Machine Learning in Computational Biology
Assignment 1

Konstantinos Konstantinidis
Student number: 7115152400017

April 5, 2025

Repo: The repository for this assignment can be found here:
`https://github.com/KonsKons26/Assignment-1`

Contents

1 Data Exploration

1.1 Preprocessing

Befor beginning with any machine learning task, we need to inspect the data and perform some rudimentary preprocessing steps. The dataset is provided in two `.csv` files, one for developing and one for validating the models. Both datasets contained no missing values, but they contained some columns with metadata, namely `Project ID`, `Experiment Type`, and `Disease MESH ID`. These columns were removed from both datasets. Finally, the values in the `Sex` column were converted to binary values, where `male` was set to 0 and `female` to 1.

1.2 Data Exploration

All analyses were performed on the **development** dataset.

First, I decided to plot the distribution of the BMI values. The distribution is shown in Figure 1. The histogram contains the BMI values, while the `kde` lines show their distribution. The distribution does not seem to be normal, as it is skewed to the right and when the values for both sexes are combined, the distribution seems bimodal.

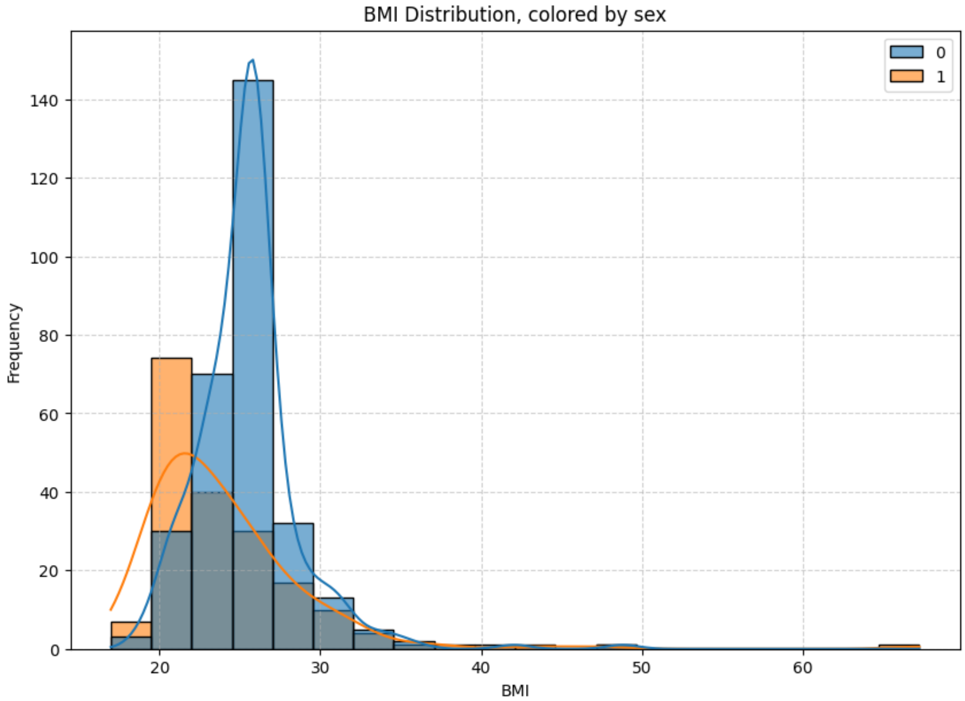


Figure 1: Histogram with a `kde` lines, colored by sex.

Another really interesting plot is the one showing the number of zeros in each feature and the mean `BMI` when the bacterial concentrations are non-zero. The plot is shown in Figure 2. The plot shows that a great number of bacteria have a lot of zero values. Also, the mean BMI (when ignoring the zeros) diverges a lot from the mean value of the whole dataset for the bacteria that have a lot of zeros. This might indicate that these bacteria are highly correlated with the `BMI` values.

