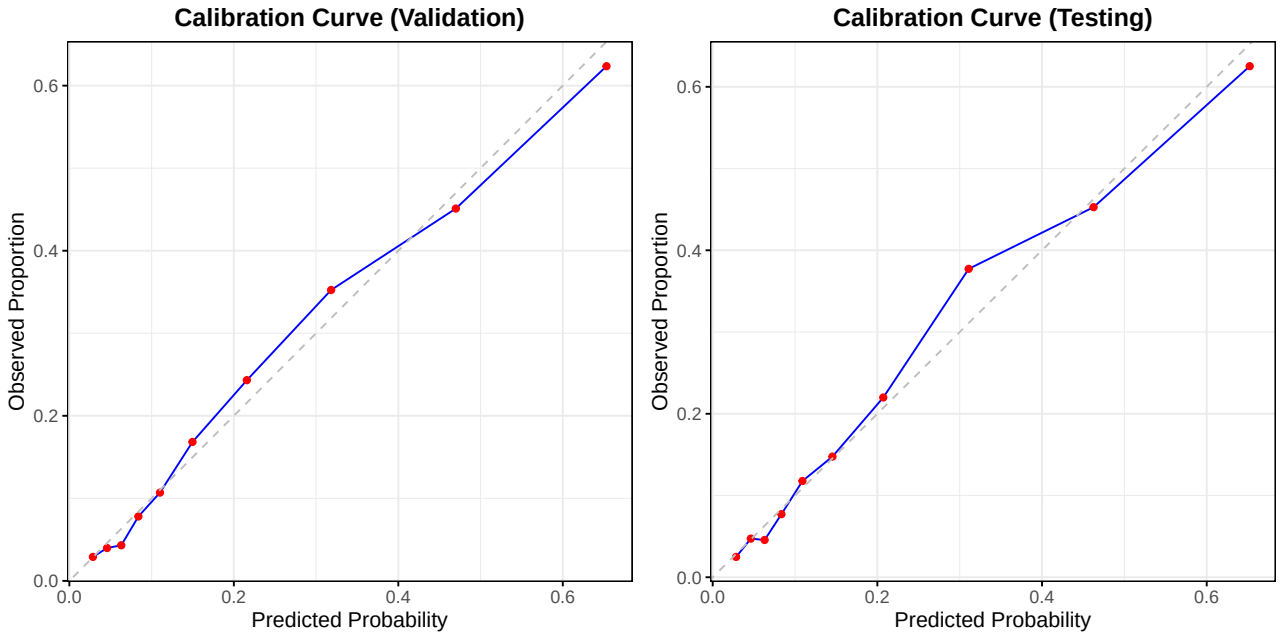

Figure 14: Calibration Curve (Logistic Regression - prob)

