Github:https://github.com/kennethxzq/Zhengqi-Xing

Catalog

1. Business and/or Situation understanding

1.1 Identify the objectives of the situation

It is mentioned in the UN Sustainable Development Strategy that there should be healthy people and promoting the well being of people of all ages.Therefore, breast cancer, as the second major health problem for women, should be taken seriously.

Cancer is a disease in which the body's cells grow out of control.After skin cancer, breast cancer is the most common cancer among women in the United States.Breast cancer, while declining over time, remains the second leading cause of cancer death in women overall.In the United States, about 250,000 cases of breast cancer are diagnosed each year, 99% of them in women.

In medical activities, machine learning and data mining can better help people to judge whether breast cancer is benign or malignant. Of course, these data come from medical devices, but it will be much higher than the correct rate of manual judgment. A suitable algorithm may have an accuracy rate of more than 95%. Therefore, we need to construct an appropriate data mining process to process these known data, so as to determine whether the tumor is benign or malignant. This can help these patients with appropriate medical treatment as early as possible, so as to achieve the right medicine.

Our main research topics are as follows:

1. Biopsy data on breast cancer tumors can be used to predict whether the tumor is benign or malignant.
2. To find out which of the variables had the greatest impact on breast cancer.
3. We need to get these results right more than 95% of the time for our purposes. Because the accuracy of successful prediction greatly affects the health of patients.

If our prediction accuracy is more than 95%, we are considered to have initially solved the problem, and the more accurate the prediction, the better.This can not only be used for clinical diagnosis, but also to give a reference to places where medical conditions are poor.This is more conducive to reducing breast cancer mortality.

1.2 Assess the situation

Demand:

A Data Mining Scenario Using CRISP-DM. This is data from a breast cancer biopsy, so even though we did it through data analysis it's still a cross-domain data mining.So we need to get a medical expert to help us determine whether our results are accurate.

Data:

The project required data mining and data analysis, so the most critical point was to find the

right data.This article quoted the Breast Cancer Wisconsin Data Set.Assume that these data sets are correct data sets (data sets collected from patients), and that the data in the data set must be true.To some extent, this data set alone can be used to predict the risk of breast cancer.In life, of course, there are other factors that can interfere with the prediction model.But the forecast is still very useful.

Risk and contingencies:

This project assumes that all other things being equal, the analysis is carried out.It may not be very accurate analysis, if accurate analysis, this needs to be a doctor's diagnosis and expert identification.Therefore, this model is suitable for theoretical inference, but not for real detection.It can tell you that the probability of disease is high, can be carried out as early as possible in the relevant aspects of the physical examination.Rather than relying solely on this model to infer whether or not you have breast cancer and your likelihood of developing breast cancer in the future.

There is a big risk with this model that if we get our predictions wrong, there will be a huge impact.For example, a patient has a benign tumor, but we test it as a malignant result.The patient thinks he has a malignancy, so he's going to have a lot of negative emotions and that's not going to help with the treatment.And a failure to properly use medical facilities to treat the disease.On the other hand, if the patient is malignant and is diagnosed as benign, treating the patient in the wrong way may miss the best time and result in the patient's death.This problem is related to life, so be sure to ask a specialist to do a second diagnosis, to ensure that the diagnosis is correct.

1.3 Determine data mining objectives

Through pre-study of data mining, translating the initial objectives of this project into data mining. The preliminary research objectives to be completed are as follows:

1. The goal of data mining is to find an appropriate model to predict the probability of breast cancer. This accuracy rate must be the higher the better for more than 95%, which is related to life and health.

2. The key factors and correlations of breast cancer were found in the model. There are many variables in our data, so we must screen out the key data for judgment.Noisy data may affect the judgment of the results and the correctness of the final prediction.

