Explanation and Analysis of Models Chosen

HARVEY KWONG and JACOB DEROSA, University at Buffalo, USA

1 Introduction

Following thorough data cleaning and exploratory data analysis, we identified 15 key predictor features. Our objective is to utilize these predictors to classify whether a given sample of *Neisseria gonorrhoeae* exhibits super resistance to specific antibiotics. In this study, resistance to azithromycin serves as the target label for our classification models. Since the target variable is binary (either true or false), we implemented and evaluated six different types of classifiers:

- K-Nearest Neighbors
- Naive Bayes
- Logistic Regression
- Support Vector Machine (SVM)
- Neural Network - (Not from Class) Harvey Kwong's custom architecture
- Extreme Gradient Boosting (XGBoost) - (Not from class)

K - Nearest Neighbors

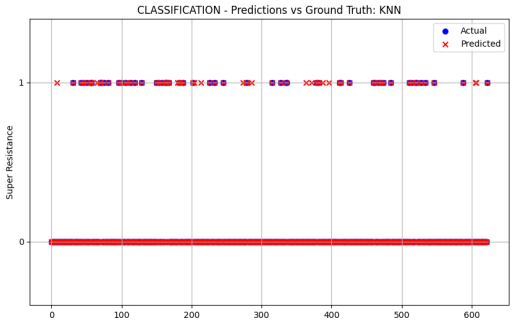
We selected K-Nearest Neighbors as one of the models for our analysis. KNN is a good fit for our problem due to the relatively low dimensionality of the dataset, where we focused on only the most significant features identified during our exploratory data analysis process. By reducing the number of features, KNN's performance improves, as it works best when our data is not at a very high dimension.

For this model, we chose to use 5 neighbors, as this configuration provided the most balanced results. Fewer neighbors made the model more sensitive to noise and overfitting, while more neighbors reduced its ability to capture important variations. Overall, KNN performed strongly, achieving approximately 96% accuracy in predicting azithromycin resistance. The extended metrics for this model are as follows:

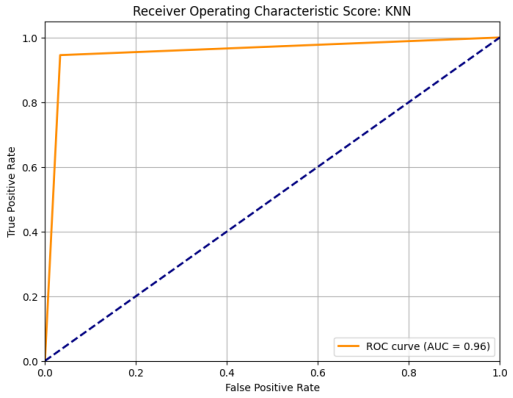
Metric	Score
Accuracy	0.9647
Precision	0.9714
Recall	0.9647
F1 Score	0.9667

Table 1. Performance Metrics for K-Nearest Neighbors

Authors' Contact Information: Harvey Kwong, harveykw@buffalo.edu; Jacob DeRosa, jderosa3@buffalo.edu, University at Buffalo, Buffalo, New York, USA.



(a) K-Nearest Neighbors Predictions vs Ground Truth



(b) K-Nearest Neighbors ROC curve

Fig. 1. K-Nearest Neighbors

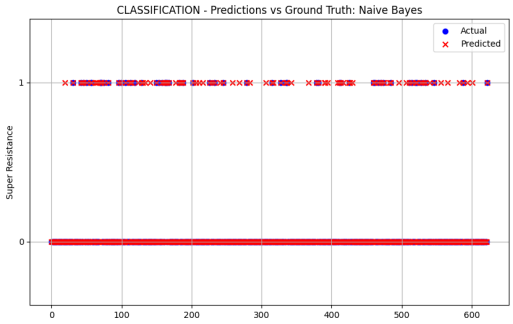
Naive Bayes

We chose to implement Naive Bayes as one of our classification models. Naive Bayes is well-suited to our problem due to its efficiency and simplicity, especially when dealing with binary classification tasks like predicting antibiotic resistance. The model assumes that the features are independent, which, while a simplification because many things are interconnected in biology – performed surprisingly well in our modelling. Given the relatively reduced number of features in our dataset, Naive Bayes was an appropriate choice as it tends to work well even with limited data.

