MATH2313 Data Science: Principle and Practice Final Project

Yunrui Liu
ID: 2023233157
liuyr2023@shanghaitech.edu.cn

Abstract

Diabetes is one of the major diseases that plague the medical community. Early detection of diabetes will be very beneficial to improving overall health. Furthermore, if a small amount of external data can be used to predict whether an individual has diabetes, this can significantly advance the patient's treatment to reduce the duration of illness. This paper uses data from health and nutrition surveys through data science analysis methods to construct a binary prediction algorithm for predicting diabetes using a small amount of human health information. The analysis uses Linear Regression, Logistic Regression, Random Forest, and PCS analysis-based ensemble algorithms, supplemented by appropriate data cleaning, preprocessing, and exploratory analysis processes to find the optimal prediction algorithm. This paper analyzes the predictability and stability of algorithms other than the ensemble algorithm, pointing out the stability of the linear algorithm and the instability of the algorithm that relies on judgment statements, as well as the good performance of multiple optimal fit ensemble algorithms on some statistics.
The Code are avaliable on https://github.com/supperXpower/MATH2313_Final_Project.

1 Problem Introduction

Diabetes is an important disease that troubles people. Being able to predict people's risk of diabetes or to use external information to determine whether they have diabetes early is very effective for subsequent diagnosis and treatment. However, without detailed testing, the physical information related to the subject that can be obtained is very limited. If the prediction of whether the disease is present can be achieved with limited information, it will be of great help to the diagnosis and treatment of diabetes.
In this project, I hope to use different algorithms to use information that can be easily obtained such as people's health indicators, nutritional levels, and family medical history to more accurately determine whether the respondents have diabetes. I hope to be able to find out which factors have a greater impact on whether they have diabetes, what kind of people are more likely to have diabetes, and implement an algorithm to predict whether a possible patient has diabetes during the data analysis process.

2 Data Introduction

This section is to answer Chapter 11, EX 27, a.
Since it is unrealistic to obtain and train data from all over the United States, I will use a set of data that I think can represent the entire United States.
The data for this study comes from the National Health and Nutrition Examination Survey (NHANES) published by the National Center for Health Statistics. This is a nationwide health survey for all ages that measures the health and nutrition of adults and children in the United States. This study uses a

set of data collected by the survey throughout 2016.
This survey selects a group of people who are considered to be representative of all Americans. The process of selecting these people will include some oversampling, including some samples that may have different health outcomes. The data collects data on the health of these respondents, and conducts health examinations (including dental examinations and body numerical measurements), laboratory tests, and dietary interviews (involving personal health, diet, social and economic characteristics, to determining the nutrient content of the diet.).
The official provides two documents in the subordinate folder, which provide detailed information and brief information about the questions in the survey represented by each variable name of the original data. The brief version provides the serial number, name and brief description of the question of each variable in the source data, while the detailed version provides a variety of detailed information such as the problem description, value range, and value meaning of the problem. In the subsequent data cleaning process, this information will provide very helpful information.
It should be noted that the data used this study is simplified by a specific function "load_diabetes_data" in the "function/load_diabetes_data.py" document. This function extracts the ID information of some respondents including the survey month and address and integrates it into a unique ID for each respondent. It also extracts a response variable of whether they have been notified of diabetes, 5 processed variables (including 5 binary variables) and 6 unprocessed variables (including 4 numeric variables, 1 binary variable and 1 categorical variable). At the same time, the variable names are greatly optimized, and the capital letter abbreviations of the original data variables are changed to lowercase letters.
In the data, the variables mean:

- "house_family_person_id" represents the information related to the survey time and address of the respondent, which is different for each person.
- "diabetes" represents whether the participant has been informed of diabetes by the doctor.
- "coronary_heart_disease" represents whether the participant has been told by a doctor that he or she has coronary heart disease.
- "hypertension" represents whether the participant has been told by a doctor that he or she has hypertension.
- "heart_condition" represents whether the participant has been told by a doctor that he or she has heart disease.
- "cancer" represents whether the participant has been told by a doctor that he or she has cancer.
- "family_history_diabetes" represents whether the participant has diabetes in their family history.
- "age" represents the participant's age.
- "smoker" represents whether the participant smokes.
- "sex" represents the participant's gender.
- "weight" represents the participant's weight (pounds).
- "bmi" represents the participant's bmi value ($\times 100$).
- "height" represents the participant's height (inches).

The observation unit of this data is the individual subject. The health and nutrition information provided by the data is what I need, so I believe that the data is relevant to our prediction.
The simplified version of the loaded data has 32,499 observation units, 13 columns, one of which is the information column, one is the response variable, and the rest are columns that I think may have predictive effects.

3 Data Cleaning and Preprocessing

3.1 Exploration for Data Cleaning

This subsection is to answer Chapter 11, EX 27, c.
I first explored the data briefly to determine which data needed to be cleaned and preprocessed.

The first step was to find out whether there were any invalid values (here includes extreme values). I checked the four numerical variables, “height", “weight", “age" and “bmi", and plotted their distributions in Figure 1. From the distribution, each of the four numerical variables has a area on the far right that is significantly different from the distribution and is abnormally prominent. After consulting the document, I learned that some outliers represent extreme values (unmeasurable, such as weight=996 means the individual’s weight is too large) and some represent unknown values (such as weight=999 means the individual doesn’t know his weight). These values cannot provide exact information, and I cannot judge their true values. In other words, assigning any value to these parts may result in deviations from reality. And these values must to be cleaned, because most of the these values are seriously exceed the range of the variable (e.g. the range of weight is 100∼299, and the range of its outliers is 996∼999). Therefore, how to deal with there outliers of numerical variables is a factor that must be considered in subsequent data processing.

Similarly, I also explored some binary variables . By reading the function that simplifies the data, I

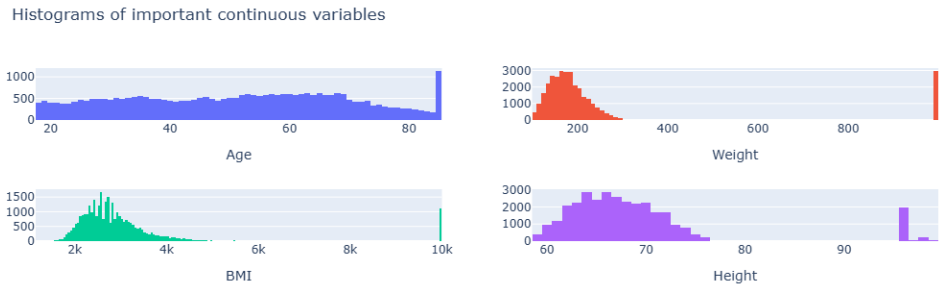

Figure 1: The distribution of numerical variables before cleaning and preprocessing

found that the values of the five binary variables processed must be only 0 and 1, so I only explored features “smoker" and “sex". From the distribution diagram in Figure 2, I found that “smoker" has invalid values 7, 8, and 9. After read the corresponding documents, I found that these values correspond to individuals who cannot provide accurate information. Because of these values, the variable cannot exist as a binary variable, so these values need to be processed. In addition, the distribution of the “sex" variable indicates that it has only two values 0 and 1, so it does not need to be cleaned or preprocessed.

The next step was to examining the missing values. Through the output of the function, I found

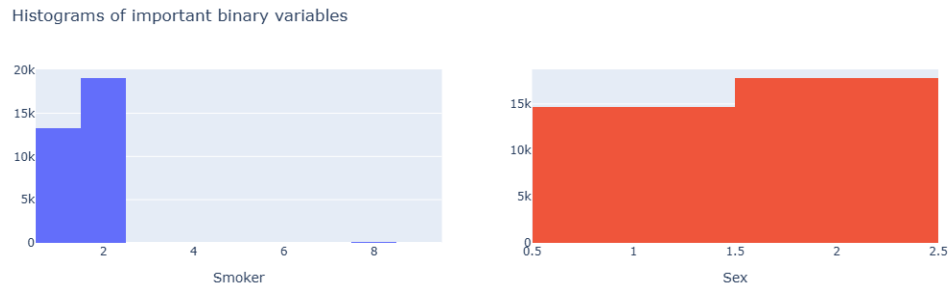

Figure 2: The distribution of categorical variables

that any variable in the simplified version of the data does not have missing values, so it can be left unprocessed (or just in case).