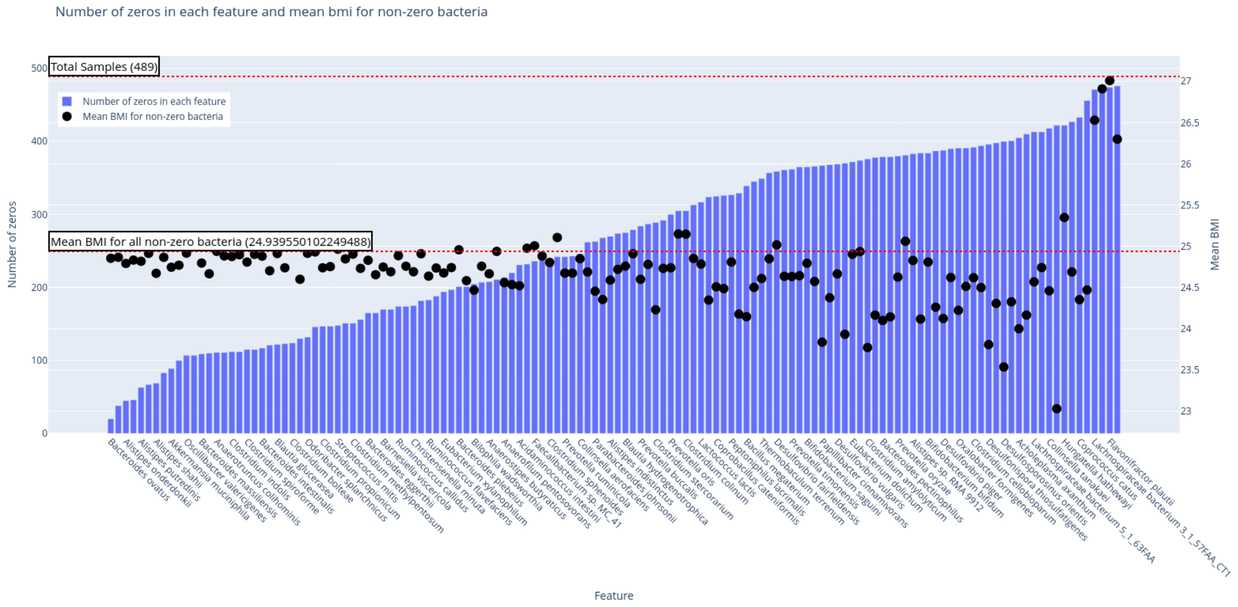


Figure 2: Barplot (sorted) showing the number of zeros in each bacterium and the mean BMI when the bacteria are not zero.

A good way to grasp the feature space is to plot the mean, min, and max BMI for each feature, as show in Figure 3. We can see that the mean values of almost all bacteria lie around 0, they have very small IQR's and very few outliers. A MinMax normalization would make the feature space very sparsely populated, so it seems like a bad choice. On the other hand, a standardization seems more optimal; even if it might lead to negative values which make no sense when talking about bacterial concentrations, our models will not mind. A plot in similar spirit is shown below (Figure 4).

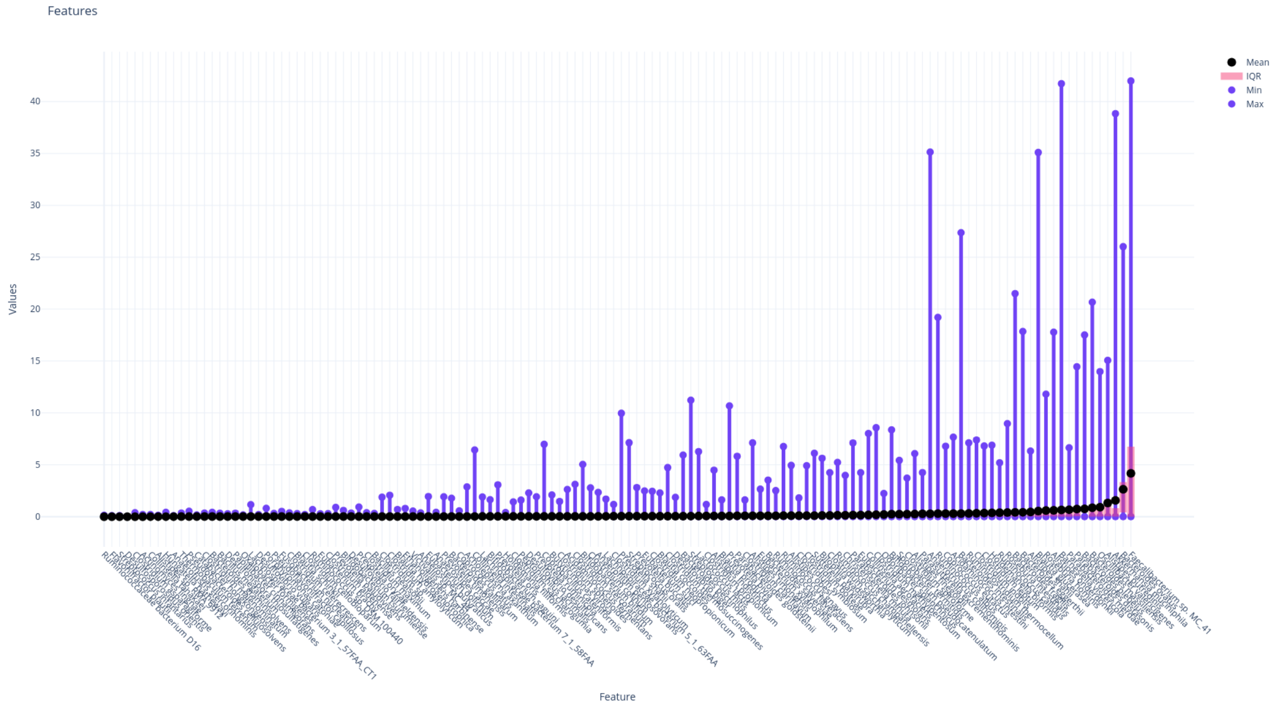


Figure 3: Feature spans: Mean, minimum and maximum, and Inter-Quantile Range IQR of each feature, (black circles, purple sticks, transparent pink bars respectively). Features sorted by ascending mean values.

