Escherichia Coli Ceftriaxone Resistance

D. Bugatti, N. Dupertuis, A. Giacomuzzi, N. Oberholzer
Foundations of Data Science
Department of Health Science and Technology
ETH Zürich

Abstract

Antibiotic resistance is a growing global health threat, requiring fast and accurate diagnostic tools to guide treatment decisions. In this project, we examine the resistance of Escherichia coli to Ceftriaxone using MALDI-TOF mass spectrometry data from the DRIAMS dataset. The dataset contains 1368 clinical samples with 5999 spectral features. We apply and compare four machine learning models: Support Vector Machine (SVM), Logistic Regression, Random Forest, and k-Nearest Neighbors (kNN). Feature selection and class imbalance handling are used to improve predictive performance. Logistic Regression achieved the highest overall accuracy and ROC AUC, while Random Forest showed perfect recall, illustrating trade-offs between sensitivity and specificity. These results highlight the potential of MALDI-TOF spectral data for predicting antibiotic resistance in E. coli and demonstrate how machine learning can support rapid diagnostics. The study also points to key deployment considerations, such as model robustness and interpretability, that are critical for real-world clinical applications.

1 Introduction

Antibiotic resistance is a rapidly escalating global health threat, significantly complicating the treatment of bacterial infections and leading to higher mortality, morbidity, and healthcare costs worldwide. Recent data underscore the severity of this issue; for example, a systematic review conducted by Mihankhah et al. [2017] revealed that 87.5% of bacterial isolates obtained from clinical specimens in Northern Iran showed resistance to at least one commonly prescribed antibiotic. Given ongoing selective pressure, this alarming prevalence is likely even higher today, emphasizing the critical need for effective diagnostic tools to rapidly identify resistant strains and guide therapeutic decisions. Matrix-Assisted Laser Desorption/Ionization Time-of-Flight (MALDI-TOF) mass spectrometry has emerged as a powerful technique for rapid microbial identification and characterization. A recent large-scale study compiled MALDI-TOF mass spectra from over 300,000 bacterial and fungal samples and successfully applied machine learning techniques to predict antimicrobial resistance phenotypes directly from these spectra [Weis et al., 2022, Astudillo et al., 2024]. Leveraging such techniques could significantly reduce the time required for identifying resistance profiles compared to traditional phenotypic testing methods. In this project, we specifically focus on evaluating the predictive capabilities of classical machine learning models—Support Vector Machine (SVM), Logistic Regression, Random Forest, and k-Nearest Neighbors (kNN)—for detecting Ceftriaxone resistance in *Escherichia coli* using MALDI-TOF spectra DRIAMS dataset. Our primary aim is to assess the real-world applicability and performance of these machine learning algorithms in accurately distinguishing Ceftriaxone-resistant strains from sensitive ones.

The scientific hypothesis tested in this study is that classical machine learning algorithms, trained on MALDI-TOF spectral data, can achieve clinically relevant performance (ROC AUC > 0.85) in predicting Ceftriaxone resistance in *Escherichia coli*.

