THE CHINESE UNIVERSITY of HONG KONG, SHENZHEN

STA 4606 / MBI6006

MACHINE LEARNING FOR BIOMEDICAL RESEARCH

Project Report

Joseph Ariel Christopher Teja

120040002

December 28, 2023

1. Introduction

Robust prognostic techniques are urgently needed to detect people who may acquire severe cases of COVID-19 as a result of the worldwide pandemic. Improving patient outcomes requires prompt interventions and efficient use of medical resources. Our work explores the complex topography of the human metabolome to find possible biomarkers linked to the severity of COVID-19 infection in answer to this demand. The objective is to construct a predictive model that can accurately classify disease severity based on metabolites, facilitating the early identification of patients requiring immediate attention. Through analyzing the predictive model, we will also be able to gain insight into the influential biomarkers that affect COVID-19 severity, which will allow us to more efficiently and cost-effectively conduct testing to determine hospital resource allocation

Our dataset consists of metabolic features measured by untargeted metabolomics by liquid chromatography-mass spectrometry. This approach measures small molecules in a system in an unbiased manner (Ma et al., 2023). However, a main obstacle in doing LC-MS metabolomics analysis is the matching between measured features and known metabolites, as unlike gene expression data, which is measured by deep sequencing, LC-MS lacks direct chemical identity information. Instead, potential matches between features and known metabolites will be primarily based on the features' mass-to-charge ratio (m/r) and retention time (RT) which will often result in multiple potentially matched metabolites for one feature and vice versa.

Our analysis explores dimension reduction techniques, overall data analysis, and several machine learning methodologies for the classification of disease severity based on human metabolic features. Throughout, we maintain a keen focus on the existence or nonexistence of batch effects and their role in influencing metabolic profiles as the dataset we is collected on different batches.

The urgency of this research is underscored by the need to develop a predictive model that not only stratifies patients based on disease severity but also informs resource allocation strategies. By finding metabolomic markers linked to severity, our work adds to the increasing body of information about COVID-19 and provides a possible tool for risk assessment and resource optimization during the continuing epidemic.

2. Materials

The dataset under investigation originates from a study aimed at developing prognostic tools for predicting the severity of COVID-19 infection. The dataset, accessible through the Metabolomics Workbench, is part of study ST001849, which aims to analyze the human metabolites in association with the severity of COVID-19 infection. The dataset consists of metabolic features from 339 COVID patients, which was collected through untargeted metabolomics from plasma, with the samples being collected at six longitudinal points. Data from this study was preprocessed beforehand and formatted into three CSV files: "data.csv," "Y.csv," and "feature_meta_matching.csv.” data.csv contains the measurement of the metabolites, where each row is a metabolic feature, and each column is a sample. Y.csv contains batch information, the outcome variable,icu, which is about whether the patient ended up in the ICU or not, information about infection status, and the day the sample was taken from the patient. To limit the scope of this study, we will only be analyzing the data of infected patients whose sample was taken on day 0. Lastly, feature_meta_matching.csv contains information regarding the biological annotation of each metabolic feature.

Initially, there were 700 samples in our dataset. After filtering for the samples of infected patients and those that were taken on day 0, we were left with 243 samples. Each sample has 1334 metabolite measurements.

3. Methods

3.1 Dimensionality Reduction

Due to the large dimension of our dataset, in order to ease data analysis, we conducted dimensionality reduction on our dataset. We utilized two methods of reducing data dimensions, PCA and T-SNE. Conducting dimensionality reduction as it allows us to quickly get an overall picture of our dataset without needing to analyze each metabolic feature individually.

