Final Project: Microbiome Data Analysis

Name: Yi Wang (yiw10)

1. Abstract

Microbes are important to human. Humans' immune system, digestive system, and appearance rely heavily on microbes. They help us digest food, kill virus, and maintain a relative stable internal environment for human beings to keep healthy. With the development of sequencing technologies, we can now study the compositions of microbes and microbial communities through sequencing analysis with 16s rDNA amplification. In the process of 16s rDNA sequencing, we will get an OTU (operational taxonomic units) table, which is the starting point of the analysis in this project.

An OTU is a representative of a specific taxa, from kingdom to species. Usually, we use 97% as the threshold to divide into different OTUs. That is, reads under the same OTUs are more than 97% in similarity. After classifying reads of samples in different OTUs, we get an OTU table, which gives the information of the number of reads each OTU contains in each sample. There is another form of OTU table, that is, to use relative abundance (calculated by reads of one sample in each OTU/total number of reads in the sample) rather than the number of reads. Thus, for the table of relative abundance, the sum of each row will 1.

In the project, we will study the relative abundance of samples, along with a few other variables, to see how they are related to human health and behavior. As the data is very sparse, we will perform some transformations to make it more adaptable to models. In part 3, we will use hierarchy clustering, k-means clustering and sparse PCA to study the OTUs, trying to find information beneath the mess data. In part 4, we will perform revised Lasso regression, decision tree regression to predict BMI based on the covariates. In part 5, we will employ logistic regression, random forests and XGBoosting to build models to predict BMI categories, and alcohol consumption frequency. Part 6 is used for a brief discussion to answer the collaborator's questions and conclusion.

Based on our analysis, we found that OTUs are most appropriate to be grouped into two clusters. The OTUs do have predictive power in predicting human BMI/BMI category/alcohol consumption frequency. However, the most predictive variable is the weight of bodies.

2. Literature review

An important feature of microbiome data is that it's compositional, which means that the sum of each sample is 1. In 2016, Pixu Shi, Anru Zhang, and Hongzhe Li in their paper, "Regression Analysis for Microbiome Compositional Data" , gave several ways to make use of such feature. They used linear models with a set of linear constraints on regression coefficients. Then, use a penalized estimation procedure to estimate the regression coefficients and select variables under the constraints. Specifically, they use two different kinds of models. The first is linear log-contrast model. That is, select a covariate as the reference component, get the values of other covariates divided the referent component, and take the log of the quotient. Then, build a linear regression model without intercept with all the covariates except the basis. However, as it's crucial to find the appropriate reference component, a new method is developed. That is, get the logarithm of each covariate, then build a linear regression model with the constraint that the sum

of all coefficients equals zero. The second is subcompositional regression model. That is, gather the covariates belong to the same taxon at a higher rank as a new covariate, use these covariates as new variables to perform linear regression with the constraints that the sum of coefficients is zero. A log-transformation is also needed here. Through these tools, they find that Oscillibacter genus is found to be associated with BMI, and demonstrates that prediction performance was better when linear constraints are imposed.

In 2020, Antoni Susin, Yiwen Wang, Kim-Anh Lê Cao, and M Luz Calle wrote a paper, "Variable selection in microbiome compositional data analysis", to study the variable selection methods for the compositional data in microbiome. In the paper, they proposed three methods. Selbal is a forward selection approach. First, find the pair of taxa whose balance most associated with the response. Then, at each step, a new taxon is added to the current balance, to improve the optimization criterion. Both Clr-lasso and Coda-lasso used the penalized regression. For Clr-lasso, first project the compositional data to a Euclidean space, then apply a penalized regression to select variables. Coda-lasso is to use a log-contrast model instead of the first step in Clr-lasso, then apply a Lasso regression with constraint that all coefficients sum to 0 to select variables. After comparison, they found that Selbal can achieve a very good result by selecting the fewest variables among the three, but it's also computationally very expensive. The rest two methods do not have a significant difference in performance. Thus, no best variable selection method found. Users should select variable selection method based on their own needs.