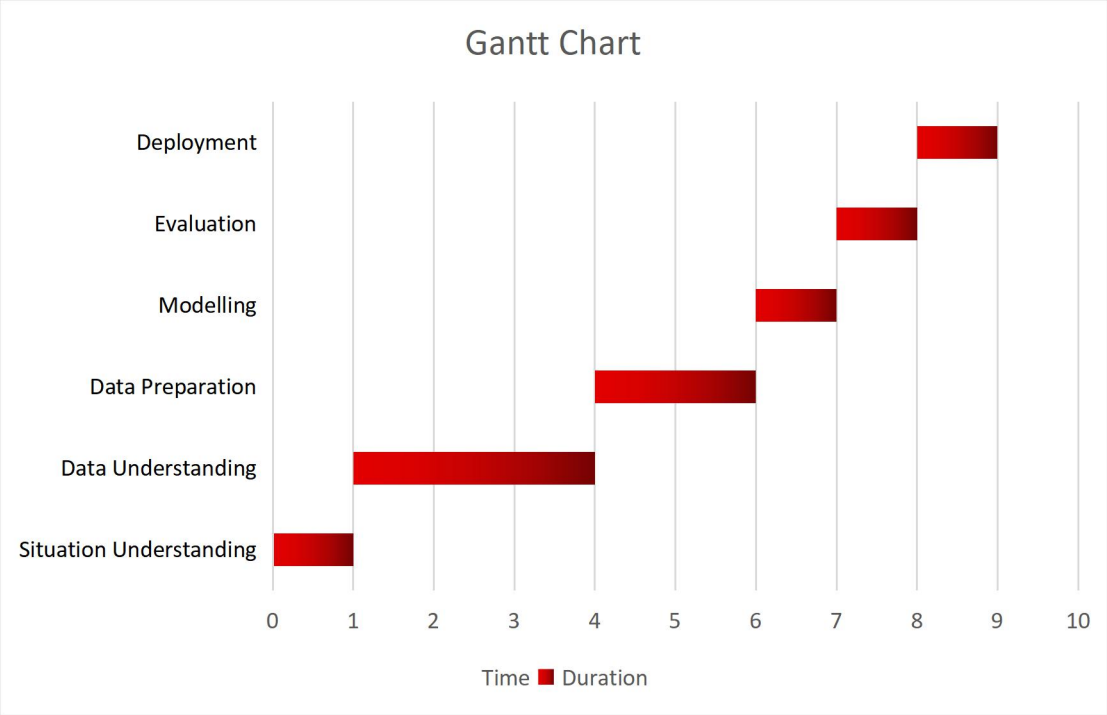
3. Search for the best model among these models. Because the data mining model we use is finally classification, classification is divided into many models, such as random forest, decision tree, KNN and so on.So choosing the right model is going to get you twice as far.

4. Seek out the success rate of the best model for predicting breast cancer. If the correct model and parameters are selected, the final prediction accuracy will be the highest.This is the time to evaluate the accuracy and it's best to do cross-validation to verify the accuracy of the results.

1.4 Produce a project plan

Phase	Time	Resources	Risks
Situation understanding	1 week	All analysts	Economic change
Data understanding	1 week	All analysts	Data problems, technology problems
Data preparation	3 weeks	Data mining consultant, some database analyst time	Data problems, technology problems
Modeling	2 weeks	Data mining consultant, some database analyst time	Technology problems, inability to find adequate model
Evaluation	1 week	All analysts	Economic change, inability to implement results
Deployment	1 week	Data mining consultant, some database analyst time	Economic change, inability to implement results

Plot 1.1



Plot 1.2

Data preparation and processing is very important, so schedule a certain amount of time to find the data and process it. One of the key success factors of data mining is the rationality of data.A good data will produce the desired results. Then it is necessary to carry out the process of modeling analysis and cyclic modeling, which will improve the accuracy of prediction step by step.

Finally, we get the correct rate we want to complete the goal of data mining.

2. Data understanding

2.1 Collect initial data

Import the data set to Jupyter notebook.

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Why did we choose to collect this data?

Because clinical trials know that these data can tell whether breast cancer is benign or malignant.These data include the area, the circumference, the radius of some slices and so on.

Where id is the patient number, and the variable diagnosis is the classification variable: benign and malignant. Everything else is an observable variable.

```
root
 |— id: integer (nullable = true)
 |— diagnosis: string (nullable = true)
 |— radius_mean: double (nullable = true)
 |— texture_mean: double (nullable = true)
 |— perimeter_mean: double (nullable = true)
 |— area_mean: double (nullable = true)
 |— smoothness_mean: double (nullable = true)
 |— compactness_mean: double (nullable = true)
 |— concavity_mean: double (nullable = true)
 |— concave points_mean: double (nullable = true)
 |— symmetry_mean: double (nullable = true)
 |— fractal_dimension_mean: double (nullable = true)
 |— radius_se: double (nullable = true)
 |— texture_se: double (nullable = true)
 |— perimeter_se: double (nullable = true)
 |— area_se: double (nullable = true)
 |— smoothness_se: double (nullable = true)
 |— compactness_se: double (nullable = true)
 |— concavity_se: double (nullable = true)
 |— concave points_se: double (nullable = true)
 |— symmetry_se: double (nullable = true)
 |— fractal_dimension_se: double (nullable = true)
 |— radius_worst: double (nullable = true)
 |— texture_worst: double (nullable = true)
 |— perimeter_worst: double (nullable = true)
 |— area_worst: double (nullable = true)
 |— smoothness_worst: double (nullable = true)
 |— compactness_worst: double (nullable = true)
 |— concavity_worst: double (nullable = true)
 |— concave points_worst: double (nullable = true)
 |— symmetry_worst: double (nullable = true)
 |— fractal_dimension_worst: double (nullable = true)
 |— _c32: string (nullable = true)
```

Plot 2.1

The data from the Kaggle.com. (UCI Machine Learning,2016)This data can be mined and analyzed.

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

If you don't know what every statistic means or what it means for breast cancer, be sure to look at what every statistic means.I was confused when I first saw this data.But a study of the data and a study of the knowledge of breast cancer will reveal what this data represents.

