Assignment #1 - Machine Learning in Computational Biology

Spring 2025

**Asimenia Giagli 7115152300004**

**Introduction**

This assignment focuses on body mass index BMI prediction based on real datasets on gut microbiome analysis. The task is formulated as a regression machine-learning methodology, with the intention of familiarization with ML methodologies, development environment, and algorithms. All three tasks and respective subtasks have been implemented in the accompanying Jupyter notebooks and Python module. The notebooks contain markdown comments regarding the objectives, process, and results commentary, however, this report contains a more detailed explanation.

**Task 1: Setting up the environment**

An anaconda environment has been created (environment file is submitted with the assignment’s code) and the code was development using JetBrains’ PyCharm IDE. A Git/Github repository was created for backups, by forking the assignment’s Github repository. Apart from the code, the Git repository further includes the data set files, saved models, and plots/figures.

**Task 2: Dataset exploration**

The scr, data, and notebooks directories have been created as per instructions. The data\_exploration.ipynb notebook contains the loading, preprocessing, visualization, and analysis operations of the given data sets. The preprocessed data sets for the development and validation sets are saved to the following files:

./data/development\_final\_data.csv

./data/evaluation\_final\_data.csv

**Note:** When a new unseen data set is to be tested with the saved models, the corresponding CSV must be imported as the “evaluation” file by this notebook (i.e., by setting its path on the data\_file\_val variable in Cell 2) along with the original “development dataset, so that scaling and initial feature reduction to take place, before being saved as “evaluation\_final\_data.csv”.

*General information regarding the data sets and unnecessary columns*

The development set contains 489 data instances, while the validation set contains 211 instances. The data files contain 141 columns, of the above:

* One of the columns contains the BMI target values.
* A graph with numbers and a bar

  Description automatically generatedThe Project ID column contains information regarding the project under which each data instance originated. As it is shown in Figure III-1, most values correspond to two projects, so this index can be dropped without losing valuable information.

*Figure III‑1: Distribution of values of the Project ID column.*

* Disease MESH ID and Experiment type columns contain only a single value for all columns so they can be ignored.
* The Sex column contains the two values “Male” and “Female” which are converted to 0/1 in this file.
* The rest of 137 columns contain gut microbiome indices, which present the features considered for regression (alongside Sex).

*Visualization of BMI*

|  |  |
| --- | --- |
| A graph of a distribution of bmi values  Description automatically generated | A graph of a distribution of bmi values  Description automatically generated |
| *Figure III‑2: Visualization of the distribution of BMI values. (Left) The values are displayed using narrow bins for higher precision. (Right) The values are taxonomized into the categories: Underweight, Normal Weight, Overweight, Moderate Obesity, Severe Obesity, and Very Severe Obesity.* | |

A visualization of the target BMI values is given in Figure III-2. Most values are centered in the [20,35] interval, which indicate that the sample is relatively unbiased, since most instances correspond to individuals characterized as normal weight or overweight (but not obese).

*Correlations of indices*

A graph of heat map

Description automatically generated with medium confidence

*Figure III‑3: Correlation among different indices and Sex column.*

A visualization of the target BMI values is given in Figure III-3. There are sparse positive (and negative) correlations between features for 35 randomly selected columns. Since it is difficult to identify the usefulness of specific columns for the regression tasks, further feature extraction and analysis steps have been implemented, which will be described later.

*Standardisation of feature values*

Another characteristic of the features is that they follow diverse ranges of values which is typically a setback in ML algorithms. As a solution for this, each feature is *independently* standardized, so that the mean value of the feature values is equal to 0 and their standard deviation is 1 as

where is the standardised value of the -th instance of feature , is the mean of feature , and is the standard deviation of feature . Only the features of the development set are used to compute and (i.e. “fit” the scaler object), while the pre-fitted process is also applied to the validation set. The BMI index is not standarised to the purpose of keeping the original units of measuremens (kg/m2) when reporting RMSE and MAE values during evaluation.

*Correlation of features with the BMI-index*

Since correlations among features have been identified, a further investigation has been carried out to understand how each feature correlates to the target BMI values, as depicted in Figure III-4, where the indices of the development set are ordered according to their correlation coefficient with the target values. There are strong positive and negative correlations by some indices, indicating that the dataset contains useful information for BMI estimation.