3. Unsupervised learning

3.1 Address missing data for unsupervised learning

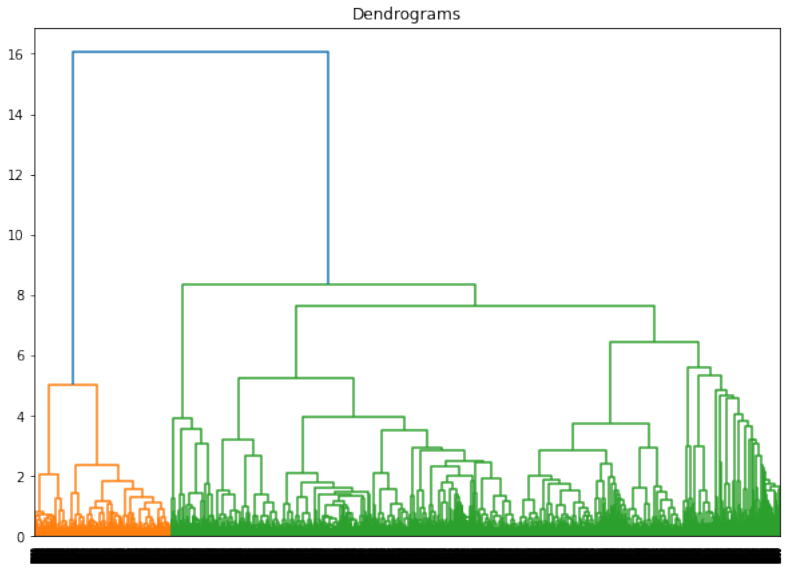
The original dataset is very sparse, and many covariates are only specified at the genus level. Thus, we cluster those variables to the level of genus, and add all values that share the same genus level of the same sample to create new covariates. Then, drop the covariates that are not specified at the species level to get a new data. After the step, the sparsity decreases from 99.42% to 98.45%.

Based on information we have, it's hard to tell whether a zero means the data is missing, or that the sample does not contain such an OTU. We also found that each OTU occurred at least once in the whole dataset. Thus, we assume that a zero means the sample does not contain the OTU. So, we leave zeros as they are for unsupervised learning.

3.2 Hierarchy clustering

We first tried hierarchy clustering, performed with the scipy.cluster.hierarchy package in python. Hierarchy clustering is a method that tries to group samples into a hierarchy of clusters. We will use agglomerative approach here. That is, each sample starts as its own cluster. As moving up to higher hierarchy, clusters will be grouped together to make a new bigger cluster. The process ends when all samples are in the same cluster.

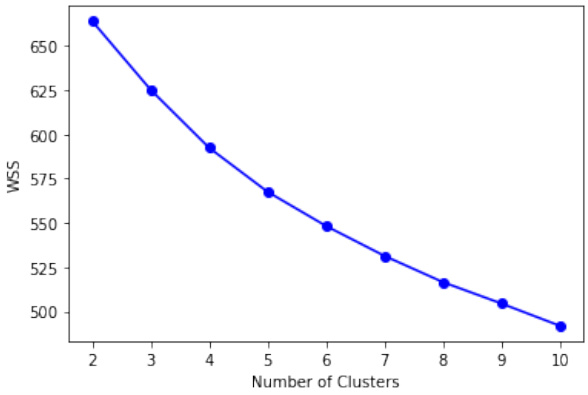
Since our data is in the same scale, we choose to use the Euclidean distance. The merge strategy of our hierarchy clustering algorithm, that is, how we will divide samples into different clusters, is to minimize the sum of variance within all clusters, similar to k-means clustering.
Below is the result of hierarchy clustering. Besides the messy part at the bottom, we can see that we have two big clusters, and three to five smaller clusters. Based on the result, we can see two points. First, the OTUs do have share some similarities that can be merged. Second, the number of clusters can be 2~5, depending on the objective.

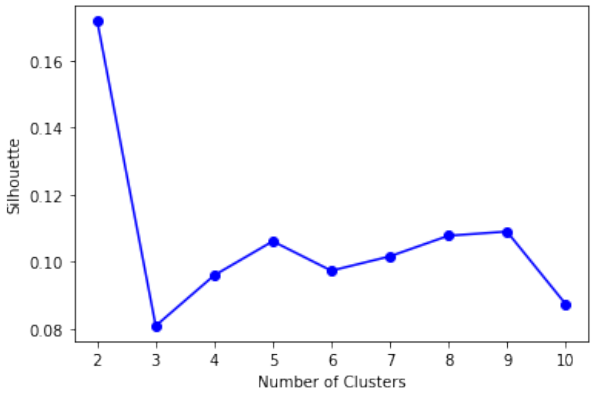


3.3 K-Means clustering

K-Means clustering is a method that tries to cluster samples into k clusters based on the objective of minimizing WSS (within cluster sum of squared errors). Thus, a key point of k-means clustering is to choose an appropriate k value. We chose it through the Elbow method and Silhouette method. Elbow method is to calculate the WSS associated with different k values, then choose k for which is a turning point for WSS. After the value of k, WSS decreases much slower. Silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation). (Source: Wikipedia) The range of Silhouette value is between -1 and 1. The higher, the better.