2.2 Describe the data

It has 569 sets of data. Computation of features from digital images of fine needle aspiration (FNA) of breast masses. They describe the characteristics of the nuclei present in the image. (plot2.2)

Description:

All data includes various areas of tumor slices, including cross-section, radius, and maximum and minimum conditions. In addition, there are some data about the average value of these data, the worst-case value and the best-case data. These data can help us to judge whether the tumor is malignant or benign. Next, there are some specific descriptions of these data.

It has many types of data:

1) ID number

2) Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

a) radius (mean of distances from center to points on the perimeter)

b) texture (standard deviation of gray-scale values)

c) perimeter

d) area

e) smoothness (local variation in radius lengths)

f) compactness (perimeter^2 / area - 1.0)

g) concavity (severity of concave portions of the contour)

h) concave points (number of concave portions of the contour)

i) symmetry

j) fractal dimension ("coastline approximation" - 1)

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800

8 rows × 32 columns

Plot 2.2

The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

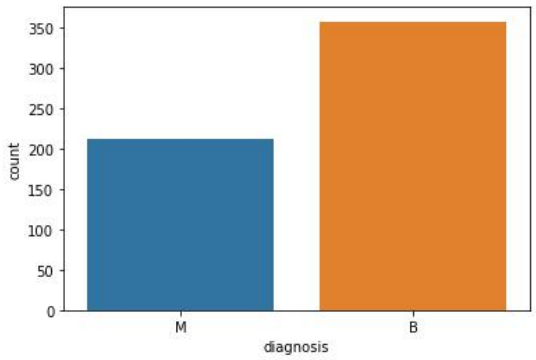
All feature values are recoded with four significant digits.

Missing attribute values: none

Class distribution: 357 benign, 212 malignant

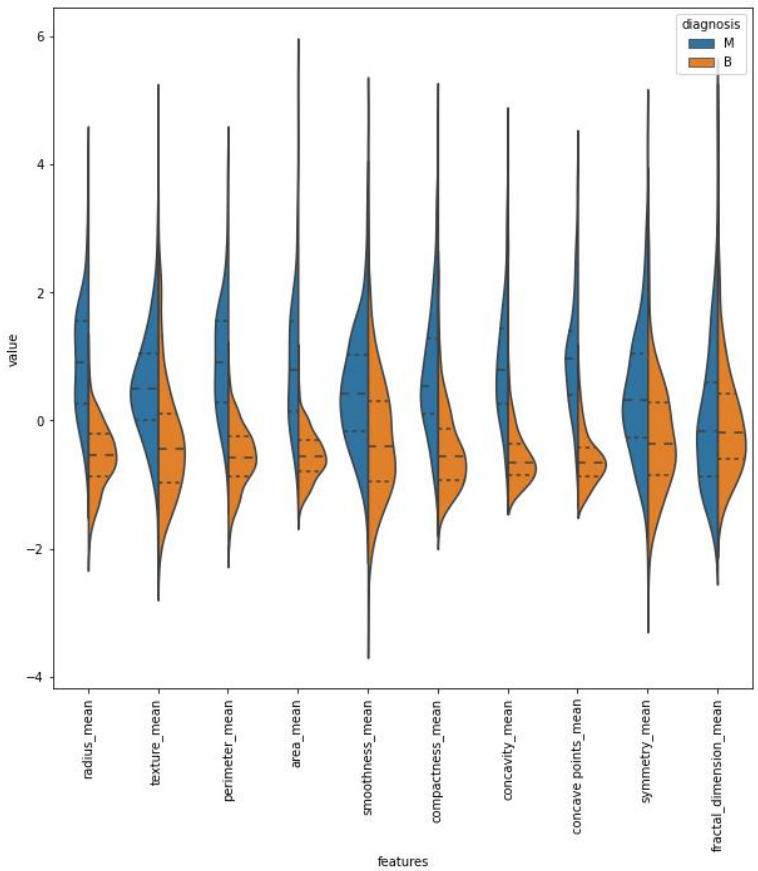
2.3 Explore the data

First of all, our most important thing is to find the key variables: malignant and benign. So I will list this data in a separate bar graph to show that blue is malignant and orange is benign. We found that the proportion of malignant tumors is relatively low, indicating that most people have breast cancer tumors that are benign, indicating that timely judgment can better help us treat benign tumors. Because in comparison, benign tumors are better to be treated.