3.1.1 Principal Component Analysis (PCA)

PCA is a widely used statistical technique for dimensionality reduction. It works by transforming a high-dimensional dataset into a lower-dimensional space while preserving the maximum amount of variance in the data (Shlens, 2014). PCA achieves this by identifying the principal components, which are linear combinations of the original features. These components are orthogonal to each other and ordered by the amount of variance they explain in the data. By selecting a subset of the principal components, we can effectively reduce the dimensionality of the dataset while retaining the most essential information. PCA is beneficial for identifying the main patterns and relationships in the data. The specific algorithm is as follows:

1. Standardization
2. Covariance matrix computation
3. Computation of the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Select principal components
5. Data transformation in new dimensional space

3.1.2 t-Distributed Stochastic Neighbor Embedding (T-SNE)

T-SNE is a nonlinear dimensionality reduction technique that focuses on preserving the local structure and clustering of the data points (Van der Maaten & Hinton, 2008). It is especially effective for visualizing high-dimensional data in a lower-dimensional space. T-SNE works by constructing a probability distribution over pairs of high-dimensional data points, where the similarity between points is measured by their proximity. It then constructs a similar probability distribution in the lower-dimensional space, minimizing the divergence between the two distributions. By doing so, T-SNE maps the high-dimensional data points into a lower-dimensional space, emphasizing the similarities between nearby points and preserving the local structure of the data. This allows T-SNE to reveal inherent clusters or groups within the dataset.

3.2 Batch Effect Analysis

We also conducted batch effect analysis to test whether batch plays an important role in our study and whether we should take it into consideration. To analyze the batch effect, we implemented several different methods. First, we analyzed the scatter of the principal components to see whether samples of different batches mix with each other or they are separated according to the batches. Second, we observed the distribution of several individual features, particularly those which have the most different statistical properties across batches, measured through the variance. Third, we conducted a chi-squared test to test whether there is a statistically significant association between batches and the icu variable. Lastly, we observed the distribution of each principal component across batches to see whether there are notable discrepancies across batches.

3.3 Classification Algorithms

Lastly, we used three classification models, which we trained to predict the severity of the COVID case based on the metabolic features and deduce which features most indicate COVID severity. We chose Random Forest Classifier, XGBoost Classifier, and partial least squares (PLS) regression models.

3.3.1 Random Forest Classifier:

Random Forest Classifier is a popular ensemble learning algorithm that combines multiple decision trees to make predictions. It works by constructing a multitude of decision trees and aggregating their predictions to determine the final classification (Breiman,2001). Each decision tree in the random forest is trained on a random subset of the data and a random subset of features. During the training process, the trees learn to split the data based on different features, resulting in a diverse set of classifiers. The final prediction is made by majority voting or averaging the predictions of individual trees. Random Forest Classifier is known for its ability to handle high-dimensional data, handle missing values, and reduce overfitting.

3.3.2 XGBoost Classifier:

XGBoost (Extreme Gradient Boosting) Classifier is an optimized implementation of gradient boosting, a powerful machine learning algorithm. It is designed to create a strong predictive model by combining weak models, typically decision trees, in an additive manner. XGBoost builds the model in a stage-wise fashion, where each new tree is trained to correct the mistakes made by previous trees (Chen & Guestrin, 2016). It uses a gradient-based optimization approach to find the optimal weights for each tree. XGBoost incorporates regularization techniques to prevent overfitting and provides hyperparameter tuning options to optimize the model's performance .