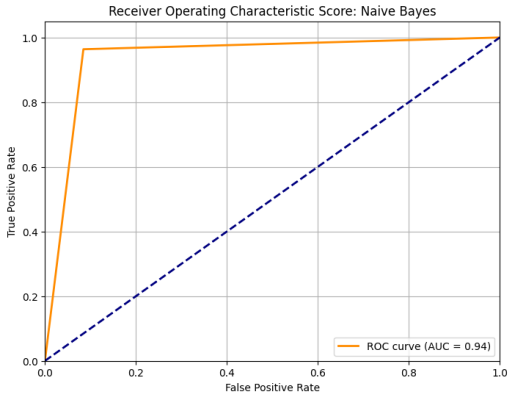
Additionally, Naive Bayes requires minimal computational resources, which makes it advantageous for fast predictions. Although it does not capture feature interactions, it still correctly predicted azithromycin resistance approximately 91% of the time. Our extended metrics for this model are as follows:

Metric	Score
Accuracy	0.9197
Precision	0.9545
Recall	0.9197
F1 Score	0.9298

Table 2. Performance Metrics for Naive Bayes



(a) Naive Bayes Predictions vs Ground Truth



(b) Naive Bayes ROC curve

Fig. 2. Naive Bayes

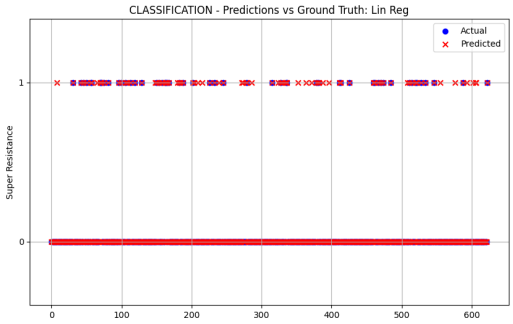
Logistic Regression

We chose to include Logistic Regression as one of the models for our classification task. Logistic Regression is a commonly used algorithm for binary classification problems like ours, where the goal is to predict azithromycin resistance. One of the main advantages of Logistic Regression is its simplicity and interpretability, as it provides clear probabilistic outputs. This makes it particularly useful for understanding the relationship between the predictor variables and the target label. Though in our model, we thresholded the predicted values. Values above 0.5 went to positive, and those below went to negative.

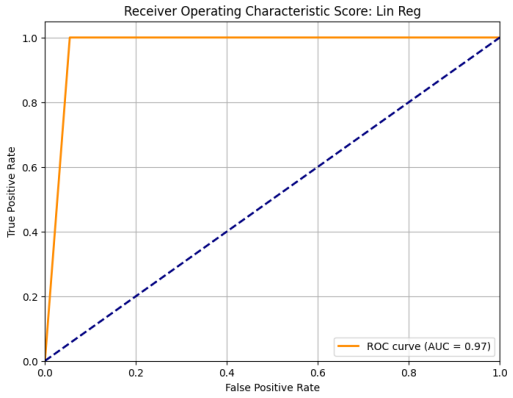
Since our dataset contains well-processed and balanced features, Logistic Regression was able to achieve a solid performance without much additional complex tuning. Despite using a max iteration limit of 10, the model demonstrated reliable results, with an accuracy of around 95%. Our extended metrics for this model are as follows:

Metric	Score
Accuracy	0.9502
Precision	0.9681
Recall	0.9502
F1 Score	0.9550

Table 3. Performance Metrics for Logistic Regression



(a) Logistic Regression Predictions vs Ground Truth



(b) Logistic Regression ROC curve

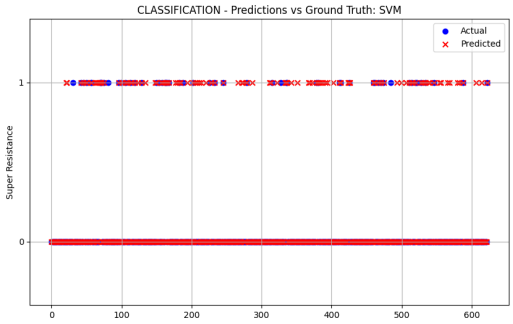
Fig. 3. Logistic Regression

Support Vector Machine (SVM)