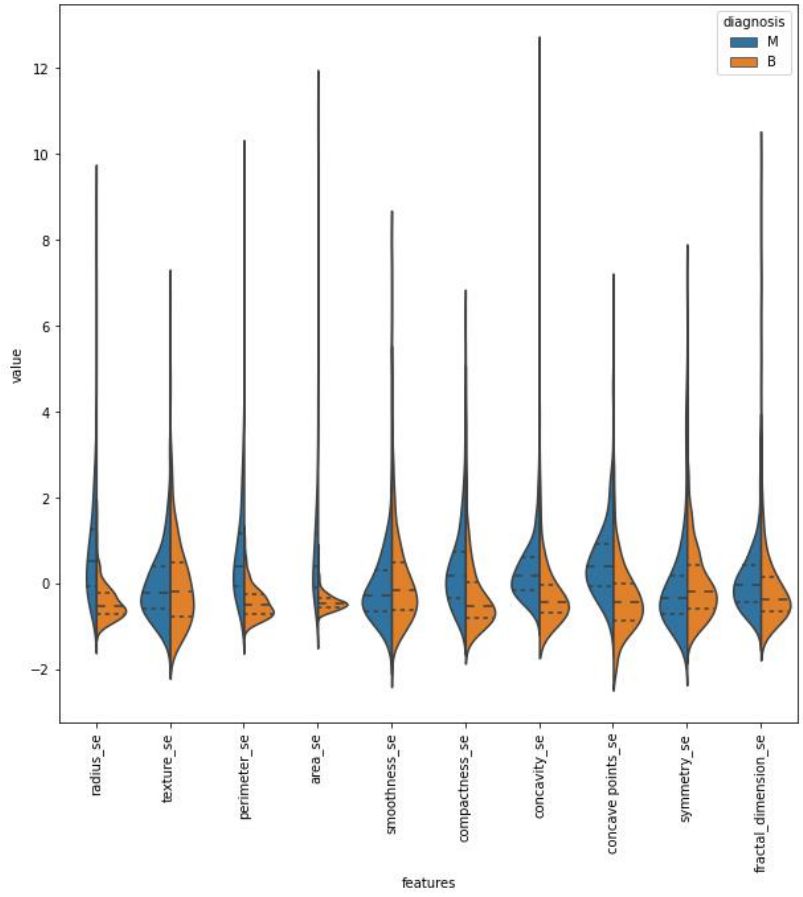
Plot 2.3

In order to better visualize the data, we used Violin or Swarm graph to represent it. Since we have many data variables, we divide them into three groups to make them clearer. Each group consists of ten features.



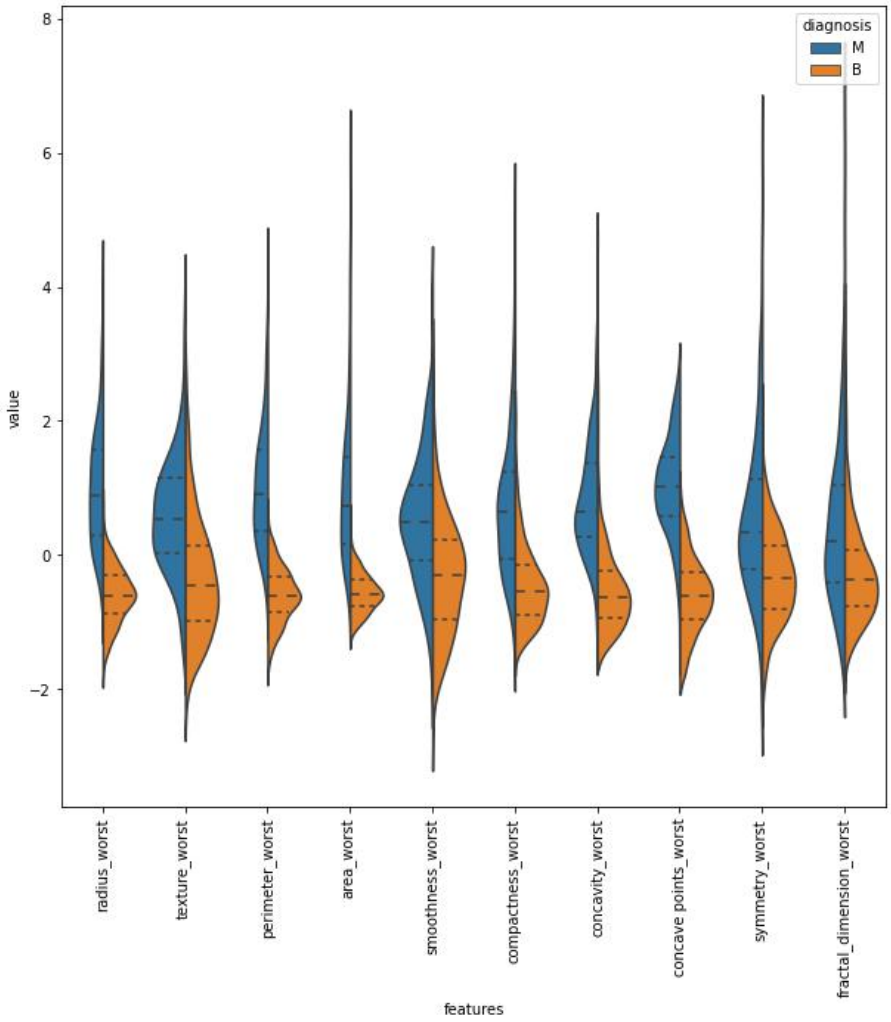
Plot 2.4

Lets interpret the plot above together. For example, in texture_mean feature, median of the Malignant and Benign looks like separated so it can be good for classification. However, in fractal_dimension_mean feature, median of the Malignant and Benign does not looks like separated so it does not gives good information for classification.