3.3.3 Partial Least Squares (PLS) Regression:

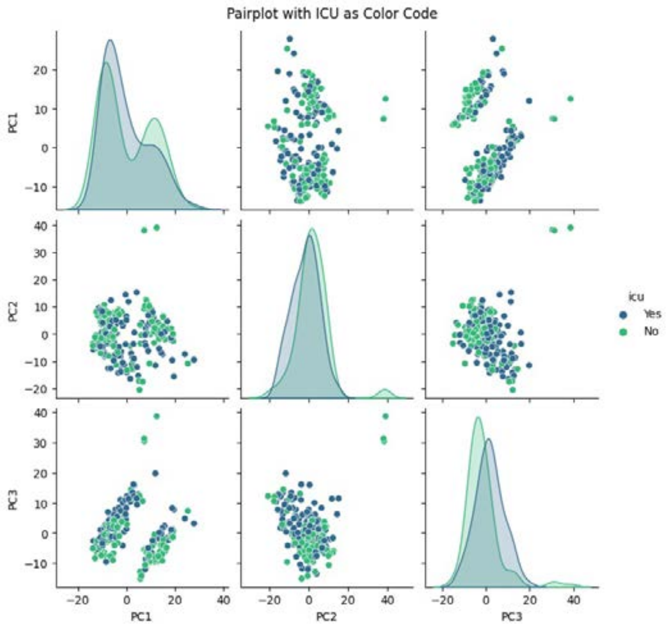
Partial Least Squares (PLS) Regression is a statistical method used to model the relationship between a set of independent and dependent variables. It is particularly useful when dealing with datasets with many predictors and potential collinearity among them. It works by extracting latent variables, known as components, from the independent variables that explain the maximum covariance with the dependent variable (Geladi & Kowalski, 1986). These components are obtained through a process of iterative linear regression, where each component is a linear combination of the original predictors. PLS Regression aims to find a reduced set of components that capture the most relevant information in the data while minimizing the prediction error.

4. Results

4.1 Data Analysis on Dimensionally Reduced Data

4.1.1 PCA Results

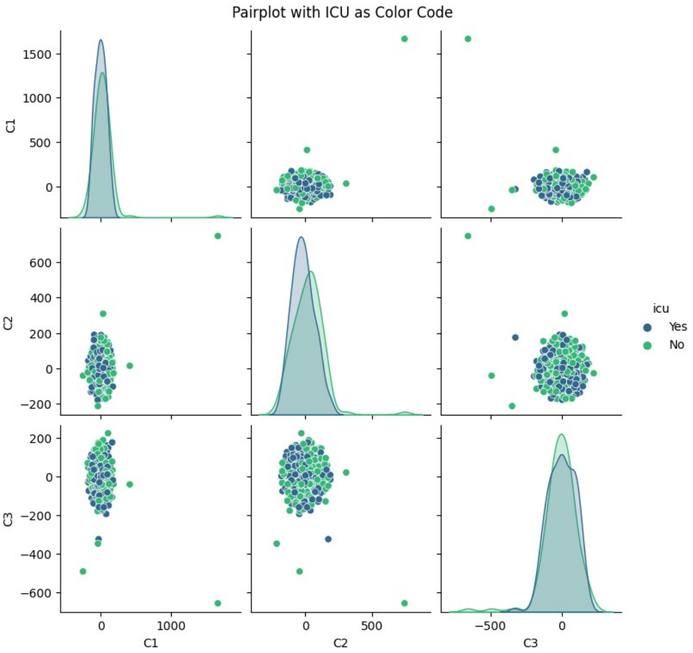
For ease of visualization, we only selected 3 principal components. However, the three selected components only managed to explain around 16.22% of the variance. PC1 explained around 7.4% of the variance, PC2 explained around 4.7% of the variance, while PC3 explained around 4% of the variance.



As can be observed in the plot above, there does not seem to be any obvious separation between people who went to the ICU and those who did not. However, we can see that there exists segregation by some other factors. From this graph we can conclude that, at least based on the principal components, there does not seem to be any obvious separation between the different classes.

4.1.2 t-SNE Results

As PCA only managed to capture around 16% of the variance we explored dimensionality reduction using another method, we chose t-SNE. As with PCA, we selected three components for ease of visualization.

