# Big Data Science project

## 1 Results task 1: Classification - Antimicrobial resistance detection with mass spectrometry

### 1.1 Dataset splitting

All three given datasets were combined and missing values were analyzed. As can been seen in Table 1, they are up to 59% for some of the antibiotics. In order to maximize the available data, I have decided to split the dataset by antibiotic for the resistance prediction and combine the training data for species classification.

| Antibiotic | # Missing values | Percentage |
|---|---|---|
| Ciprofloxacin | 144 | 1.24% |
| Ceftriaxone | 189 | 1.62% |
| Cefepime | 3921 | 33.66% |
| Piperacillin | 6851 | 58.82% |
| Tobramycin | 3928 | 33.72% |

Table 1: Missing values for each antibiotic

| Feature | Importance |
|---|---|
| 6257 | 0.354 |
| 7742 | 0.284 |
| 1615 | 0.277 |

Table 2: Feature importance according to Decision Tree Classifier

As one of the models is a Multi-Layer Perceptron (MLP), for which cross-validation (CV) is limited due to time and computational resource constraints, I did not perform it to ensure comparability across all models. Otherwise, I would have performed CV for the Random Forest (RF) and Support Vector Machine (SVM).

### 1.2 Problem 1: species prediction

### 1.3 Dataset cleaning and exploratory data analysis

The training, validation and test of all antibiotics was combined for this problem. No outliers were found using Isolation Forest and Local Outlier Factor, so the features were scaled with z-score normalization. Binning method was not used in order to have higher resolution of the spectra.

The data is imbalanced, with the majority class comprising almost 57%, as can be seen in Figure 1, a. The PCA visualization in Figure 1, b indicates limited separation, with an overlap among the species based on the first two components. In contrast, the t-SNE visualization shows a clearer separation, with distinct clusters for each species (Figure 1, c).

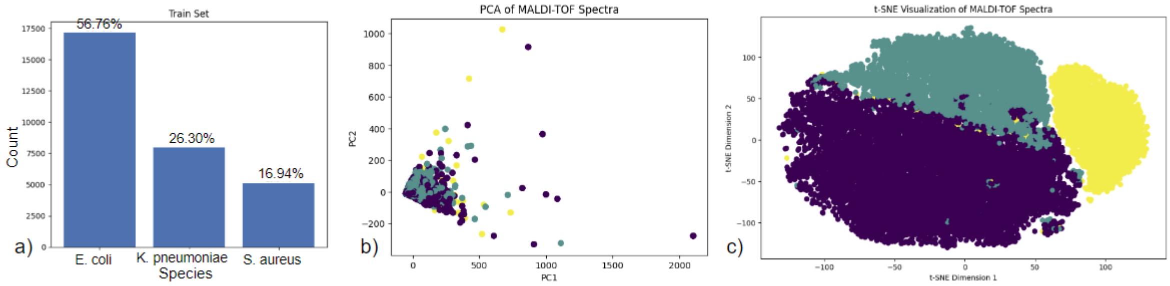

Figure 1: a) The training data for species prediction. b) PCA and c) t-SNE of training data

#### 1.3.1 Model comparison with all the features

In **RF**, the following hyperparameters were varied and 10 configurations were tried:

- The number of trees in the forest: randomly from 50 to 200
- The maximum depth of the tree: No max, 10, 20, 30, 40, 50

The best configuration had the depth of 40 and 70 trees. The advantages of RF are its interpretability and ability to be parallelized.

In the protocol, I initially mentioned using logistic regression. However, I later realized it is not the best option because the features cannot be considered independent, which is one of the assumptions of logistic regression. Since **SVM** does not have this assumption, it became my new choice. The following hyperparameters were varied and 5 configurations were tried:

- Regularization parameter C: 0.001, 0.01, 0.1, 1, 10, 100, 1000
- Kernel: linear, RBF (Radial Basis Function)

The number of configurations was rather limited because of computational cost of SVM. The best configuration had the C of 0.1 and RBF kernel. As a result, both the RF and SVM models demonstrated perfect scores on the test data when using all the features, as shown in Figure 2. However, the SVM model required a longer computation time, as indicated in Table 3, because it cannot be parallelized and its algorithm is more complex.

