Prediction of Age and Sex from Blood Metabolites

Project in Bioinformatics

Fall 2019/2020

Author: Tomáš Sládeček

Supervisor: Palle Villesen Fredsted

The code is available on: https://github.com/tsladecek/PiB_Blood_Metabolites

Contents

1 Introduction

Levels of blood metabolites as shown in previous studies change with age [1] and also with sex [11]. This study takes a look at a particular dataset from KarMeN (Karlsruhe Metabolomics and Nutrition) study of 441 blood metabolites with underlying information about sex, age and menopausal status for women. The KarMeN paper [11] shows a clear signal between the metabolite levels and these personal characteristics.

To model the effects of metabolites on Sex and Age, the KarMeN study used simple methods like PLS (Partial Least Squares), SVM (Support Vector Machines with linear kernel) and GLM (Generalized Linear Models). The goal of this project was to replicate the results and possibly improve them using several different machine learning approaches, that allow more hyperparameter tuning.

1.1 Data for Sex and Age Prediction

The raw data consisted of 308 individuals with 441 measured metabolite levels. For the prediction of Sex only individuals within the age range of 36-80 were selected, to ensure balanced classes (males = 99, females = 100). Two male and seven female individuals had several missing values for the metabolites and were thus removed from the dataset.

To maintain a similar ratio of males/females in Training and Test dataset a Stratified Sampling was performed, which can be conveniently accessed using the `scikit-learn` library, with 80% of the data used in training and the remaining 20% for estimating the error on unseen data.

The Stratified sampling works only with categorical data, which means that to perform it for the Age label a discretization step was necessary. The labels were split into five quantile bins (0 - 0.2, 0.2-0.4, 0.4-0.6, 0.6-0.8 and 0.8-1) on which the sampling was performed.

To account for the effects of Sex on metabolite levels, the prediction of Age was performed on male and female samples separately. The dimensions of the datasets can be seen in Table 1 together with the label distributions on Figure 1.

Table 1: *Dimensions of Data sets used for modelling*

	Sex	Age (male)	Age (female)
Train	152	136	96
Test	38	34	25

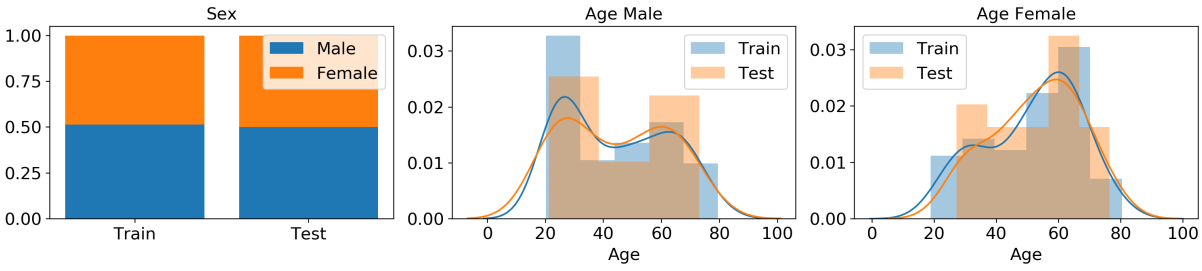

Figure 1: *Train/Test label ditributions*

1.1.1 Collinearity Issue

A frequent issue with high dimensional data is a common variation of two or more predictors, also called multi-collinearity. Collinearity imposes difficulty in distinguishing the variables truly associated with the response. To see if it is present in our dataset, we calculated the (absolute) correlation of each pair of predictors and generated a heatmap - Figure 2.

There are several regions with highly correlated variables. However, this is expected because of how the data was collected and ordered. For example, there are many types of Phosphatidylcholine/Sphingomyelin (bright area in the left bottom) or sugars and fatty acids (upper right).

One way to measure the collinearity of a system is by estimating the Variance Inflation Factor (VIF) which regresses a variable X_j onto the rest of the predictors and calculates the amount of explained variance, ie.

$$X_j \sim \beta_1 X_1 + \beta_2 X_2 + ... + \beta_{j-1} X_{j-1} + \beta_{j+1} X_{j+1} + ... + \beta_p X_p$$

VIF is then calculated as:

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}} \tag{1}$$

where $R^2_{X_j|X_{-j}}$ is the explained variance of the model above [6]. However this does not work well with high dimensional dataset, since if $p > n$ we are going to perform perfect linear fits, meaning that $R^2_{X_j|X_{-j}} = 1$. Thus techniques with strong regularization or dimensionality reduction methods are required and were used in this project.

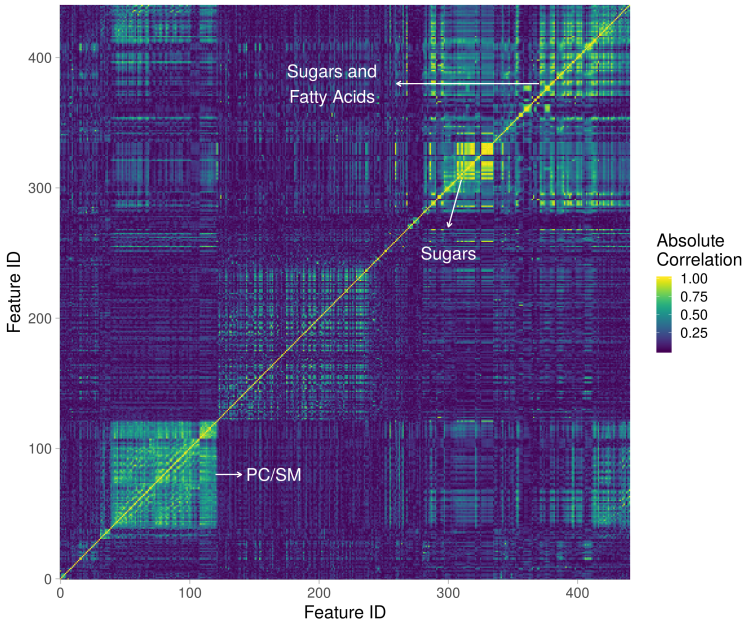

Figure 2: *Correlation heatmap. Color of a bin is correspondent to the absolute correlation between feature i and j in the dataset. All feautures were standardized.*

2 Methods

The main goal of this report was to compare the performance of several frequently used machine learning techniques for the task of classification (Sex prediction) and regression (Age prediction). The models we used can be divided into 3 following subcategories as shown in Table 2.

The "stable" category refers to models that produce the same model each time they are trained. The unstable models, such as tree-based methods, generate different structures/models due to their stochastic properties (such as randomly choosing a subset of predictors). The final category represents models that first reduce the dimensions (create linear combinations of the dimensions) and then these new directions are used to fit the model.

2.1 Preprocessing

Many of the models and techniques (KNN, PCA, LDA, etc.) are sensitive to the scales at which the features were measured. Thus it was necessary to scale them first. In practice, two techniques are used - standard normalization

$$\mathbf{x}_{Norm} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{sd(\mathbf{x})} \tag{2}$$

and "Min-Max" scaling which forces the values to pack in a certain range, usually [0, 1]

$$\mathbf{x}_{MinMax} = \frac{\mathbf{x} - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \tag{3}$$

where the bold letters represent vectors.

Since there is no direct way of telling which one to use, both methods were performed to see if there was any difference between them. The potential advantage of Min-Max scaling is the suppressing effect of outliers by decreasing the variance of the variable. However, some methods (like PCA) require the information about the variances and thus Min-Max scaling is not appropriate [10].

For each model, a hyperparameter grid search was performed and its performance was evaluated using 10-fold Cross-Validation. For the Ensemble methods (Boosting and RandomForest), an additional step of repeated cross-validation on the best 20 models was necessary to decrease the uncertainty of their performance.