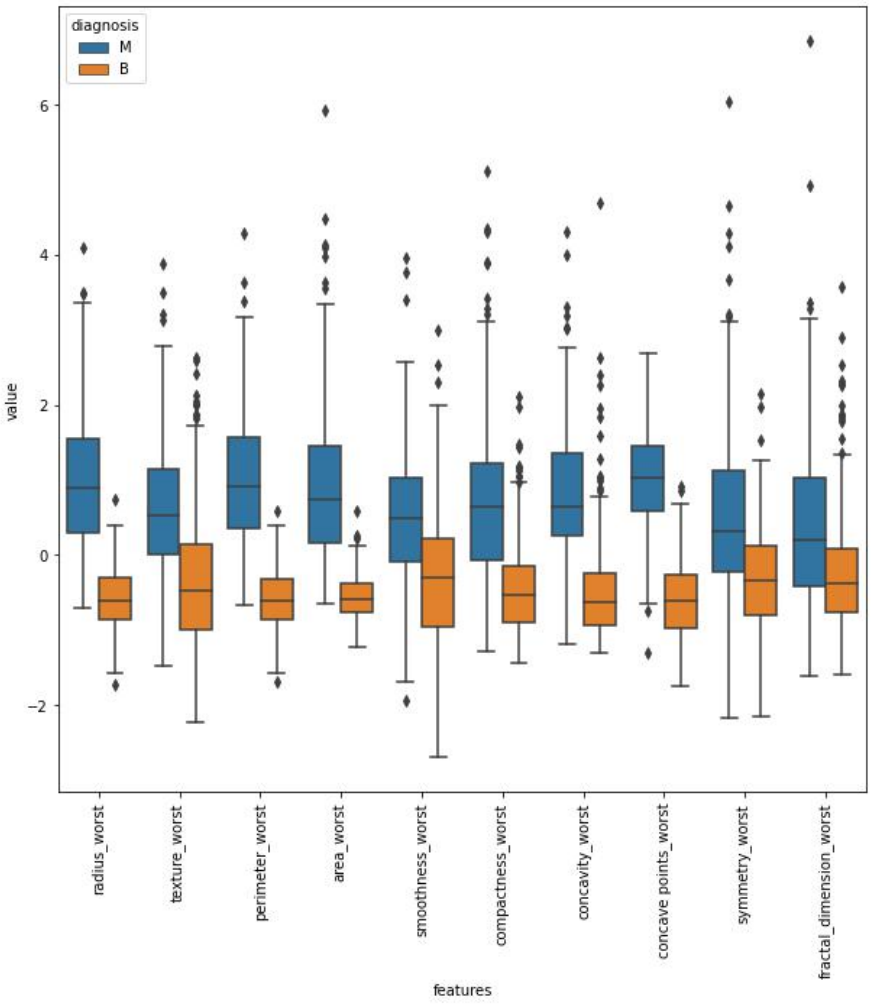
Plot 2.5

The above are the ten characteristics of the second group. We can find that some variables can distinguish benign from malignant well, but some variables are very vague. So the next operation may involve retaining only the more important features instead of all the features.



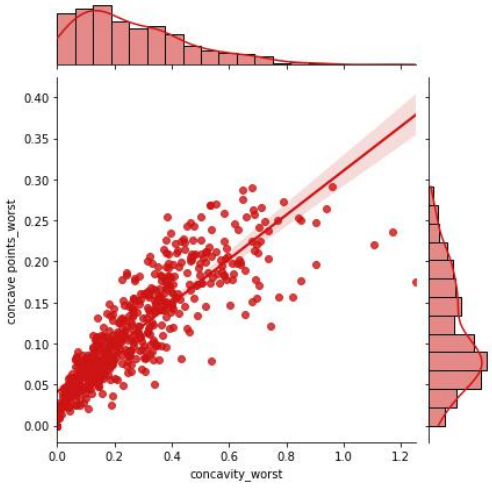
Plot 2.6

This is the data of the last group of characteristics, which is different from the second group in the figure. But looking at the picture, we can still find that some of the data features are more obvious, while the other parts are not very obvious. Some of these features, such as convex points_worst, symmetry_worst, and fractal_dimension_worst, are not very beneficial to our analysis results.