### 1.3.2 Dimensionality reduction

For feature selection (FS), a Decision Tree Classifier was used. It identified three important features: 6257, 7742, and 6845 (Table 2). Models were subsequently tested using these three features, as well as subsets consisting of the two most important features and the single most important feature. RF outperformed SVM in both performance and time, as shown in Figure 2 and Table 3.

**Visualization of top three important features** I first used PCA and t-SNE to visualize the separation of the data. In Figure 3, the separation is clear in both PCA and t-SNE.

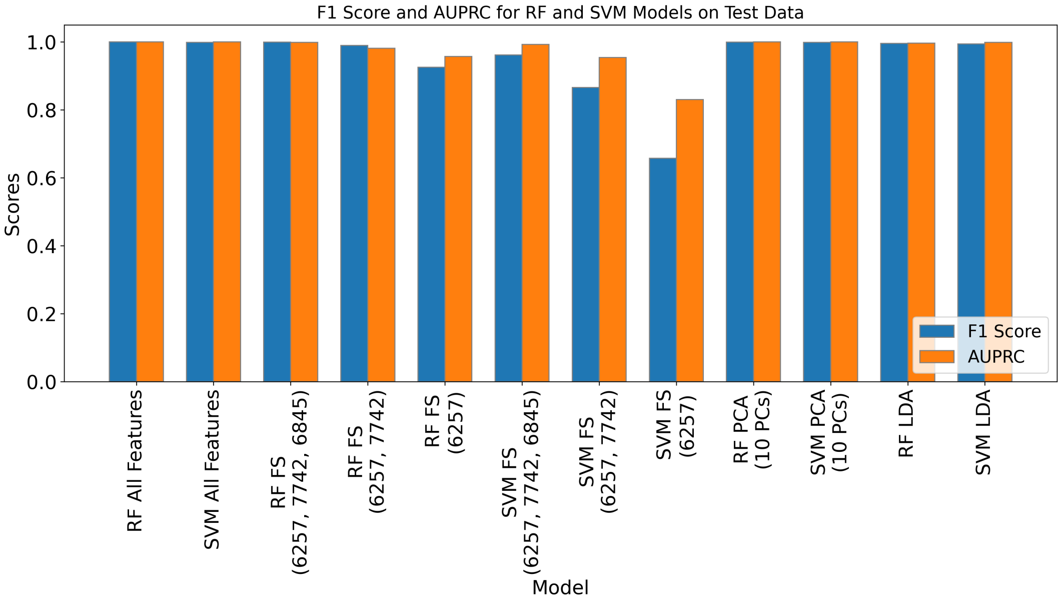

Figure 2: The summary of RF and SVM performances in problem 1

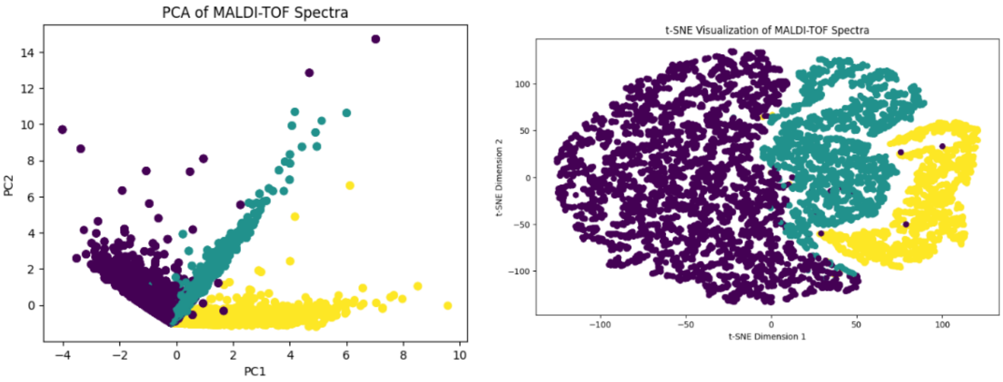

Figure 3: PCA (left) and t-SNE (right) of top three important features

I also constructed boxplots to observe how the top three features are distributed across the species. These boxplots in Figure 4 (left) provided a clear picture, illustrating which feature is characteristic of each species.