The next step was to check whether the format of each column of data is unified. Through the output of the function, I found that each column only outputs one format when I check the data format of elements in every column, so this does not need to be processed.

The next step was to check the column names. As mentioned earlier, all column names have been converted from uppercase abbreviations to lowercase variable names with the same meanings of original variables, so this does not need to be processed.
The next step was to check the data format of each column. Since binary variables can be represented by int, and the function that generates data also guarantees the value of binary variables that only includes 0 and 1, the corresponding output has also been shown that all except "house_family_person_id" are in int format ("house_family_person_id" is in "object" format that stands for the string format), so item does not need to be processed.
The final step was evaluating data completeness. Here I mainly checked the number of independent values of "house_family_person_id" to ensure that no respondents appear repeatedly. The function printed the number of corresponding non-repeating values that corresponds the number of rows, which means that each row corresponds to a unique respondent, and no duplicate data appears. The completeness of the data has also been verified.
In summary, in the section of data cleaning and preprocessing, I mainly need to deal with outliers and extreme values of 4 numerical variables and 1 binary variable.

3.2 Data Cleaning and Preprocessing

This subsection is to answer Chapter 11, EX 27, c.
The biggest problem with outliers and extreme values is that they cannot be directly included in the calculation and do not provide any exact information. No matter how these extreme values and outliers are handled, they are all based on specific assumptions, but the assumptions themselves may have a certain degree of deviation from reality. Therefore, my goal in this process is to reduce the number of assumptions in the data as much as possible to ensure that the data set itself is more authentic. However, some assumptions can be implemented in the verification of stability as the perturbation of the dataset.
Therefore, my final decision was to remove observation units that cannot provide exact information, that is, in the default processing function, delete individuals with outliers in "weight" (>299), "height" (>76), "bmi" (>9994), "age" (>84), and "smoker" (>2). Overall, this approach deletes about 13% of individuals, but it ensures that all data are true and valid, and the data retained represent information for a population that is measurable and within normal range.
In addition, the code for removing missing values was added, just in case. Also, the variables in "smoker" and "sex" were treated with -1 (that is, the original values 1,2 were changed to 0,1), making them more in line with the definition of binary variables.

3.3 Data Segmentation

This subsection is to answer Chapter 11, EX 27, b.
Since the survey was conducted in the same year, also did not explain whether there were any obvious characteristics among participants in different months, there is also no continuous survey of the same individual over time, so there is no need to divide the data set by time. Further more, the independent information of the observation unit is the relevant ID, and the information in the ID is the survey month and address which are not considered to be related to diabetes itself (no specific address is provided, only the number, not regional), so there is no need for specific grouping to divide the data set.
So the data set was randomly divided here, and the simplified data set was divided into a training set, a validation set, and a test set in a 6:2:2 ratio using random sampling. After the data set is divided, each data set was cleaned and preprocessed.

4 EDA

This section is to answer Chapter 11, EX 27, d.
In this section, I explored and analyzed the data to find some patterns in the data.
First, I checked the basic situation of the data after cleaning and processing (the data set is not divided currently), printed the column name and data size, ensured that the data cleaning and preprocessing has been completed and the column name is readable. After that in preparation for clustering, I copied a data set and standardized all the numerical variables in it.

As done in the exploration before data cleaning, I plotted the histogram of numerical variables to observe the distribution of them in Figure 3. The results show that all numerical variables except “age" show approximate Gaussian distributions. This distribution is the most common distribution, which can illustrate the authenticity of the data source. The distribution of the data conforms to the law of large numbers.

I then showed the distribution of the numerical variables with specific labels in Figure 4. From the

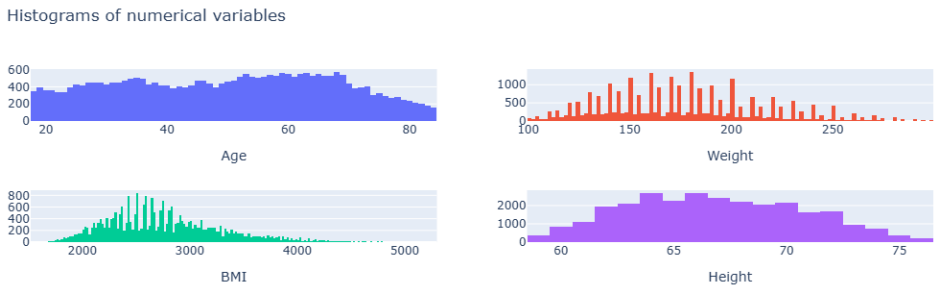

Figure 3: Distribution of numerical variables

output, the expected value of the distribution of “age" with diabetes is significantly higher than that of the distribution of “age" in people without diabetes, and the distribution of the “age" variable with diabetes is also very close to the Gaussian distribution. The expected values of the distribution of “weight" and “bmi" with diabetes are also slightly higher than those without diabetes. From this part, I can preliminarily believe that the population with diabetes is mostly older and heavier.

I printed these four variables again in box plots in Figure 5. From the box plot, from the overall

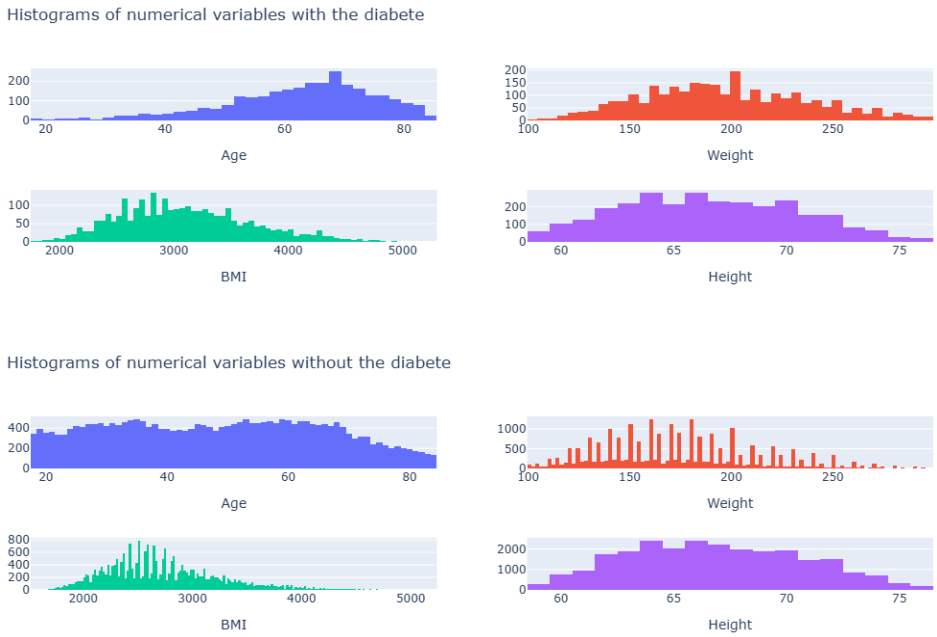

Figure 4: Distribution of numerical variables with and without diabetes

distribution, although the previous “age" distribution is very close to the combination of uniform distribution and Gaussian distribution, the overall range is relatively uniform and stable. It can be

considered that the age survey is more consistent with the age distribution of the population. "weight" and "bmi" have many outliers that are much larger than the mean, which to some extent widen the entire distribution. However, considering the range of values of the Gaussian distribution, I cannot say that these points do not conform to the distribution. Moreover, these points are taken from real surveys, even if they deviate far from the center of the data distribution, they do represent a real situation. Above all, these values are not processed.

