RESEARCH

Project-1

Sujan Darai (sujad96@zedat.fu-berlin.de), Matanat Mammadli (matanam94@zedat.fu-berlin.de), Samra Hamidovic (samrah96@zedat.fu-berlin.de) and Ibraimi Ibraim (ibraimii@hu-berlin.de)

Full list of author information is available at the end of the article

Abstract

Goal of the project: Analysis and result replication of eight different machine learning algorithms on metabolomics data.
Main results of the project: Comparison and evaluation of machine learning algorithms based on their performances on the same dataset.
Personal key learning: Creating a virtual environment and installing the dependencies required to run the code.
Estimated working hours: 10 hours.
Project evaluation: .....
No. of words: About 1100 words.

Keywords: Metabolomics, Principle component regression, Support vector machine

1 Introduction

Metabolomics is one of the many omics approaches which are used in the next-generation sequencing era in view of dissecting the molecular mechanisms behind gene, protein or metabolite function and their implication in disease. While until very recently modern biology and data science in general were relying on statistcs aming at drawing conclusions about a population based on principles of inference and model building by knowing the model in advance, modern data science and bioinformatics are shifting towards approaches which have to account for the huge amount of data produced through omics and other data collection procedures. More precisely, machine learning strategies are based on the idea of building a model only after learning from data by using computational methods. Until a short while ago, Partial least squares (PLS) machine learning algorithms were used in metabolomics due to their capacity to handle high-dimensional information and give interpretable information. Due to the nonlinear structures in metabolomics data, advanced nonlinear machine learning strategies like Random forest (RF), Support vector machines (SVM), and Artificial neural networks (ANNs) are picking up considerable attention. Despite their capacity, these models have been neglected in metabolomics due to computational complexities and cost reasons. This paper compares the predictive performances of different machine learning algorithms on metabolomics datasets. [2].

2 Goal of the project

The primary focus of this project is to run the code and compare the predictive performances of the 8 different machine learning algorithms on one data set. It also includes the optimization of the code if necessary and extracting the important features of the machine learning models. The goal also includes the reproducibility of the performance of the model with previously studied papers.

3 Data and preprocessings

We have chosen the MTBLS136 dataset which consists of a serum LC-MS datasets. These datasets consist of 949 named metabolites associated with postmenopausal hormone use. Multiple metabolites were compared in the group of women including 332 estrogen users and 337 estrogen plus progestin users versus 667 non-users. In this research, metabolite levels were compared between estrogen-only and estrogen-plus progestin users. This means it consists of the metabolite concentration of users that is supposed to be due to estrogen-only versus estrogen-plus-progestin hormones [3].

In data preprocessing, all the required packages were imported and the dataset was loaded in the jupyter notebook. The data is then divided into two tables, namely: DataTables and PeakTables using the function $cb.utils.load\_dataXL$. The PeakTable consists of null values more than 20 percent removed using the $perc\_missing$ column. The DataTable2 was created using DataTable having class values 0 and 1 only. The binary class column was then assigned to an outcome vector Y. The data is then split into train and test datasets using $train-test-split$ from DataTable2 and Y. Xtrain was introduced by taking the names of metabolites from PeakList and then converted into logarithmic value and then scaled using function $cb.utils.scale$. The missing values were then imputed using the K-Nearest Neighbor technique with the k=3 neighbors. The XTestKnn was produced using Datatest similar to the XTrainKnn dataset.

4 Methods

The machine learning algorithms used in this project are described below.

4.1 Principle component regression

Principle component regression (PCR) is the combination of two machine learning algorithms, principle component analysis (PCA) and multiple linear regression (MLR). In PCA, the X matrix is rotated and projected into lower dimensional space depending upon the orthogonal covariance such that the principle components describe the direction of maximal variance in X where the principle component acts as independent variables. The directins are carried by the corresponding eigenvectors ofthe covariance matrix, while the variance is carried by the corresponding eigenvalues. In PCR, coefficients from PCA and MLR are calculated separately, combined and reduced to linear regression for the prediction analysis. The only single hyper-parameter in this algorithm is the number of principle components used [2].