From the two plots below, we can see that for WSS, there is no obvious turning point, which means we cannot choose k values based on the method. Still, we can see from the plot that WSS decreases fastest when k=2. As k goes up, the speed of WSS decreases becomes slower. In Silhouette plot, it is highest when k=2, which means the appropriate k value in our case is 2. But it's noticeable the value of Silhouette when k=2 is still not large.



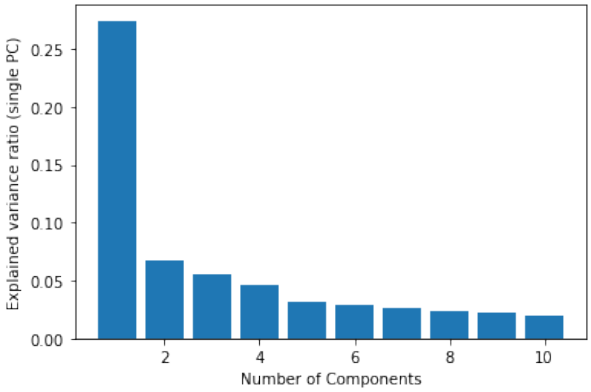


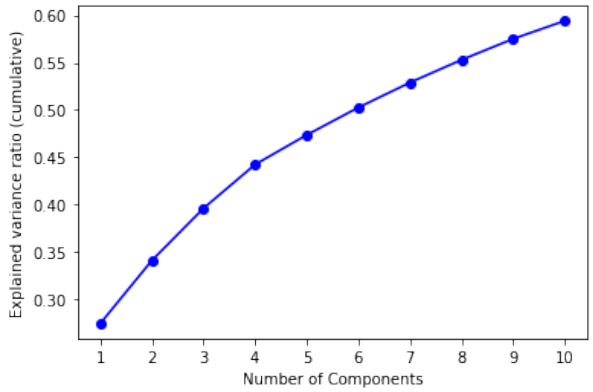
The final k-means clustering result in our case is as follows. We have 6378 samples in the first cluster and 3133 samples in the second cluster. The WSS of the final cluster is 664.

3.4 Sparse PCA

PCA, principal component analysis, is a method to transform high-dimensional datasets to lower-dimensional ones while retaining most of information contained in the original dataset. Sparse PCA, sparse principal component analysis, is a revised version of classic PCA. It improves classic PCA by finding linear combinations of a few covariates instead of all of them. As the number of covariates of our dataset is very large, sparse PCA is a more appropriate approach compared to classic PCA.

We plotted the result of the first 10 PCs. PCs are ranked based on the ratio of original data variance they explained. From the result below we can see that apart from the first PC, most PCs do not explain much variance of the original data. Even the first PC only explains less than 30% of the variance of the original data. We can also see that, after the fifth PC, the ratio of variance explained by a PC does not change much. This echoes our previous results. It's noticeable that the sum of the 10 PCs only explains about 60% of variance of original data, which means that our data is really scattered.





3.5 Questions and Summary

Q1: What is the level of sparsity and how does that affect the clustering results?

The sparsity score, the number of zeros in a matrix divided by the total number of elements in the matrix, is 0.98 in our dataset. So, it's a very sparse dataset.

From the result above, we can see that the clustering is not very successful. For the k-means clustering, the WSS is 664, which is very large, meaning the samples in the same cluster are very diverse. For the sparse PCA, the sum of ratio of variance explained by the first 10 PCs is only around 0.6. It's possible that the OTUs are not related to each other, but it may also be the result of a very sparse dataset.

Q2: Are there any underlying clusters based on OTU information?

It's obvious that we can divide the samples into two clusters. From the above result, we can see that, under this clustering approach, our samples are well placed. Whether to group samples into more clusters (up to 5) depends on the needs of the problem.