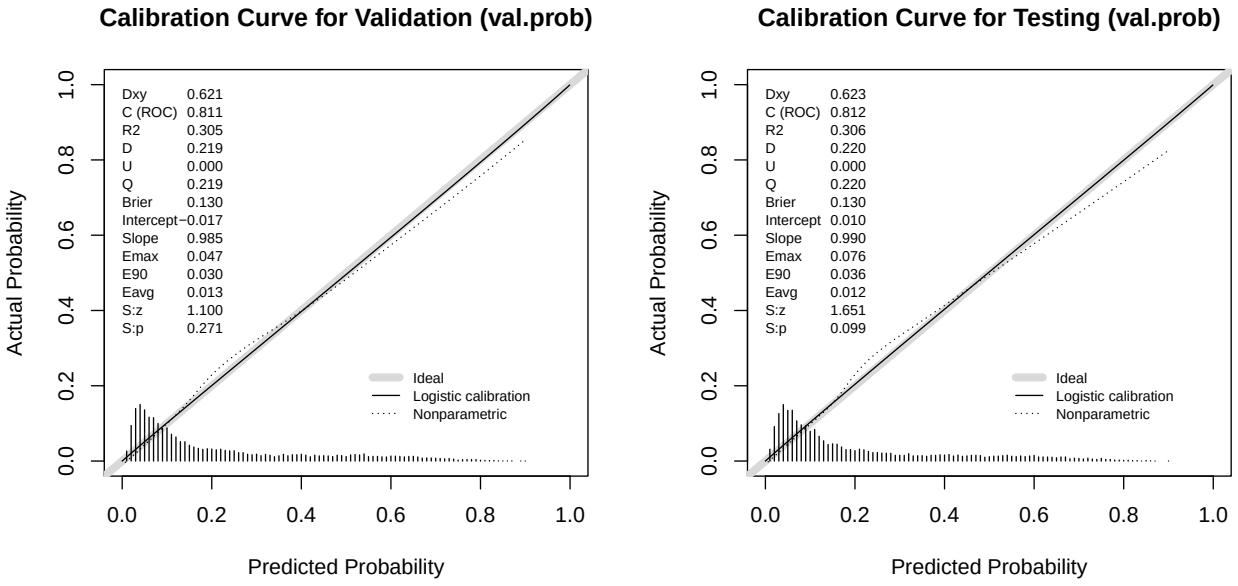

Figure 15: Calibration Curve (Logistic Regression - val.prob)

5.2.4 Random Forest Model

Performance Evaluation for Random Forest Model:

Model Accuracy: The Random Forest model achieved a validation accuracy of 0.8273 and a test accuracy of 0.83, indicating a strong ability to classify UTI-positive and UTI-negative cases. The minimal performance drop from validation to test suggests that the model generalizes well without significant overfitting.

AUC-ROC Analysis: The Receiver Operating Characteristic (ROC) Curve assesses the model's ability to distinguish between positive and negative cases. The validation AUC-ROC is 0.8755, and the testing AUC-ROC is 0.8738, confirming that the model has strong discriminative ability. The small gap between validation and test scores indicates stability and reliability. However, the Random Forest model's AUC-ROC is slightly lower than the XGBoost model (above 0.93), suggesting that XGBoost may be a more effective classifier.

Precision-Recall AUC Analysis: The Precision-Recall (PR) Curve is particularly useful for evaluating model performance on imbalanced datasets. The validation AUC-PR is 0.6134, while the testing AUC-PR is 0.6136, demonstrating consistent performance across different datasets. While slightly lower in comparison to AUC-ROC, the Random Forest model outperforms XGBoost in AUC-PR (approximately 0.59), indicating better precision and recall trade-offs, which may be beneficial in handling class imbalance.

Overall, the Random Forest model performs well in predicting UTI diagnosis, providing a balance between accuracy and interpretability. However, further optimizations may enhance performance, particularly in distinguishing positive cases with higher precision.