4.2 Linear kernel support vector machine

The purpose of a Linear kernel support vector machine is to find out the best hyperplane in a multi-dimensional space from the set of hyperplanes to separate N data points of a given matrix. The hyperplane is chosen such that maximizes the margin of discrimination. It has a single tuning hyper-parameter called C (”Cost” in Python library) [2].

4.3 Hyper-parameter optimization and cross-validation

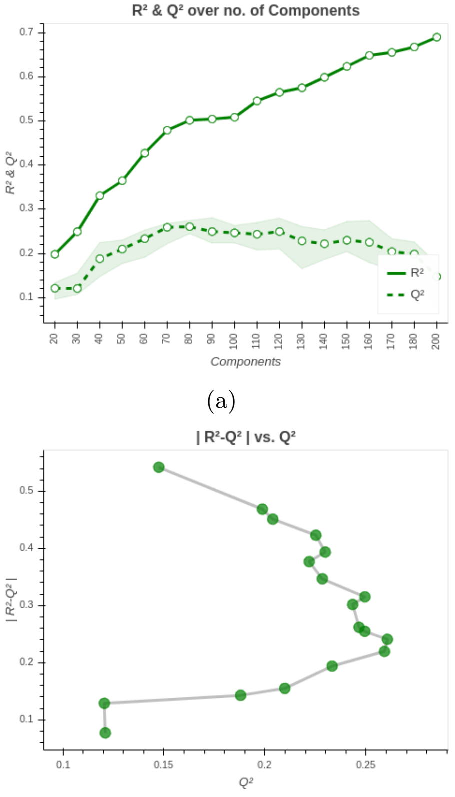
Each model was optimized differently depending upon model type i.e. linear search for single and grid search for double hyper-parameters. The model is then fivefold cross-validated using 10 Monte Carlo repartitions and plots of $|R^2-Q^2|$ vs. Q^2 were generated to measure the optimized hyper-parameter values (R2 is the coefficient of determination for the full data set, and Q2 is the mean coefficient of determination for cross-validated pre-diction data across the 10 MC repartitions[1].

4.4 Evaluation

The model was trained with the training dataset using an optimized hyper-parameter and then tested with the test dataset to measure the test performance of the model by taking the AUC values. However, the models are still prone to bias in performance due to the sampling bias. For that reason, bootstrap evaluation is done in both training and test data to measure the confidence interval in the uncertainty of prediction [2].

5 Results and Discussions

We have used PCR and SVM-lin methods only due to word limitation.

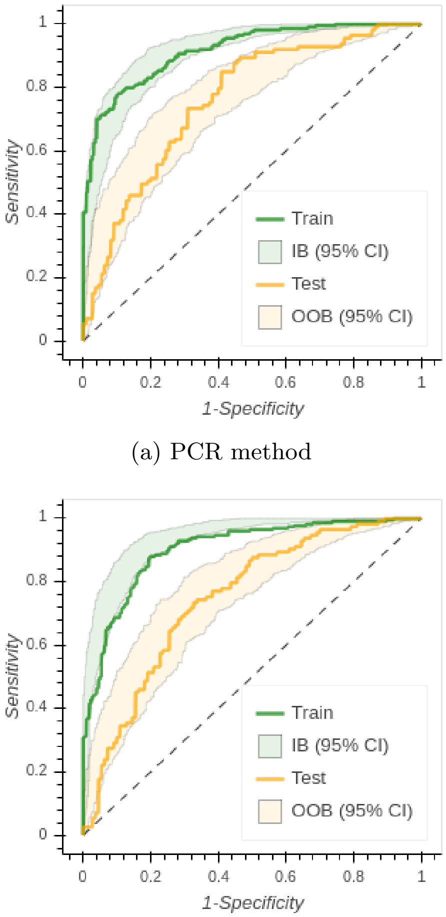


(a)

(b)

Figure 1: Optimization of hyper-parameter in PCR. a) $|R^2 - Q^2|$ vs. Q^2 curve b) R^2 and Q^2 vs. no. of components curve

From the figure 1, curve $|R^2 - Q^2|$ vs. Q^2 is used to measure the optimized hyperparameters by taking the point of inflection of the outer convex hull. From the figure, the optimal number of components where the curve of R^2 and Q^2 changes the shape is around 80.