Since the dataset contained more than twice the number of features than observations and high collinearity problem, dimensionality reduction methods were performed. For Classification Principal Component Analysis was performed with LDA and QDA methods creating predictions from the

Table 2: *Machine learning approaches used in this study*

	Classification	Regression
Stable	Logistic regression LDA, QDA, KNN Support Vector Machines (SVM)	Ridge, Lasso
Unstable	Tree ensemble (Boosting, RandomForest)	
Dimensionality reduction	PCA	PLS

transformed principal components, while for regression we used the method of Partial Least Squares.

It is noticeable (top row in Figure 6), that the first two principal components can separate the classes to a certain extent. This is good evidence and reason for proceeding with the modeling.

2.2 Metrics

For every machine learning problem, special care is required, especially when evaluating its performance. Sometimes having an extremely conservative model is a good idea, in other cases, the opposite might be better. Below is a summary of the metrics that were used to evaluate the performance of the classifiers and regression models.

2.2.1 Classification Metrics

The core in evaluating the performance of a classifier is a confusion matrix, which uses real and predicted values to calculate the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). An example is shown in Figure 3 where the task was to create a binary classifier for deciding whether a certain observation is male or not.

Every metric below is essentially a unique combination of these four counts and provides a different view on our problem.

- Accuracy

  Accuracy is simply the ratio of correctly classified observations, ie.

$$Accuracy = \frac{TN + TP}{TN + TP + FP + FN} \tag{4}$$

  It works well for data that have balanced classes and if having some number of false positives or false negatives is not a big issue. If that is the case, one of the following should be used.

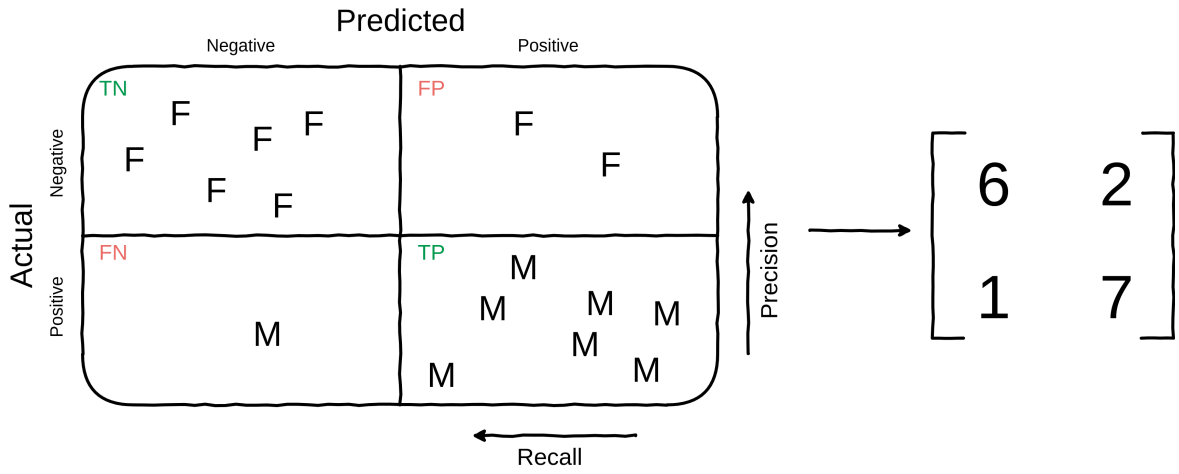

Figure 3: *Confusion Matrix sketch inspired from [4] representing a classifier for a taks of distinguishing whether an observation is a Male or not*

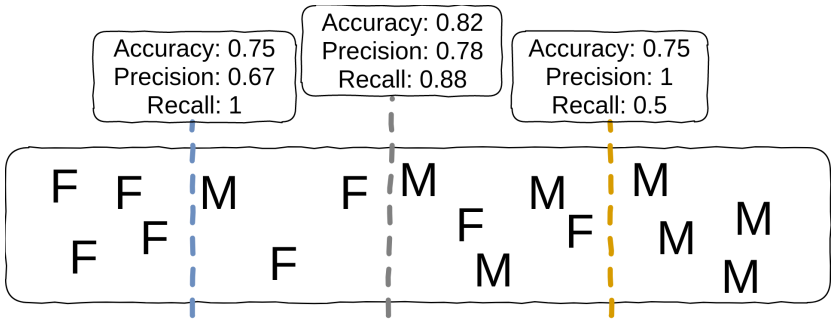

Figure 4: *Classifier thresholds depicting the precision/recall trade-off together with accuracy results.*

- Precision

  Precision measures the accuracy of the positive predictions [4]:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

  It tells us how much should we believe our classifier when it says that an object belongs to class "Male". However, precision is not very useful on its own, since we might have a very "picky" classifier that chooses to only predict one observation for the predicted class. If this prediction is correct, then this would result in perfect precision (= 1).

- Recall

  Sensitivity/Recall/True positive rate

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

  tells us about the ratio of correctly classified observations that actually belong to the tested class. Figure 3 and 4 provide some intuition behind precision and recall. To obtain perfect recall, the classifier has to assign every observation to the tested class. This would generate a large number of false positives but zero false negatives.

  There is a trade-off between precision and recall, which is not necessarily bad. For example, if we were interested in spotting shoplifters in a grocery store, we would prefer a system with high recall than high precision [4].

- F1 score

  If we are interested in a classifier that has high both precision and recall, we could train in on their harmonic mean, or F1 score

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{TP}{TP + \frac{FN+FP}{2}} \tag{7}$$

  Real-world data have some variation that can not (and should not) be explained by a model. Problems with perfectly separated classes are rare, meaning that we expect a certain proportion

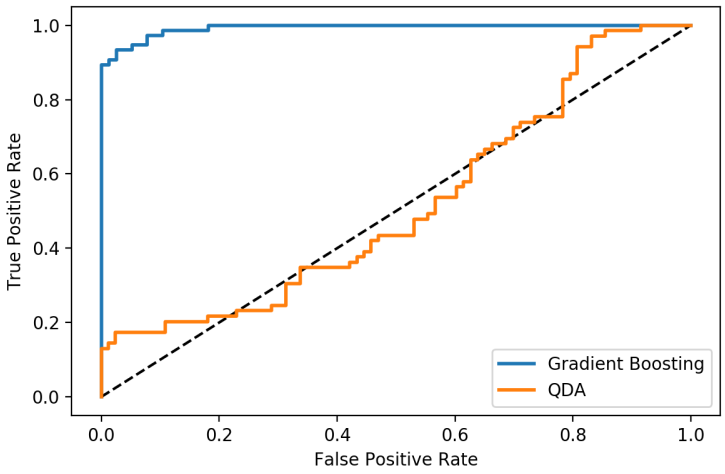

Figure 5: *ROC curve with two different classifiers. The dotted line represent the null, random classifier. The orange line is a trained QDA model and blue line is classifier based on Gradient Boosting.*

of observations to be misclassified. F1 score will be high only if both precision and recall are high and low otherwise.

- Matthews Correlation Coefficient

  An alternative to the F1 score is a Matthews Correlation Coefficient, which works well even in the imbalanced classes scenario. It can be computed from the confusion matrix, where the numerator consists of inner values of the confusion matrix, and denominator of its row and column sums [7]:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{8}$$

- ROC-AUC

  Receiver Operator Curve calculates True Positive Rate (Recall/Sensitivity - second row of the confusion matrix in Figure 3) for different values of False Positive Rate ($\frac{FP}{FP+TN}$ - first row of the confusion matrix in Figure 3). The trade-off is present again - increasing TPR also increases FPR. For a random classifier, the TPR will be roughly equal to FPR and will thus produce a linear curve. Figure 5 displays two models predicting the sex of an individual. Gradient boosting performs well, it stays on the left side, far off the diagonal. This means that the area under its curve will be high (close to 1). On the other hand, QDA does not seem to perform much better than a random classifier (dotted line) and so we expect the area under its curve to be close to 0.5.