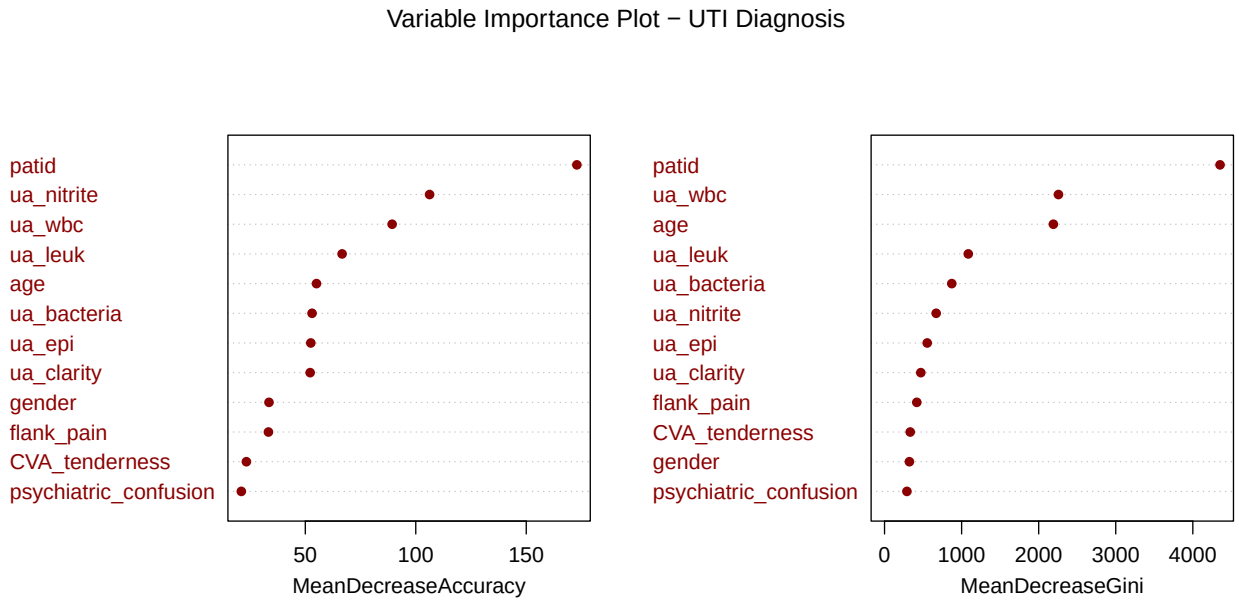

Figure 16: Variable Importance Plot

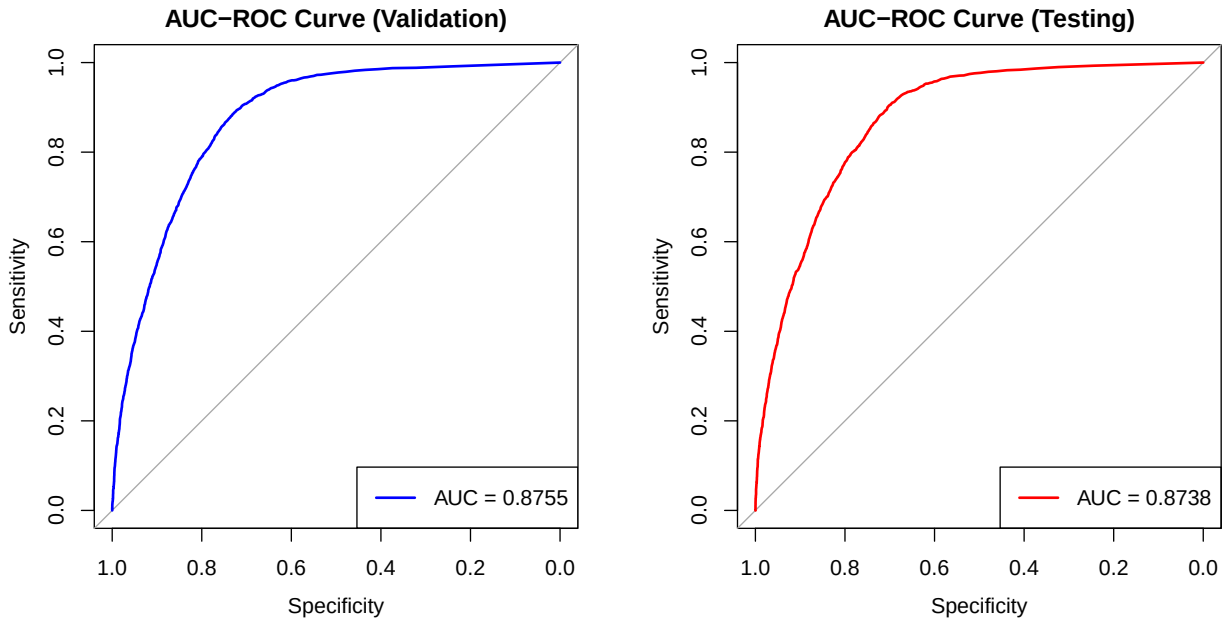

Figure 17: AUC-ROC Curve (Random Forest)