The LASSO method, however, was computationally too heavy, so I performed PCA and LDA and compared performances of the RF and SVM models after dimensionality reduction. After PCA (with 10 PCs, explained 90% of the variance) and LDA, the performance was the same as before (Figure 2), but the training time significantly decreased (Table 3).

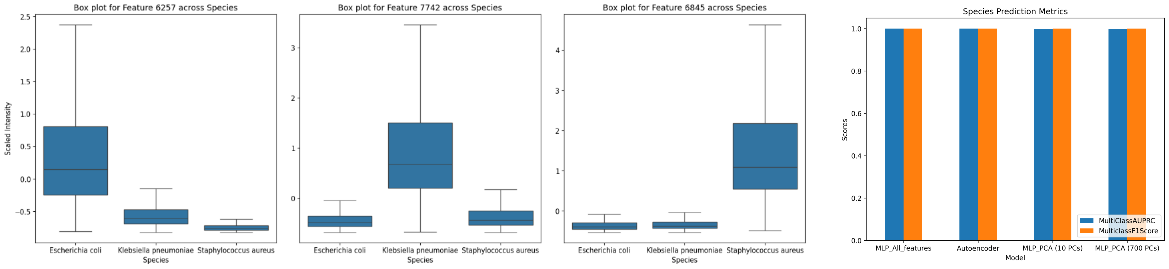

Figure 4: Three top features across the species in the training data (left). The summary of the MLP performance for species prediction (right).

| | RF | SVM |
|---|---|---|
| All features | 1 min 13 sec | 4 min 29 sec |
| FS (6257, 7742, 6845) | 0 min 0 sec | 0 min 33 sec |
| FS (6257, 7742) | 0 min 0 sec | 0 min 50 sec |
| FS (6257) | 0 min 0 sec | 1 min 17 sec |
| PCA (10 PCs) | 0 min 4 sec | 0 min 5 sec |
| LDA | 0 min 2 sec | 0 min 2 sec |

Table 3: Execution times of RF and SVM models (problem 1)

| Model | Time |
|---|---|
| MLP All Features | 19 min 45 sec |
| MLP AE | 10 min 43 sec |
| MLP PCA (10 PCs) | 8 min 30 sec |
| MLP PCA (700 PCs) | 8 min 57 sec |

Table 4: Execution times for best MLP model (combined problem)

## 1.4 Problem 2: antibiotic resistance prediction

| Model | Ciprofloxacin | Ceftriaxone | Cefepime | Piperacillin | Tobramycin |
|---|---|---|---|---|---|
| RF All Features* | 0 min 2 sec | 0 min 2 sec | 0 min 3 sec | 0 min 2 sec | 0 min 3 sec |
| SVM All Features | 14 min 46 sec | 13 min 41 sec | 6 min 21 sec | 2 min 6 sec | 6 min 11 sec |
| RF FS | 0 min 0 sec | 0 min 0 sec | 0 min 1 sec | 0 min 1 sec | 0 min 1 sec |
| SVM FS | 0 min 29 sec | 0 min 9 sec | 0 min 5 sec | 0 min 2 sec | 0 min 16 sec |
| RF PCA | 0 min 0 sec | 0 min 1 sec | 0 min 1 sec | 0 min 1 sec | 0 min 1 sec |
| SVM PCA | 2 min 8 sec | 1 min 44 sec | 0 min 44 sec | 0 min 10 sec | 0 min 37 sec |
| RF LDA | 0 min 0 sec | 0 min 0 sec | 0 min 1 sec | 0 min 1 sec | 0 min 1 sec |
| SVM LDA | 0 min 2 sec | 0 min 2 sec | 0 min 0 sec | 0 min 0 sec | 0 min 0 sec |

Table 5: Execution times of RF and SVM models (problem 2). *For all the RFs here, the number of parallel jobs is equal to 100

For each antibiotic, no outliers were found using Isolation Forest and Local Outlier Factor, and features were scaled with z-score normalization. The training data was imbalanced in each antibiotic as demonstrated in Figure 5. The same models and dimensionality reduction techniques used in the previous classification were also applied to this problem. The performance on test data and execution times of the models are summarized in Figure 6 and Table 5, respectively. It can be seen that RF is faster because of parallelization, but for some antibiotics SVM outperforms RF in terms of performance. However, in most cases they demonstrate similar results.