2.2.2 Regression Metrics

In regression, the task is to find and mimick a relationship between our features and the response. This usually amounts to finding a hyperplane that is as close to the available data as possible. Similarly to

classification, different metrics can be used, each of them conveying slightly different information.

- R^2

  The explained variance tells us how much better is our model than a null expectation. The "null" of the model is usually the mean of the label and our model is competing against that. It can be calculated as the error the model makes (Residual Sum of Squares) vs the error against null (Total Sum of Squares):

  $$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \tag{9}$$

  R^2 can in theory range from $-\infty$ to 1. When it is close to zero it means that it does not perform much better than the null. If it is less than zero than there is something severely wrong with the model (eg. a line perpendicular to the least-squares line would have negative R^2).

  R^2 on its own is not very useful since it does not tell us anything about the distances of observations from predicted lines. For this purpose, the next two metrics are appropriate.

- Mean Absolute Error

  The easiest way to measure the models' performance is by finding the absolute vertical distances of the observations from the predicted (hyper)plane:

  $$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}| \tag{10}$$

  Mean Absolute Error treats every error linearly. If, however, we would like to penalize the distant observations and favor the closer ones, the mean squared error might be more useful.

- (Root) Mean Squared Error

  By taking the squares of the residuals we give more weight to observations that are far away from the prediction. However, this has an immediate side-effect that even a very good model can have high MSE when there are a few outliers.

  $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{11}$$

  To get MSE to the same scale as the true labels, we usually compute its root - RMSE.

2.3 Models and Their Parameters

Over the years many models were developed that use different ideas to perform the same task. From the most basic rigid ones (linear regression) to highly parametric models (such as gradient boosting or neural networks). Simple models are relatively easy to interpret and easy to train. However, their performance usually does not reach the performance of other models with well-picked parameters.

Such models are usually more difficult to train since one has to search through hyperparameter space with more dimensions. A summary and introduction to the models used in this study is given in the list below.

2.3.1 Regression Models (Ridge, Lasso, Logistic Regression)

Simple linear regression is a fair choice for many datasets. However, for large dimensional datasets, we would like to be able to distinguish the features that are truly associated with the response and weight them accordingly. This increases the bias of a model but the resulting model will likely perform better on real data because of the decreased variance.

For the case of linear regression:

$$\hat{y} = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

we would like the predictors ("betas") that are not associated with the response, to be removed from the model (ideally). This only increases the variance of the model, which is problematic when predicting new unseen data. We can thus create a budget (s) for the "betas" and fit the model with this new objective:

$$\underset{\beta}{\text{minimize}}\ RSS \ \text{ subject to } \sum_{j=1}^{n} |\beta_j|^r \leq s \tag{12}$$

which is equivalent to

$$\underset{\beta}{\text{minimize}}\ RSS + \alpha \sum_{j=1}^{n} |\beta_j|^r \tag{13}$$

This latter way is what is used in most modeling packages. The regularization parameter α imposes an obstacle for the model to deal with. With low α values the resulting model would not differ much from simple linear regression, while for higher alphas the coefficients of unrelated predictors will get more shrunk than the important ones. How much depends on the way we calculate the sum of coefficients in equations 12 and 13.

For the case of L1-regularization ($r = 1$), some coefficients might be removed from the final model. L2-regularization ($r = 2$) is more benevolent and usually does not set any coefficient equal to zero [6]. The former of these two options is also called LASSO (Least Absolute Shrinkage and Selection Operator) and the latter Ridge.

Logistic regression is one of the most well-known methods for binary classification. The probability of an observation belonging to one of the classes (Y) is given by the logistic function:

$$P(X_i = Y) = \frac{e^{\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}}}{1 + e^{\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}}} \tag{14}$$

Regularization in the `scikit-learn` package is done with a "C" parameter, which is an inverse to the α parameter in the Ridge regression. This means that low values of C result in strong regularization.

The advantage of logistic regression lies in its simplicity and interpretability which is an outcome of its rigidity. However, it can be very unstable for well-separated classes and is also rarely used for

multi-class prediction [6].

2.3.2 Support Vector Machines

Support Vector Machines are powerful models for both classification and regression. The basic idea is to fit an as wide "street" as possible through the parameter space while minimizing the number of margin violations (soft margin classification). The task is reversed for regression - we are trying to create a narrow "street" with as many observations on it as possible. Several hyperparameters control the flexibility and regularization of the model, which effect can be also seen in Figure 6 [4].

- **Kernel**

  Using the kernel trick for SVMs one can create margins of various shapes. The most used kernels are Linear, Polynomial, Sigmoid or Gaussian. The 3 latter ones bring more flexibility to the model and also more hyperparameters that need to be tweaked for the model to work properly. The basic idea behind the kernels is to create higher dimensional space, in which the data would be linearly separable.

- **C**

  The C parameter (in classification setting) controls the width of the street. With smaller values, the street is going to be wider and thus there will be more margin violations. This is not necessarily wrong, because a margin violation does not mean misclassification. Usually, models with lower C value will generalize better than a more variant model with a narrow street.

- $\boldsymbol{\epsilon}$

  The ϵ is only used for regression, where it controls the width of the street and the C parameter controls the number of observations outside the margin.

2.3.3 Linear and Quadratic Discriminant Analysis

Discriminant analysis methods assume each class being distributed according to some function (eg. normal/Gaussian distribution). Thus the model has to learn (for each class) the locations of these multivariate distributions $\boldsymbol{\mu} = (\boldsymbol{\mu_1} + \boldsymbol{\mu_2} + ... + \boldsymbol{\mu_k})$, where $\mu_i = (\mu_{i1} + \mu_{i2} + \mu_{ip})$, their covariance matrix/matrices $\boldsymbol{\Sigma}$ and prior probability distribution $\boldsymbol{\pi} = (\pi_1, ..., \pi_k)$ (if not given beforehand). If the desired decision boundary is supposed to be linear, than the covariance matrix is the same for all classes - Linear Discriminant Analysis. Quadratic Discriminant Analysis relaxes this condition and allows each class to have its own covariance matrix.

An observation is then assigned to a class for which

$$P(Y = i|X = x) = \frac{\pi_i f_i(x)}{\sum_{j=1}^{K} \pi_j f_j(x)} \tag{15}$$

is highest, where $f_i(x)$ is the density of the i-th distribution at location x [6].