Next, I calculated the correlation coefficients between the variables. I showed a list of correlation

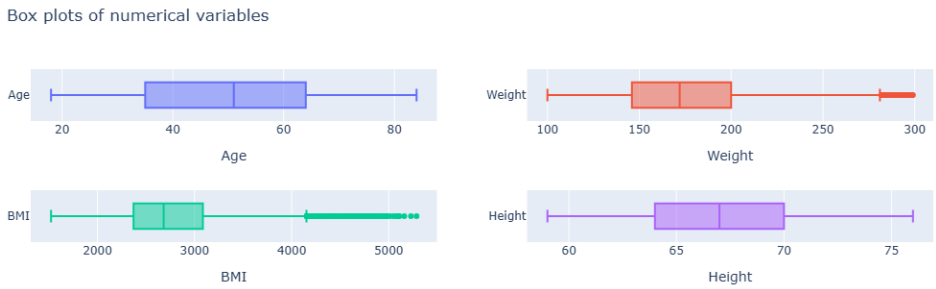

Figure 5: Box plots of numerical variables

coefficients between all variables and “diabetes", as well as the coefficient graph of correlation coefficients between all variables in Figure 6. I found that the correlation coefficients between all variables and “diabetes" are not high, and even the highest positive correlation value is only 0.29. However, even so, I still believe that these are likely to be related to “diabetes", because there is only this information in the current data set. This may also mean that there are many determinants that work together to determine whether a person has diabetes, making the relevance of a single factor low.

It should be noted that the correlation coefficient between “bmi" and “weight" is very high, and

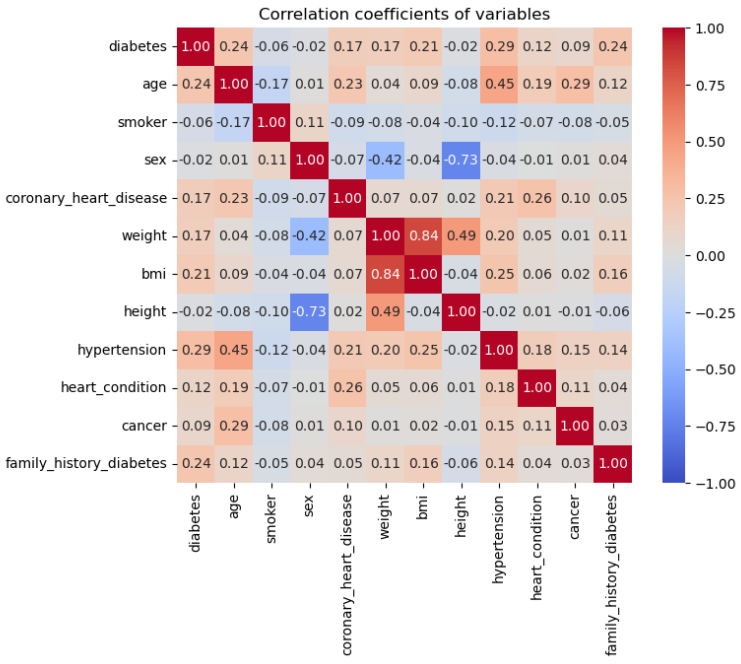

Figure 6: Correlation between variables

both are numerical variables. Considering that the calculation of bmi itself is related to weight and

height, I will delete the “bmi" variable in subsequent analysis and predictions and use the remaining 10 variables for prediction. Although I found that “sex" and “height" have a strong correlation, but considering that “sex" is a binary variable, it is difficult to replace each other, I don’t intend to delete any of them.

Next, some basic information of the variables was output, including mean, median, standard deviation, MSE and MAE. Through preliminary observation, it should be noted that only about 10% of the processed data are with diabetes, which means that even if all people are directly predicted to be without diabetes, the accuracy rate will be about 89%. In the subsequent analysis, it is necessary to pay attention to the accuracy value and compare it with this reference value. There are some differences in the distribution of other binary data. For example, the proportion of smokers and the proportion of different genders are basically the same (close to 1:1), while the vast majority of the data do not have cancer and heart disease, and their data proportion and standard deviation are not much different from those of diabetes. However, it has been verified above that the correlation between these variables is not high, so this may just be a coincidence, or it is possible that the incidence rates of some diseases are very similar.

Next, I tried to discover patterns in data through clustering. I chose two commonly used clustering algorithms, hierarchical clustering and Kmeans clustering, as well as the Gaussian mixture model, considering that numerical variables are close to Gaussian distribution. Since this is a binary classification problem, the number of categories for all clustering algorithms is 2. I used the standardized data for clustering and expressed the category distribution of all clusters. From the clustering results in Figure 7 and 8, considering the ratio of different labels of the response variable, the effects of the three clustering methods seem to be average, and the ratio of the number of clusters under the three methods is very close to 1:1. But when I remove all binary variables and then cluster them, I find that the results are just the opposite. The results of hierarchical clustering are closer to the proportion of diabetic patients, while the other two models don’t have such a result. From these analyses, it is preliminarily judged that the influence of numerical variables on whether or not the disease is quite large. When only considering numerical variables, it is indeed possible that individuals with similar numerical values have similar disease conditions. In order to explore the possibility of correlation, I plotted the distribution of each numerical variable under different clustering methods in Figure 9 and 10. Contrary to the previous conclusion, “age" does not show a very obvious difference in clustering. The age distribution of the two categories in all clusters is very close, while the distribution of the other two numerical variables shows obvious differences in different clusters (and the trends are basically the same). Therefore, it is currently impossible to conclude which variable will have a greater impact on the prediction results.

Finally, I used “height" and “weight" as the horizontal and vertical coordinates to display the

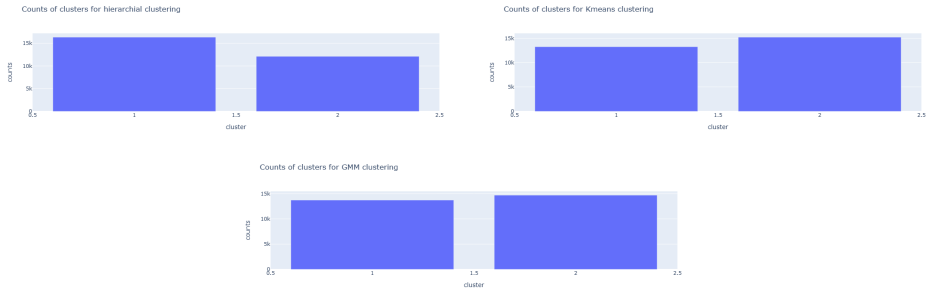

Figure 7: Counts of clusters of different clustering methods h-cluster, Kmeans and GMM

clustering results in the form of coordinate graphs in Figure 11 and 12. Judging from this result, the main effect of binary variables on clustering results is to dilute the classification boundaries. Without considering binary variables, the results obtained by the three clustering methods basically have a clear dividing line, and the two clustering results with binary variables have a certain degree of overlap, which is roughly consistent with the distribution of people with or without diseases obtained from the previous distribution plots. This also shows the role of binary variables in clustering or subsequent classification. It can help the algorithm improve the accuracy of judgment when there is a large overlap in certain features of the actual classification results. It is difficult to get accurate results relying solely on numerical variables.