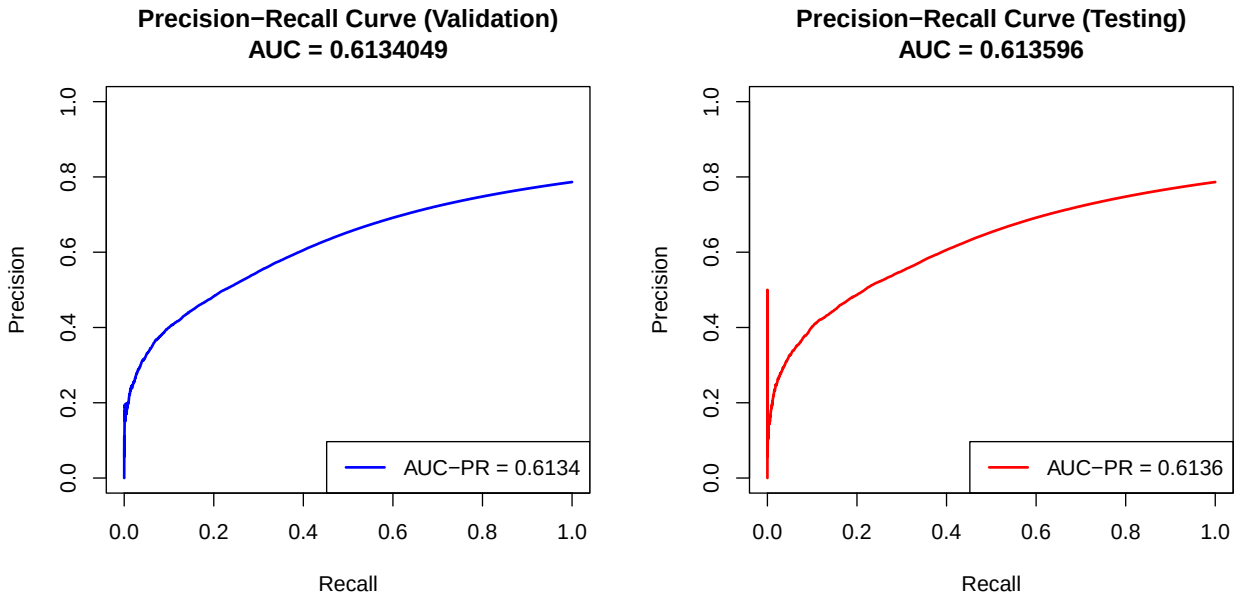

Figure 18: Precision-Recall AUC Curve (Random Forest)

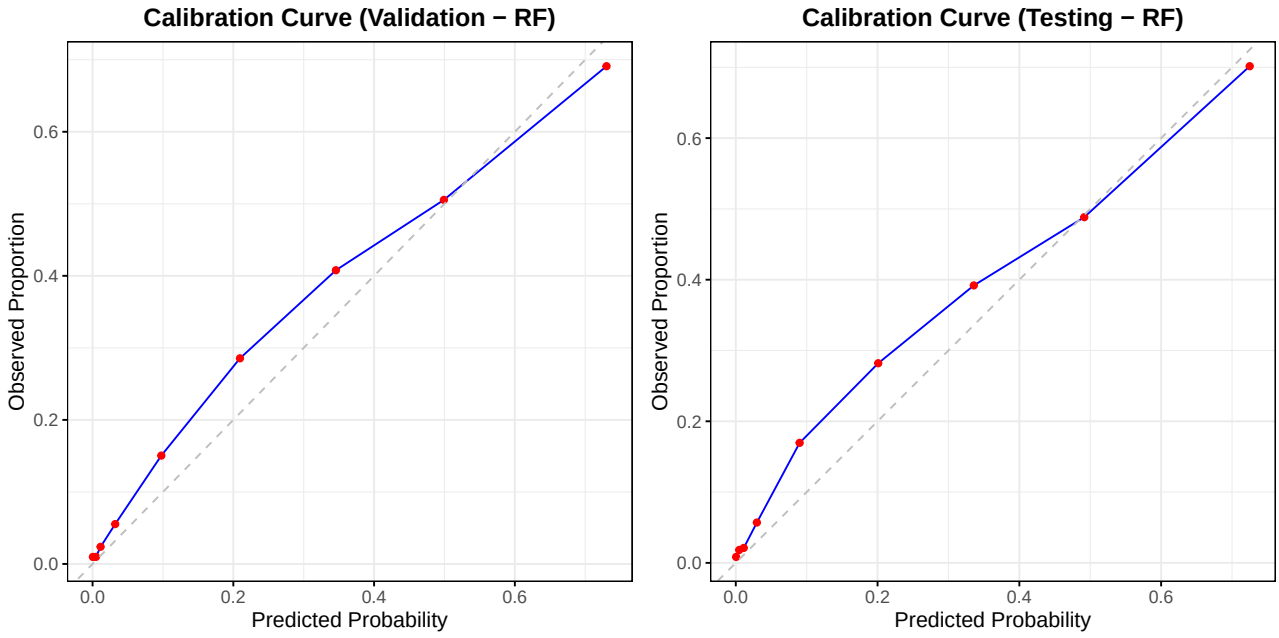

Figure 19: Calibration Curve (Random Forest)