For cases when the number of features is large (as in our case) the empirical estimation of covariance matrices is not sufficient. A regularization term can be added which results in a covariance matrix of form:

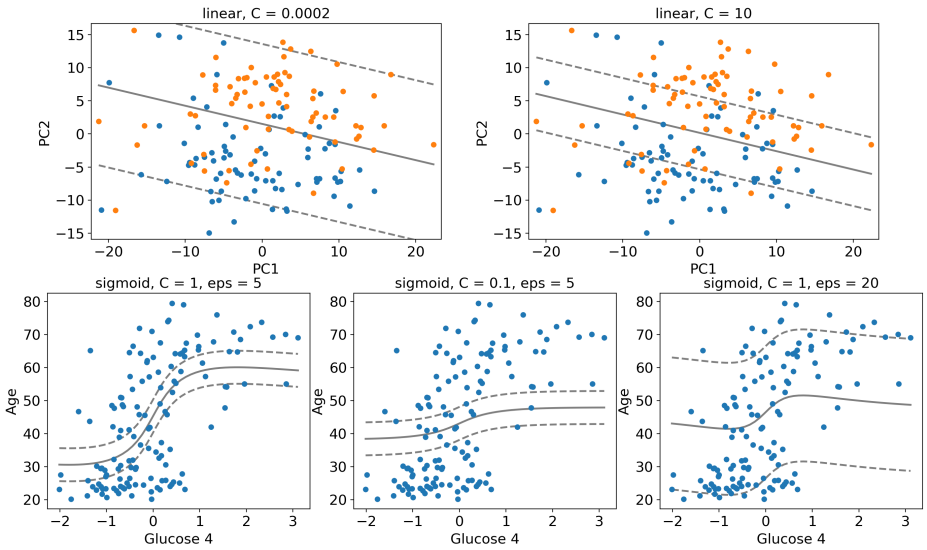

Figure 6: *Tweakable SVM hyperparameters in classification setting (upper row) or regression (lower row). For the classification first two principal components were chosen with labels representing Sex of individuals. For regression we chose one of the 441 predictors that was highly correlated with Age in males.*

$$\boldsymbol{\Sigma_s} = (1-t)\boldsymbol{\Sigma} + t\boldsymbol{I}$$

where t is the shrinkage parameter (or regularization parameter) and $\boldsymbol{I}$ is the Identity matrix.

LDA can be also used as a dimensionality reduction method. It performed very well in the Sex prediction and was able to find an axis in the space with well-separated projected classes - Figure 7

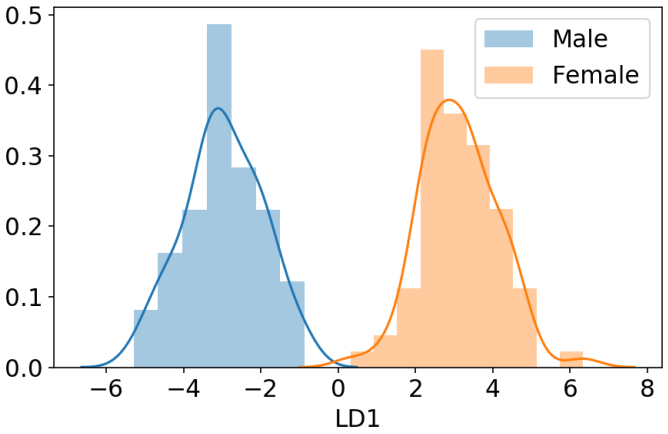

Figure 7: *Class distributions on the first LD component*

2.3.4 k-Nearest Neighbors

KNN is a simple classification method, where in every point in space the algorithm looks at k closest observations and assigns that point to the class for which the probability

$$P(Y = i|X = x) = \frac{1}{K} \sum_{i \in N} I(y_i = i) \tag{16}$$

is the highest. N in the equation above is the set of k points closest to observation x and I is the indicator function that is equal 1 only if its condition is fulfilled and 0 otherwise.

The only parameter that controls the flexibility of the model is the number of neighbors, where by increasing its value, the decision boundary becomes more rigid ($\implies$ higher bias) [5].

2.3.5 Tree Ensemble methods (Boosting, RandomForest)

Tree methods are based on splitting the predictor space into simplified regions with some predicted value. To generate a split, the algorithm looks at every predictor and finds the one that minimizes the error function the most.

An unregularized tree can overfit the data very easily by simply having each training instance represented as a leaf. One way to obtain a more robust model is to use the bootstrapping idea - generate several datasets based on a certain fraction of the original dataset (repeated sampling) and fit a tree to each dataset.

Bootstrapping can still suffer from tree correlation. If there are few strong predictors than all the bootstrapped trees will look very similar. One simple way to fix this is by reducing the predictor space at each split. This, together with the bootstrapping idea is a Random Forest.

To further constrict the model one can make the tree shallower (tree depth) or by setting the minimum number of observations in each leaf or at each split.

A different way of using trees is in sequential learning. We can fit a simple (shallow) tree to the weights associated with each observation (AdaBoost) or the residuals (Gradient Boosting, XGBoost - Extreme Gradient Boosting). In this way, the model slowly learns about the data and improves upon its predecessors. The amount by which each tree is allowed to alter the predictions is called the learning rate (λ), which is usually set to values around 0.1. The final prediction is a sum over all the possible generated trees:

$$\hat{f}(x) = \sum_{i=1}^{N} \lambda \hat{f}_i$$

where $\hat{f}_i$ is the prediction given by i-th tree.

2.3.6 Dimensionality Reduction (PCA, PLS)

We expect many of the features in a dataset to be redundant and contributing only noise to the final result. Another problem is multicollinearity which makes it hard to find the responsible variables for a given achieved result.

PCA (Principal Components Analysis) finds directions in the predictor space that satisfy two conditions:

- Point projections on a line given by this direction (eigenvector) have as high variance as possible
- The direction of the line is perpendicular to every other direction

PLS is a supervised dimensionality reduction method, that finds directions in the space that explain not only the predictor space but are also related to the response. PLS has the advantage of reducing the predictor space. However, it usually does not perform better than regularized linear models, because it increases the variance of the model [6].

3 Results and Discussion

3.1 Prediction of Sex

As already illustrated in the LDA example in Figure 7, the two Sexes have different locations of their metabolite distributions. Figure 8 illustrates the "confidence" of a subset of models in predicting the Sex of a person (more specifically it shows the probability that a certain observation belongs to the Male class). The full figure with all models can be seen in Appendix A- Figure 15.

This Figure works very well in complementary with Figure 9 which shows the performance of all the models according to the metrics explained in the Methods section. The model confidence Figure shows how certain a model is about the predictions while this information is hidden in the "metric figure". According to the different metric performances, it seems like logistic regression performs better than any other model. However, this is not that obvious in Figure 8. It is likely (as can be seen in Figure 7) that the decision boundary is linear and the classes are well separated. In these instances, logistic regression is known to produce unstable models [6]. LDA does not mind this restriction and a classifier based on it performs very well.

Regarding the different metrics, it depends on the task. For a basic classification, Accuracy, Area under ROC and F1 score work well. Since the classes we worked with were balanced, the correction by the Matthews Correlation Coefficient is not necessary. Precision and recall are useful for very specific tasks, so unless formulated, the other metrics should be sufficient.

For the tree ensemble methods, the metric results look almost the same. The differences can be seen in the full model confidence figure - Appendix A: Figure 15, where the Gradient Boosting imitated the decision boundary the best.

In terms of model choice, LDA and Logistic Regression seem to perform well and there is no need for less interpretable methods such as ensemble trees. The test accuracy of these two models was about 95% and 97%. The mean CV-Accuracy was 96% and 95% for LDA and Logistic regression respectively. The authors in the article were able to achieve 95% accuracy (possibly only CV, because there is no notion of the creation of the test dataset). Our results are not much better than theirs, at least in terms of CV performance.