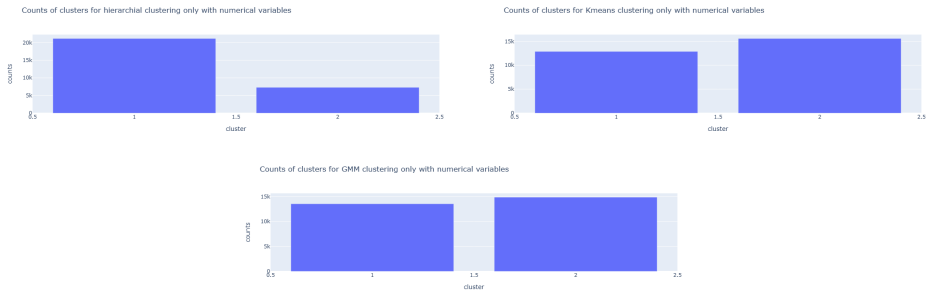

Figure 8: Counts of clusters of different clustering methods h-cluster, Kmeans and GMM with only numerical variables

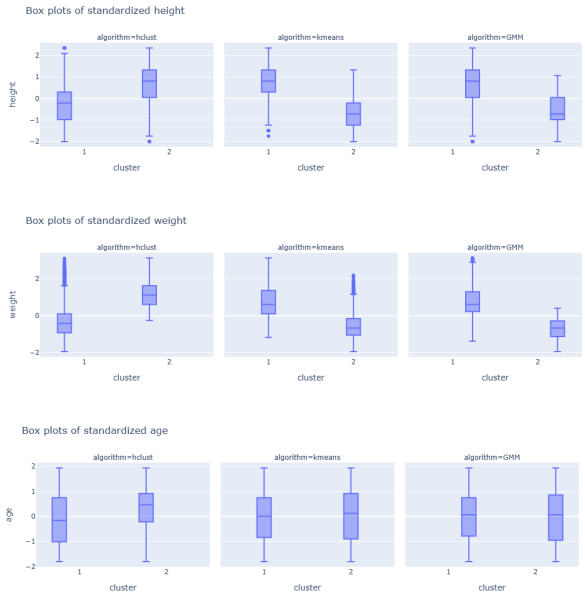

Figure 9: Boxes of variables in different clusters of “height", “weight" and “age"

5 Prediction Models

5.1 Linear Regression

This subsection is to answer Chapter 11, EX 27, e.
Linear Regression is the most commonly used and basic fitting method. Linear Regression assumes that the response variable and the feature follow the simplest linear relationship, that is:

$$response\ variable = intercept + \sum_i b_i \times feature_i \tag{1}$$

It uses this as the basic model for fitting and prediction. The method generally used for Linear Regression is the least squares method to minimize the sum of square difference between predictions and ground truths. Since the variables generated by Linear Regression are continuous values, and our needs are binary values, I need to train a threshold value to predict that the prediction above the threshold as has diabetes and the results below the predicted value as healthy. This threshold value needs to make the accuracy highest under the current prediction result.
The training of this threshold is carried out through the validation set, that is, to find the threshold value that makes the prediction accuracy of the validation set the highest. Since this threshold is automatically selected and directly applied to the PCS analysis process, it is not shown.

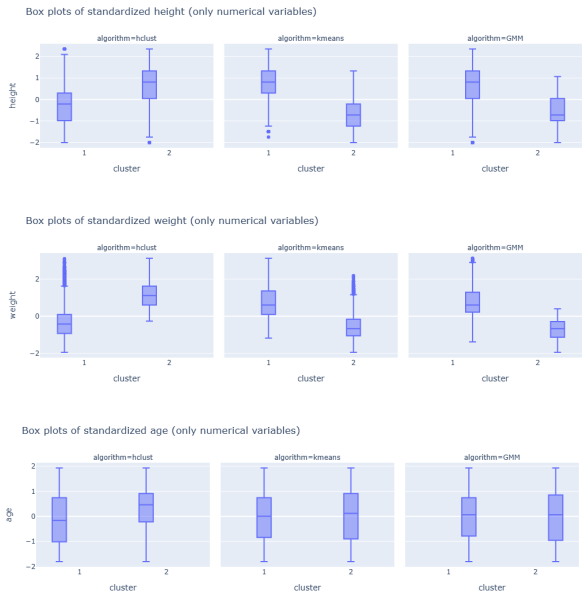

Figure 10: Boxes of variables in different clusters of “height", “weight" and “age" with only numerical variables

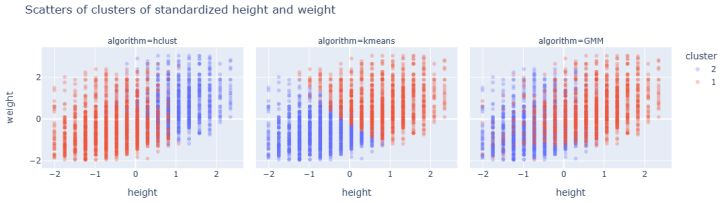

Figure 11: Scatters of samples in different clusters methods

5.2 Logistic Regression

This subsection is to answer Chapter 11, EX 27, e.
Logistic Regression is a predictive model designed specifically for binary prediction problems. Logistic Regression is similar to Linear Regression, but it converts a unbounded numerical value of Linear Regression into a bounded probability value between [0,1] by:

$$p = \frac{1}{1 + e^{-(intercept + \sum_i b_i \times feature_i)}} \tag{2}$$

so that the result obtained by regression becomes the probability of predicting disease and health. However, the calculation of the optimal parameters for Logistic Regression is very complex, and in most cases an iterative algorithm is used to optimize the parameters.
The direct prediction result of Logistic Regression is the classification label (of course, the probability value can also be obtained), and its threshold is calculated inside the function, so no additional training parameters are required.

5.3 Decision Tree

This subsection is to answer Chapter 12, EX 21, a.
Decision Tree is an algorithm for classification based on multiple judgment sentences. The Decision Tree determines the classification of data points through multiple judgment statements. It passes all data points downward from the root, assigns data points to different child nodes based on the results

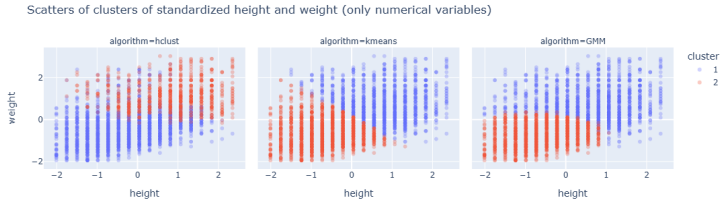

Figure 12: Scatters of samples in different clusters methods with only numerical variables

of judgment statements related to certain feature values, and repeats the corresponding operations in the child nodes. When a data point reaches a leaf node, a prediction for the data point is generated based on the value or category of the training set on the leaf node.
The Decision Tree is optimized using the CART algorithm (also represented in the code as CART), which minimizes the variance or Gini impurity of the segmentation of each child node as much as possible. The training of Decision Trees includes the hyperparameters of maximum tree depth and minimum number of sample splits, as well as a threshold. In the actual code, some values are pre-selected for the first two hyperparameters, and finally a model is trained for each parameter combination, and the three parameter combinations with the highest accuracy in the validation set are selected as the final parameters. These parameters will be used in PCS.

5.4 Random Forest

This subsection is to answer Chapter 12, EX 21, a.
Random Forest is an ensemble algorithm that combines multiple Decision Trees. It uses bootstrap sampling to train each Decision Tree with different versions of samples. When generating binary predictions, it uses the highest vote share of each sample as the prediction or generates the prediction probability in the form of probability by calculating the mean of the probabilities generated by the Decision Tree (this project uses the latter). When there are enough Decision Trees, the final prediction result can be closer to the ground truths.
The training of Random Forest involves three hyperparameters: the number of Decision Trees, the maximum number of variables and the maximum depth of the tree, and the threshold for the final prediction. Consistent with the training of Decision Trees, the possible values of the first three hyperparameters are pre-selected, and each hyperparameter combination is trained once. Each threshold of each model is predicted on the validation set, and the parameter combination with the highest accuracy is selected. This parameter combination will still be used in PCS.

5.5 Ensemble Algorithm

This subsection is to answer Chapter 13, EX 21, a and b.
The ensemble algorithm is a collection of multiple algorithms trained with multiple versions of data sets. By training algorithms by each version of dataset, it will generate multiple ($|dataset\ versions| \times |algorithms|$) fits that can be used to predict. The ensemble algorithms used in this project can be divided into two categories: one is to select the best fit in the validation set among various fits as the final fit, the final prediction is the prediction of the final fit; and the other is to select multiple best fits in the validation set as final fits and predict by combining the predictions of these fits that is similar to the Random Forest.
When training the ensemble algorithm, since multiple training data are required, the training set needs to be changed, and these changes themselves will also be considered as perturbations on the training set. Therefore, the training of this part of the algorithm will be carried out simultaneously with the stability analysis of the perturbation in the PCS process.

