
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*.

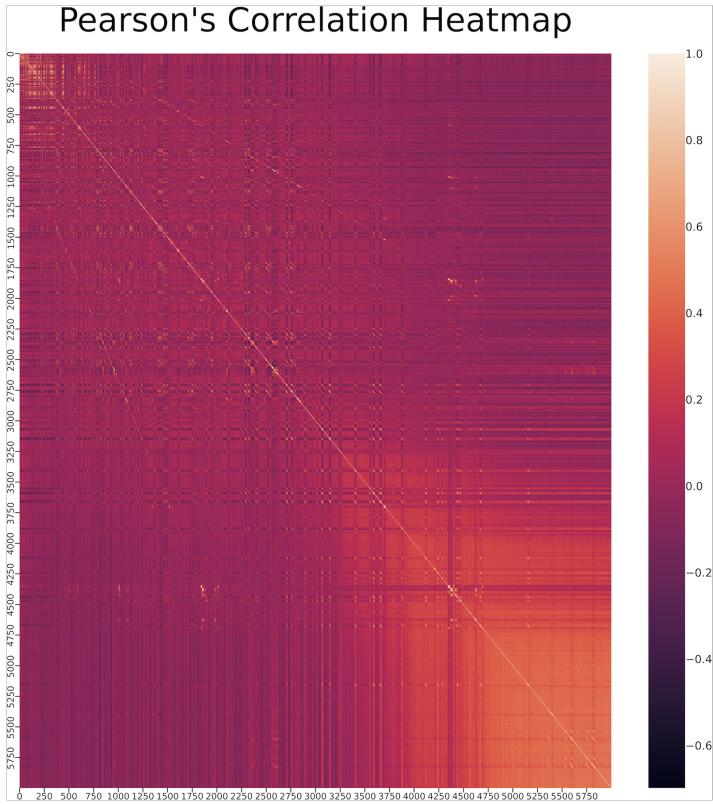


Figure 1: Heatmap of Pearson’s Correlation Matrix

38 2 Methods

39 2.1 Data Source

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

44 2.2 Data Preprocessing

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

50 2.3 Feature Selection

51 As the dataset contains more features than samples (5999 features vs. 1368 samples), a feature
 52 selection needed to be performed before fitting with either a filter, wrapper or embedded method.
 53 To ensure that some features can be removed without losing too much information, we created
 54 a Pearson’s correlation matrix of the features and plotted its heatmap (Figure 1) with matplotlib
 55 [Hunter, 2007] and seaborn [Waskom, 2021].
 56