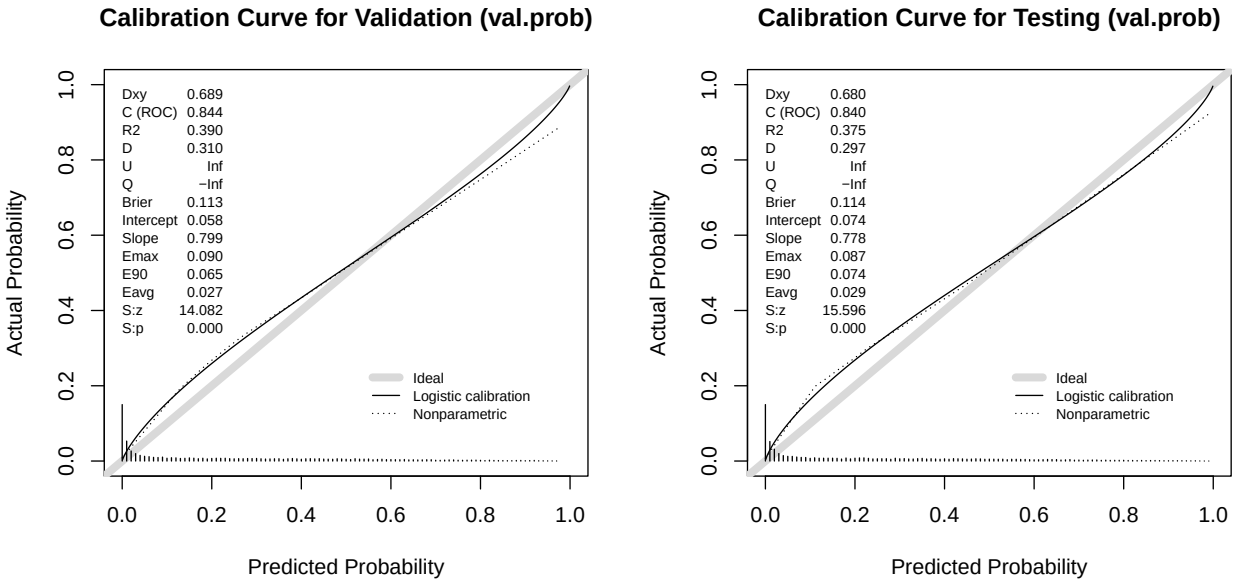

Figure 20: Calibration Curve (Random Forest - val.prob)

5.2.5 XGBoost

Performance Evaluation for XGBoost Model:

The XGBoost model demonstrates strong overall classification performance, with validation accuracy of 88.9% and test accuracy of 89.6%, indicating good generalization to unseen data. Compared to the Random Forest model, which had slightly lower accuracy, XGBoost appears to make better predictions on both validation and test sets. However, as with Random Forest, accuracy alone does not fully capture the model's performance, particularly when dealing with class imbalance, where correctly classifying the majority class can mask errors in predicting the minority class. Despite this, the consistency between validation and test accuracy in XGBoost suggests that overfitting is not a significant concern, making it a robust choice for this classification task.

In terms of discriminatory power, XGBoost achieves a validation AUC-ROC of 0.9348 and test AUC-ROC of 0.9564, outperforming the Random Forest model, which had a lower AUC-ROC. This suggests that XGBoost is better at ranking positive instances higher than negative instances, meaning it provides stronger separation between classes. The improved test AUC-ROC in XGBoost further highlights its ability to generalize better than Random Forest. However, while AUC-ROC provides a useful overall measure of performance, it does not consider the impact of class imbalance, as it evaluates both classes equally. This means that despite a strong AUC-ROC, the model may still struggle in cases where correct classification of the minority class is crucial.

The Precision-Recall AUC, which specifically evaluates the model's performance in identifying positive cases, provides a clearer picture of how XGBoost handles class imbalance. With a validation AUC-PR of 0.5934 and test AUC-PR of 0.5878, XGBoost performs moderately well but does not show a significant advantage over Random Forest, which had similar PR AUC values. This suggests that, while XGBoost improves overall ranking ability (as seen in AUC-ROC), it does not necessarily lead to a significant boost in precision-recall tradeoffs. The relatively lower PR AUC values indicate that the model may still struggle with balancing precision and recall, potentially leading to missed positive cases (low recall) or too many false positives (low precision). As a result, further optimization (e.g., class weighting, threshold tuning, or resampling techniques) may be necessary to improve positive class detection.