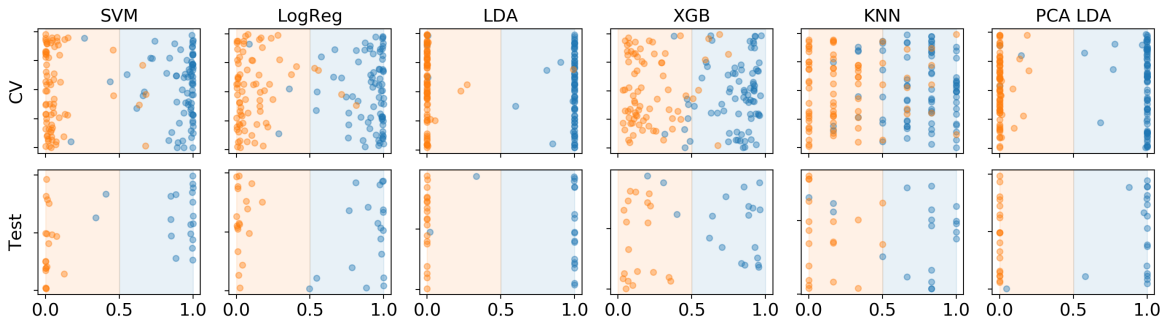

Figure 8: *Classification performance of 6 different models. Full figure with all models can be accessed in Appendix A- Figure 15. The blue regions are the true Male regions and orange are true Female regions. Dots represent individual observations and their probability of being classified as Male.*

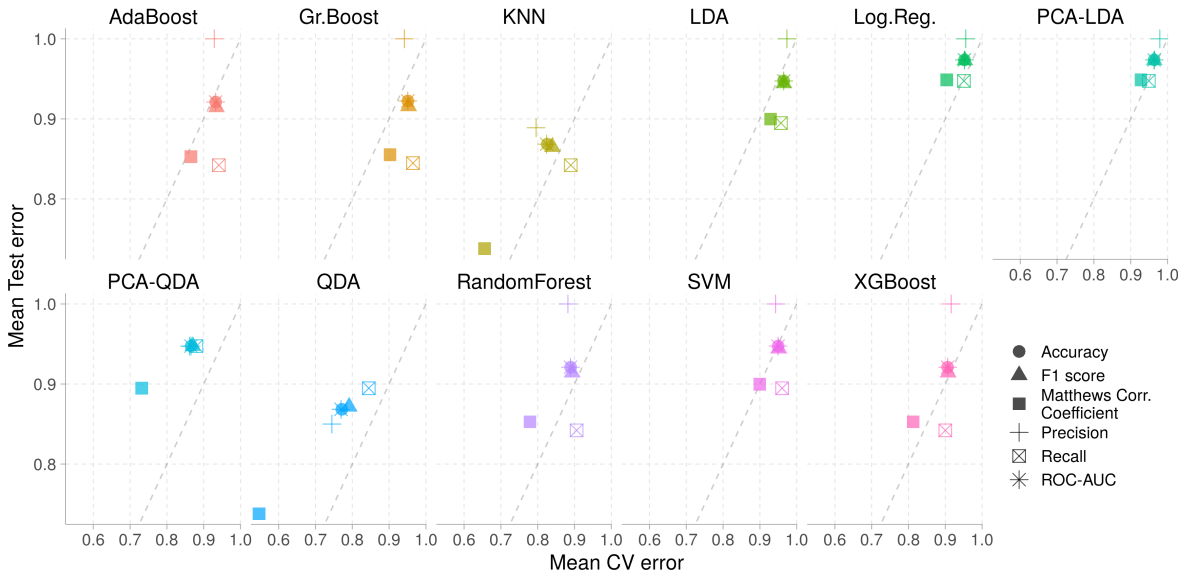

Figure 9: *Performance of the Modelling techniques according to six different classification metrics. Ideally the points would lie on the grey dotted line what would mean that the classifier performs well on cross-validated and test data.*

For all the methods (except QDA, KNN, and PCA based) it is possible to extract the impact a feature has on the response prediction, either through weights (SVM, LDA, Logistic Regression) or feature importances (ensemble tree methods). Since all of the models seek the best way to separate the two classes a good assumption is, that they should assign high absolute weights/feature importances to the same predictors. With this in mind, the predictors were ranked for each modeling technique and finally, a median of these ranks was taken. In the KarMeN article, the authors performed similar steps but with a mean instead of a median. The median has the advantage of not getting inflated by

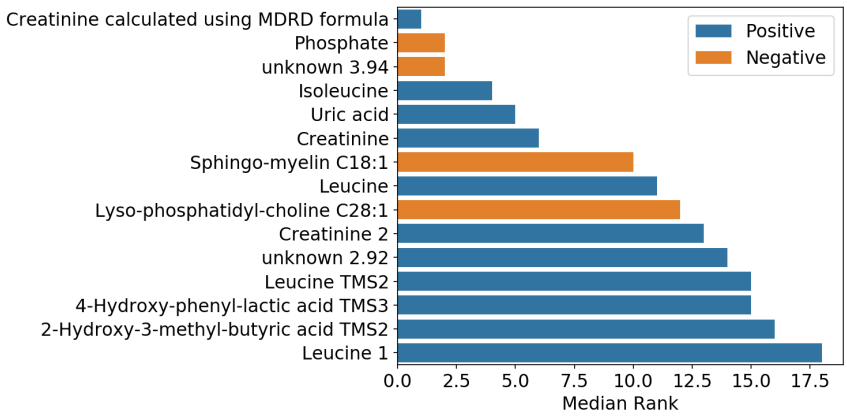

Figure 10: *Best Features for the Prediction of Sex. For every model, all the features were ranked according to their weight/importance for predicting the response. Then median rank for every feature was computed. Furthermore, the direction of association was obtained from models assigning positive and negative weights to the features - SVM, LDA, and logistic regression*

large outlier values and thus seemed like a better choice for this task.

Furthermore, the weights also carry information about the direction of the association to the response which is represented by two different fill colors.

For example, this means that Creatinine was picked by most of the models as a very good predictor and is present in higher amounts in male blood. Similarly, Phosphate seems to be also very good at discriminating between the two classes but there is more of it on average in female blood.

So, does it make any biological sense?

It is hard to tell. Most of the best predictors do not have recorded sex-dependent differences. However at least Creatinine and Uric Acid occur at different levels in males and females. Creatinine is a muscle waste product and thus is correlated with the muscle amount in a body. Since men are on average more muscular the women, it is reasonable to expect higher Creatinine levels in male bodies [14]. Uric acid (side product of purine decomposition) also has different normal blood levels in males and females [3] and thus should have a say in the classification.

The KarMeN study found almost the same set of best predictors, with Creatinine, Phosphate, Uric Acid and Sphingomyelin among the best ones.

3.2 Prediction of Age

Prediction of age is a regression task and thus requires different learning models. In addition to the techniques used for both classification and regression - such as SVMs or tree methods, we looked at