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

62 **2.4 Class Imbalance Handling**

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

73 **2.5 Dataset Split**

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

76 **2.6 Machine Learning Models**

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

79 **2.6.1 Support Vector Machine (SVM)**

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

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

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

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

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

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

100
101
102

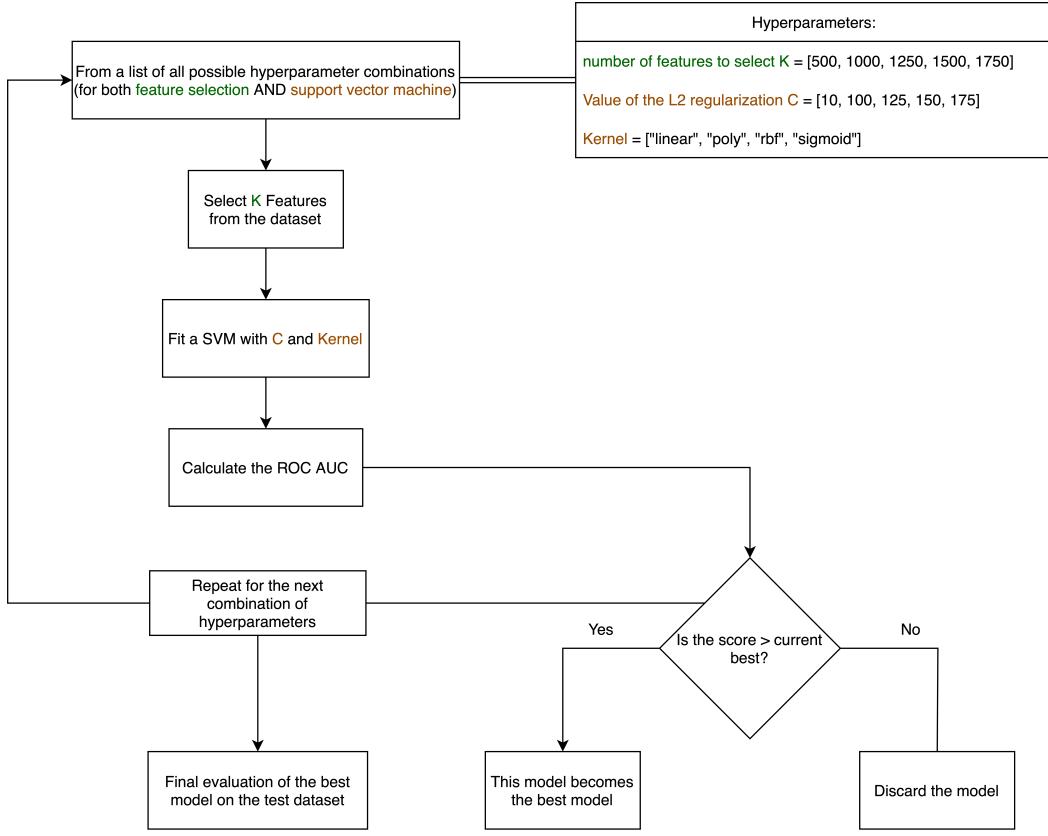


Figure 2: Flowchart of the implemented SVM Model

103 2.6.2 Random Forest (RF)

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

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

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

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

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

120 2.6.3 Logistic Regression

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

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

126 **2.6.4 k-Nearest Neighbors (kNN)**

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

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

138 **2.7 Evaluation Metrics**

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

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

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

150 **3 Results**

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

155 **3.1 SVM**

156 The model that performed the best during cross validation was an SVM with rbf kernel and C
157 coefficient equal to 125 fitted on 1500 out of the 5999 total features. After retraining the model on
158 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 (Table 1): Accuracy is

Table 1: Evaluation Metrics for the SVM Model

Metric	Value
Accuracy	0.9087
Precision	0.9408
Recall	0.9464
F1 Score	0.9436

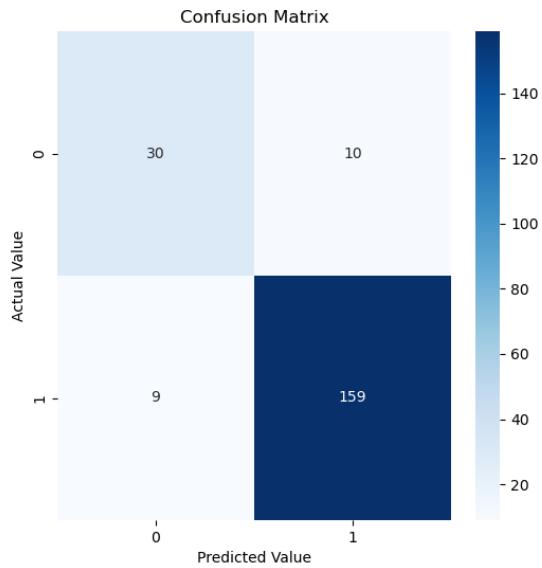


Figure 3: Confusion Matrix for the SVM Model

159 the lowest of the scores, and this is explained by looking at the confusion matrix. The model is quite
 160 good at predicting positive bacteria samples, but performs quite worse in predicting negative samples.
 161 As accuracy gives the same importance to both positive and negative samples, it is comprehensibly
 162 the lowest score. The calculated ROC AUC score was instead a 0.9149. Overall, the model performed
 163 reasonably well, however there still might be space for improvements on the hyperparameters. Since
 164 the filtering process was itself part of the model and caused the model to perform better, it is not only
 165 fair, but also right to compare the model with specific feature selection with the other models, which
 166 have selected their features in a different way.
 167

168 **3.2 Random Forest**

169 The optimal hyperparameter combination found through cross-validation was:

- 170 • `n_estimators = 500`
171 • `max_depth = 20`
172 • `min_samples_split = 5`

173 The optimized Random Forest model was retrained on the entire training set (1178 samples) and
174 evaluated on the test set (208 samples). Figure 4 illustrates the confusion matrix for the Random
175 Forest classifier on the test dataset.