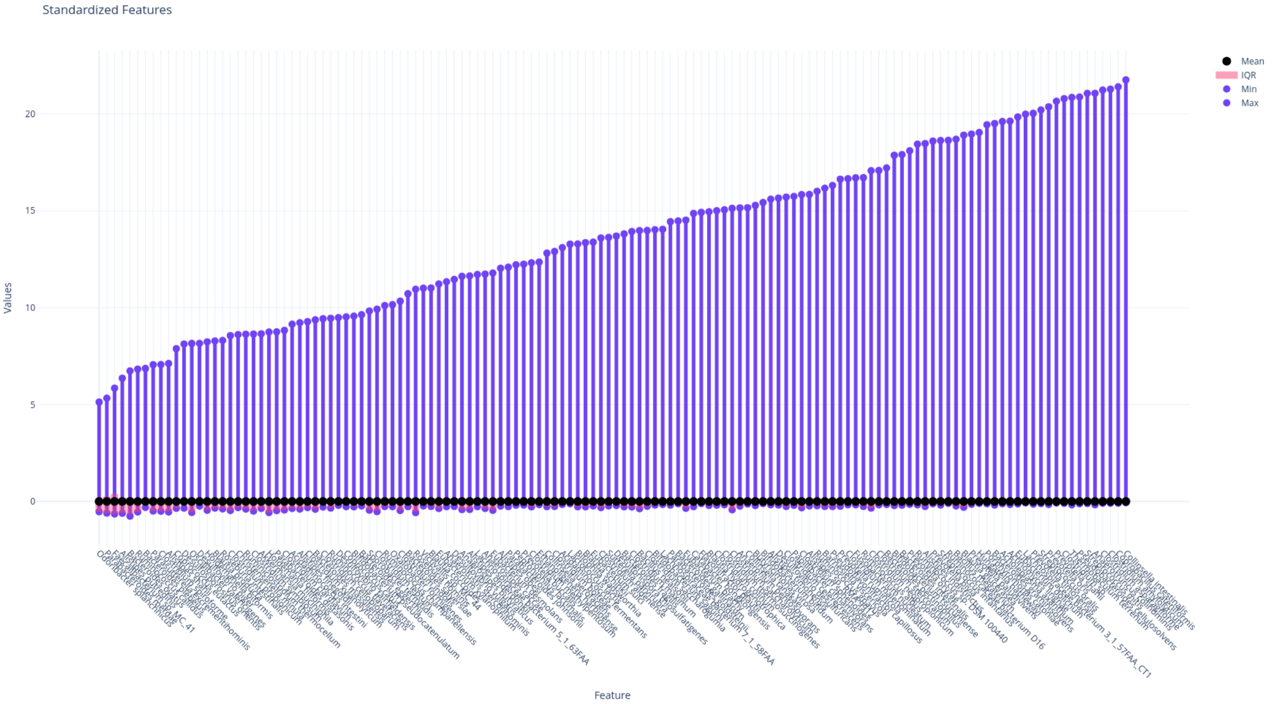


Figure 4: Features — Standardized: Mean, minimum and maximum, and Inter-Quantile Range IQR of each feature, (black circles, purple sticks, transparent pink bars respectively). Features sorted by ascending max values.

Another importand metric when aiming for a regression model is to check the correlation of the features with the target values. Pearson's correlation coefficient measure linear relationship between the two variables, while Spearman's and Kendall's can capture non-linear relationships, as long as they are monotonic. In either case though, our dataset does not show any type of correlation (Figure 5).