A graph with blue lines

Description automatically generated

*Figure III‑4: Correlation of dataset features with the BMI target value.*

Despite simple correlation indices not being enough to fully characterize the usefulness of a feature (since they do not account for joint effects of features and only capture linear correlations), a preliminary feature selection was carried out by selecting the top 10 features with highest absolute correlation coefficients, as shown in Figure III-5. The intuition is that the selected attribute set maintains most of the predictive power of the data. The distributions of those features are given in Figure III-6. The standarisation process resulted in single-mode distributions (apart from the Sex feature) with few outliers. The correlations among them are shown in Figure III-7. As expected, there are strong correlations among them, indicating that this feature set can be further reduced. For this reason, and for consistency with the assignment, while CSV files containing only those features (and the BMI index) for the development and test set are saved, they are not used during training. Instead more advanced feature selection methods are applied, as will be discussed later.

A graph of blue rectangular bars

Description automatically generated with medium confidence

*Figure III‑5: Features with the highest absolute correlation with BMI values that are selected as an initial feature selection step.*

A diagram of a violin plot

Description automatically generated with medium confidence

*Figure III‑6: Distribution plot of selected features after standardisation.*

A graph with blue squares

Description automatically generated

*Figure III‑7: Correlations among features that exhibit the highest (positive or negative) correlations with BMI values.*

*Preprocessing of the validation set*

The same procedures of data loading, conversion of “Sex” values to numerics, removal of non-numeric types, and standardisation of features (excluding BMI) based on the pre-fitted formula have been carried out for the validation set as well. Optionally, a version with only the top 10 features is saved in parallel).

**Tsk 3: Training and evaluation of BMI predictors**

*Training*

*Considered Regression Models*

The three regression algorithms considered are the “ElasticNet”, which is a linear regression algorithm combining L1 (lasso) and L2 (ridge) regularization penalties, Support Vector Regression (SVR), which is a version of kernel-based support vector machines trained for regression, and “BayesianRidge”, which is a special case of Bayesian linear regression with Ridge regularization. From a theoretical perspective, kernel SVMs are powerful models, so SVR is expected to perform better, while BayesianRidge is expected to be better in uncertainty quantification since it inherently incorporates regularization as prior distributions, which reduce overfitting.

*Training methodology*

For each regressor, the following three models/processes were implemented:

1. A baseline model was trained on the complete development dataset (i.e., no hyper-parameter selection in the algorithm). All training is performed with the objective of root mean squared error (RMSE) minimization and 10-fold validation is applied on the development set for reporting RMSE values in all cases.
2. Feature Selection (FS) was applied to determine the lowest set of features that increase the regression efficiency. The F-test was used as a measure of quality of each feature, which is defined as:

were is the F-test value of the feature , is the number of data instances, is the true BMI value for this instance, is the predicted BMI value for this instance by training a univariate regression model using only the *j*-th feature as input, while is the mean of the target values. The features with the highest F-test values were kept, the baseline regressor was trained on them, the RMSE over 10-fold cross validation was examined. Different values (2, 3, 5, 8, 10, 15, 20, 50) were tried and the one that resulted in the lowest RMSE for each regressor was selected. Therefore three distinct “selector” objects were “fitted” and saved as part of the pipelines.

1. Parameter-tuning was then performed to the baseline model being trained on the FS-processed dataset. A grid search strategy was adopted where for each model by trying all combinations of predefined values for certain parameters. Specifically,
   * For the SVR, different values were examined for the regularization parameter (which controls the strength of regularization), epsilon parameter (defines a threshold distance on the residual prediction errors below which, no penalty is induced by the loss function), and gamma (the kernel coefficient). Before the final search, different kernels were tried, with the Radial Basis Function (RBF) kernel found to performing the best.
   * For the ElasticNet, different values were tried for the strength of the regularization and the ratio between the L1 and L2 penalties.
   * For BayesianRidge, the hyper-parameters, , and are investigated. The respective pairs of those control the gamma prior distributions on (regularization) and (inverse precision) of the Gaussian model with conjugate Gamma prior.

*Evaluation*

*Performance Metrics*

Three key metrics were used: The RMSE, the Mean Absolute Error (MAE) that gives information on the true residuals between predictions and true values, and is thus interpretable, as well as the coefficient, which is defined as

explains the proportion of the variance in the dependent variable that is predictable from the independent variables, and can therefore be used for quantification of uncertainty: Values closer to 0 indicate that the model does not explain any of the variance in the data.

*Stable and generalizable estimates*

Since the validation set is small, a bootstrap approach has been implemented to give more diverse validation scores: For a predefined number of trials, samples (with replacement) are sampled from the validation set. The metrics are computed on each individual sample and are then averaged.