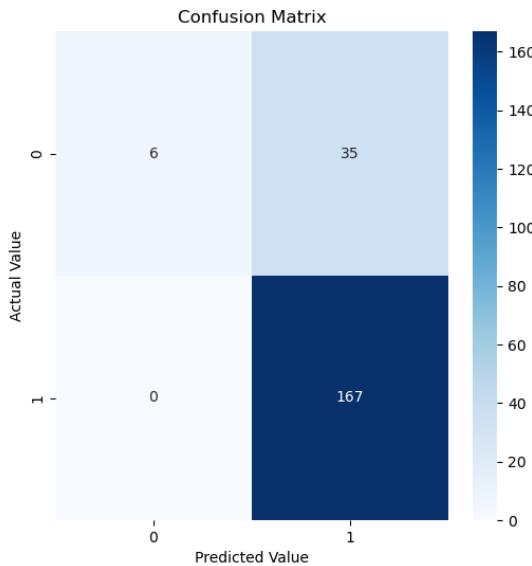


Figure 4: Confusion Matrix for the RF Model

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

Table 2: Evaluation Metrics for the RF Model

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

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

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

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

186 **3.3 Logistic Regression**

187 The confusion Matrix for Logistic Regression gave the following results (Figure 5):

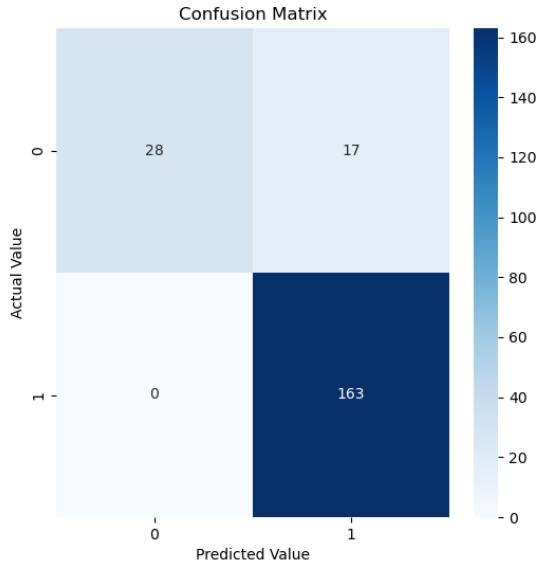


Figure 5: Confusion Matrix for the LR Model

188 From these values, we calculate:

Table 3: Evaluation Metrics for the LR Model

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

189 These metrics from Table 3 confirm that logistic regression did not just guess correctly by chance,
190 but was able to learn meaningful patterns from the dataset.

191 A particularly interesting finding emerged when reducing the number of features: even with only the
192 10 most important features (instead of the original 100), the prediction performance remained nearly
193 constant. This illustrates two key points:

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

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

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

203 [Hosmer et al., 2013]

204 **3.4 K Nearest Neighbors**

205 To assess the performance of the KNN classifier, we analyzed the confusion matrix (Figure 6) and
206 computed key metrics.