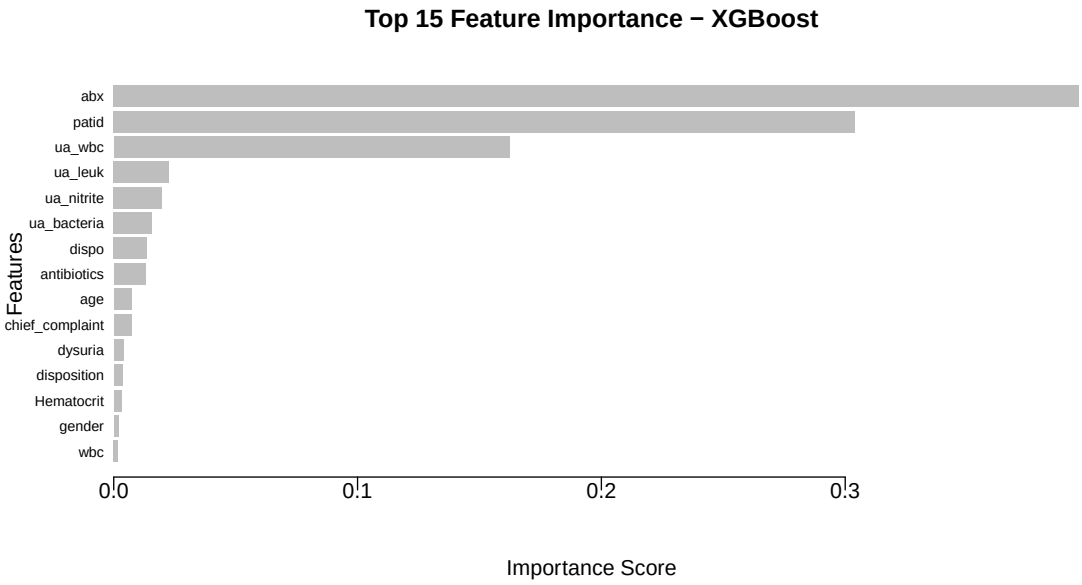

Figure 21: Feature Importance Analysis

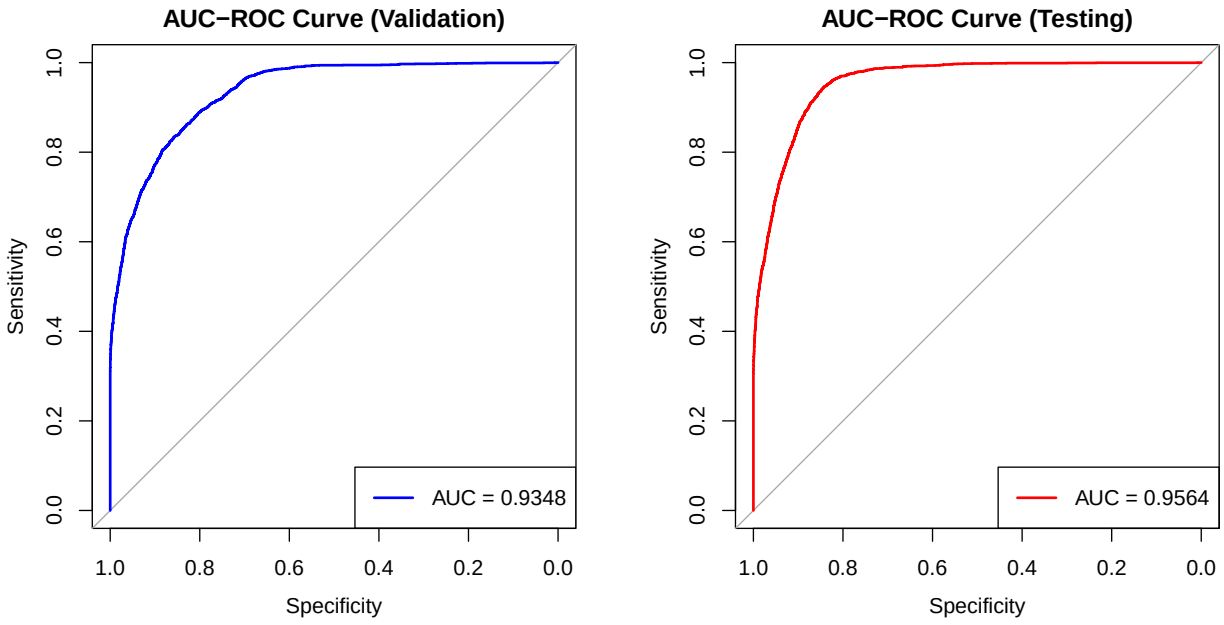