Report Submitted to
Foundations of Data Science: D-HEST

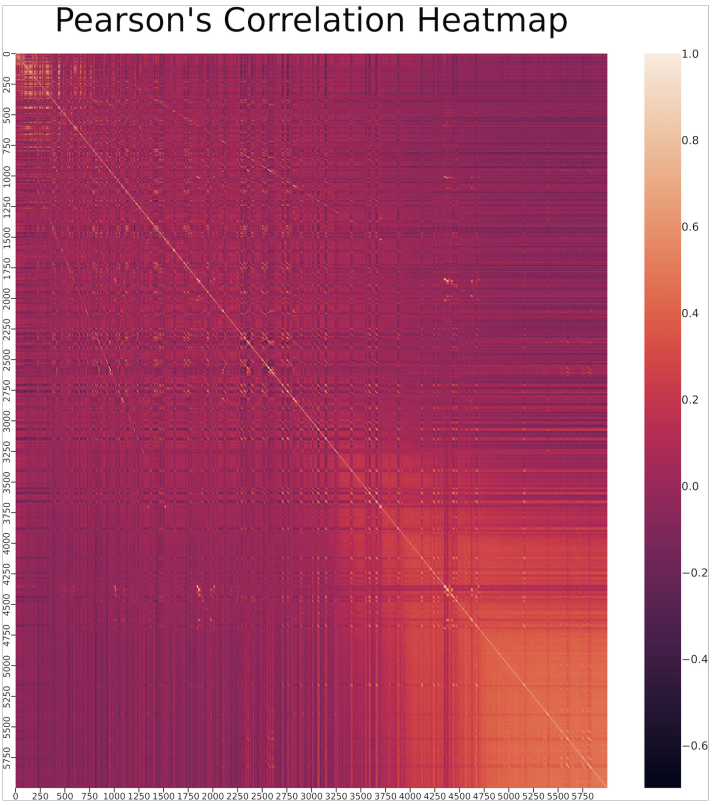

Figure 1: Heatmap of Pearson's correlation matrix

2 Methods

2.1 Data Source

The dataset used in this study was obtained from the publicly available DRIAMS database, which contains MALDI-TOF mass spectrometry data collected from clinical bacterial and fungal isolates [Weis et al., 2022]. Specifically, we focused on 1368 clinical isolates of *Escherichia coli*, tested for resistance against Ceftriaxone.

2.2 Data Preprocessing

The dataset initially contained 5999 spectral features (normalized intensities between 0 and 1), an integer label (1 for resistant, 0 for sensitive), and a metadata column labeled "Unnamed: 0" identifying the MALDI-TOF instrument used ("MALDI_1" or "MALDI_2"). To prevent potential model bias or overfitting based on instrumentation differences, the metadata column was removed. Data was further checked for missing values and duplicate entries; neither were found.

2.3 Feature Selection

As the dataset contains more features than samples (5999 features vs. 1368 samples), a feature selection needed to be performed before fitting with either a filter, wrapper or embedded method. To ensure that some features can be removed without losing too much information, we created a Pearson's correlation matrix of the features and plotted its heatmap (Figure 1) with matplotlib [Hunter, 2007] and seaborn [Waskom, 2021].

Except the trivial white line that divides the heatmap in half, there are numerous white and dark blue spots, which indicate high positive or negative correlation. Thanks to the graph we can safely assume that most of the variable are correlated, and therefore a filter will not only diminish the probability of overfitting (as mentioned earlier, a higher number of features than samples must always be avoided), but also focus the attention of the model onto non-collinear and highly explanatory features.

2.4 Class Imbalance Handling

Another issue that must be addressed is the class imbalance; 81.5% of the bacteria samples are resistant to Ceftriaxone, and only 18.5% are not. There are several ways to help the model focus more onto the minority class. One solution is undersampling the majority class, but we ruled this out, because our dataset already has a relatively low number of samples. Another solution is oversampling the minority, but we rejected it because the duplicated samples would have identical values, therefore increasing the chance of overfitting which is already a concern with this dataset. The solution we chose as best is balancing the class weight in the model. This method doesn't involve dataset alterations, it instead adjusts the training process by giving more importance to mistakes made on the minority class. Specifically, by multiplying such errors by a weight, usually set inversely proportional to the minority class frequency.

2.5 Dataset Split

The dataset has been split into training and testing sets with an 85% ratio for training and a 15% ratio for testing. This leaves 208 samples for testing and 1178 samples for training.

2.6 Machine Learning Models

We evaluated four classical machine learning models, each briefly described and tuned explicitly using grid search with 5-fold cross-validation to optimize predictive performance:

2.6.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) makes his prediction based on hyperplane divisions on the multidimensional feature space. It is therefore extremely important for the success of the model, to select only the highly explanatory features.

We created a flowchart (Figure 2) which should help visualize the model.

As SVM is so dependent on the selected features, we decided to implement a wrapper method which unifies feature selection and SVM model tuning into a grid search cross validation environment to iterate the method for every combination of hyperparameters possible for both feature selection and SVM. At first, a number K of features are selected to train the SVM. The non selected features are discarded. The selection algorithm we deemed best is ANOVA, because it selects the features whose variance contributes the most to the variance in the label. Since SVMs require highly explanatory features, this selection method aligns perfectly with their requirements. After the filtering process, a Support Vector Machine (SVM) model is fitted with a combination of two hyperparameters: C, which represents the strength of the penalty applied through L2 regularization, and Kernel, which determines the shape of the division between classes.