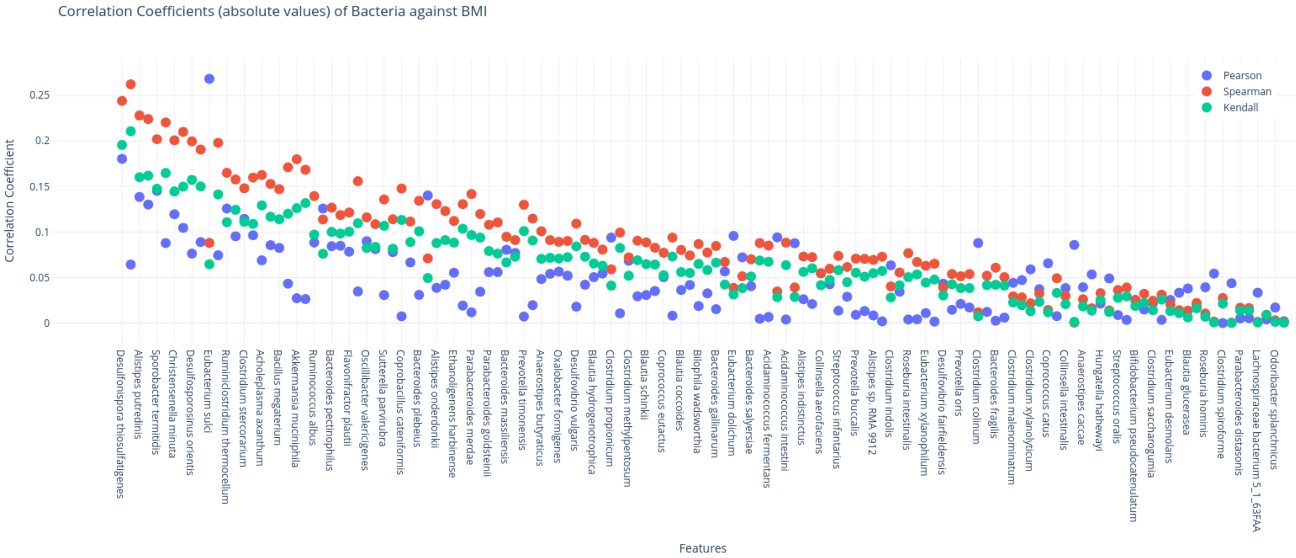


Figure 5: Pearson's r, Spearman's ρ, and Kendall's τ absolute correlations of the bacteria with BMI (blue, red, and green respectively), sorted by descending order.

Taking the 5 most correlated bacteria with BMI, and plotting them against each other in a pairplot (Figure 6), we can see that they are highly uncorrelated, indicating that they at least will carry different information to the regression problem.

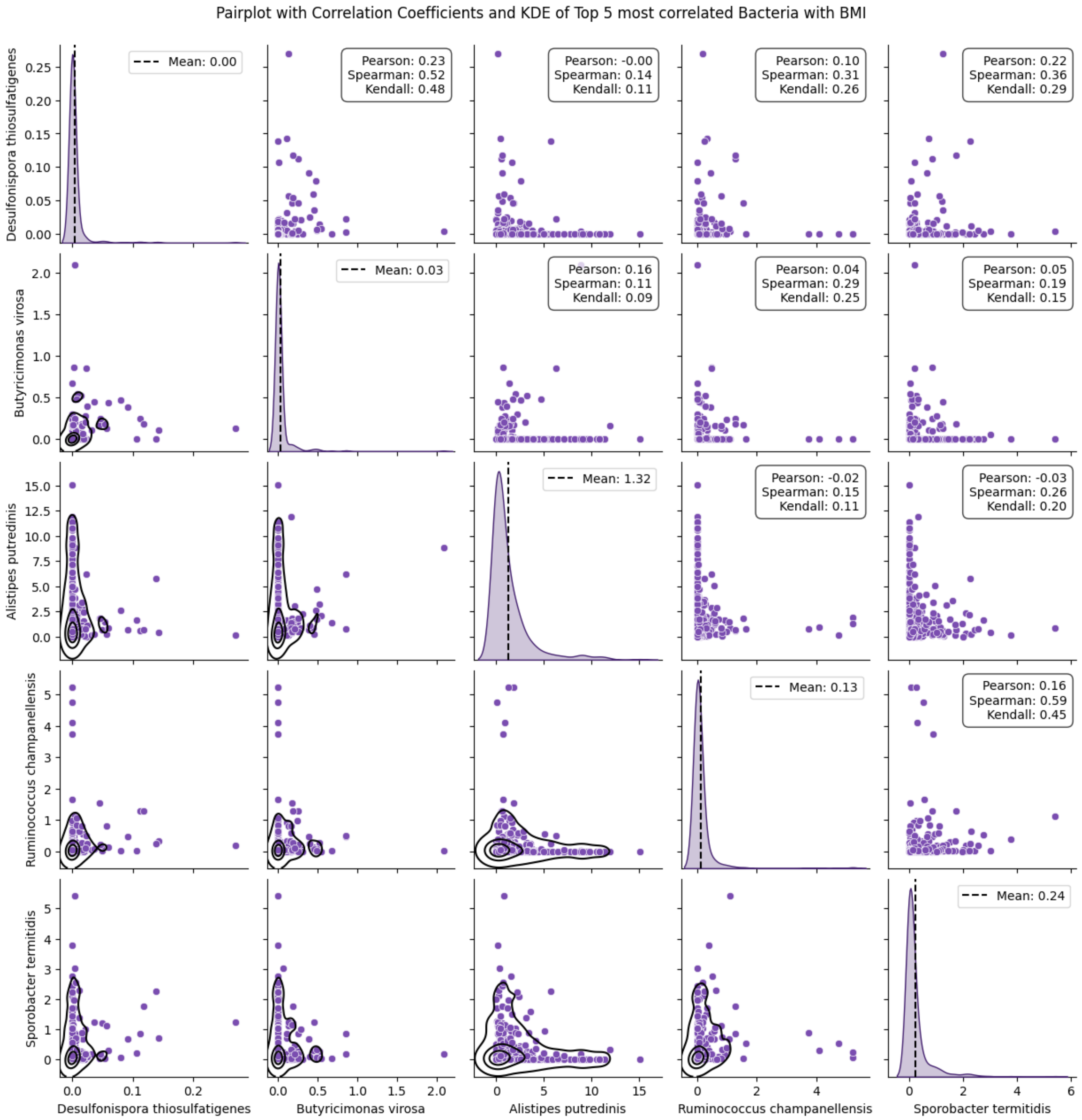


Figure 6: Pairplot of the top 5 bacteria based on correlation with BMI, plotted against each other.