Figure 22: AUC-ROC Curve (XGBoost)

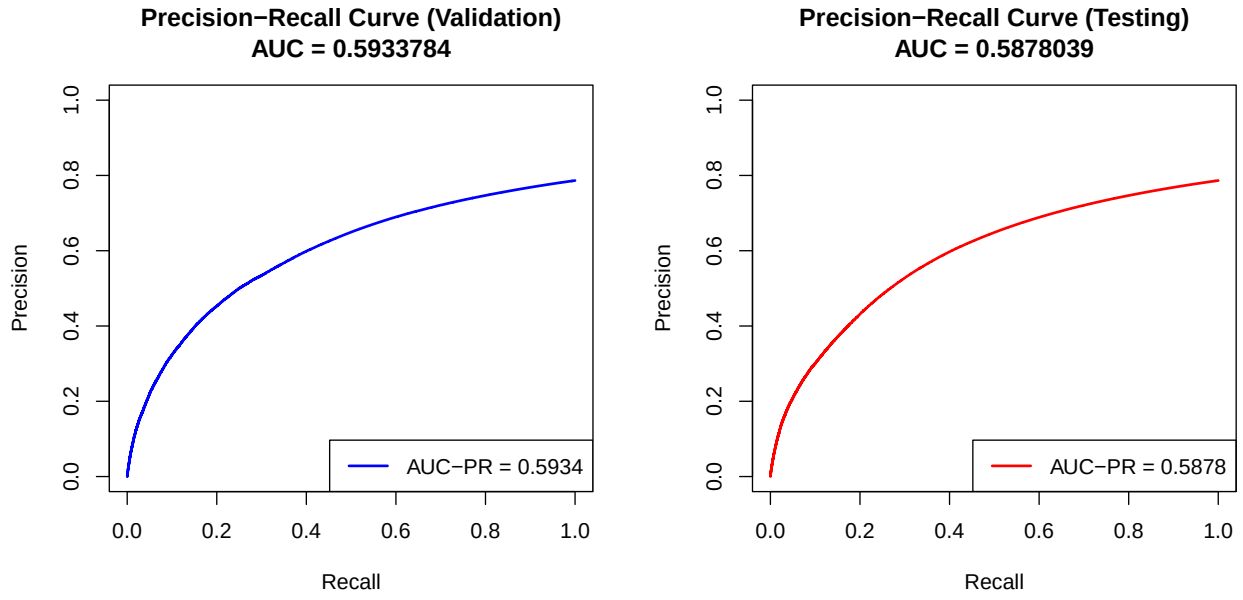

Figure 23: Precision-Recall AUC Curve (XGBoost)