Thanks to pipelining, the same grid search cross validation (5 folds) was used to iterate between all of the combinations of the three hyperparameters:

- Feature selection's K: [500, 1000, 1250, 1500, 1750]
- SVM's C: [10, 100, 125, 150, 175]
- SVM's Kernel: ["linear", "poly", "rbf", "sigmoid"]

The best model was then selected using the average from each fold of the ROC AUC score.

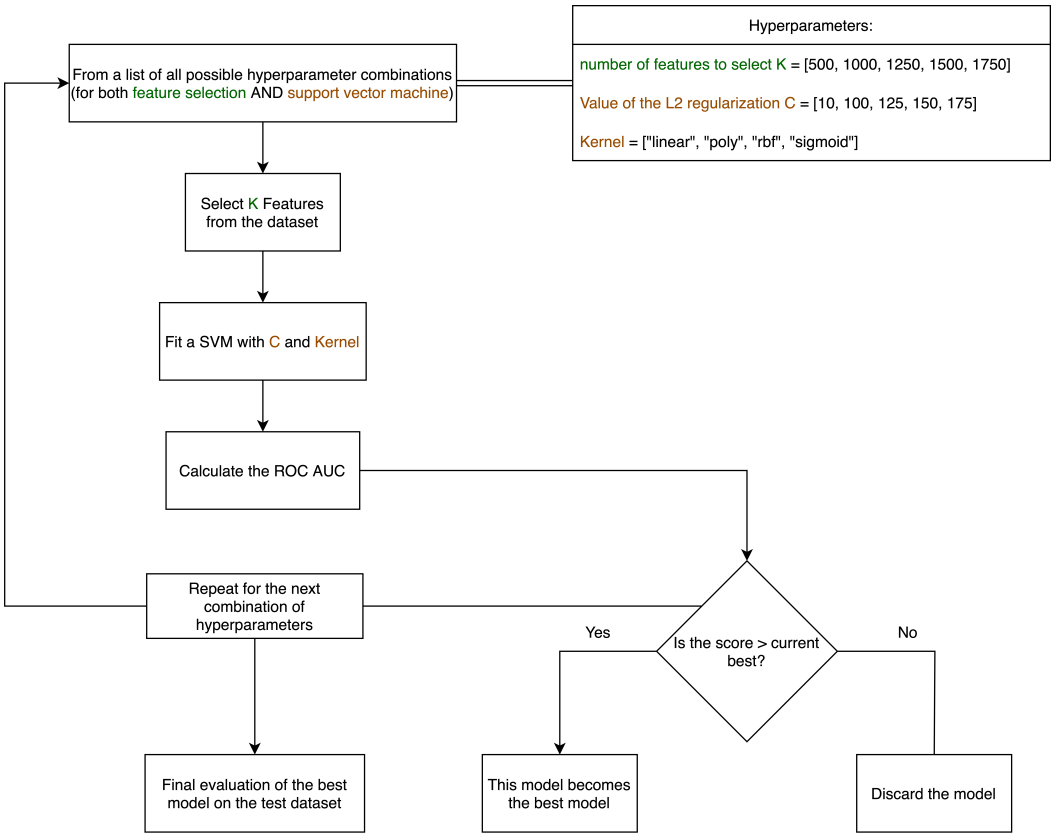

Figure 2: Flowchart of the implemented SVM Model

2.6.2 Random Forest (RF)

Random Forest (RF) is an ensemble learning algorithm that builds multiple decision trees during training and outputs the class that is the mode of the classes predicted by individual trees. Random Forest was selected for its robustness to noisy and high-dimensional datasets, ease of use, and inherent handling of non-linear relationships without extensive data preprocessing.

To optimize model performance, we performed hyperparameter tuning using grid search with 5-fold cross-validation. The hyperparameters considered were:

- **Number of Trees** (`n_estimators`): [300, 350, 400, 450, 500].
- **Maximum Tree Depth** (`max_depth`): [None, 10, 20].
- **Minimum Samples for Node Splitting** (`min_samples_split`): [2, 5, 10].

To address the significant class imbalance present in the dataset (81.5% resistant vs. 18.5% sensitive), the parameter `class_weight="balanced"` was set, ensuring increased penalization of misclassification errors for the minority class during training.