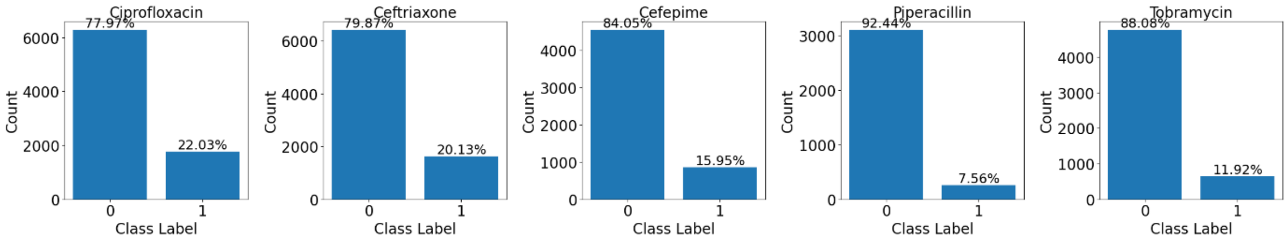

Figure 5: The training data for predicting antibiotic resistance.

## 1.5 Combined problem

For the combined task, MLP neural network with a multihead architecture was developed. In this context, "combined" means that the model has two outputs: one for multiclass classification to predict species and

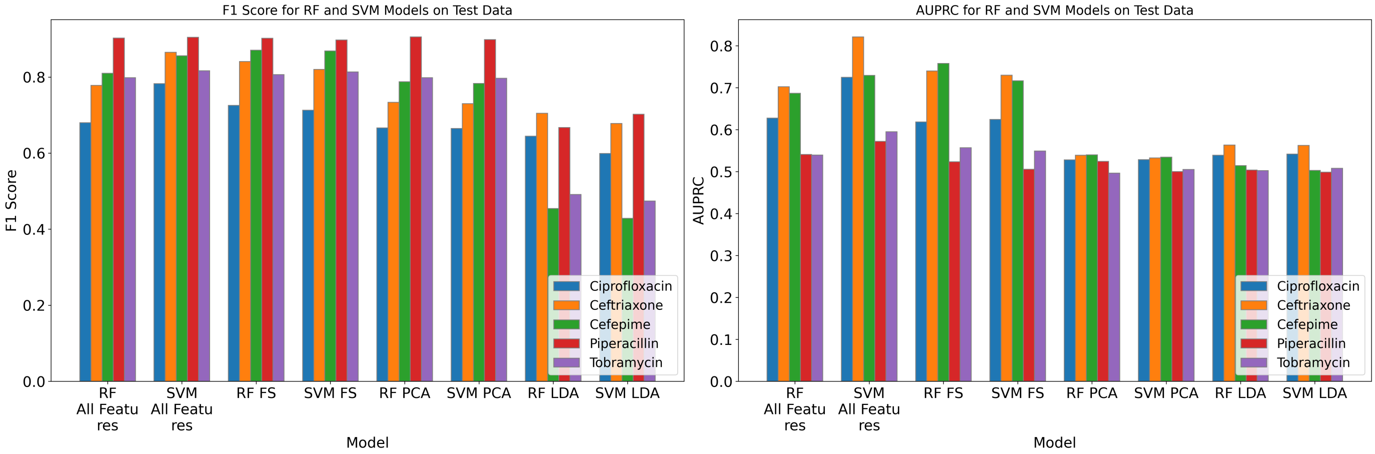

Figure 6: The summary of RF and SVM performances in problem 2.