One final attempt to capture the inner structure of our features is to use dimensionality reduction techniques to try and project the data in a way that would be beneficial to us. For example, using PCA, one can find the orthogonal axes that span the directions of maximum variance in the data, allowing us to reduce redundancy and potentially visualize high-dimensional patterns in a lower-dimensional space. After performing PCA, I retained the first 5 principal components and plotted them in pairs, Figure 7. The same process was followed with t-SNE and UMAP (*the results can be seen in the notebook* `data_exploration.ipynb`, *they will not be included here as their results are similar to the ones already presented*). No matter the technique used, no discernible structure can be observed.

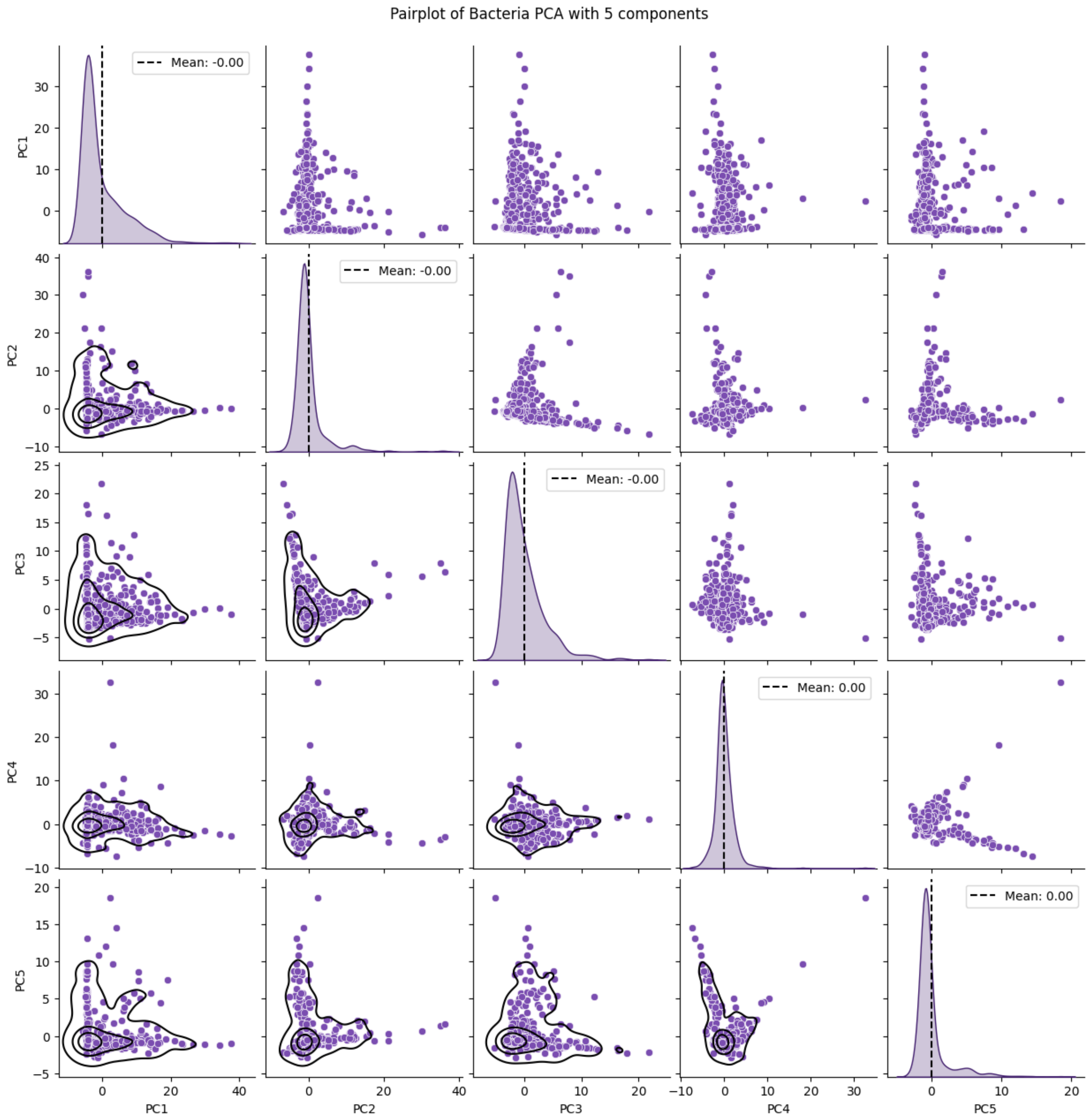


Figure 7: PCA: Pairplot of the first 5 principal components.