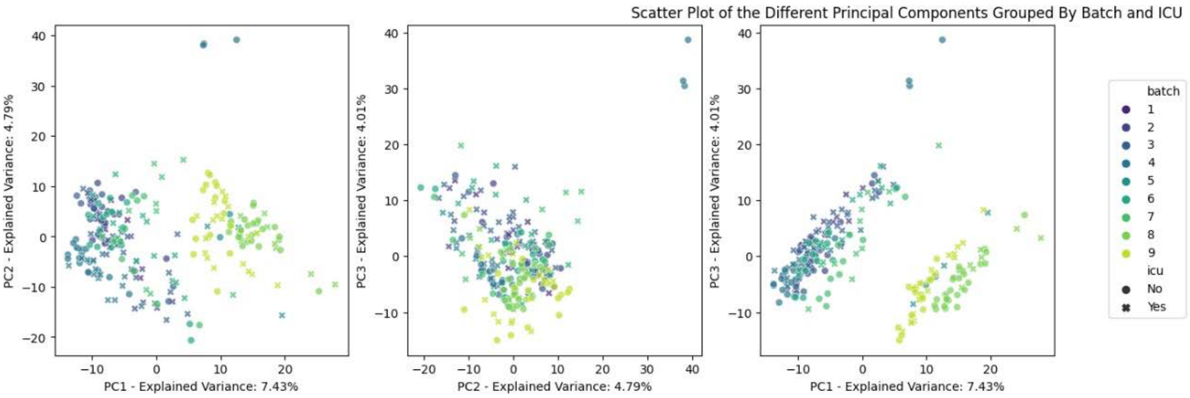


As with the PCA results, we are unable to find obvious separation based on the t-sne results alone. The data seems to be well-mixed.

4.2 Batch Effect Analysis

As the dataset was collected in batches, we want to check whether there are significant discrepancies between the batches and whether it has an effect that might skew the results of our predictions later on.

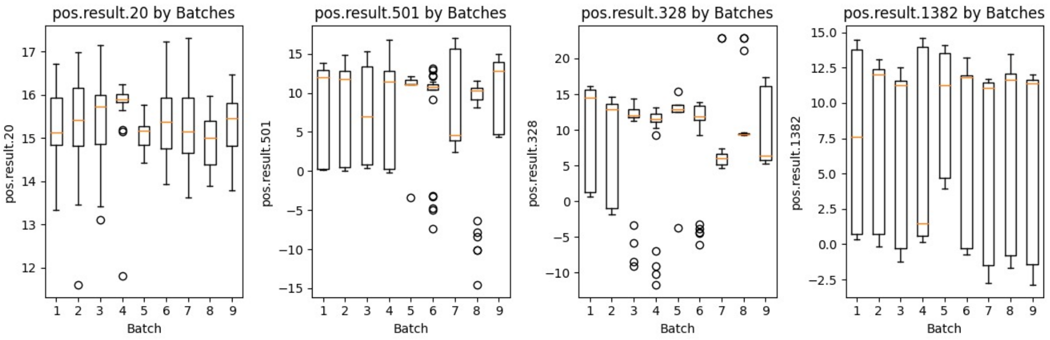
4.2.1 Scatter Plot on Principal Components



As can be seen from the scatter plots above, we can observe sample differences due to different batches from the scatterplot above. In particular, we can observe it easily in the first PC. In the first PC, we can see that batches 7-9 separates itself from most of the other batches. This supports the argument that batch effects exist in this dataset.

4.2.2 Box Plots of Several Individual Metabolic Features

To see whether batch effects exist, we also picked out several features which have the most variance across batches (pos.result.501, pos.result.328, pos.result.1382) as well as one random feature (pos.result.20). If there exist noticeably significant differences between the statistical distributions of these features across batches, it further supports the argument that batch effect exists.