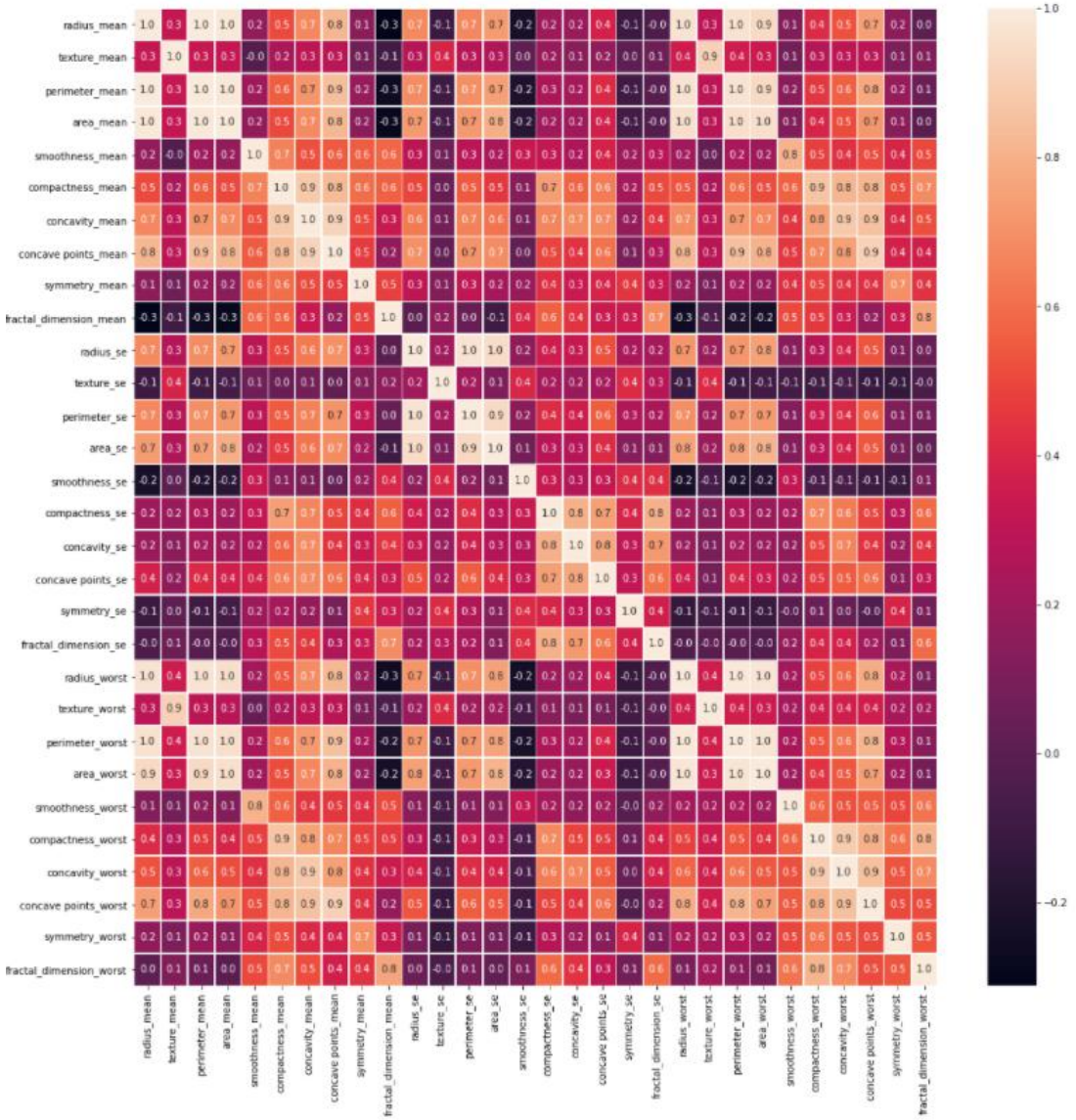
Plot 2.7

It can be further found from the above figure that the last two variables are indeed not conducive to our judgment of the results. To further compare the two features, we use a joint graph. Look at the following union diagram, and it is indeed relevant.But we don't know for sure, it's just an idea. And we can see from the graph that they do have some correlation, and it's almost linear.



Plot 2.8

Of course, this is just a relationship between two variables, so let's have all the variables present in one graph.



Plot 2.9

This is a very good case for an illness judgment, because we know that it's the significant variables that tend to get sick that tend to be larger than the average, so can we see from that that the distribution of the outstanding numbers is skewed to the left.It turns out that there are some right-skew values that affect the distribution of the mean and then they're more likely to get sick.If all the data is normally distributed then it's not going to be accurate for what we're predicting. Diseases are often caused by a number of salient variables.

From this graph (Plot2.4), we can find that most of the variables in it are related to the final result.Through the depth, we can know the influence of each part on the final classification result and the relationship between the data.It shows that the data can explore the final conclusion, which is why we choose this data.As mentioned earlier, this data set has been proven by the medical field to have some basis for whether breast cancer is benign or malignant.This chart shows that each variable has a certain degree of influence on the final result.In the following

modeling, we will select the more important data to make the prediction, and delete the irrelevant data or the data that will have a negative impact on the result.