1.3 Conclusion

1. First we can observe that BMI (dependent variable) does not follow the normal distribution, so a simple linear regression model like the least squares will likely not work. One interesting thing to notice is that the male participants have a higher BMI than the female participants, which will likely play a role in the final model.

2. When plotting the number of bacteria with zero values and the mean BMI when the bacteria are not zero we get something interesting. The bacteria with many zero values, show a different BMI compared to the rest.

3. The bacteria lie close to 0 with only a few outliers as seen by their means and Inter Quantile Ranges (IQR). A MinMax normalization would make the features very sparce in the range $[0, 1]$, so a Standardization seems like a better approach.

4. Using Pearson's r, Spearman's ρ, and Kendall's τ correlation coefficients we see that there is not a single bacterial species that is highly correlated (either positively or negatively) with BMI. Plotting a pairplot of the 5 most correlated features with BMI we can see that those features are not highly correlated between themselves so at least they –in theory– carry different information and using them would be helpful.

5. When using either of the three dimensionality reduction methods (PCA, t-SNE, UMAP) no visible clusters form.

2 Regression

2.1 Baseline models

The first step in the regression task is to create a baseline model. I created three baseline models, one for each type of regressor: Elastic Net, SVR, and Bayesian Ridge. The model training was performed by first fitting a standardizer to the training set (after splitting the data into training and testing sets) and then transforming the training and testing sets. The models were trained using the training set with no cross-validation, and the results were evaluated using the testing set.

2.2 Feature selection

The next step was to perform feature selection and train the models again, using the selected features. I decided to use a method that is model-dependent, so for each of the three model types, I performed feature selection using different methods, then I calculated the scores of the models using the selected features and I chose the best performing feature selection method for each model.

The feature selection methods I used are:

- **SelectKBest**: I used the `SelectKBest` method to select the top k features based on two different metrics:
  - `f_regression`: This method uses the F-statistic to select the features.
  - `r_regression`: This method uses the Pearson's r correlation coefficient to select the features.
- **VarianceThreshold**: I used the `VarianceThreshold` method to select the features with the highest variance.

All methods are implemented in the `sklearn` library. The first stip of the process is to perform feature selection using these methods and then train the models using the selected features using a K Fold cross-validation with 5 folds. After that, I selected the method that yielded the best score for each model, based on the following scoring function (Equation 1).

$$\text{score} = \frac{\text{mean}(R^2)}{(\text{mean}(RMSE) + \text{mean}(MAE))/2} \tag{1}$$

I used this scoring function to try and balance the maximization of the r^2 score with the minimization of RMSE and MAE. The selected feature selection methods for each model are:

- **Elastic Net**: `SelectKBest` with the `f_regression` metric and $k = 20$, yielding 20 features.

- **SVR**: `SelectKBest` with the `r_regression` metric and $k = 20$, yielding 20 features.
- **Bayesian Ridge**: `VarianceThreshold` with a threshold of 0.1, yielding 57 features.

The selected features for each model are shown in Figure 8. The features are sorted by the times they were selected by for each model type. As was expected, *Host age* was selected for all models, but *Sex* was not selected by `SVR` which used the `r_regression` metric.