5.5.1 Perturbations of Dataset

Before creating a new data set, I need to clarify the data cleaning and preprocessing work I did before. Each step taken previously will affect the composition of the dataset. In this section, I processed three other numerical variables and the binary variable “smoker". And after the EDA section, I decided to

drop the “bmi" feature due to a high correlation between it and “weight", that made me in prediction section, when I cleaning the data, I first dropped the “bmi" (This is why I said earlier that I processed three numerical variables).
Since the disturbance itself means both the change of the data set and the addition of some uncertainty, in the process of building a new data set, I consider whether the originally deleted observation units can be put back into the data set in a special way. In this way, the data set will involve more and more complete observation units.
Therefore, I will use the following different disturbance methods, which can provide 32 different data sets:

- 1. Handling extreme values. In the “height", “weight" and “age" features, there are values such as 96, 996, and 85 respectively. The first two represent extreme values that are too high, while the latter indicates that the age is over 85. All of them don’t mean exact values. Here I assume that the reason the investigators recorded the values in this way is that the extremely high values and the changes in these extreme values within the extreme range will no longer have any impact on the prediction, so they can be classified into one category. Therefore, I assign the observation samples with these values to the upper limit of the normal value of the corresponding feature + 1, and add a new binary variable to determine whether the corresponding feature of the observation unit is such an extreme value. Through this perturbation, three binary variables will be added to the original data set, and the observation samples that were temporarily deleted due to extreme values will also be returned to the data set.
- 2. Handling multi-classification items. This perturbation is for the "smoker" feature. Since the 7, 8, and 9 contained in this variable can all represent “uncertain whether to smoke", the “smoker" can be divided into three categories: smokers, non-smokers, and unsure smokers. This perturbation will use one-hot encoding to divide smoker into three binary variables, each representing a smoking situation, and the original “smoker" will be deleted, so this perturbation will increases 2 binary variables.
- 3. Handle unknown values. Take “height" as an example, where the three values 97, 98, and 99 can also represent unknown values. At this time, it is necessary to assign a value to this feature of a specific observation sample to return it to the data set. Here I decided to assign the mean of this feature to each person in this situation, and give the corresponding standard deviation perturbation, that is, the assignment process is to assign a normal distribution random number with the feature mean as the mean and the feature standard deviation as the standard deviation, so that it can be closer to the level of most people (It is assumed here that the corresponding features of these observation units are within a general range), and at the same time, it is hoped that this method can reduce the gap with the true value in the form of probability. This perturbation will not increase the number of features, but will change the observed values of some units.
- 4. Do not remove the “bmi" feature. Considering the calculation method of bmi, which contains non-first-order terms, it can still be considered to be added to the training features. It should be noted that since “bmi" is a numerical variable, if “bmi" is selected to be included in the dataset, other perturbations that will affect it will be performed together.
- 5. Standardization. Subtract the mean of each numerical variable and divide it by its standard deviation.

In this way, I got 32 new data sets (one of which is the same as the original training set). These data sets can be considered as different versions of datasets or perturbed datasets. These training sets will be used to train the ensemble algorithm and perform stability analysis at the same time.

5.5.2 Selection of Ensemble Algorithm

I first analyzed the results of each fit and made a histogram of each fit on four values before the selection. From the results in the histogram, the significances of True Positive Rate (TPR) and True Negative Rate (TNR) are not great, because the values of these two are relatively extreme in this environment, and the performance of Logistic Regression is significantly different from other algorithms. And because the difference in accuracy is relatively small (within 0.01), I chose to use the area under the ROC curve AUC for algorithm selection.

In the process of using the ensemble algorithm for single prediction fitting, I selected the algorithm with the largest AUC in the validation set as the selection algorithm, and also select the corresponding version of the test set to do the test. In the process of using the ensemble algorithm for ensemble prediction, I and selected the fits in the top 25% of AUC for ensemble prediction. Since it is a binary response variable, after obtaining the prediction results of each algorithm, the one with the most observation samples (that is, the value greater than 16) is selected to generate the final prediction result. When calculating ROC, the probability mean of all predictions is used.

6 Results and PCS Analysis

It should be noted that since I did not preset any random number seeds in the code (mainly reflected in the division of the data set), the results obtained from each run will basically vary to varying degrees. For example, the Decision Tree algorithm does not have a very good effect in most cases, but occasionally there will be a TPR value of 0.25 or even higher, and there will also be a TPR of 0. Therefore, all the results analyzed in the report are obtained from a certain run and are for reference only.

6.1 Results and Evaluations for Prediction on Validation Set

This subsection is to answer Chapter 11, EX 27, f; Chapter 12, EX 21, b and Chapter 13, EX 21, c d and e.
I combined the predictions of the first four algorithms on the validation set and obtained the following table in Figure 13 and the overall ROC curves in Figure 14. It can be seen that the accuracy difference of the four algorithms is not large, though there's a small increase compared with the algorithm that predicts all as negative (without diabetes), and the TNR difference is not large. In this case, the values of TPR and AUC are more important, and these are the statistics that differ greatly among the four algorithms. In this prediction result, Random Forest achieved good results after hyperparameter screening, with the highest TPR. Under this data structure, it is more necessary to predict more diabetic patients because the prevalence rate is only slightly more than 10% that would make the predictions overwhelmingly more likely that the patient would not have diabetes. Logistic Regression can achieve the highest area under the ROC curve (AUC), that is, its overall prediction effect is better. From the ROC curve, the results of the four algorithms are not different obviously, and only the ROC curve area of the Decision Tree algorithm looks smaller.
I also made distribution graphs in Figure 15 and histograms of the prediction probabilities in Figure

	Accuracy	rMSE	True Positive Rate	True Negative Rate	AUC
LS	0.909378	0.301034	0.142857	0.989525	0.864926
LOG	0.909905	0.300158	0.153989	0.988943	0.869450
CART	0.906744	0.305379	0.063080	0.994956	0.841400
RF	0.908500	0.302489	0.178108	0.984869	0.863413

Figure 13: Prediction evaluations of four algorithms

16 of all algorithms. From the probability distribution graph, I can roughly see the approximate prediction range and corresponding distribution of the prediction categories of different algorithms, and I can also roughly see the location of the best thresholds of some algorithms. Among them, the thresholds of Logistic Regression, Decision Tree and Random Forest are basically around 0.1 (about 0.11), while the threshold of Linear Regression algorithm is around 0.15, but the output of Linear Regression will produce negative values, and its value range is positive and negative infinity, so it is not appropriate to compare their thresholds directly. As for their prediction histograms, it can be basically seen that the confusion part of the prediction is basically in the probability interval of $0.1 \sim 0.35$. The overlap of the two distributions in this interval also shows the difficulty of predicting in this topic.
Regarding Decision Trees and Random Forests, I made an permutation importance map in Figure 17. From this figure, I can see that the four features "family_history_diabetes" "hypertension" "age" and "weight" are the most important features for prediction in these two methods. That is, in the algorithm that uses judgment statements as classification criteria, these four features are the factors that have

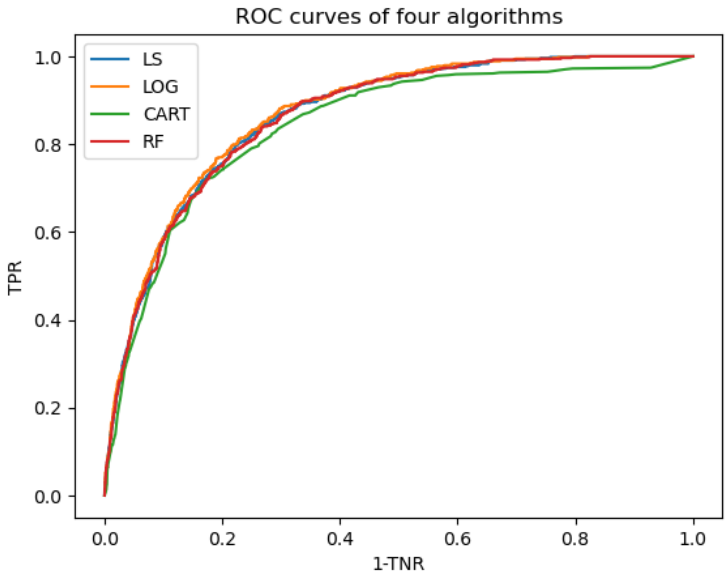

Figure 14: ROC curves of four algorithms

the greatest impact on whether the observed unit is sick.