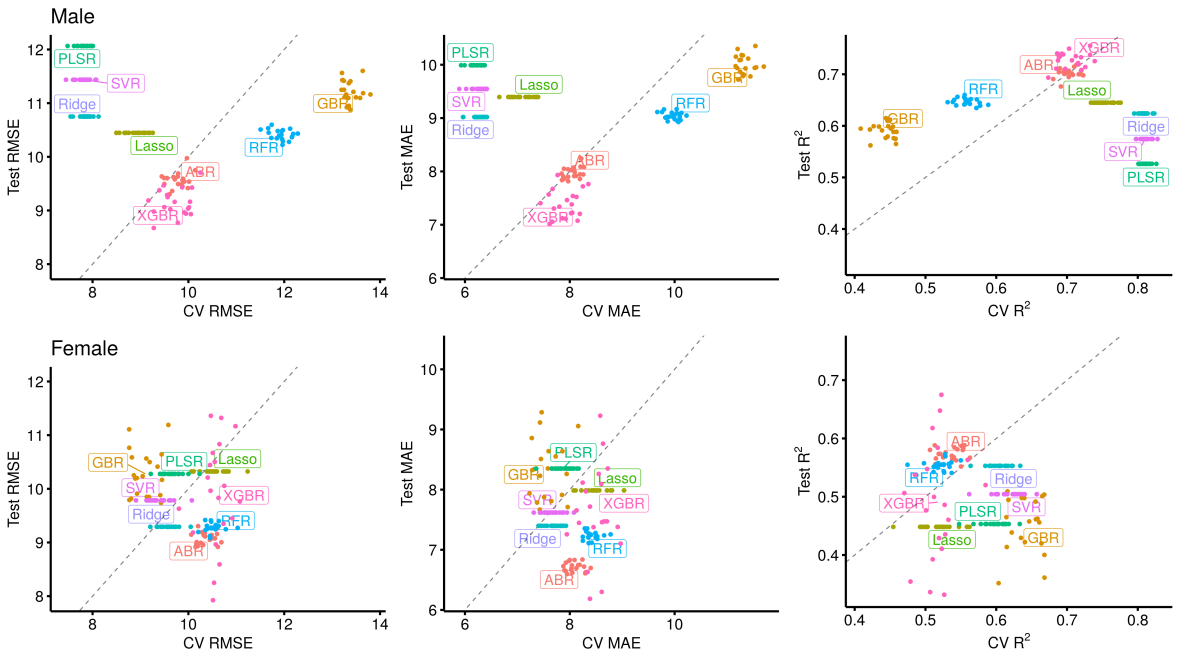

Figure 11: *Model performance according to MSE (RMSE), MAE and R^2. The upper plot represents results for Male individuals and the lower plot for Females. We performed repeated cross-validation to get an idea of how much the results change when using different observations as the training set. The tree methods produce different model every time they are fitted (even on the same data) and thus repeated measuring of test error was required.*

regularized linear models and PLS.

Similarly to the prediction of Sex we measured the performance of each model based on three different metrics- MSE, MAE and R^2. The strengths and weaknesses are presented in the Method section of this report, but in short, the R^2 tells us about the amount of variation we are capturing, MSE is the mean squared difference between real and predicted response and MAE is the absolute difference. RMSE (the root of MSE) is more influenced by outliers than MAE and is usually higher than MAE. The only case when MSE can be lower than MAE occurs when most of the residuals are in range (-1, 1).

Additionally a plot of predicted values vs. real ones is provided for both males - Figure 12 and females - Figure 13. These two figures work nicely together and show interesting facts about the models and data itself.

First, looking at the metric plot (Figure 11) the male age prediction seems to be more precise - the spread of both CV predictions and test predictions is much lower than in females (lower part of the figure). This is caused partly by the smaller female sample size but likely is also a result of hormonal fluctuations in female bodies as a consequence of the menstrual cycle. In addition, many women in the dataset were both pre and post-menopause. Thus it would make more sense to create a model for these two cases separately. The classification of females based on the postmenopausal status is possible, and we were able to reach 90% accuracy. However, this should be taken with a pinch of salt since the datasets generated for females with menopausal status classification task were even smaller than for the age prediction. But if there is ever going to be more data, it would be worthwhile building a stacked regression model, with menopausal classification at the bottom level and age regression at the top level.

The metric results for males show an interesting phenomenon: it seems like the linear models performed well on cross-validated data set but poorly on unseen test data. The tree methods (especially

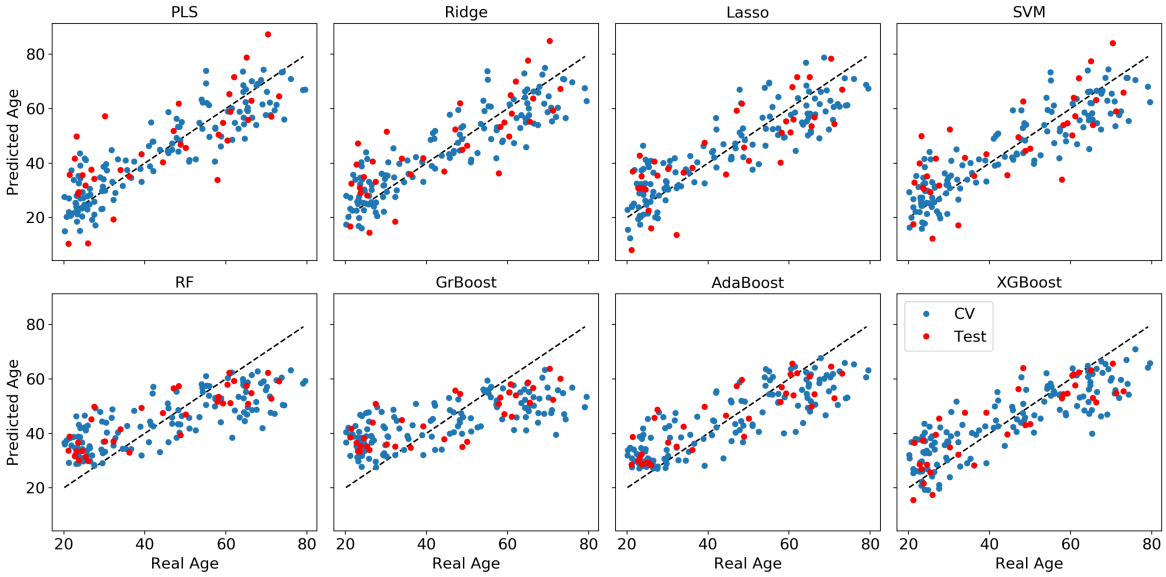

Figure 12: *Cross-validation (blue) and Test set (red) predictions for males*

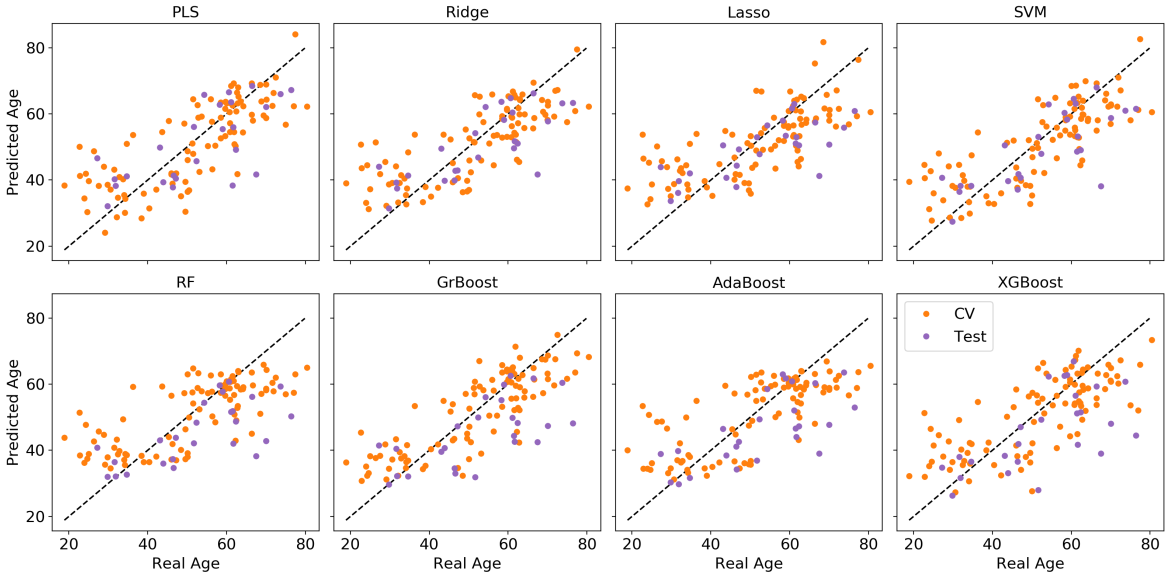

Figure 13: *Cross-validation (orange) and Test set (purple) predictions for females*

AdaBoost and XGBoost) were able to generalize much better. From this, one could conclude that the linear methods are overfitting. However, the figure depicting predicted vs. real provides more information on this problem. There, it looks like the linear methods worked well and the true relationship is linear. The residuals are more or less evenly distributed around 0 and the same holds for test data (Appendix B, C).

It looks like the tree methods were unable to capture the relationship and underfitted the data - for smaller ages the predictions are higher than they should be and for high ages, they are lower. Similarly to the Sex prediction, if only one of these plots were provided, we would have incomplete information about the performance of the models.