another for binary classification to predict antibiotic resistance. This multihead architecture allows the model to simultaneously handle both tasks, potentially enabling the model to leverage shared information and improve learning for each task. A hyperparameter optimization (HPO) using random search was conducted and 100 configurations were evaluated, each with 100 epochs and a patience of 15 on a validation loss. The Weighs and Biases platform was used for logging and the results of HPO is available via https://wandb.ai/reginaib/BDS_task_comb/sweeps/4b5wf0d0?nw=3ojp12pczen . The following parameters were varied in the combined task:

- Number of Hidden Layers: 1, 2, 3, 4
- Hidden Layer Size: 16, 32, 64, 128, 256, 512, 768, 1024
- Learning Rate: 0.00001, 0.00003, 0.0001, 0.0003, 0.001
- Dropout Rate: 0.01, 0.1, 0.2, 0.3, 0.4, 0.5

The optimal (based on validation loss) MLP model configuration included 4 hidden layers, each with 128 units. The model was trained with a learning rate of 0.00003 and a dropout rate of 0.1. Using the best set of hyperparameters, the model was trained with features obtained after dimensionality reduction using autoencoder (AE, reduction to 100 dimensions) and PCA (10 PCs explained 91% of the variance, while 700 PCs explained 99% of the variance). I also attempted to perform dimensionality reduction using LDA and Decision Tree Classifier, utilizing only the species labels since the resistance labels contain missing values. However, it appears inappropriate to use only a single output variable for dimensionality reduction in the combined problem. Runs are available via https://wandb.ai/reginaib/BDS_task_comb_train?nw=nwuserreginaib. The performances and time are summarized in Figures 4 (right), 7 and Table 4. To sum up, MLP did not outperform previous models, probably because of small dataset or more work is needed in terms of HPO. Specifically in the case of piperacillin, which had least data available (Table 1), the performance was the lowest in the case of MLP.

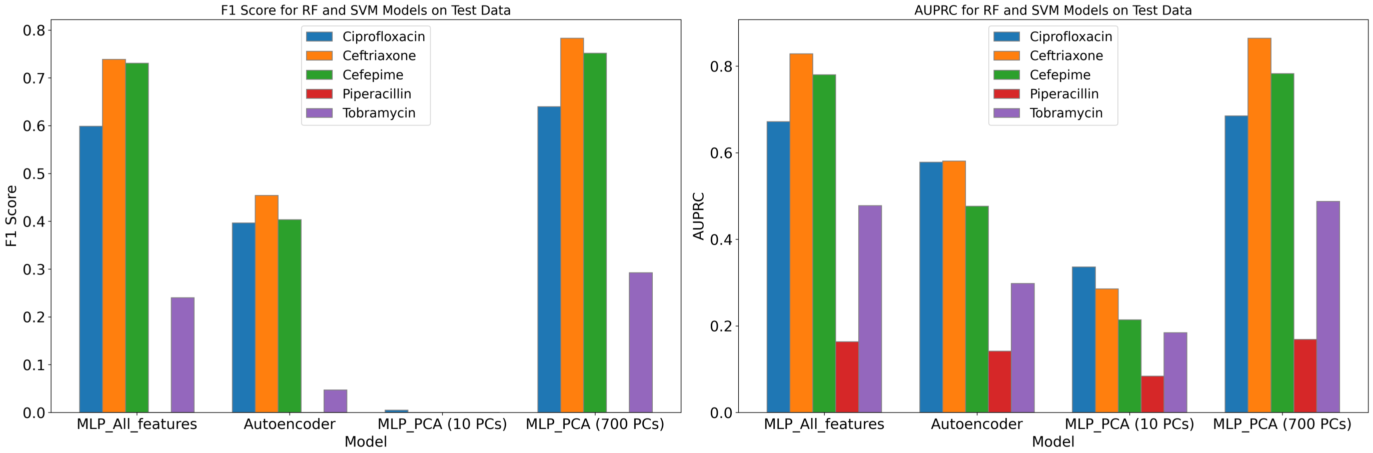

Figure 7: The summary of best MLP performance for antibiotic prediction.