Monte Carlo Experiment: Logistic Regression and Random Forest

Aaliya Merchant, Aaron Tsui, Nikhil Prabhu, Shiraz Rehmani

MSDS 460: Decision Analytics

Professor Thomas W. Miller

November 17th, 2024

Abstract

This study investigates the application of machine learning models, specifically Logistic Regression and Random Forest, on a drug prescription dataset. The dataset of 200 records contains patient information, including age, gender, blood pressure, cholesterol levels, sodium-to-potassium ratio, and drug type. The research employs a combination of exploratory data analysis, simulations, and model evaluation to assess model performance in both Python and R environments. Key metrics recorded included accuracy, confusion matrix, classification reports, and training time. They were recorded across many Monte Carlo simulations to test model robustness.

Keywords: Machine Learning, Drug Prescription, Model Evaluation, Monte Carlo

Introduction

With the increasing use of machine learning techniques in healthcare, the ability to analyze large-scale patient data for drug prescription optimization has become a critical area of study. This paper explores the performance of two commonly used ML algorithms– Logistic Regression and Random Forest. Through a structured analysis that includes preprocessing the data, exploratory data analysis, model training, and simulation-based evaluations, the study aims to assess the efficacy and robustness of both models in predicting appropriate drug prescriptions.

The dataset was processed and analyzed using both languages to facilitate a comparative performance assessment. This study aims to provide actionable insights into the advantages and disadvantages of each tool for ML applications in healthcare. The findings from this analysis will help inform future decision-making when it comes to tool selection for data-driven healthcare

applications, emphasizing the importance of balancing model performance with computational efficiency.

Literature Review

The application of machine learning in healthcare has grown exponentially in recent years, with a particular focus on optimizing drug prescriptions (the purpose of this study), patient diagnosis, specialized treatment plans as well as mitigating medical errors. Rajkomar et al. (2019) discuss and emphasize ML applications in healthcare, including personalized medicine and drug prescription optimization. Albanese and Hug (2018) highlight how ML models improve existing prescription accuracy and reduce the risk of human error, negatively impacting patients.

Machine Learning implementation is showing significant promise in mitigating medication errors as well. Sendak et al. (2023) demonstrate the potential of ML systems that outperform traditional rule-based systems revealing 68.2% of unique alerts potentially saving over $1.3 million by preventing adverse events. Zhan et al. (2019) employed ML methods for outlier detection to enhance the accuracy and effectiveness of medication error alerts in inpatient settings resulting in 85% valid alerts and 43% influencing changes in medical orders. These studies underscore the transformative potential of implementing machine learning in clinical leading to improved patient outcomes and more efficient healthcare delivery.

Methods

Python and R were used for data preprocessing and performance comparison. Exploratory data analysis (EDA) employed various methods, including plots such as a pairplot to visualize relationships between features, density plots to illustrate the distribution of sodium-to-potassium ratios across different drug categories, and pie charts to show the

distribution of categorical variables such as gender, blood pressure levels, and drug types. Additionally, heatmaps were generated to analyze cross-tabulations between features like sex, blood pressure, and cholesterol with the drug class. These visualizations offered a comprehensive view of the dataset, emphasizing important patterns and relationships. The visualizations were aesthetically appealing especially Python in comparison to R.

Two models were employed: logistic regression and random forest. For both models, the data was split into training and testing sets with an 80-20 ratio. The logistic regression model was evaluated using metrics such as accuracy, confusion matrix, and classification reports. A Monte Carlo experiment was conducted with 1,000 simulations, recording the training time and accuracy across iterations. Similarly, the random forest model used hyperparameter tuning, and the Monte Carlo experiment was conducted with 100 simulations to assess model stability. The difference in the number of simulations was intended to evaluate the stability of each model more thoroughly.

Lastly, visualizations were applied to illustrate key aspects of the data and model performance, including the confusion matrix. These visualizations not only highlighted the accuracy of the models but also compared the performance of Python and R, providing insights into the relative effectiveness of each scripting language in handling the models.