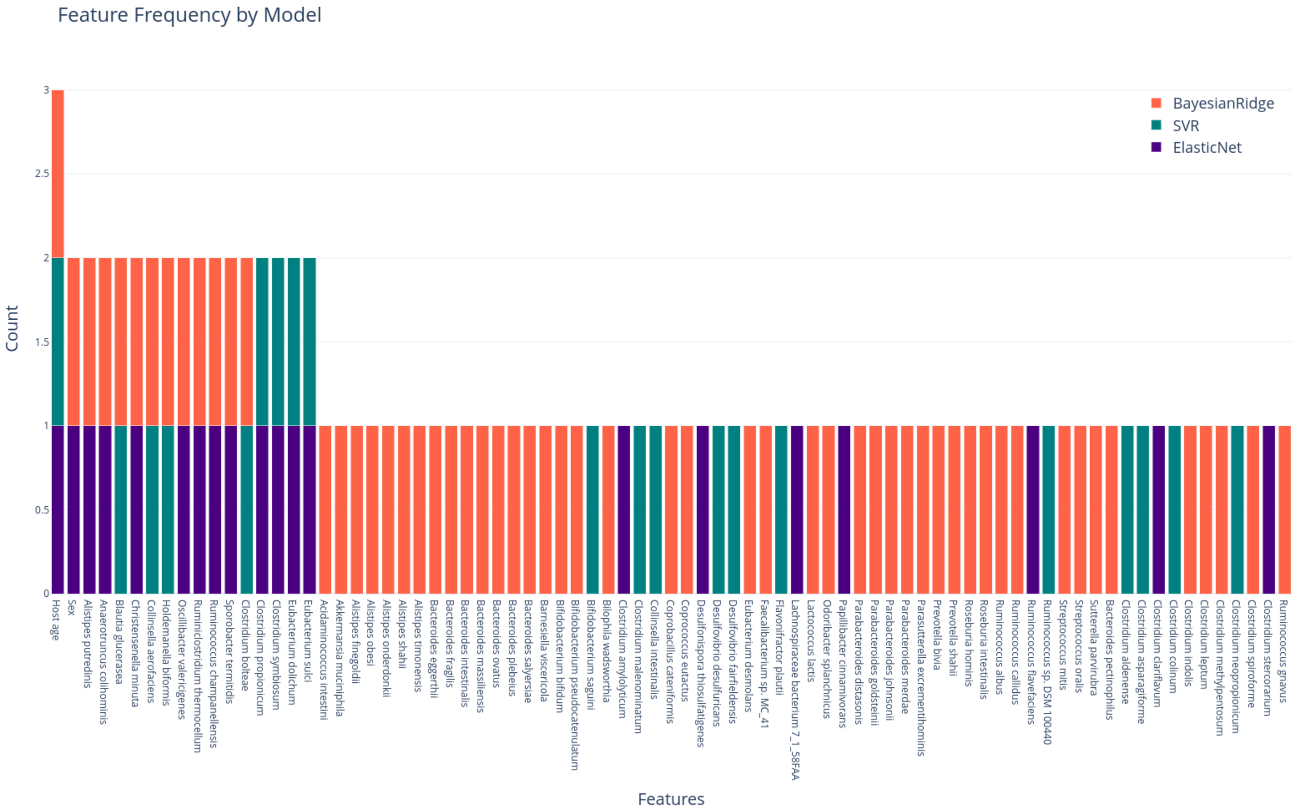


Figure 8: Selected features for each model.

2.3 Tuning

The final step was to tune the models using the selected features. I used a `GridSearchCV` with 5 folds with negative mean squared error, negative mean absolute error, and R^2 as scoring metrics.

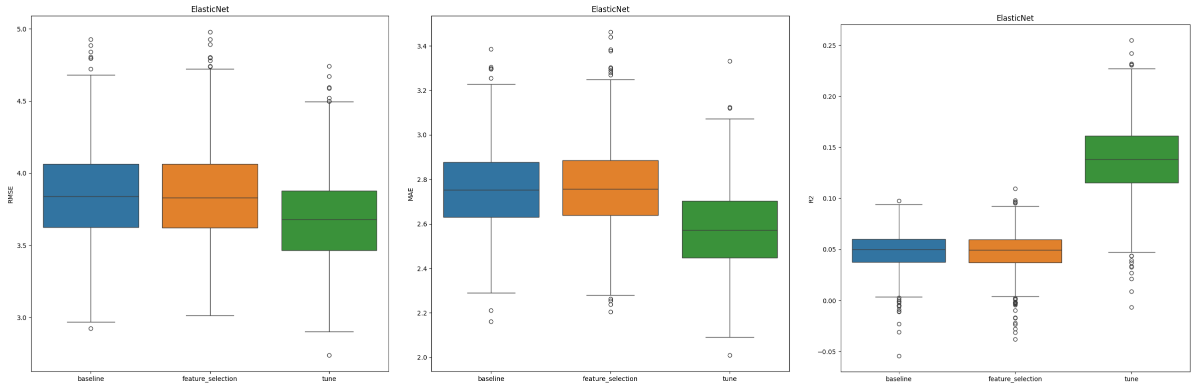
The hyperparameter spaces for each model are summarized in the following table (Table 1).

Model	Parameter	Values	Picked value
ElasticNet	α	`linspace`(0.1, 1.0, `grid_size`)	0.98
	`l1 ratio`	`linspace`(0.1, 1.0, `grid_size`)	0.1
	`tolerance`	[1e-3, 1e-4, 1e-5, 1e-6, 1e-7]	0.001
SVR	`kernel`	['rbf', 'linear', 'poly', 'sigmoid']	'rbf'
	`degree`	[2, 3, 4]	2
	γ	['scale', 'auto']	'auto'
	`coef_0`	`linspace`(0.0, 1, `grid_size`)	0.0
	`tolerance`	[1e-3, 1e-4, 1e-5, 1e-6, 1e-7]	1e-7
	`C`	`linspace`(0.1, 1.0, `grid_size`)	1
	ϵ	`linspace`(0.0, 1.0, `grid_size`)	0.222
BayesianRidge	`tolerance`	[1e-3, 1e-4, 1e-5, 1e-6, 1e-7]	0.001
	α_1	`linspace`(1e-3, 1e-9, `grid_size`)	0.001
	α_2	`linspace`(1e-3, 1e-9, `grid_size`)	1e-9
	λ_1	`linspace`(1e-3, 1e-9, `grid_size`)	1e-9
	λ_2	`linspace`(1e-3, 1e-9, `grid_size`)	0.001
	`compute_score`	[True, False]	True

Table 1: Hyperparameter spaces for each model. The `grid_size` is the number of points in the grid for each hyperparameter, set to 50 by default. The `linspace` function used provided by `numpy` is returning evenly spaced numbers over a specified interval. The values are presented exactly as they were passed in the `Python` functions for easier comparison with the code base.