Q3: Summary of this part

In this part, we applied hierarchy clustering, k-means clustering and sparse PCA to study the OTU table. Clustering is always a starting point of data analysis. By studying samples within each cluster, we can get a more in depth understanding of data. And also, there are algorithms that are built specifically for sparse data like sparse k-means. It's slightly different from the classic k-means we applied here. However, as time is limited, we will not go deep in these questions.

4 Supervised learning (Regression)

4.1 Address missing data for supervised learning

For the missing values in the predictors, if the variable is categorical, we leave the missing values as they are; if the variables is numeric, we fill the missing values with the mean of the variable. For response variables, BMI, BMI category and alcohol consumption frequency, we fill in the missing values based on the following rules:

a. Missing values of BMI: calculate the mean BMI value associated with each value of alcohol consumption frequency, then fill in the missing values with mean value based on the corresponding alcohol consumption frequency value.
b. Missing values of BMI category: fill in based on the CDC published relationship between the value of BMI and BMI category.

c. Missing values of alcohol consumption frequency: first find the most frequent alcohol consumption frequency value associated with BMI category, then fill in the missing values based on the corresponding BMI category value.
d. For missing values, we cannot fill in based on the above rules, delete them.

Since most features are not specified at the species level, we will ignore the species and only classify the OTU to the genus level. Thus, we rename all OTUs to the genus level, then add the columns with the same name together to one column with the new name.

We will drop covariates that appear in less than 1% samples. To maintain the compositional nature of microbiome data, we will delete samples that do not satisfy the nature after dropping the covariates.

4.2 Strategies to process the compositional data in sparse context

The strategies we used to process the compositional data are learned from the paper, "Regression Analysis for Microbiome Compositional Data", Pixu Shi, Anru Zhang, and Hongzhe Li, 2016.

The composition nature can be explained as follows. Suppose we have an $n \times p$ matrix X, n samples and p covariates. The composition nature of data makes each row of X to be a simplex $\{(x_1, \cdots, x_p): x_j > 0, j = 1, \cdots, p \ and \ \sum_{j=1}^{p} x_j = 1\}$.

We used two strategies to make use of the composition nature of data. The first is to get the logarithm of each covariate and make a constraint on the coefficients that the sum of all coefficients equals zero. The following is an example in the context of a linear regression.

$$Y = Z\beta + \varepsilon, \qquad 1_p^T \beta = 0$$

$$where \ 1_p = (1, \cdots, 1)^T \in R^p, Z = (z_1, \cdots, z_p) = (log x_{ij}) \in R^{n \times p}, and \ \beta = (\beta_1, \cdots, \beta_p)^T$$

The second strategy is to select a covariate as the reference component, get the values of other covariates divided the referent component, and take the log of the quotient. We will also explain it with a linear regression example.

$$Y = Z^p \beta_{\backslash p} + \varepsilon$$

$$where \ Z^p = \left\{ \log \left(\frac{x_{ij}}{x_{ip}} \right) \right\}, \ \beta_{\backslash p} = (\beta_1, \cdots, \beta_{p-1})$$

Z^p is an $n \times (p-1)$ matrix with the pth covariate as the reference component.
We cannot make constraints on coefficients for some algorithms we applied below. For those algorithms, we use the second strategy. For the models that we can make the constraint, we use the first strategy.

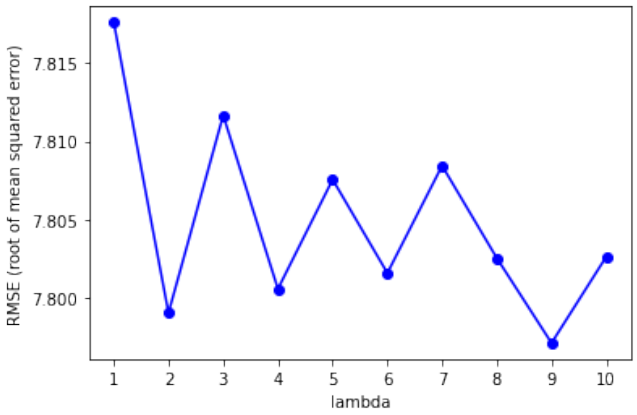
As our data is sparse, we need to make some modifications before we can apply these two strategies. We add 1 to each element in the dataset. As $log1 = 0$, the modification does not change the sparsity of the matrix. The sum of a row is 1 plus the number of covariates, which is still a constant. Thus, the modification does not change the composition nature, either. We will make the modification first, then apply the appropriate strategy.