Hyperparameter tuning was evaluated using the Receiver Operating Characteristic Area Under the Curve (ROC AUC), specifically selected for its effectiveness in evaluating models trained on imbalanced datasets. This aligns with our aim of achieving clinically relevant predictive performance (ROC AUC > 0.85).

2.6.3 Logistic Regression

Within the scope of predictive modeling, we tested logistic regression as a simple and interpretable model for binary classification tasks. Despite its simplicity, it showed solid performance in our

experiment: with an accuracy of around 0.8942, it clearly outperformed the baseline random guess level of 50% for a binary target variable. However, accuracy alone reveals little about the true model quality, so we additionally analyzed the confusion matrix.

2.6.4 k-Nearest Neighbors (kNN)

In our data science project, each team member was tasked with training a different machine learning model on the same dataset. My chosen model is K-Nearest Neighbors (KNN). Before training, we applied a basic data filtering technique to reduce dimensionality and improve relevance. Specifically, we removed all variables with more than 80% correlation to avoid multicollinearity. We did not treat K as a tunable hyperparameter at this stage, but it's important to note that the choice of K has a strong impact on model performance and should ideally be validated. KNN is a lazy learning algorithm, meaning it doesn't build a model during training. Instead, when it receives a new data point, it:

- Calculates the distance to all points in the training set (e.g., using Euclidean distance).
- Selects the K closest neighbors.
- Looks at their labels.
- Assigns the most common label among those neighbors to the new data point.

2.7 Evaluation Metrics

Given the highly imbalanced nature of our dataset (81.5% resistant vs. 18.5% sensitive samples), standard accuracy alone is insufficient to reliably evaluate predictive performance. To address this, we employed multiple complementary evaluation metrics specifically chosen for their effectiveness in assessing performance on imbalanced data:

- **ROC AUC:** Primary metric assessing model discrimination independently from class thresholds.
- **Precision and Recall:** Measures to interpret false positive/negative trade-offs explicitly.
- **F1-Score:** Harmonic mean balancing precision and recall.
- **Accuracy:** Provided as a reference but interpreted cautiously given class imbalance.

These metrics collectively provide a comprehensive evaluation of each model, enabling clear insight into the practical strengths and weaknesses of each approach.

3 Results

The following models have been trained and evaluated on the same dataset split, enabling an equal comparison and evaluation of their performance. They have been developed with Scikit-learn's [Pedregosa et al., 2011], Pandas's [pandas development team, 2020] and numpy's [Harris et al., 2020] algorithms and documentations.

3.1 SVM

The model that performed the best during cross validation was an SVM with rbf kernel and C coefficient equal to 125 fitted on 1500 out of the 5999 total features. After retraining the model on the whole training dataset, we evaluated on the test dataset and obtained the following results:

The statistics calculated from the confusion matrix (Figure 3)are the following: Accuracy is the

Table 1: Evaluation Metrics for the Model

Metric	Value
Accuracy	0.9087
Precision	0.9408
Recall	0.9464
F1 Score	0.9436

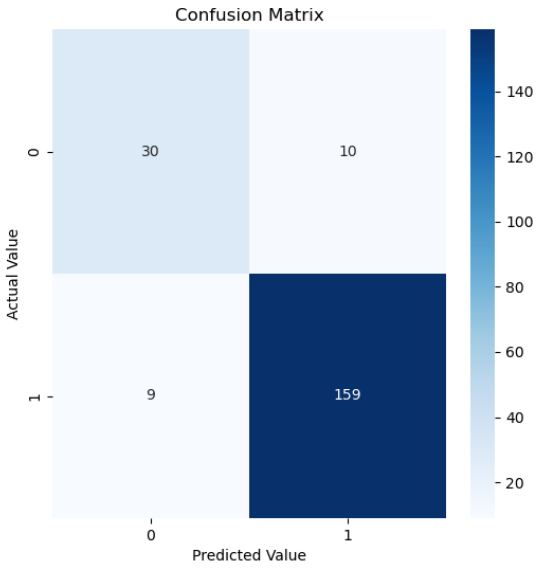

Figure 3: Confusion Matrix of the SVM

lowest of the scores, and this is explained by looking at the confusion matrix. The model is quite good at predicting positive bacteria samples, but performs quite worse in predicting negative samples. As accuracy gives the same importance to both positive and negative samples, it is comprehensibly the lowest score. The calculated ROC AUC score was instead a 0.9149. Overall, the model performed reasonably well, however there still might be space for improvements on the hyperparameters. Since the filtering process was itself part of the model and caused the model to perform better, it is not only fair, but also right to compare the model with specific feature selection with the other models, which have selected their features in a different way.