2.4 Results

The results of the whole process can be seen in the boxplots below (Figure 9, Figure 10, and Figure 11). The models, after being trained with the development set, were tested on teh validation set. For a better calculation of metrics, I opted to using bootstrapping/resampling. Resampling, as the name implies, resamples (randomly) values from the data set in each different loop, leading to different combinations of the data set; I used 1000 loops, so the below boxplots show the metrics calculated 1000 times with resampling.

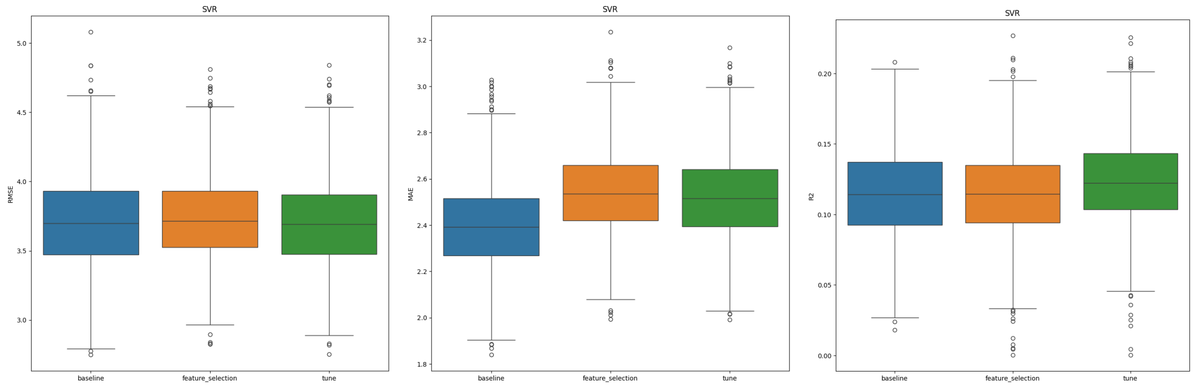


(a) RMSE

(b) MAE

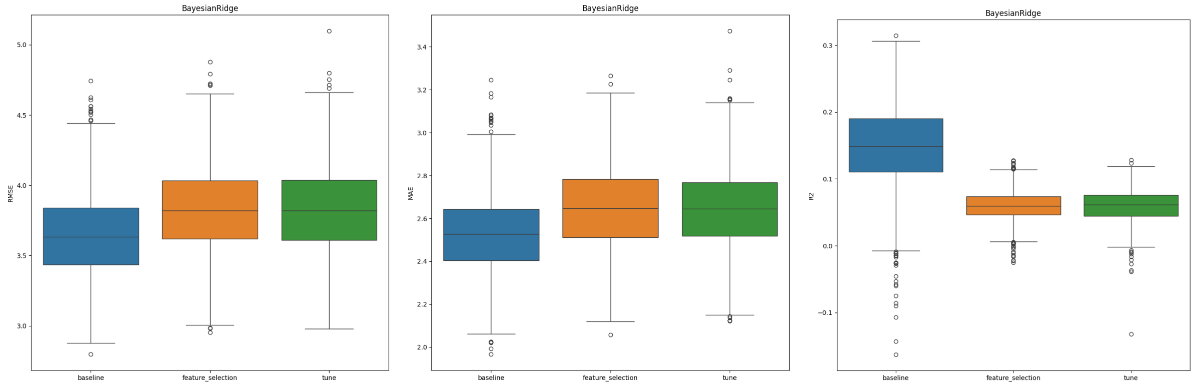
(c) R^2

Figure 9: Elastic Net: Results based on the validation set, RMSE, MAE, and R^2. Blue box for the baseline model, orange box for the feature selection model, and green box for the tuned model.



(a) RMSE (b) MAE (c) R^2

Figure 10: SVR: Results based on the validation set, RMSE, MAE, and R^2. Blue box for the baseline model, orange box for the feature selection model, and green box for the tuned model.



(a) RMSE (b) MAE (c) R^2

Figure 11: Bayesian Ridge: Results based on the validation set, RMSE, MAE, and R^2. Blue box for the baseline model, orange box for the feature selection model, and green box for the tuned model.

The first thing we notice, is that for all cases, either feature selection, or tuning, leads to similar results as the baseline model. For some cases even, the models seem to perform worse after tuning or feature selection (for example SVR has higher MAE, Figure 10b; Bayesian Ridge has smaller R^2, Figure 11c).

The scores for all models are very similar, with RMSEs between 3.5 and 4, MAEs around 2.6, and R^2 scores between 0 and 0.2. Elastic Net seems to be the only model that displays any improvement after tuning. SVR shows similar scores, while Bayesian Ridge shows worse scores after tuning.

The model I select as the **best**, considering it's speed, predictive ability and simplicity is **Elastic Net** with feature selection and tuning.