4.3 Lasso regression with constraints

Lasso regression is a penalized regression algorithm. It added a shrinkage of the $l1$ norm of parameters to the loss function of linear regression. It can perform both variable selections, as the coefficients of covariates can be shrunk to 0, which means the covariates are dropped, and regularization, preventing overfitting. As we have many covariates, Lasso is worth trying. Also, we will make a constraint on the model that the sum of coefficients is 0.

We tuned the value of λ from 1 to 10. An interesting fact is that we do not have coefficients equal 0. All covariates are included in the model. We plotted the RMSE (root of squared error), a metric to evaluate the result of prediction, against λ. The result showed that overall, the performance of model does not change much as the value of λ changes. The best performance was reached when $\lambda = 9$.

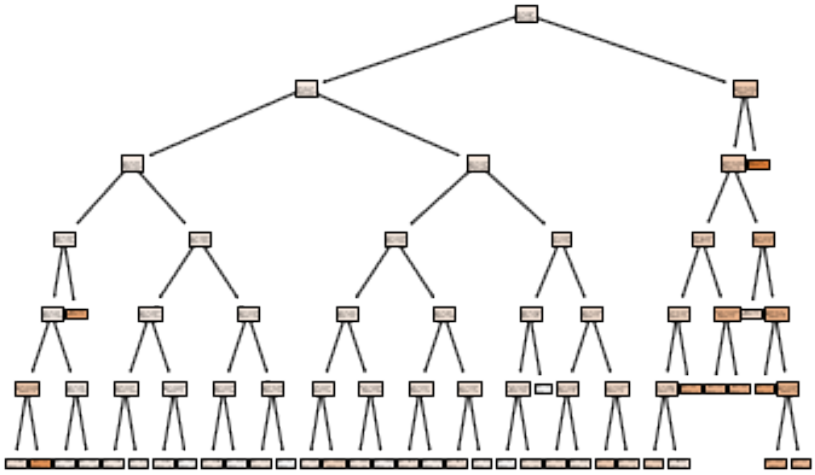
We applied the $\lambda = 9$ in our final model and got the RMSE as 7.8.



4.4 Decision tree regression

Decision tree is a statistical model that can be used both in regression and classification. It uses a tree like structure to predict the target based on the inputs of covariates. A key hyperparameter for decision tree model is the max depth of the tree, i.e., how many decision notes the tree can have at most. A decision tree may stop when the node cannot be splitted, or the max depth is reached. A too large max depth of may incur overfitting, while a too small the max depth of tree will harm the predictive power of the tree.

Based on our result, the best performance tree has the max depth equal to 6. The RMSE predicting BMI value of this tree is 8.2. We also plotted the final decision tree. However, as the space is limited, it's not very clear.



5 Supervised learning (classification)

5.1 Logistic regression

Logistic regression is a very basic and powerful statistical model. If you do not know which model to use when facing classification problems, logistic regression is always worth trying. Logistic regression is used to model the probability of a certain class of the target variable occurs. It's most commonly used in modeling binary target variables. However, multiclass logistic regression is also very popular. It can be thought as K (number of classes in the target) binary logistic regression problems. Multiclass logistic regression can be implemented automatically in python. Only need to be careful about the values of parameters.

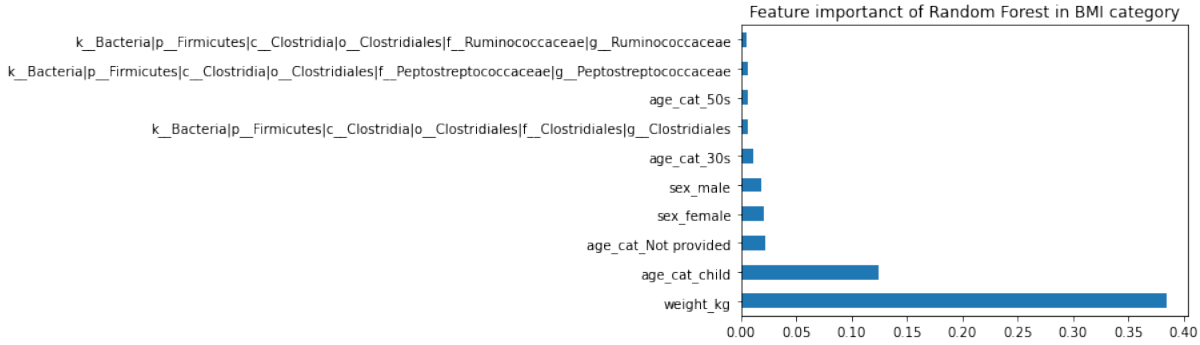
In our models, as both BMI category and alcohol consumption frequency are multiclass variables, we used multiclass logistic regression model. We employed accuracy as the metric to evaluate our result. The accuracy of BMI category of our test is 0.69 while that of alcohol consumption frequency is 0.33.