For the ensemble algorithm, I select the fit for predictability based on the validation set and test it on the test set. Logistic Regression (usually) has a very good AUC among all other algorithms. From the AUC, I can know that Logistic regression has better prediction results than other algorithms totally, so the fit selected in the end is likely to be Logistic Regression trained by a version of training set. When using the ensemble algorithm for ensemble prediction, multiple different fits will be selected based on the Figure 18. Most of them belong to Logistic Regression and Random Forest algorithms, and a few belong to Linear Regression. The results of two different integrated algorithms are shown in Figure 19. It can be seen that the ensemble algorithm using multiple fits has improved accuracy, TPR and TNR compared to the best fit selected. Of course, changing the conditions for selecting the algorithm may have different results.

6.2 Stability Analysis for Bootstrap Samples

This subsection is to answer Chapter 11, EX 27, f and Chapter 12, EX 21, b.

In this part, I used the bootstrap method to resample the training set with replacement, obtained 100 sets of training data, and used each set of data to train each algorithm to obtain a new model to find the impact of rearrangement of the training set. Each new model predicted on the same validation set, and I plotted the ROC curves of all models of each algorithm in Figure 20, as well as the distribution of TPR, TNR and Accuracy in Figure 21. From the ROC curve, I can conclude that the Decision Tree algorithm is more unstable than the other three algorithms when the training set changes, and its ups and downs will be larger and more chaotic. From the distribution of the remaining three statistics, the fluctuations of TPR, TNR and accuracy of the Random Forest algorithm are relatively larger. Linear models that rely on calculations such as Linear Regression and Logistic Regression are always in a relatively stable state, while the other two algorithms that rely on judgment statements are more susceptible to changes caused by disturbances in the training set. Regarding Linear Regression and Logistic Regression, I additionally output their coefficient changes under this condition in Figure 22, that is, the coefficient distribution under 100 sets of different training data. A very interesting result is that the variables corresponding to the most stable coefficients in the two algorithms are “family_history_diabetes" “hypertension" “age" and “weight", which correspond to the most important variables in Decision Trees and Random Forests, while the coefficients of several variables with some but not high importance fluctuate relatively greatly. This also shows to some extent that I can get the importance of predictive variables by training algorithms.

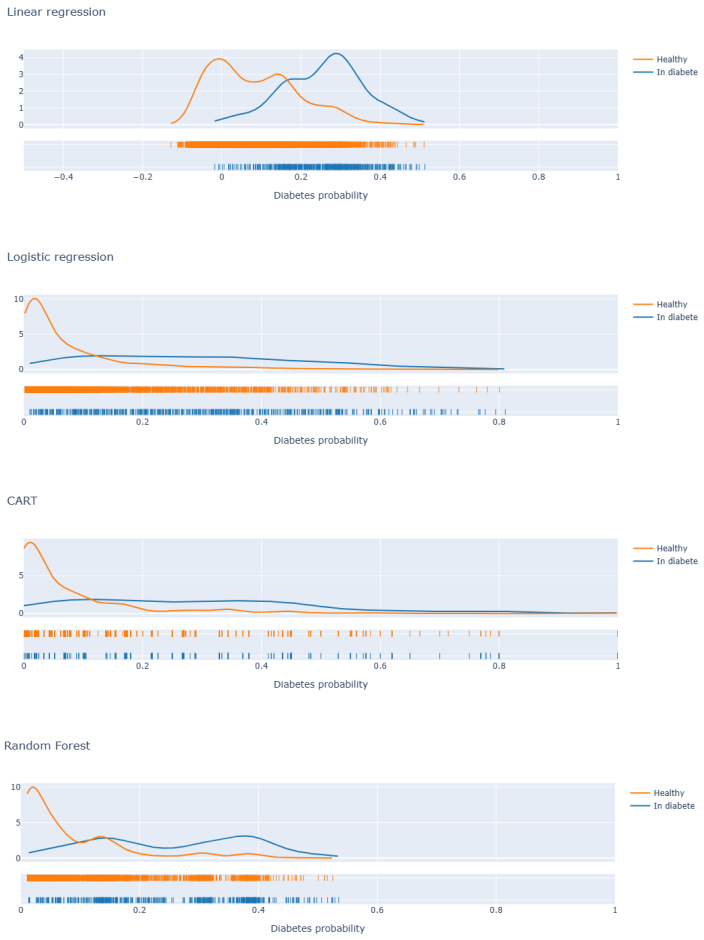

Figure 15: Probability density curves of four algorithms

6.3 Stability Analysis for Perturb

This subsection is to answer Chapter 11, EX 27, f and Chapter 12, EX 21, b.
In this section, I use the various perturbations mentioned in Section 5.5.1 to test stability to find the impact of data cleaning and preprocessing methods. I use the trained hyperparameters to train the corresponding algorithms on these 32 data sets, and each algorithm will have 32 different fits. I predicted the corresponding versions of the validation set for all fits, and finally got the ROC curves for each algorithm in Figure 23 and the distribution of the other three statistics in Figure 24. Similar to the result of bootstrapping the dataset, only the ROC curve of the Decision Tree algorithm showed some fluctuations, while the ROC curves of the remaining algorithms were quite stable, and the TPR and TNR of the Random Forest also had the largest fluctuations. Unlike the previous results, the accuracy of all algorithms fluctuated greatly when the data was perturbed. It can be preliminarily considered that the stability of the prediction results is not good enough in this set of data or under the default cleaning and preprocessing method. The introduction of extreme values and unknown values will shake the original prediction results. Judging from this result, there is still room for improvement in this project.

7 Conclusion and Discussion

This study conducted a data science analysis on whether people who participated in the National Health and Nutrition Survey had diabetes.

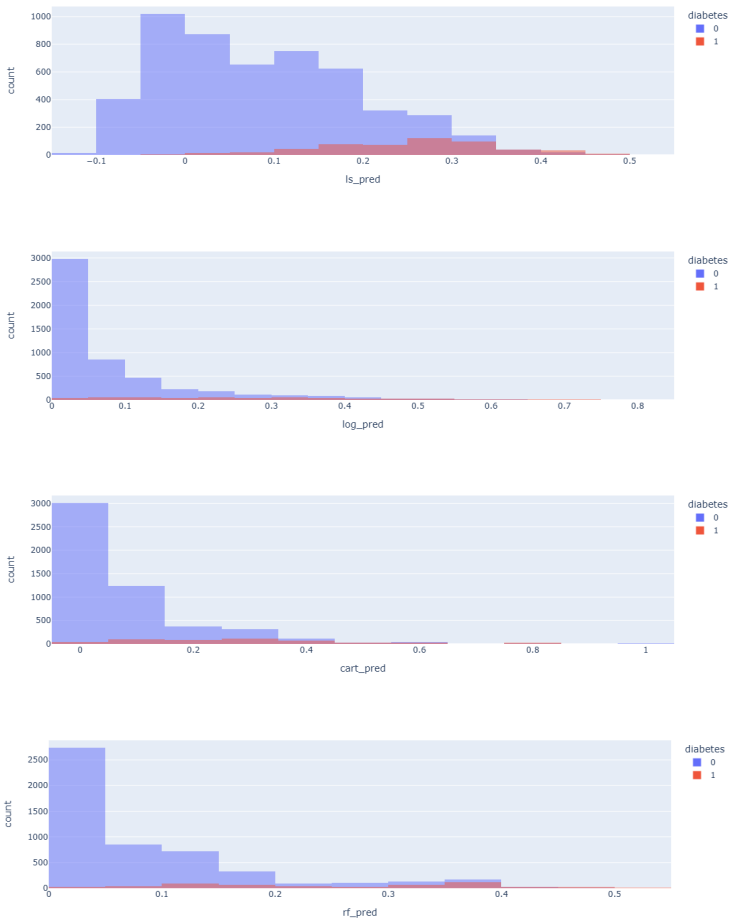

Figure 16: Histograms of prediction probabilities of four algorithms

This study used simplified data from the 2016 National Health and Nutrition Examination Survey. First, the data was explored and background mined, and the data collection process and each variable in the data were briefly understood, and it was determined that this was a binary prediction problem. The next step was to simply explore the data, find invalid values, missing values, etc. between variables, and use this to determine the default data cleaning process of the data, and then use random methods to divide the data before performing data cleaning.