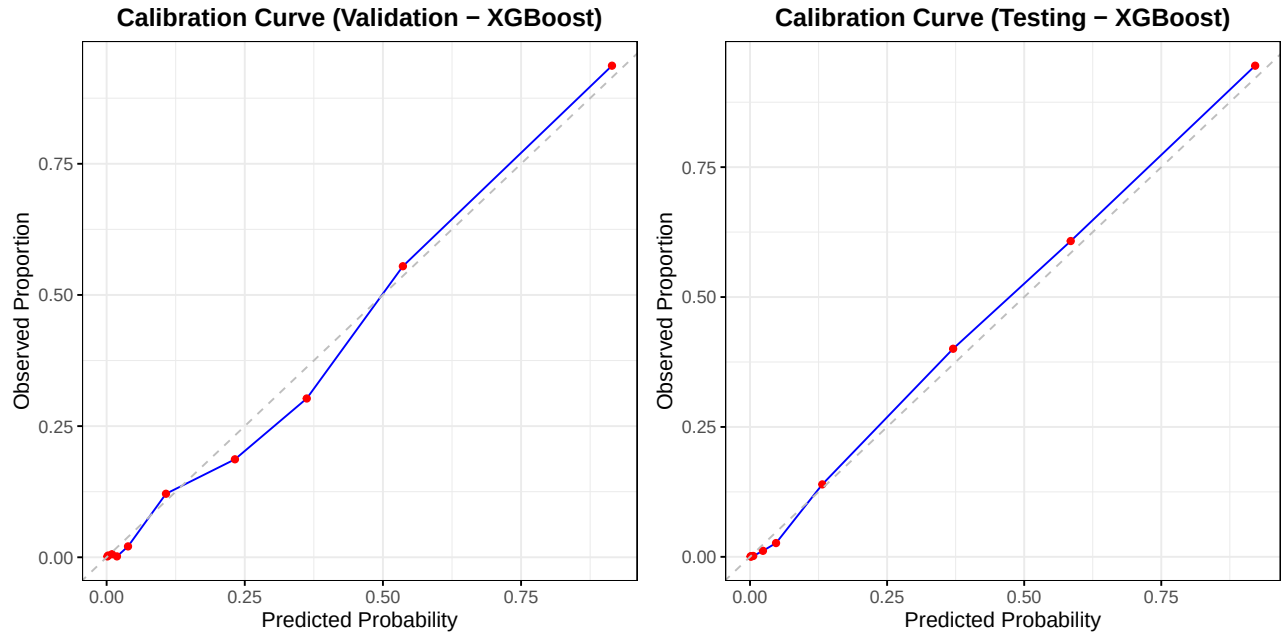

Figure 24: Calibration Curve (XGBoost - prob)

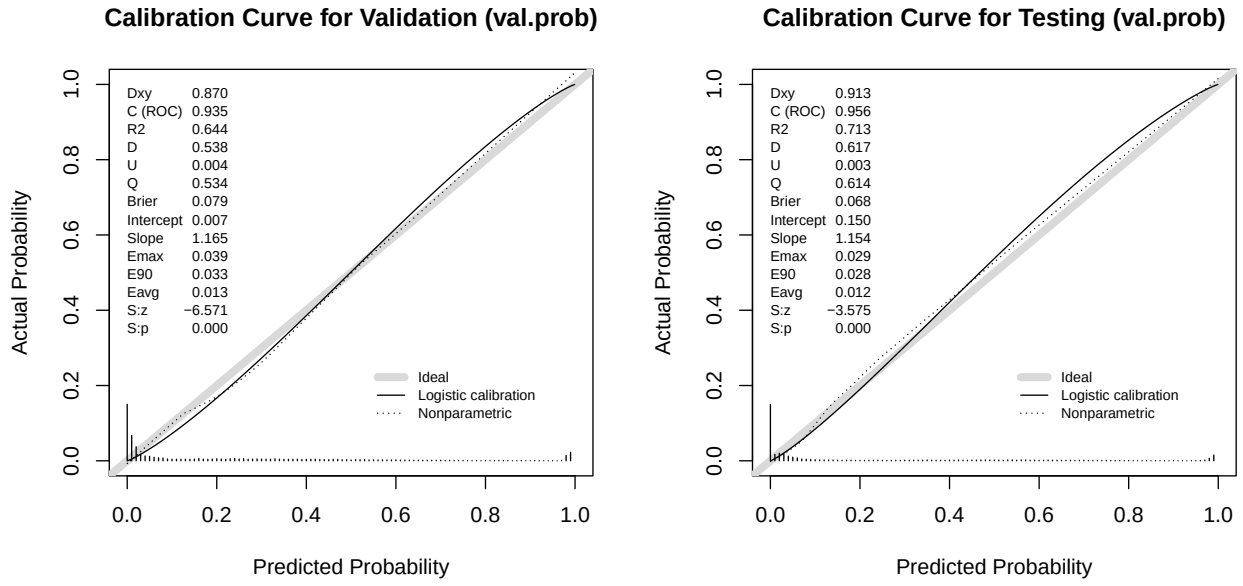

Figure 25: Calibration Curve (XGBoost - val.prob)