Results

The comparison between Python and R in terms of performance metrics reveals distinct advantages and considerations for each language, particularly in the context of data analysis and machine learning. In the recent implementation, Python demonstrated superior efficiency in loading data, completing the task in just 0.0042 seconds compared to R's 0.0060 seconds.

However, when it comes to training models, R outperformed Python in Logistic Regression, achieving an average training time of 0.0048 seconds versus Python's 0.0338 seconds. For Random Forest, both languages showed comparable performance, with R slightly faster at 0.0100 seconds compared to Python's 0.0108 seconds. The overall benchmark implementation was significantly quicker in R, taking only 10.03 seconds compared to Python's 41.07 seconds.

In terms of accuracy, Python's Logistic Regression achieved an accuracy of 87.50%, while R's accuracy was notably lower at 15.98%. This discrepancy highlights that while R may excel in speed, it may not always deliver the best predictive performance for certain models. The reason for this case may be due to the amount of simulations that were processed for both languages. When increasing the simulations the model was favored towards Python but conversely, for Random Forest, R achieved an accuracy of 98.92%, slightly higher than Python's 98.65%, indicating that R can effectively capture complex patterns in data with fewer simulations.

The confusion matrix results further illustrate the performance of both models. For Logistic Regression in R, the confusion matrix showed that it correctly classified 3 instances of drugA and 3 instances of drugB, but struggled with other categories, particularly drugX and DrugY, where it misclassified several instances. In contrast, the confusion matrix for Python's Logistic Regression indicated strong performance, with 15 correct classifications for drugA and 6 for drugB, demonstrating its effectiveness in distinguishing between categories.

For Random Forest, R's confusion matrix indicated perfect classification for drugA and drugB, with 4 correct classifications for drugA and 3 for drugB. Python's Random Forest also performed well, achieving similar results with 15 correct classifications for drugA and 7 for

drugB. Both implementations showed high accuracy in identifying DrugY, with R classifying all 18 instances correctly.

When choosing between Python and R, several factors come into play. If the primary goal is rapid data processing and model training, R may be the better choice due to its faster execution times. However, if model accuracy, particularly for Logistic Regression, is a priority, Python might be more favorable. Additionally, the choice may depend on the specific requirements of the project, such as the need for extensive data manipulation capabilities, which Python excels at, or the availability of statistical packages, where R has a strong foothold.

Conclusion

When choosing between Python and R for machine learning, consider project needs: R excels in data processing and model training speed, while Python is preferred for accuracy, especially in Logistic Regression. Python is better for extensive data manipulation, whereas R offers strong statistical packages for specialized analyses.

This study compares the efficiency and effectiveness of Logistic Regression and Random Forest models in both Python and R, with a focus on their application in drug prescription analysis. The benchmark results highlight significant differences in model training times, data processing speeds, and overall performance, offering a nuanced understanding of each tool's strengths. Moreover, organizations should also weigh factors such as model interpretability, ease of use, and the cost of implementation and maintenance. Both Python (Raheem, 2024) and R (Biecek, 2021) offer powerful Explainable AI tools like LIME and SHAP, which provide insights into model predictions and improve transparency. However, Python's larger community and

documentation base may make it more cost-effective for development and integration, whereas R may require more specialized resources.

In conclusion, this study not only examines the technical performance of Logistic Regression and Random Forest models in Python and R but also considers broader implications for real-world applications. Decision-makers in healthcare must balance these findings with their specific constraints and objectives to make informed choices that optimize both performance and cost.