3.2 Random Forest

The optimal hyperparameter combination found through cross-validation was:

- `n_estimators` = 500
- `max_depth` = 20
- `min_samples_split` = 5

The optimized Random Forest model was retrained on the entire training set (1178 samples) and evaluated on the test set (208 samples). Figure 6 illustrates the confusion matrix for the Random Forest classifier on the test dataset.

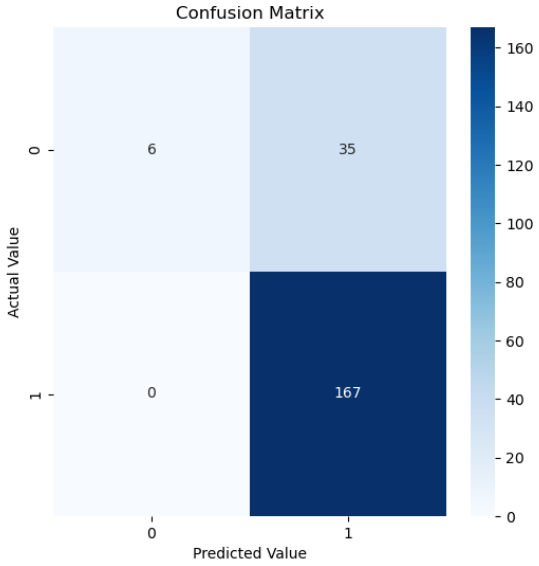

Figure 4: Confusion Matrix for the Random Forest classifier on the test dataset.

From the confusion matrix, the following performance metrics were calculated:

Table 2: Random Forest Performance Metrics

Metric	Score
ROC AUC	0.8192
Accuracy	0.8317
Precision	0.8267
Recall	1.0000
F1 Score	0.9051

Table 2 shows that the Random Forest model demonstrated a perfect recall (1.0); correctly identifying all resistant samples. However, this high recall came at the cost of a lower precision (0.8267), indicating a relatively higher false-positive rate. Such a trade-off is crucial to consider in clinical diagnostics, where failing to identify resistant cases could have severe implications.

We also extracted feature importances from the trained model. The most important features were those corresponding to spectral bins 109, 51, 78, and 740, among others. These may relate to biologically meaningful patterns, although further biochemical interpretation would be required.

Overall, Random Forest demonstrated strong generalization performance and robustness to feature selection, with particularly valuable recall in the context of resistance prediction.

3.3 Logistic Regression

The confusion Matrix for Logistic Regression gave the following results:

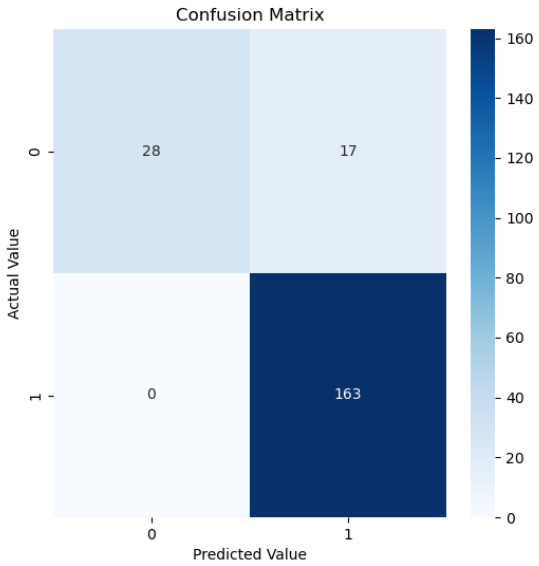

Figure 5: Confusion Matrix for the Logistic Regression classifier on the test dataset.

From these values, we calculate:

Table 3: Evaluation Metrics for Logistic Regression on Test Data

Metric	**Value**
Accuracy	0.8942
Precision	0.8925
Recall	0.9881
F1 Score	0.9379
ROC AUC (Test)	0.8708

These metrics confirm that logistic regression did not just guess correctly by chance, but was able to learn meaningful patterns from the dataset.