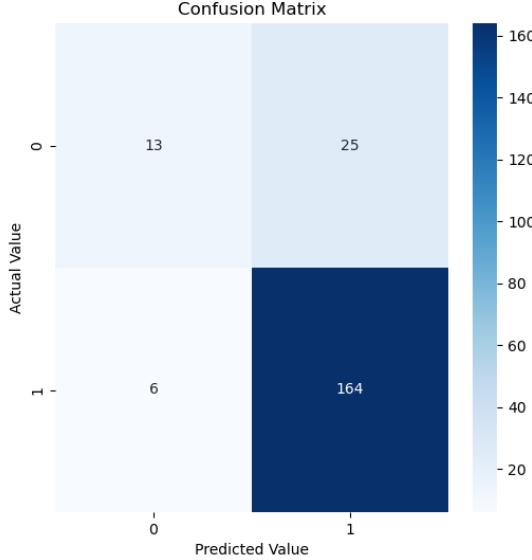


Figure 6: Confusion Matrix for the KNN Model

207 From this, we calculate:

Table 4: Evaluation Metrics for the KNN Model

Metric	Value
Accuracy	0.8510
Recall (Sensitivity) for Class 1	0.8549
Precision for Class 1	0.9821
F1 Score	0.9141
ROC AUC	0.7771

208 Table 4 shows that the model performs well at detecting positive cases (class 1), with a very high
209 recall of 96.5%. This makes it suitable for applications where missing a positive case is costly.
210 However, it misclassified 29 out of 40 negative cases (class 0), correctly identifying only 11, which
211 shows a weakness in detecting negatives. Despite a strong F1-score and precision, the ROC AUC
212 of 0.709 indicates that KNN struggles to distinguish between classes overall. KNN achieves high
213 accuracy and excellent performance on positive cases, but it is not reliable for detecting negative
214 cases. Its predictive power is skewed toward class 1, which is important to consider depending on the
215 use case. [IBM, n.d.]

216 **4 Discussion**

217 **4.1 Interpretation of Results**

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

222 Logistic Regression emerged as the top performer, achieving the highest ROC AUC (0.911) and F1
223 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.

274 **References**

- 275 C. A. Astudillo, X. A. López-Cortés, E. Ocque, and et al. Multi-label classification to predict
276 antibiotic resistance from raw clinical maldi-tof mass spectrometry data. *Scientific Reports*, 14:
277 31283, 2024. doi: 10.1038/s41598-024-82697-w. URL <https://www.nature.com/articles/s41598-024-82697-w>.
- 278 Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David
279 Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti
280 Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández
281 del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy,
282 Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming
283 with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2.
284 URL <https://doi.org/10.1038/s41586-020-2649-2>.
- 285 David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. Wiley,
286 Hoboken, NJ, 3rd edition, 2013.
- 287 J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):
288 90–95, 2007. doi: 10.1109/MCSE.2007.55.
- 289 IBM. What is k-nearest neighbors (k-nn)?, n.d. URL <https://www.ibm.com/think/topics/knn>.
290 Accessed: 2025-06-15.
- 291 Abbas Mihankhah, Rahem Khoshbakht, Mojtaba Raeisi, and Vahideh Raeisi. Prevalence and
292 antibiotic resistance pattern of bacteria isolated from urinary tract infections in northern iran. *J Res
293 Med Sci*, 22:108, 2017. ISSN 1735-1995 (Print); 1735-7136 (Electronic); 1735-1995 (Linking).
294 doi: 10.4103/jrms.JRMS{_}889{_}16.
- 295 The pandas development team. pandas-dev/pandas: Pandas. February 2020. doi: 10.5281/zenodo.
296 3509134. URL <https://doi.org/10.5281/zenodo.3509134>.
- 297 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Pretten-
298 hofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and
299 E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*,
300 12:2825–2830, 2011.
- 301 Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):
302 3021, 2021. doi: 10.21105/joss.03021. URL <https://doi.org/10.21105/joss.03021>.
- 303 Caroline Weis, Aline Cuénod, Bastian Rieck, Olivier Dubuis, Susanne Graf, Claudia Lang, Michael
304 Oberle, Maximilian Brackmann, Kirstine K. Søgaard, Michael Osthoff, Karsten Borgwardt, and
305 Adrian Egli. Direct antimicrobial resistance prediction from clinical maldi-tof mass spectra using
306 machine learning. *Nature Medicine*, 28(1):164–174, 2022. doi: 10.1038/s41591-021-01619-9.
307 URL <https://doi.org/10.1038/s41591-021-01619-9>.
- 308