Appendix

Tables

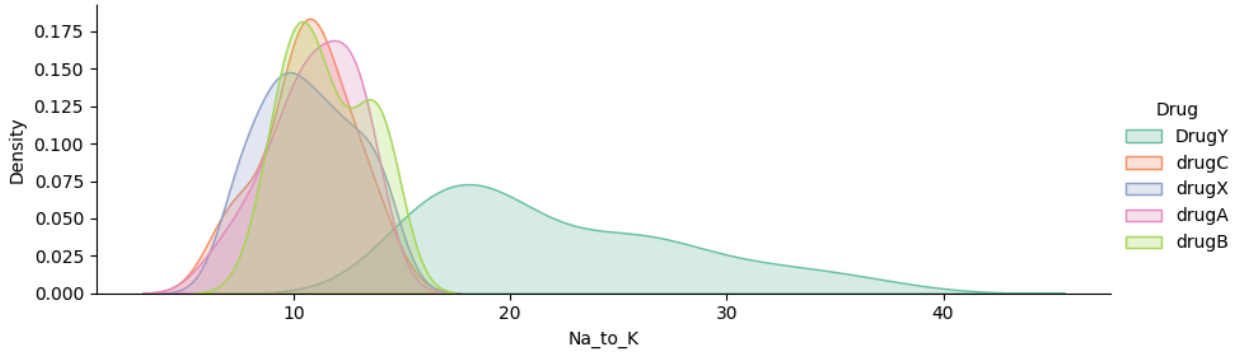
Comparison of Performance Metrics Between Python and R

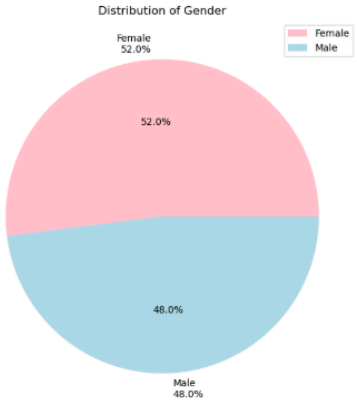
Metric	Python	R
Time taken to load data into the system	0.0042 secs	0.0060 secs
Average training time for Logistic Regression	0.0338 secs	0.0048 secs
Average training time for Random Forest	0.0108 sec	0.0100 secs
Time taken to implement the benchmark	41.07 sec	20.03 secs

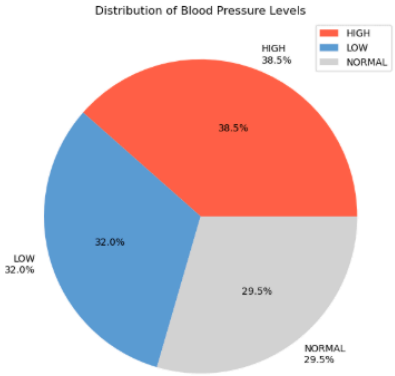
Comparison Model Accuracy Between Python and R

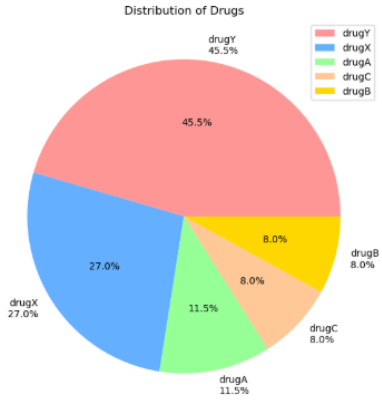
Metric	Python	R
Average accuracy for Logistic Regression	0.8750	0.1598
Average accuracy for Random Forest	0.9865	0.9892