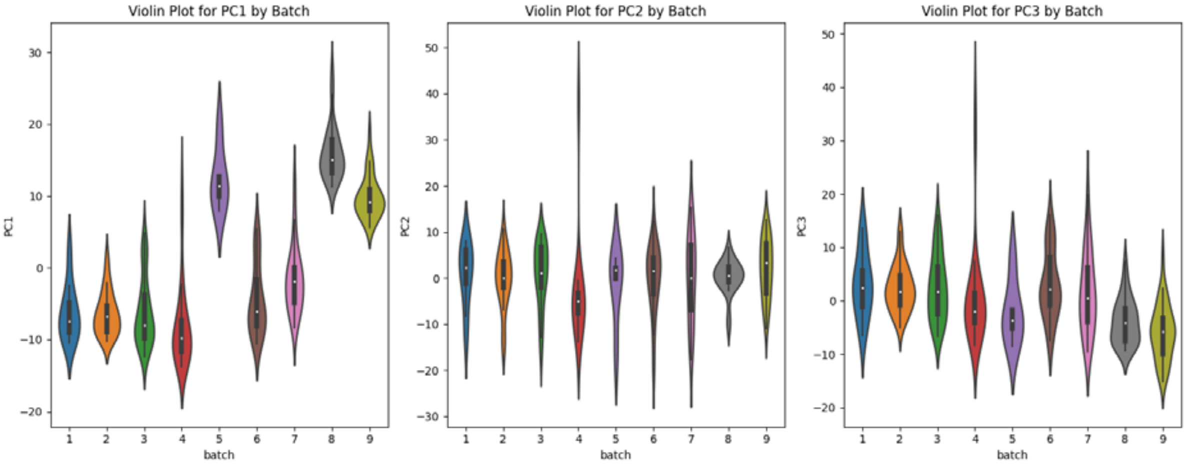
We can observe that for pos.result.501 and pos.result.328 specifically, there are significant differences in the statistical distributions of the samples across batches. This supports the argument that batch effects exist.

4.2.3 Chi-Squared Test

We conducted a chi-squared test between the batch feature and the icu variable. It resulted in a chi-squared statistic of 17.467 and a P-value of 0.0256. This result indicates that at a 5% significance level, there is a statistically significant association between batches and the ICU variable.

4.2.4 Principal Components Violin Plot

We also measured the statistical distributions of the principal components across batches to see whether there are significant discrepancies across batches. We visualize this using a violin plot.



As can be seen, particularly for PC1, we can observe noticeable differences for batches 5,8, and 9. This supports our scatter plot result and further strengthens the argument that there is a significant batch effect in our dataset.

4.3 Classification Algorithms Performance Results

We split the dataset into an 8:2 train test split while doing 5-fold cross-validation for all three algorithms. We will use accuracy on the test data as the primary metric to evaluate the model performance.

Model	Random Forest	XGBoost	PLS
Accuracy	69.39%	69.39%	67.35%

As we can see from the table above, all three models perform similarly, with Random Forest and XGBoost having the highest and similar accuracies.

4.4 Most Influential Features

As both Random Forest and XGBoost provide an easy way to get the feature importance, we will use the feature importance of those two models to see our most influential features and their corresponding KEGG IDs. We select the top 10 features of both models.

4.4.1 Random Forest Top 10 Features

The top 10 features for Random Forest and their corresponding KEGG IDs are:

Feature	KEGG ID
pos.result.1098	C00188
pos.result.2476	C06205
neg.result.898	C00800, C00257
pos.result.138	C00041, C00213, C00099, C00133, C00546
pos.result.114	C06178
pos.result.3603	C04840
neg.result.1267	C00299
pos.result.2934	C05452
neg.result.830	C16358, C16353
pos.result.922	C00847

4.4.1 XGBoost Top 10 Features

The top 10 features for XGBoost and their corresponding KEGG IDs are:

Feature	KEGG ID
neg.result.355	C02362

pos.result.128	C00058, C02218, C00804
neg.result.140	C01353
pos.result.2174	C00386, C05340
pos.result.1098	C00188
pos.result.1043	C00077, C00515
pos.result.1163	C00579
pos.result.889	C13747, C07480
pos.result.2042	C00262
neg.result.1579	C04188, C04582

The feature importance of Random Forest and XGBoost are quite different. However there are 1 feature and KEGG metabolite ID which is in both algorithms' feature importance which is feature pos.result.1098 which corresponds to KEGG ID C00188

4. Discussion

In this study, we analyzed metabolic feature data and their relation with COVID severity. We managed to identify key metabolites which have a large influence in the determination of COVID severity. We also conducted thorough analysis on the existence of batch effects in our dataset.

References

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Geladi, P., & Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, *185*, 1-17.

Ma, G., Kang, J., & Yu, T. (2023). Bayesian Functional Analysis for Untargeted Metabolomics Data with Matching Uncertainty and Small Sample Sizes. *arXiv preprint arXiv:2312.03257*.

Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, *9*(11).