The data we worked with had many more predictors than observations. In this case, linear models that use all of them to fit a hyperplane through the entire space are very likely to find a model that fits the training data together with its irreducible errors. Even a strong regularization does not seem to help unless some features are removed. Lasso generalized the best of all the methods because it removed more than 90% of the features (keeping 35 out of the 441 original). The tree methods that we used were very simple - especially AdaBoost producing only trees with depth = 3 and XGBoost with depth = 2. As it turns out, splitting the space into these simple regions seems more to avoid overfitting.

The results for females are less obvious since the CV/Test errors are more spread as can be seen in Figure 11. Interestingly, for females not only tree models underfitted the training data but also regularized linear models. Decreasing α parameter might help, but can also quickly result in a model that is overfitting. Nevertheless, Ridge regression was able to generalize relatively well and also has relatively evenly distributed residuals (Figure 12, Appendix C Figure 17). And, if we do not mind slight underfitting, than AdaBoost might also be a good choice. The other boosting models have high CV and Test error variance compared to the other models. This makes it hard guarantee stable

Table 3: *Performance of the two best models for the Age prediction in Males and Females according to three measured metrics. ABR stands for AdaBoost regression and XGBR for XGBoost regression.*

	R^2	RMSE	MAE	
Male (ABR)	0.7 ± 0.002	9.57 ± 0.032	7.97 ± 0.024	Test
	0.7 ± 0.003	9.78 ± 0.043	8.05 ± 0.032	CV
Male (XGB)	0.73 ± 0.003	9.14 ± 0.057	7.38 ± 0.058	Test
	0.71 ± 0.004	9.72 ± 0.07	7.91 ± 0.055	CV
Female (ABR)	0.57 ± 0.002	9.08 ± 0.026	6.71 ± 0.017	Test
	0.54 ± 0.004	10.37 ± 0.042	8.09 ± 0.034	CV
Female (Ridge)	0.55	9.29	7.4	Test
	0.6 ± 0.003	9.58 ± 0.042	7.68 ± 0.033	CV

performance on new data.

The results in numbers can be seen in Table 3

The article reports best R^2 results of 0.7 for males and 0.6 for females. Again, these are likely just cross-validation estimates, which by looking at figure 11 should be taken with some discretion. In our case if we were only interested in CV R^2 estimate, we could pick, for example, SVMs for male and easily achieve 80%. Our best models, which were picked according to both Test and CV error show similar results as the ones in the study; for males and also females.

One theory why AdaBoost performed so well might be, that it actually sets some feature importances to zero. In this way, it is similar to the Lasso regression. The gradient boosting methods and random forests always include all predictors, even the not associated ones (like the ridge approach).

Similarly to Sex prediction we extracted the feature weights/importances from different models, ranked the and took the median of the ranks - Figure 14. Almost all best predictors were positively associated with age, both in males and females.

Some of them (such as PseudoUridine, Choline, Isocitric Acid, Ornithine, Cholesterol or Potassium) are good Age predictors no matter of Sex, but some show a strong association only in one of the two Sexes. Several sugars (known and unknown) also showed a strong association, which supports the claim of the lower glucose tolerance in older individuals [12].

The differences in Ornithine levels were noticed in skin collagen [13], and therefore there might be a correlation between it and the blood levels. The paper suggests a connection between Arginine

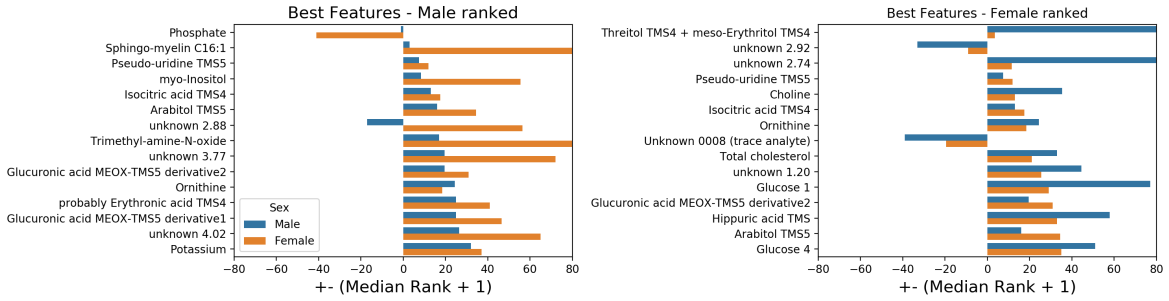

Figure 14: *Ranked features for Age prediction in Male and Female. The sign of the rank depends on the direction of association of the particular feature. Additionally each rank is increased by one, so that if a rank is 0 (such as Phosphate in males) it would be still visible in the plot*

and Ornithine levels since Ornithine is created during Arginine break down. Ornithine is also a key member in the Urea cycle together with Arginine and Citrulline [8]. However, Arginine's median rank is high- 176 and 194.5 for males and females specifically and Citrulline's too (169.5 and 172.5), which is interesting because it seems like that the levels of these variables should be connected (at least to some extent).

Cholesterol is well known for its positive association with age [2] and thus its strong influence on age in both men and women is not surprising.

Potassium levels should be also associated with age since the adequate intake is higher for males than females and is different for young and older adults [9]. Interestingly in the prediction of Sex, it was among the worst models with a median rank of 285.

Compared to the KarMeN paper (see [11]) we were able to find almost the same set of best features, which supports our conclusions.

4 Conclusion

This report summarizes and builds upon findings by a paper [11] with a similar task in mind. For both regression and classification, we used and compared several widely used machine learning techniques. We performed a hyperparameter grid-search for each of them and evaluated them using several different metrics. Finally, we were able to extract information about the strength of the association of each feature to the selected target label. We found several metabolites with a strong association with the response which were in most cases also found in the KarMeN study.The performance of our best models did not exceed the performance of the models in the study.

For the prediction of Sex, the Linear Discriminant Analysis together with Logistic Regression performed well and should also generalize well on new data. For age prediction in males, the boosting methods performed the best while the Ridge regression and AdaBoost seem like the most reasonable choices for females (without the menopausal status correction).

In terms of scaling, we noticed a difference between the two methods and the tasks. For classification, most Machine Learning methods favored the Min-Max scaling, while for regression the Standard Normalization was preffered.

References

[1] Romanas Chaleckis et al. "Individual variability in human blood metabolites identifies age-related differences". In: *PNAS* 113.16 (2012), pp. 4252–4259. DOI: www.pnas.org/cgi/doi/10.1073/pnas.1603023113.

[2] Jenna Fletcher. *What should my cholesterol level be at my age?* 2017. URL: https://www.medicalnewstoday.com/articles/315900.php#levels-and-age.

[3] Amber Erickson Gabbey and Rachel Nall. *Uric Acid Test (Blood Analysis)*. 2017. URL: https://www.healthline.com/health/uric-acid-blood.

[4] Aurelien Gerón. *Hands on Machine Learning with Scikit-learn and Tensorflow*. O'Reilly, 2019. ISBN: 978-1-492-03264-9.

[5] George Ho. *Linear Discriminant Analysis for Starters*. 2017. URL: https://eigenfoo.xyz/lda/.

[6] Gareth James et al. *An Introduction to Statistical Learning*. Springer, 2013. ISBN: 978-1-4614-7137-0. DOI: 10.1007/978-1-4614-7138-7.

[7] David Lettier. *You need to know about Matthews Correlation Coefficient*. 2017. URL: https://lettier.github.io/posts/2016-08-05-matthews-correlation-coefficient.html.

[8] PubChem. *L-Ornithine*. URL: https://pubchem.ncbi.nlm.nih.gov/compound/L-Ornithine.