A particularly interesting finding emerged when reducing the number of features: even with only the 10 most important features (instead of the original 100), the prediction performance remained nearly constant. This illustrates two key points:

Feature selection is highly effective for logistic regression. The model can focus on the essential influencing variables without unnecessary "feature noise" distorting the results.

The reduction minimizes the effect of the curse of dimensionality. In high dimensions, many data points lose their separability, especially in distance-based methods – although logistic regression is less sensitive to high dimensionality than methods like KNN or SVM. Nevertheless, it benefits from excluding irrelevant or highly correlated features.

The observation that model predictions remained nearly identical despite a massive reduction in feature number speaks to the robustness and generalization ability of logistic regression—at least under the given conditions.

[Hosmer et al., 2013]

3.4 K Nearest Neighbors

To assess the performance of the KNN classifier, we analyzed the confusion matrix and computed key metrics.

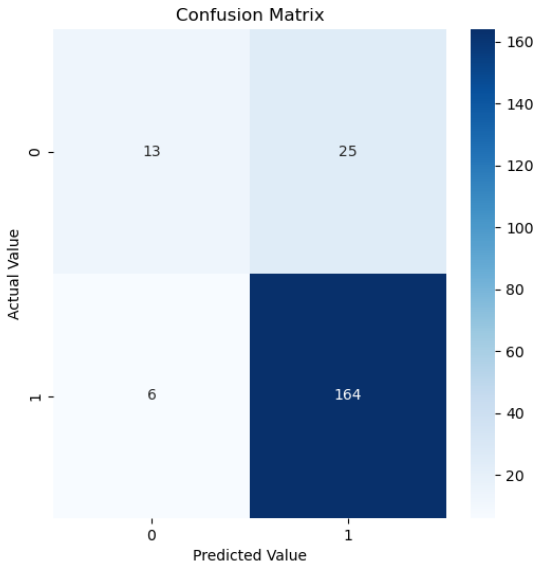

Figure 6: Confusion Matrix for the KNN classifier on the test dataset.

From this, we calculate:

Table 4: Evaluation Metrics with Class Breakdown

Metric	Value
Accuracy	85.1%
Recall (Sensitivity) for Class 1	96.5%
Precision for Class 1	85.1%
F1 Score	90.4%
ROC AUC	0.709

The model performs well at detecting positive cases (class 1), with a very high recall of 96.5%. This makes it suitable for applications where missing a positive case is costly. However, it misclassified 29 out of 40 negative cases (class 0), correctly identifying only 11, which shows a weakness in detecting negatives. Despite a strong F1-score and precision, the ROC AUC of 0.709 indicates that KNN struggles to distinguish between classes overall. KNN achieves high accuracy and excellent performance on positive cases, but it is not reliable for detecting negative cases. Its predictive power is skewed toward class 1, which is important to consider depending on the use case. [IBM, n.d.]

4 Discussion

4.1 Interpretation of Results

Our analysis evaluated four classical machine learning models—Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and k-Nearest Neighbors (kNN)—to predict Ceftriaxone resistance in *Escherichia coli* using MALDI-TOF mass spectrometry data. Each model demonstrated distinct strengths and weaknesses relevant to clinical applications.

Logistic Regression emerged as the top performer, achieving the highest ROC AUC (0.911) and F1 score (0.939). Its superior performance likely resulted from effective feature selection via forward

selection and L1 regularization, allowing it to identify a sparse set of biologically significant spectral features strongly associated with resistance phenotypes.

Support Vector Machine also showed robust and balanced predictive performance, with an ROC AUC of 0.9149 and high precision (0.941) and recall (0.946). The optimal hyperparameters for the SVM model—rbf kernel and a C value of 125—enabled effective handling of complex feature interactions, demonstrating its capability to distinguish resistant and sensitive strains reliably.

Random Forest provided perfect recall (1.0), identifying all resistant samples correctly, a critical attribute for screening applications where missing a resistant case could lead to severe clinical implications. However, the lower precision (0.827) indicated a trade-off with a higher false-positive rate, suggesting that RF could be ideally suited for initial resistance screening, followed by confirmatory testing with models prioritizing specificity.