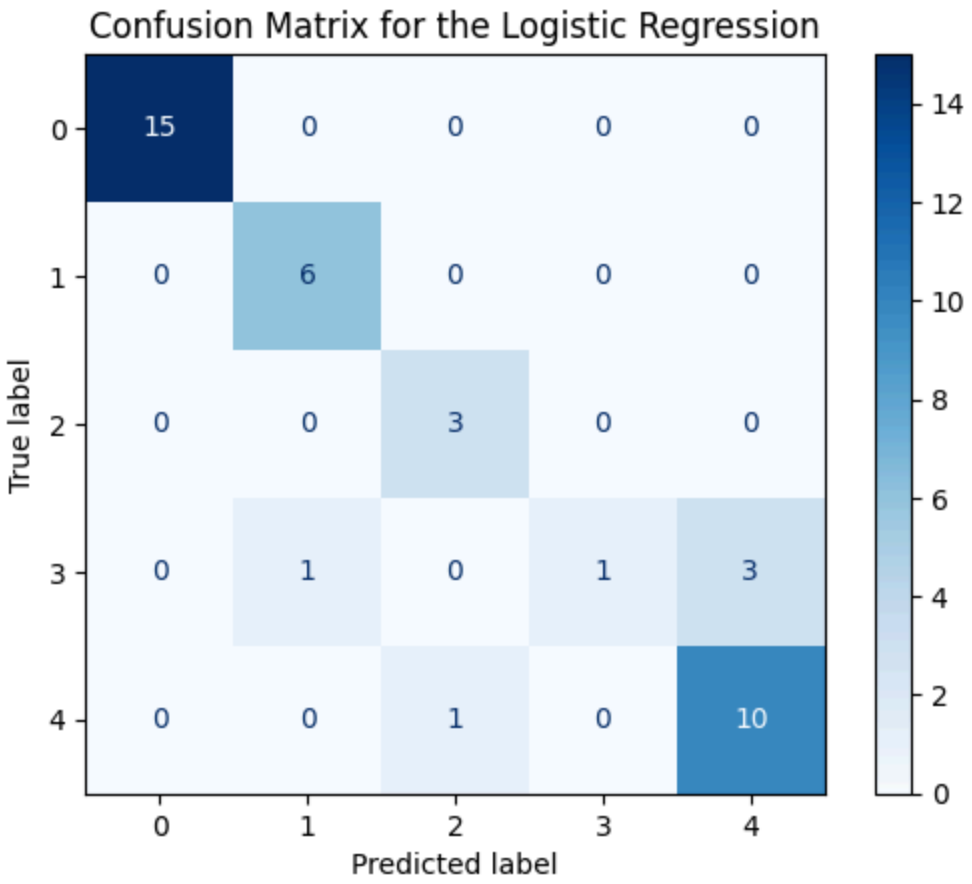
Visualizations in Python

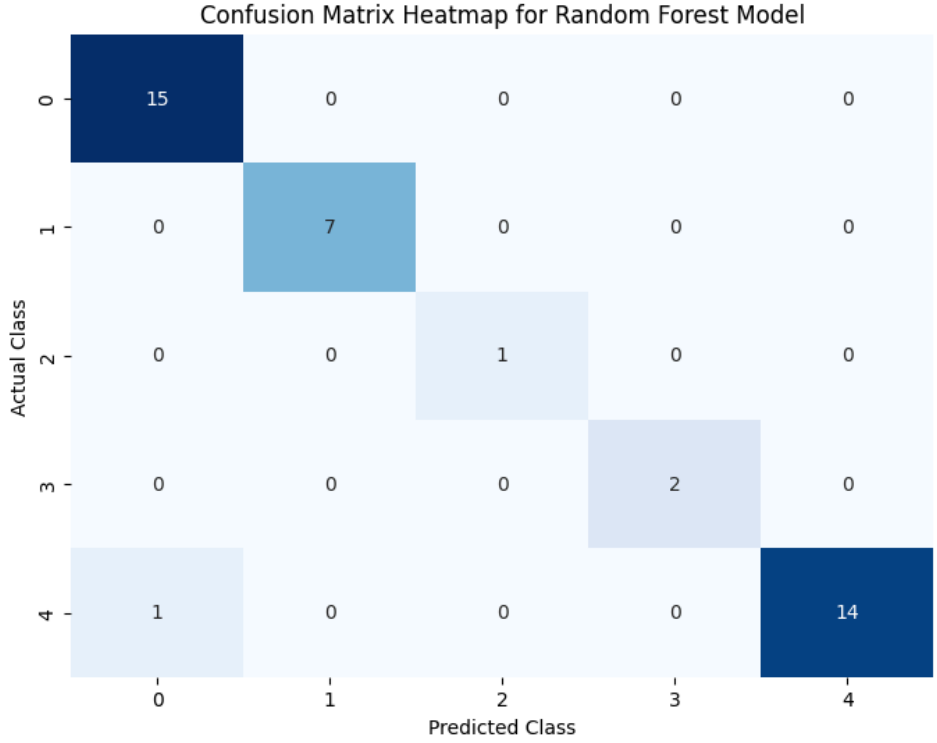




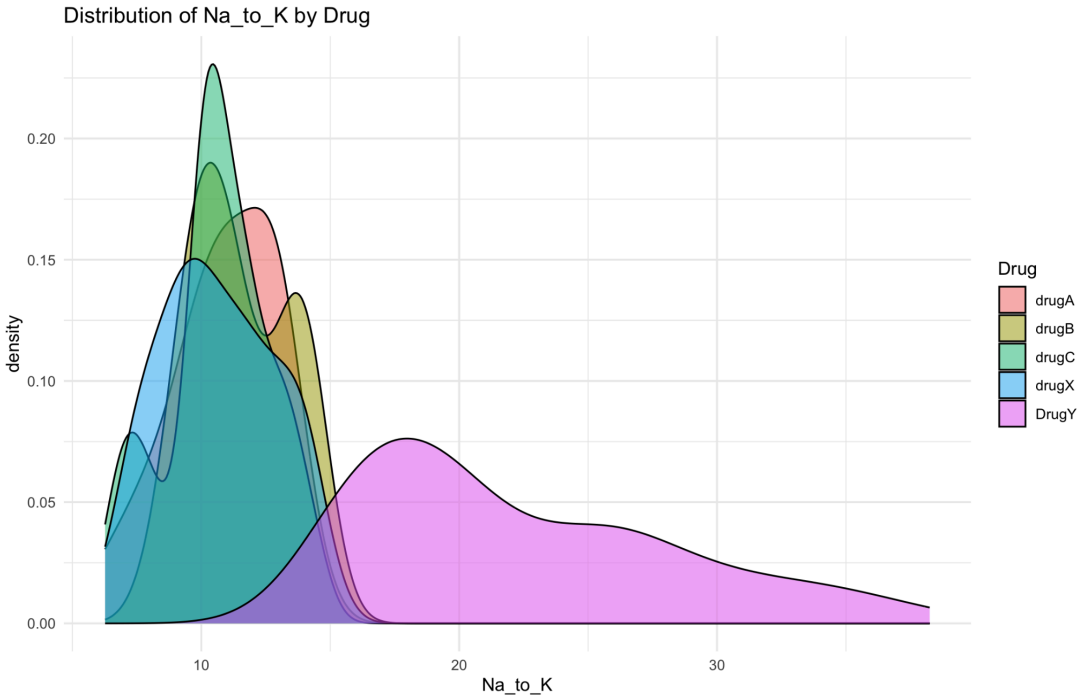




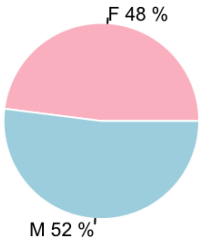

Confusion Matrix for the Logistic Regression
True label
Predicted label
0 1 2 3 4
15 0 0 0 0
0 6 0 0 0
0 0 3 0 0
0 1 0 1 3
0 0 1 0 10
14 12 10 8 6 4 2 0


Confusion Matrix Heatmap for Random Forest Model
Actual Class
Predicted Class
0 1 2 3 4
15 0 0 0 0
0 7 0 0 0
0 0 1 0 0
0 0 0 2 0
1 0 0 0 14