*Performance of regressors*

The performance of the three baselines is given in Figure IV-1, the effects of feature selection are given in Figure IV-2, while gains obtained from parameter tuning are shown in Figure IV-3. Examining each figure independently, it can be inferred that SVR outperforms the other two regression algorithms in all stages and metrics, as previously predicted. In particular, it provides smaller RMSE and MAE, as well as higher values, that indicate that the variance is better explained by this model. The RMSE and MAE metrics correlate, as expected, so we focus on MAE performance for the rest of this analysis.

The lowest MAE, obtained by the SVR after feature selection and tuning, is of the order of 2.38 kg/m2, which is tolerable in absolute terms. As a reference, the obesity classes displayed in Figure III-2 span intervals of 5 or 10 kg/m2. The Q1-Q3 distribution ranges of the MAEs of SVR indicated by the outlines of the green box in the aforementioned figures is also characteristically lower than all other methods in all cases, indicating that for about 75% of the testing set, SVR provides lower MAE. Even the highest errors (excluding outliers) do not exceed the MAE of the second-best performing approach (BayesianRidge).

Comparing the performance gains brought by each training state of the procedure, first it needs to be addressed that each stage provides only incremental performance gains. In fact, the choice of the regression algorithm is much more important. For this case of study, the SVR baseline has a MAE of 2.44 kg/m2, which is reduced to 2.42 kg/m2 upon feature selection, while the final parameter optimization is finally reduced to 2.38 kg/m2. This 0.06 kg/m2 MAE gain is less important compared for example the 0.17 kg/m2 MAE difference between the unoptimized SVR baseline and the final BayesianRidge model (of 2.61 kg/m2 MAE).

To explain this behavior, the following arguments can be made: The kernels of the SVMs are used to identify high-dimensional spaces where the data points are better represented. This projection has the property of being relatively independent of “low-information” features by the data set, assuming reasonable kernel parameters. Regarding the minimal effect of the parameter tuning, at first, the default parameter of the sklearn models have been chosen so as to provide reasonable performance across a wide variety of datasets. The standardisation pre-processing is helpful in that goal, as multiple scaling parameters might need changes to account for wider or shifted value ranges. Finally, the grid-search approach selected in this assignment is not necessarily the most effective hyper-parameter tuning procedure. Other approaches such as Bayesian Optimization or bijective methods might prove more effective, since they employ elaborate optimization procedures to determine appropriate values from ranges of candidate parameter values (instead of selecting available ones from discrete, user-defined sets).

|  |  |  |
| --- | --- | --- |
| A chart with different colored squares  Description automatically generated | A chart with different colored squares  Description automatically generated with medium confidence | A diagram of a graph  Description automatically generated with medium confidence |
| *Figure IV‑1: RMSE (left), MAE (middle), and (right) performance of the ElasticNet (blue), BayesianRidge (orange), and SVR (green) baselines.* | | |

|  |  |  |
| --- | --- | --- |
| A chart with different colored rectangles  Description automatically generated | A graph of a graph showing different colored squares  Description automatically generated with medium confidence | A graph showing different colored squares  Description automatically generated with medium confidence |
| *Figure IV‑2: RMSE (left), MAE (middle), and (right) performance of the ElasticNet (blue), BayesianRidge (orange), and SVR (green) baselines after feature selection.* | | |

|  |  |  |
| --- | --- | --- |
| A chart with different colored squares  Description automatically generated with medium confidence | A graph of a graph showing different colored squares  Description automatically generated with medium confidence | A graph showing different colored squares  Description automatically generated with medium confidence |
| *Figure IV‑3: RMSE (left), MAE (middle), and (right) performance of the ElasticNet (blue), BayesianRidge (orange), and SVR (green) baselines after feature selection and parameter tuning.* | | |

*Evaluation on unseen data*

The best performing model (SVR with FS and tuning) has been retrained on the merged development and evaluation datasets. The pre-trained feature extraction method is embedded to it as part of the pipeline and the model is saved under ./final\_models/winner.pkl using the joblib library. The final cell of the model\_analysis nodetbook provides an example code of how to use it on unseen data. It is again noted that the unseen CSV dataset must be processed by the data\_exploration notebook for elimination of non-numeric features and standardisation based on the development set, before being processed by the loaded model/pipeline. The provided load\_data\_from\_csv() function also converts the CSV into appropriate numpy arrays.