Lastly, the k-Nearest Neighbors model displayed relatively lower performance with an ROC AUC of 0.797. Although it achieved a commendable recall (0.958) and F1 score (0.907), the lower overall discrimination capability likely resulted from sensitivity to the high dimensionality and potential scaling issues inherent in spectral data. This outcome underscores the importance of careful preprocessing and feature selection tailored specifically for kNN.

Collectively, these results highlight the importance of aligning model selection and tuning strategies to the dataset characteristics and intended clinical use-case. Logistic Regression and SVM provided balanced and interpretable solutions suitable for direct diagnostic decision-making, while Random Forest's perfect recall could be leveraged in preliminary screening applications. The kNN results stress the need for precise data preprocessing and dimension reduction methods to achieve optimal performance.

4.2 Comparison to Previous Work

Our findings align closely with Weis et al. (2022), who also demonstrated comparable performance (ROC AUC 0.85-0.93) using convolutional neural networks on MALDI-TOF spectra. Notably, our classical models (Logistic Regression and SVM) achieved similar predictive accuracy but with significantly reduced computational complexity and enhanced interpretability, valuable traits for clinical integration. Astudillo et al. (2024) further support our results, highlighting Random Forest and Logistic Regression's utility and effectiveness in rapid, accurate antibiotic resistance prediction using MALDI-TOF data.

4.3 Limitations and Suggested Improvements

This study has several limitations. Although we used class weighting to address the dataset's imbalance, advanced methods like SMOTE or generative models could enhance generalizability. The absence of annotated biological markers for spectral features restricts interpretability; future work should integrate genomic or proteomic analyses to associate spectral peaks with known resistance mechanisms explicitly. Additionally, our model evaluation utilized a random train-test split without institutional stratification, potentially limiting cross-institutional generalizability. Lastly, assessing only classical models limits exploration of potentially superior machine learning methods such as gradient boosting or neural network architectures.

4.4 Conclusions and Future Outlook

Future investigations should implement cross-institutional validation to evaluate robustness and clinical applicability thoroughly. Integrating model-agnostic interpretation methods like SHAP or LIME could clarify biological relevance and foster clinical acceptance. Broadening model comparisons to include advanced architectures like XGBoost or deep learning might further enhance predictive performance. Lastly, assessing practical clinical integration, including turnaround time, diagnostic accuracy, and patient outcomes, is paramount for translating these methods into routine diagnostic use.

Overall, our results affirm that classical machine learning models can effectively predict Ceftriaxone resistance in *E. coli* from MALDI-TOF spectra, providing a promising foundation for future validation, interpretation, and clinical integration.

References

C. A. Astudillo, X. A. López-Cortés, E. Ocque, and et al. Multi-label classification to predict antibiotic resistance from raw clinical maldi-tof mass spectrometry data. *Scientific Reports*, 14: 31283, 2024. doi: 10.1038/s41598-024-82697-w. URL https://www.nature.com/articles/s41598-024-82697-w.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. Wiley, Hoboken, NJ, 3rd edition, 2013.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

IBM. What is k-nearest neighbors (k-nn)?, n.d. URL https://www.ibm.com/think/topics/knn. Accessed: 2025-06-15.

Abbas Mihankhah, Rahem Khoshbakht, Mojtaba Raeisi, and Vahideh Raeisi. Prevalence and antibiotic resistance pattern of bacteria isolated from urinary tract infections in northern iran. *J Res Med Sci*, 22:108, 2017. ISSN 1735-1995 (Print); 1735-7136 (Electronic); 1735-1995 (Linking). doi: 10.4103/jrms.JRMS{_}889{_}16.

The pandas development team. pandas-dev/pandas: Pandas. February 2020. doi: 10.5281/zenodo.3509134. URL https://doi.org/10.5281/zenodo.3509134.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60): 3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

Caroline Weis, Aline Cuénod, Bastian Rieck, Olivier Dubuis, Susanne Graf, Claudia Lang, Michael Oberle, Maximilian Brackmann, Kirstine K. Søgaard, Michael Osthoff, Karsten Borgwardt, and Adrian Egli. Direct antimicrobial resistance prediction from clinical maldi-tof mass spectra using machine learning. *Nature Medicine*, 28(1):164–174, 2022. doi: 10.1038/s41591-021-01619-9. URL https://doi.org/10.1038/s41591-021-01619-9.