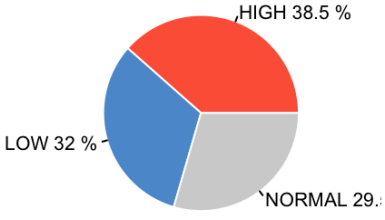
Visualizations in R



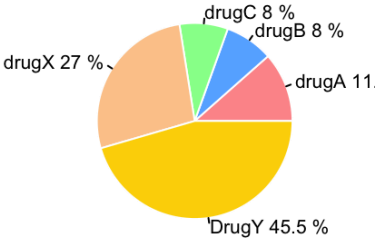
Distribution of Gender

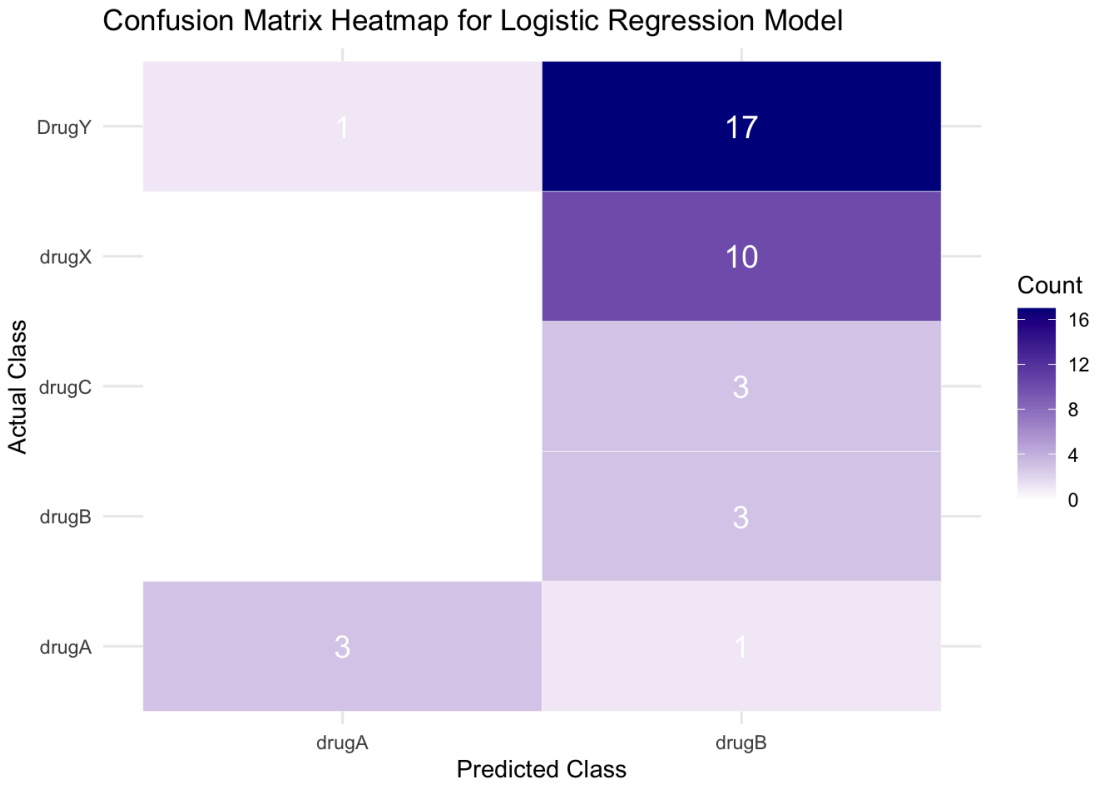


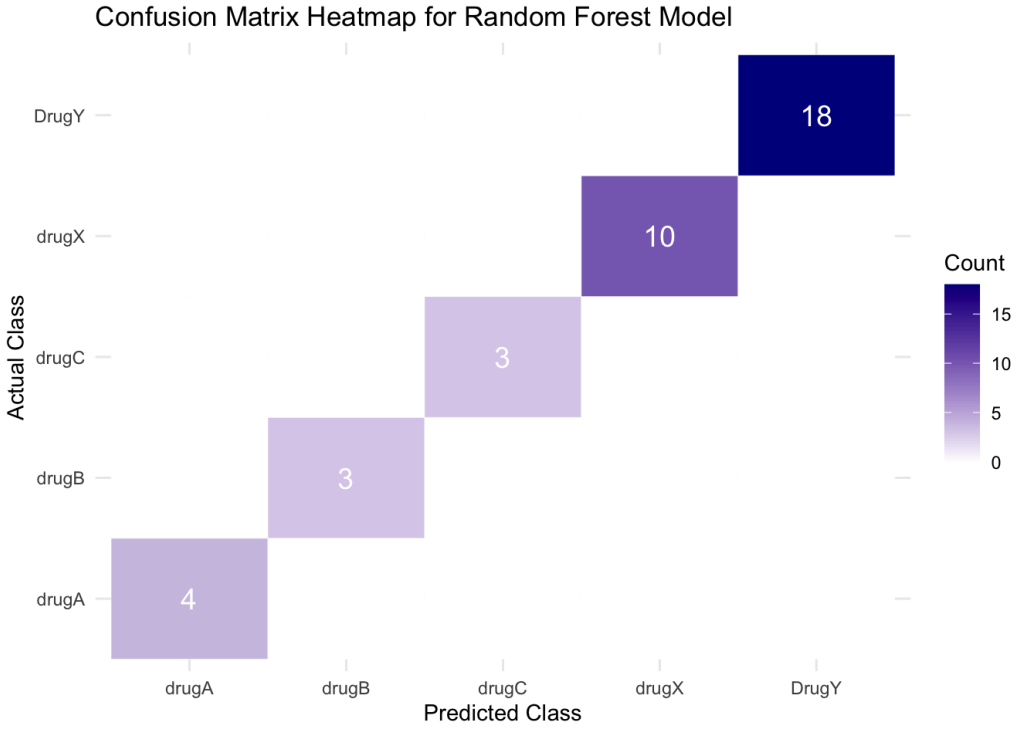
Distribution of Blood Pressure Leve



Distribution of Drugs




Confusion Matrix Heatmap for Logistic Regression Model
DrugY
drugX
drugC
drugB
drugA
Actual Class
1
17
10
3
3
3
1
drugA
drugB
Predicted Class
Count
16
12
8
4
0


Confusion Matrix Heatmap for Random Forest Model
DrugY
drugX
drugC
drugB
drugA
Actual Class
18
10
3
3
4
drugA
drugB
drugC
drugX
DrugY
Predicted Class
Count
15
10
5
0

References

Rajkomar, A., J. Dean, and I. Kohane. "Machine Learning in Medicine." *New England Journal of Medicine* 380, no. 14 (2019): 1347-1358. https://doi.org/10.1056/NEJMra1814259.

Albanese, E., and S. Hug. "Applications of Machine Learning in Healthcare: Challenges and Opportunities." *Journal of Healthcare Informatics Research* 3, no. 4 (2018): 220-232. https://doi.org/10.1007/s41666-018-0022-5.

Raheem. (2024, October 22). *Unveiling the Black Box model using Explainable AI(Lime, Shap) Industry use case.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/10/unveiling-the-black-box-model-using-explainable-ai-lime-shap-industry-use-case/

Biecek, P. (2021, December 16). R packages for eXplainable Artificial Intelligence - ResponsibleML - Medium. *Medium*. https://medium.com/responsibleml/r-packages-for-explainable-artificial-intelligence-7b3536423d2b

Sendak, M., Gao, M., Nichols, M., & Zhou, Y. (2023). Evaluating a machine learning system for identifying medication errors: Potential for enhanced clinical decision support. *ScienceDirect*. https://doi.org/10.1016/j.jbi.2023.103805

Zhan, A., Wang, P., Ouyang, H., & Fang, X. (2019). Clinical Decision Support System: A Probabilistic Machine Learning Approach to Reducing Medication Errors and Increasing Clinical Utility. *Journal of the American Medical Informatics Association*, 26(12), 1560-1565. https://doi.org/10.1093/jamia/ocz156