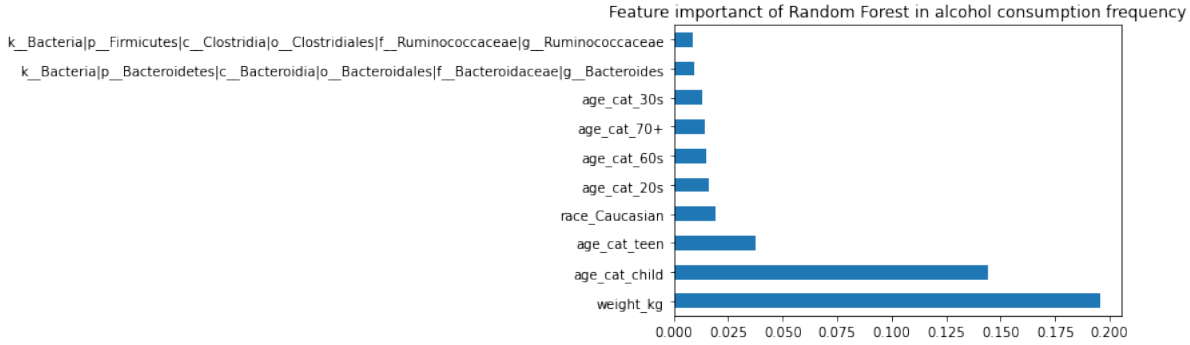
5.2 Random forest

Random forest is an ensemble learning method that can be used both in regression and classification. It's a set of simple decision trees built with the random subsets of covariates. The output of a random forest is either the mode of the outputs of the set of decision trees (in classification problems), or the mean of the prediction of the outputs of the set of decision trees (in regression problems).

An important hyperparameter for random forest algorithm is the number of decision trees built. We found the value of the hyperparameter by trying different values to find the one that can maximize accuracy. Based on our result, the best performed number of trees in BMI category problem is 600 while in alcohol consumption frequency problem is 500. After applying the two values, the final accuracy of BMI category is 0.60 and of alcohol consumption frequency problem is 0.32.

Below are the feature importance plots of random forest models. We only plotted the most important 10 features. It's interesting to see that in both problems 'weight_kg' is the most important variable. Age also plays an important role. What's noticeable is that, some OTUs are also important in prediction the target variables.