[9] Harvard School of Public Health. *What should my cholesterol level be at my age?* 2017. URL: https://www.medicalnewstoday.com/articles/315900.php#levels-and-age.

[10] Sebastian Raschka. *About Feature Scaling and Normalization – and the effect of standardization for machine learning algorithms*. July 2014. URL: https://sebastianraschka.com/Articles/2014_about_feature_scaling.html.

[11] Manuela J. Rist et al. "Metabolite patterns predicting sex and age in participants of the Karlsruhe Metabolomics and Nutrition (KarMeN) study". In: *PLoS ONE* 12(8) (2017). DOI: https://doi.org/10.1371/journal.pone.0183228.

[12] David R. Sell, Nannette R. Kleinman, and Vincent M. Monnier. "Longitudinal determination of skin collagen glycation and glycoxidation rates predicts early death in C57BL/6NNIA mice". In: *FASEB* 14.1 (2000). DOI: https://doi.org/10.1096/fasebj.14.1.145.

[13] David R. Sell and Vincent M. Monnier. "Ornithine Is a Novel Amino Acid and a Marker of Arginine Damage by Oxoaldehydes in Senescent Proteins". In: *Annals of the New York Academy of Sciences* 1043.1 (2007), pp. 118–128.

[14] *What Is Creatinine?* URL: https://www.davita.com/education/kidney-disease/symptoms/what-is-creatinine.

5 Appendix

5.1 Appendix A: Sex prediction - Model Confidence

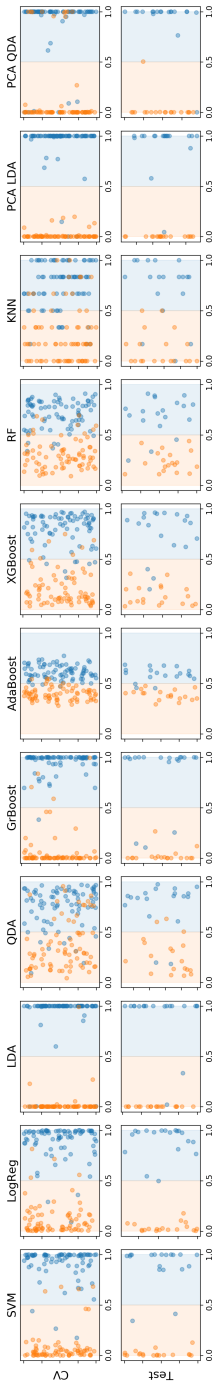

Figure 15: *Model Confidences. True male regions are filled with blue color while true female regions with orange. The position of the points on y-axis is uniformly distributed to minimize the overlapping of points.*

5.2 Appendix B: Residuals Male

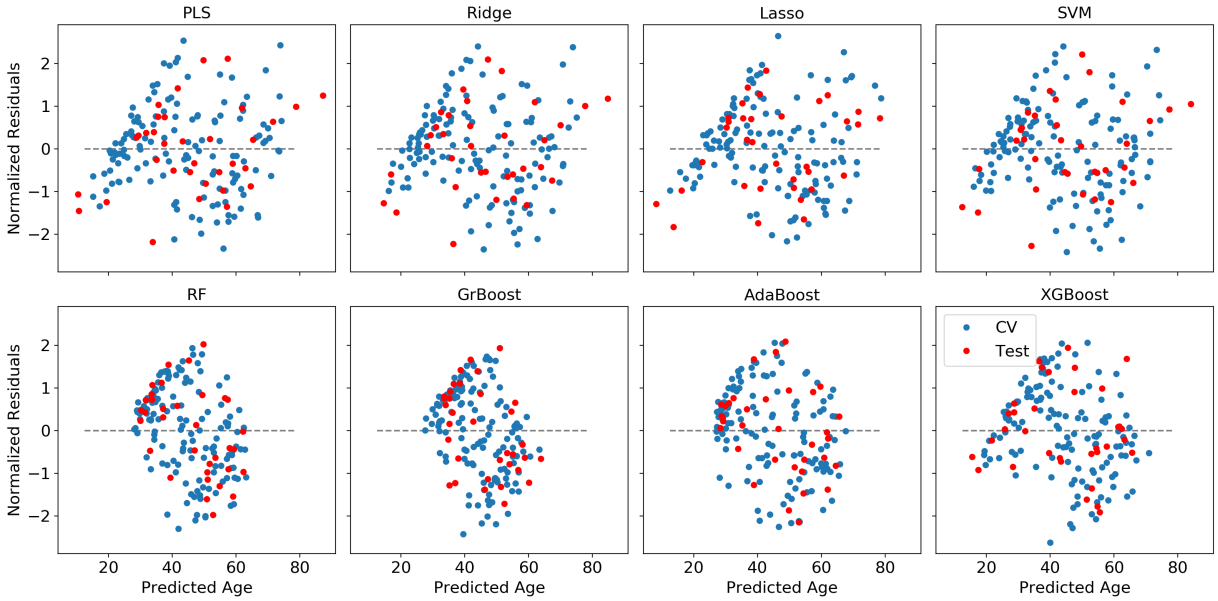

Figure 16: *Residual Plots for different models for prediction of Age in Male individuals*

5.3 Appendix C: Residuals Female

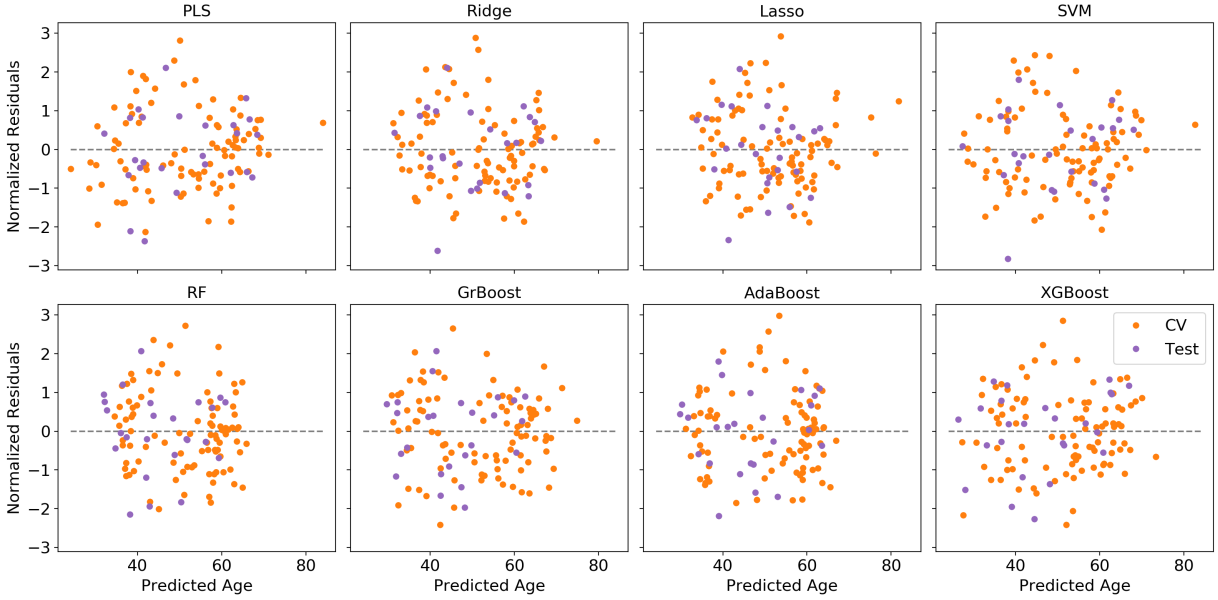

Figure 17: *Residual Plots for different models for prediction of Age in Female individuals*