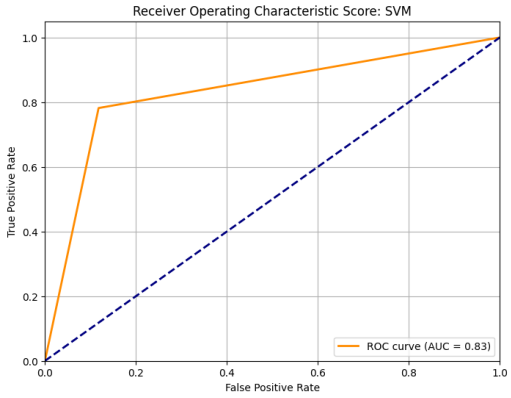
We chose to include support vector machines as one of the models for our classification task. SVM is a powerful algorithm for binary classification problems. It works by finding an optimal hyperplane that separates the classes with the largest margin, making it effective in cases where the data is not linearly separable. SVM is well-suited to this problem as it can handle non-linear decision boundaries using various kernels. For our implementation, we used the default polynomial kernel and set the maximum number of iterations to 10 to ensure computational efficiency while maintaining acceptable performance. Despite the iteration limit, SVM performed reasonably ok, achieving an accuracy of approximately 87%. Our extended metrics for our SVM model are as follows:

Metric	Score
Accuracy	0.8732
Precision	0.9249
Recall	0.8732
F1 Score	0.8911

Table 4. Performance Metrics for Support Vector Machine



(a) SVM Predictions vs Ground Truth



(b) SVM ROC Curve

Fig. 4. Support Vector Machine

Neural Network

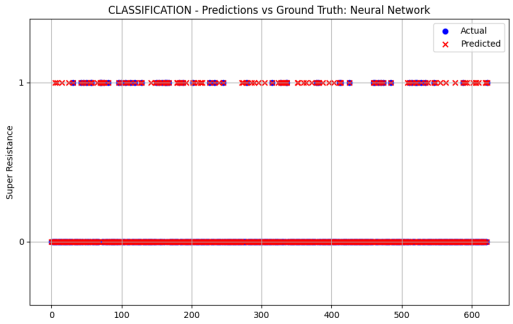
We included a custom Neural Network as one of the models for our classification task. Neural networks are highly flexible and capable of capturing complex patterns in data, making them suitable for problems where non-linear relationships between features and the target label exist. Given the relatively simple structure of our dataset, we designed a straightforward architecture to avoid overfitting and any other unnecessary complexities.

Our custom architecture consists of two hidden layers with 8 and 4 neurons, respectively, both using the RELU activation function. For the output layer, we used a single neuron with a sigmoid activation function, which is the norm for binary classification tasks. We compiled the model with the Adam optimizer and binary cross-entropy loss, ensuring efficient training with a learning rate of 0.1. Adam will automatically adjust the learning rate as it sees fit.

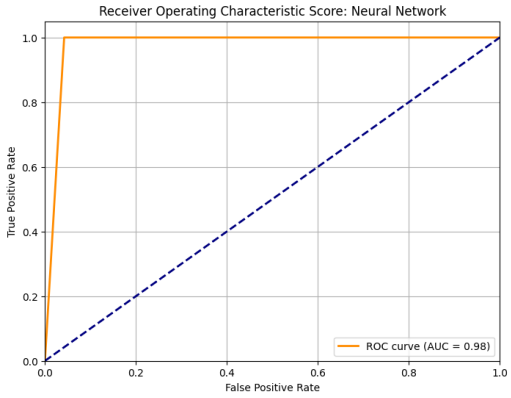
Despite the simplicity of the architecture, the neural network performed fine, achieving an accuracy of approximately 90%. We trained the model for 10 epochs, to enable easier comparisons with our other models which have had their maximum iterations set to 10. Our neural network did not perform as well as logistic regression despite having a more complicated structure and presumably higher computational cost. It is likely better to use logistic regression on our dataset compared to our neural network with our current choice of hyperparameters. i expect that if we were to increase the number of hidden units and epochs, we would see much better performance, though at a higher cost. Our extended metrics for this neural network model are as follows:

Metric	Score
Accuracy	0.9053
Precision	0.9543
Recall	0.9053
F1 Score	0.9192

Table 5. Performance Metrics for Neural Network



(a) Neural Network Predictions vs Ground Truth



(b) Neural Network ROC Curve

Fig. 5. Neural Network