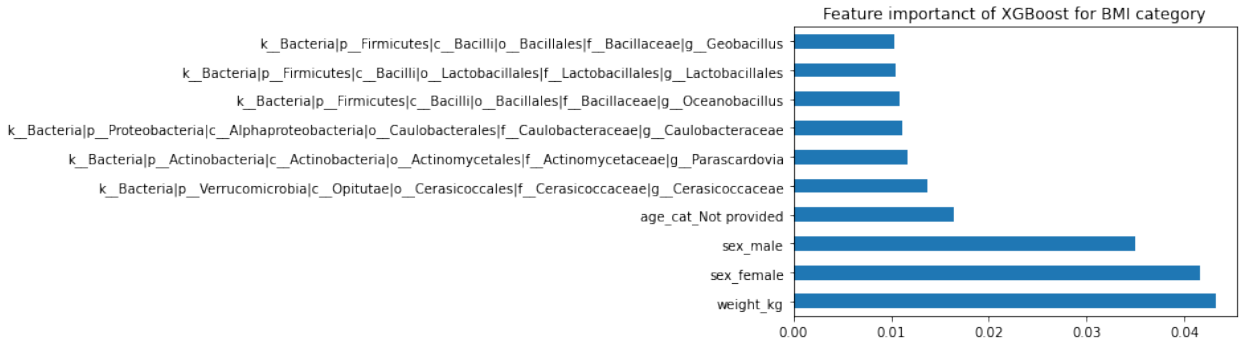


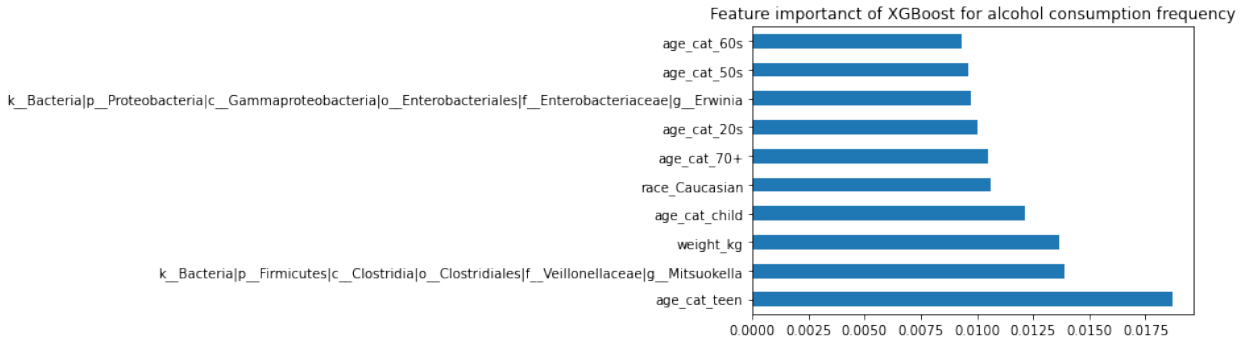
5.3 XGBoost

XGBoost, Extreme Gradient Boosting, is an open-source software library which provides an optimized gradient boosting framework for many languages. The core XGBoost algorithm is a parallel tree boosting algorithm that is very powerful. It has long been regarded as the killer in machine learning algorithms. It also can be used both in regression and classifications.

We used XGBClassifier to approach our problems. The final accuracy of BMI category is 0.75 while that of alcohol consumption frequency is 0.29.

Below are the feature importance plots of random forest and XGBoost. It's interesting to see compare the plots with that of the random forests. The XGBoost has more important OTUs in BMI category prediction than random forest. It also includes the 'sex' as an important covariate. In the alcohol consumption frequency prediction, the XGBoost makes age and weight as the most important features, which is consistent with random forest. Sadly, neither of them performed well.





6 Collaborators' questions and conclusion

Besides some extreme possibilities like the data is flawed that any analysis made based on it is not plausible, or the relationship between the covariates and the targets suddenly changed, I believe the analysis made above is at least somewhat plausible.

First, for the models building process in which include turning hyperparameters, I separated the data into three parts: training, testing, and validation. The parameter turning happened only between the training and testing set. The validation data is absolutely new to the model. Thus,

the parameter tuning process does not harm the testing environment to test the true predictive ability of the model.

Second, comparing the results of training and testing sets (see the tables below), we found that there is no significant difference. Models perform almost the same in training and testing datasets. Thus, it's unlikely that the models are overfitting.

Third, comparing the results of the different models for the same problem. We can see that the difference is not huge. Our models range from simple linear models to complex gradient boosting models. If we get close results from very different models, it's unlikely that we use an inappropriate approach to analyze the data.

Based on the three points above, I believe the results of supervised learning models are at least somewhat reliable.

For the unsupervised leaning models, it's hard to decide whether it's right or wrong. However, we can validate the results between the three different models we applied. From hierarchy clustering, we can see that an appropriate clustering approach should have 2~5 clusters. The recommended number of clusters from k-means is 2. Thus, the two results can validate each other.

Due to the limit of time, this is a very superficial analysis of the data. It's possible that if we group the data into a higher rank of taxon, the result may be better. There are also many other methods to approach sparse dataset worth trying. There are also some more complicated models that can be used to handle the composition data. These will need to be done in later research.

Model	RMSE
Lasso with constraint	7.8
Decision tree	8.2

Table1: Results of regression problems

Model	Accuracy (BMI category)	Accuracy (alcohol consumption frequency)
Logistic regression	0.69	0.33
Random forest	0.6	0.32
XGBoost	0.75	0.29

Table2: Results of classification problems