(a) PCR method

(b) SVM-Lin method

Figure 2: Receiver Operator Characteristic (ROC) curve of MTBLS136 dataset using the a)PCR model and b)SVM-lin model. The green color indicates the in-bag (IB) 95 % confidence interval and yellow color indicates the out-of-bag (OOB)95 % confidence interval

Figure 2 illustrates the full picture of ROC curves for the dataset of MTBLS136. It is also clearly seen that the sensitivity of the training dataset for both of the models is higher than the test dataset. But this is to be expected. The ROC curves are mainly used to dmosntarte the performance of the model on the test sets. Even though the models are optimized using the K-fold cross-validation, the models still suffer from over-training and are proportional to the complexity of the model. The accuracy of the PCR model using the AUC value for the train and test datasets was found to be 0.92 and 0.76 which are almost identical. For the SVM-lin model, the AUC scores for the train and test datasets were 0.9 and 0.75 respectively. The higher the coefficients are, the more the implications of the metabolite. The top 3 important features of PCR and SVM-lin are shown in Table 1.

Number	**PCR**	**SVM-lin**
1	cystine	1-linoleoyl-GPA (18:2)*
2	isobutyrylcarnitine (C4)	lysine
3	gluconate	2-aminoheptanoate

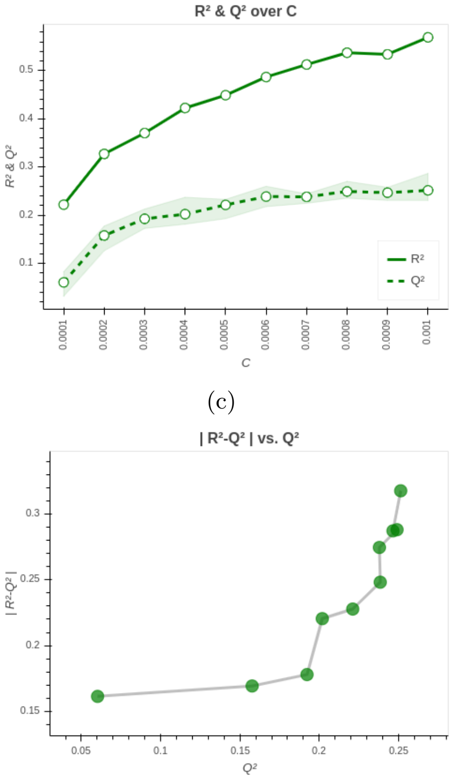
Table 1: Important features i.e. metabolites

The predicted metabolites given the machine learning based binary classification turned out to be important ones because showed to have the highest coefficients and thus make a significant impact on estrogen or estrogen plus progestin hormone function pathways. Cystine is responsible for the synthesis of glutathione and is closely related to kidney stones, neurodegenerative diseases, and cancers and is the result of uptake of postmenopausal hormone use. Inflammation and immune regulation, cardiovascular disease, and neurological disorders are associated with the level of 1-linoleoyl-GPA (18:2)* metabolites. However, there is still a big gap in information about the mechanisms of those metabolites and their relations with disease treatment. both of the models have moderate performance in prediction in metabolomic datasets.

6 Contributions

Sujan Darai: Running codes and writing report.
Ibraim Ibraimi: writing and reviewing the report.
Matanat Mammadli: writing report.
Samra Hamidovic: writing report.

7 Appendix



(c)

(d)

Figure 3: Optimization of hyper-parameter in SVM-lin method. a) $|R^2 - Q^2|$ vs. Q^2 curve b) R^2 and Q^2 vs. Cost (C) curve

Author details

References

1. Kevin MMendez. CIMCB/MetabComparisonBinaryML. https://github.com/CIMCB/MetabComparisonBinaryML. [Accessed: April 24, 2024].
2. Kevin M Mendez, Stacey N Reinke, and David I Broadhurst. A comparative evaluation of the generalized predictive ability of eight machine learning algorithms across ten clinical metabolomics data sets for binary classification. *Metabolomics*, 15:1–15, 2019.
3. Victoria L Stevens, Ying Wang, Brian D Carter, Mia M Gaudet, and Susan M Gapstur. Serum metabolomic profiles associated with postmenopausal hormone use. *Metabolomics*, 14:1–14, 2018.