The next step was to perform exploratory data analysis on the data, check the distribution, correlation and clustering results of the data, remove a feature with strong correlation between variables, and evaluate the clustering results based on the proportion of the data.

The study then carried out modeling and prediction. According to the needs of the task, four commonly used methods were selected: Linear Regression, Logistic Regression, Decision Tree and Random Forest and an ensemble algorithm related to the PCS. The study trained each algorithm on the training set, selected hyperparameters on the validation set and predicted on the validation set then obtained the prediction result table. In one of the results, Random Forest and Logistic Regression achieved the highest TPR and AUC values, respectively.

Finally, the study performed PCS analysis and trained the ensemble algorithm at the same time. The study obtained 100 new training sets through bootstrap, and trained each model with the same combination of hyperparameters, and finally obtained the corresponding ROC curve clusters and statistics, among which linear regression and logistic regression were the most stable. The study obtained 32 new data sets by changing the data cleaning and preprocessing methods, and also trained each model with the same combination of hyperparameters, not only obtaining the corresponding ROC curve clusters and statistics, but also obtaining the fits for the ensemble algorithm. Under the

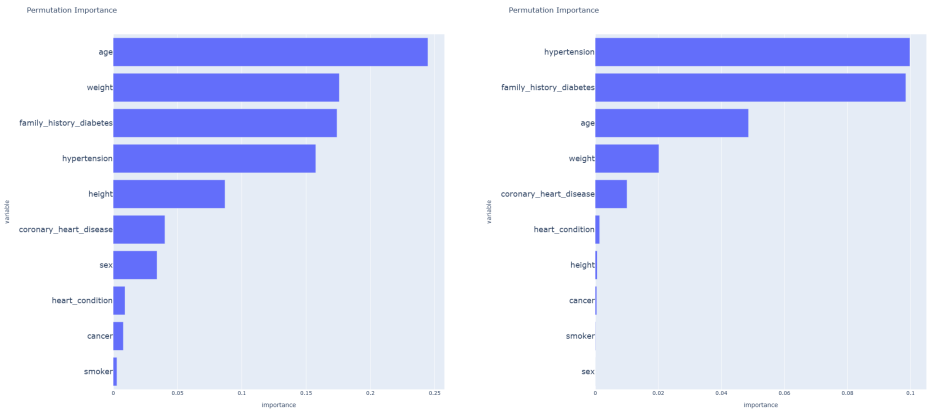

Figure 17: Permutation importance of Decision Tree (left) and Random Forest (right) algorithms

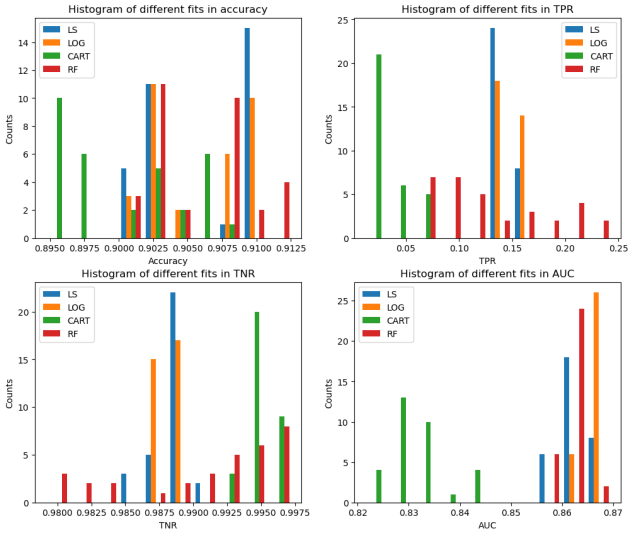

Figure 18: Accuracy, TPR, TNR and AUC of four algorithms with different versions of datasets

new datasets, the accuracy stability of all algorithms decreased. For the ensemble algorithm, the study used single optimal fit prediction and multiple optimal fit prediction respectively, and the multiple optimal fit prediction had multiple statistical improvements compared with the single optimal fit prediction.

In this project, I practiced the use of four algorithms, and became familiar with the entire process of data science analysis methods, and implemented all the codes from data cleaning-exploration-prediction-evaluation. On the basis of answering questions, Gaussian mixture models were used for clustering to explore the possible laws of data.

There is still room for improvement in this project. For example, in the data processing stage, not many changes were made due to the change in data structure. Perhaps a more complex data processing process could be tried from the original data. Clustering was evaluated only based on the data proportion, without evaluation on clustering-related statistics. There was no analysis of why the accuracy of the four algorithms was not greatly improved compared to predicting all negative (even occasionally the prediction of one algorithm was equivalent to the prediction of all negative). The parameters of the four algorithms were not fully understood, and perhaps their predictions still had room for improvement.

	Dataversion	Algorithm	Accuracy	True positive rate	True negative rate	AUC
0	Test set corresponding version	LOG	0.902443	0.149047	0.98748	0.844643

	Dataversion	Algorithm	Accuracy	True positive rate	True negative rate	AUC
0	Test set corresponding version	Ensemble	0.902937	0.149306	0.987869	0.844543

Figure 19: Ensemble algorithms' prediction results (the upper is the best one and the lower is the combination of multiple fits)

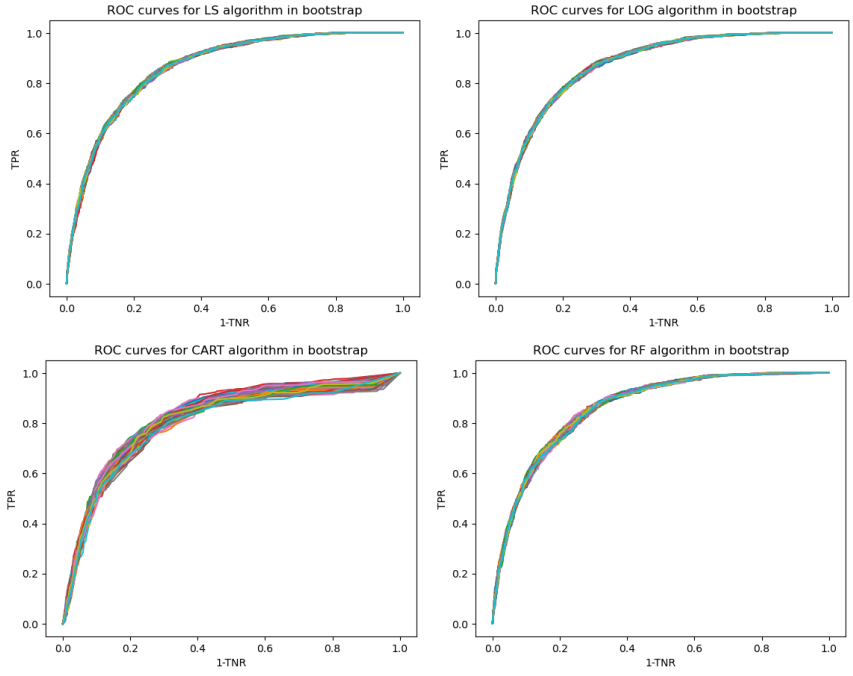

Figure 20: ROC curves of four algorithms with bootstrap

8 Codes

The codes are on https://github.com/supperXpower/MATH2313_Final_Project.

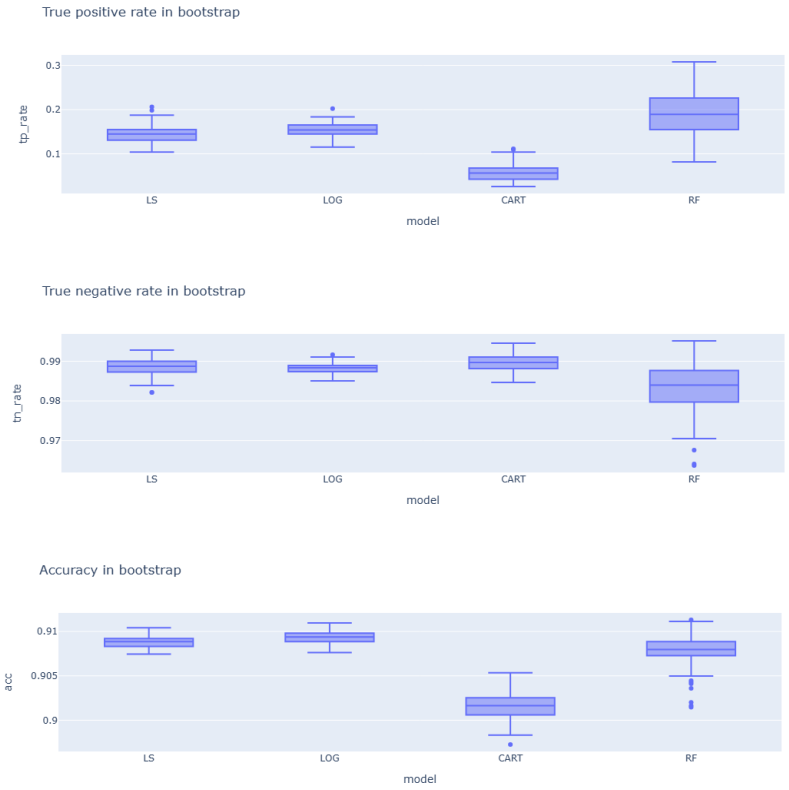

Figure 21: TPR, TNR and Accuracy of four algorithms with bootstrap

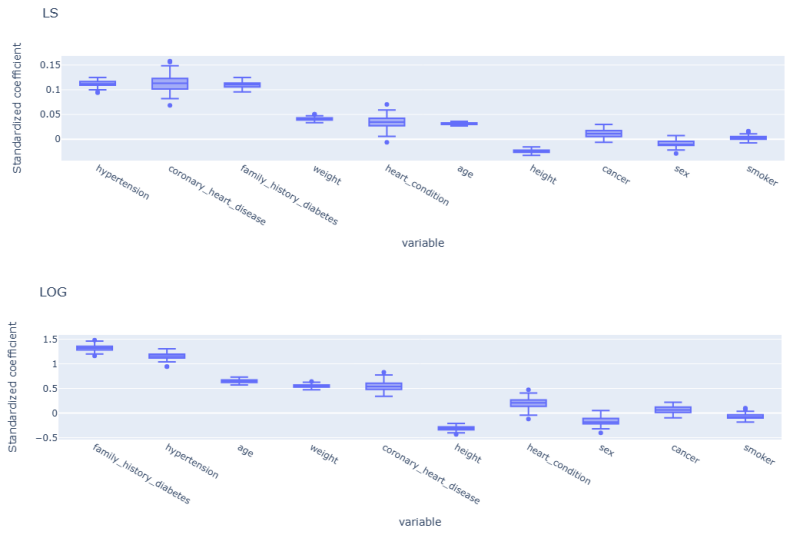

Figure 22: Coefficients of Linear Regression and Logistic Regression with bootstrap

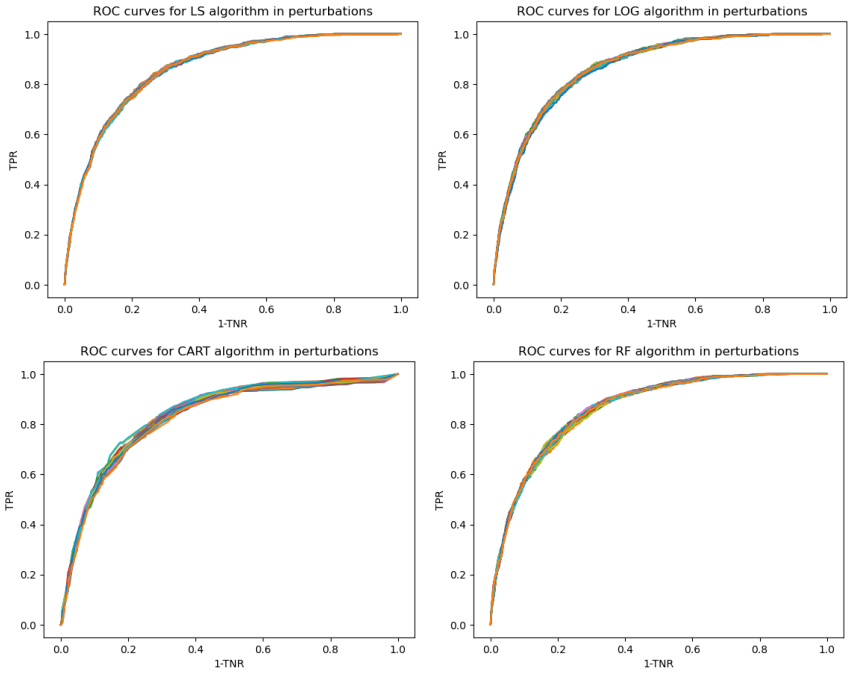

Figure 23: ROC curves of four algorithms with perturbation

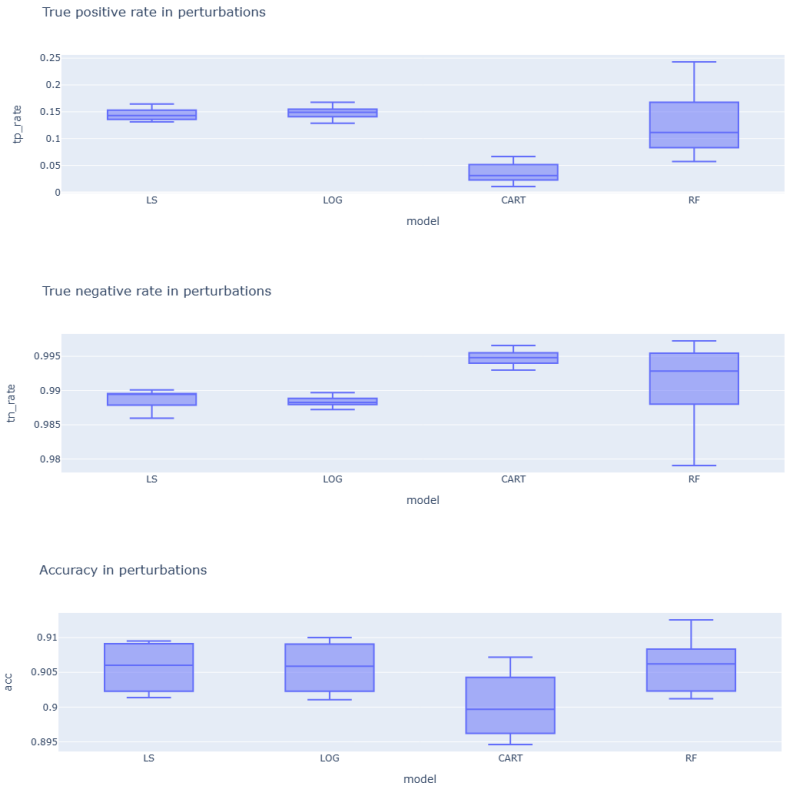

Figure 24: TPR, TNR and Accuracy of four algorithms with bootstrapperturbation