2.4 Verify the data quality

First we look for missing values in all the data. The diagram below:

```
print(pd.isna(data).sum())
```

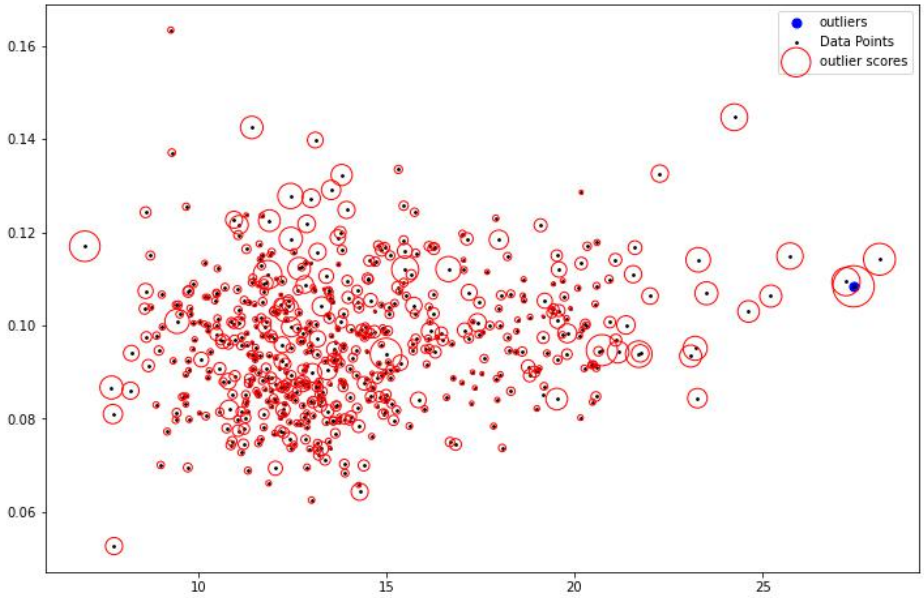
```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
Unnamed: 32                569
dtype: int64
```

Plot 2.10

After checking all the missing values, we found that all the data were complete with 569.There is no missing value in this group of data, indicating that this group of data is very good and we do not need to manually supplement the missing value, which will make the final result more accurate.

The last column named "Unaname: 32" seems like an erronous coloumn in our dataset. We might probably just drop it. Most of the columns seem to have a numeric entry. This would save our time from mapping the variables. The ID column would not help us contributing to predict about the cancer. We might as well drop it.

And then we look for outliers:



Plot 2.11

I used Python's visualized graphs to find some local outliers, indicating that although all variables in the data set were of high quality, there were no missing values. However, there is still a certain amount of outliers that will be removed in a later step. However, it can be seen from the figure that most of the values are within the normal range, indicating that this data set is very suitable for our data mining target.

Overall, the data set is of high quality and suitable for data mining. Outliers and low-correlation variables will be removed later in the procedure. The entire data is complete except that all the variables in one column are NULL. This work will be carried out in data preparation.

3. Data preparation

3.1 Select the data

We know that in the work of data mining, the most important thing is the selection of data. So this step usually accounts for more than half or even more of the total step time. Choosing a suitable data set will often make our prediction accuracy higher, and it will also make machine learning faster. Only one column of the data set we chose was discarded by us, and the other data will be used by us. And in this data set, we have already divided the best and worst values of each feature and the average value of each feature for us. This is more suitable for us to do better machine learning and deep learning.

Target selection:

First of all, the ID in the data set is the only variable, which represents the person recording the data, so it is not a predictive variable.DIAGNOSI is the result of prediction, so it is not a predictive variable.However, all the other variables may be one of the factors affecting the incidence of breast cancer, so all the other variables were set as test variables.

In addition, ID is set as a unique variable to distinguish each patient.Diagnosis is set to FLAG variable and the results are B (Benign) and M (Malignant).B is a benign tumor, which may have no impact on life, while M is a malignant tumor, which has an impact on life and may lead to death.

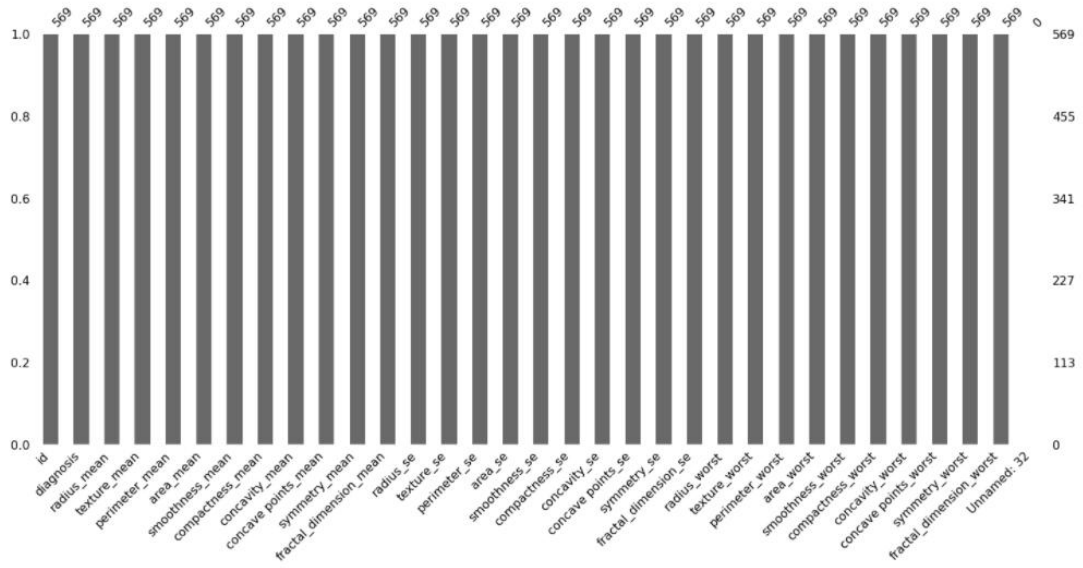
Feature selection:

Because our goal is to predict whether a breast cancer tumor is benign or malignant, we should select variables other than the two variables of ID and diagnosis. And because the quality of all our variables is relatively good, other variables can be selected. And our data set is almost perfect without the last column. Because if you look at the way the data is presented, you can see that part of the data represents the average, part of the data represents the worst case.In addition, each data has some relationship to the final classification. In addition, all the remaining feature sets not only contain the best and worst case of each feature, but also the average value of the feature, which is more conducive to our data mining work.

3.2 Clean the data

Data cleaning is related to the quality of the data. We know that not all data will produce good results for our predictions, and some variables will produce some negative results for our predictions. So what we have to do here is to extract those good features, and then use them in subsequent data mining. In addition, we have to delete some unnecessary outliers of features, because these outliers will affect the final correct rate.

Determine data integrity:



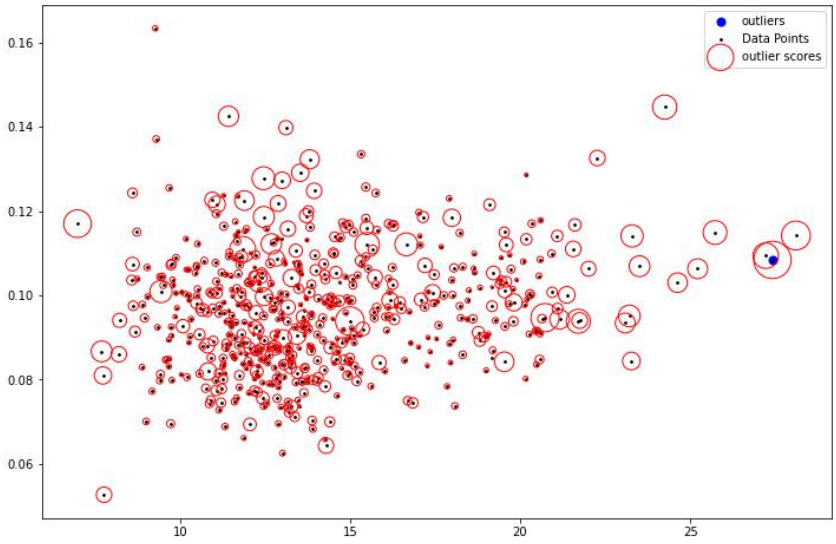
Plot 3.1

We found that all the data are 569, which matches our total. But we know that all the data in the last column is null. This is something we don't want to see, so we can ensure that all the data is complete by combining this picture with the previous picture and deleting the last column.

To deal with the missing and outlier value:

Because all variables in all rows except the last column are valid variables. So we don't need to deal with missing values. If there are really missing values, it is the last column. We will delete

the last column in the following process to keep all other features.



Plot 3.2

In addition to missing values, another anomaly is the outlier.In the figure (Plot3.2), we find that all the points circled in red are outliers.

For these outliers, we will delete them directly instead of replacing them. Because these data come from actual measurement data. If we use some mean or median for simple processing, although it may improve the correct rate in machine learning, it will lead to more serious results in actual judgment. We cannot modify this data because it does not conform to our results. For example, the data of a person who has a malignant tumor may be biased. But this is indeed possible in the actual situation. If this data is modified, it will lead to errors in its prediction results. Of course, this is a machine learning model, it needs higher accuracy, so we can't ignore the outliers. In the end, I chose to delete these outliers.

Change the feature flag:

```
df.diagnosis.replace({"M":1,"B":0},inplace=True)
df.diagnosis.unique()
```

```
array([1, 0], dtype=int64)
```

Plot 3.3

We change the data of the flag feature variable, changing M and B to 1 and 0. Among them, 1 represents that the tumor is a malignant tumor, and 0 represents that the tumor is a benign tumor. We know that in the work of data mining, the purpose of all data cleaning is to make the process of data mining clearer and more visible. Compared with letters, numbers are more conducive to the judgment of the machine, after all, computers are also composed of binary. So I converted these two variables into 0 and 1 separately.

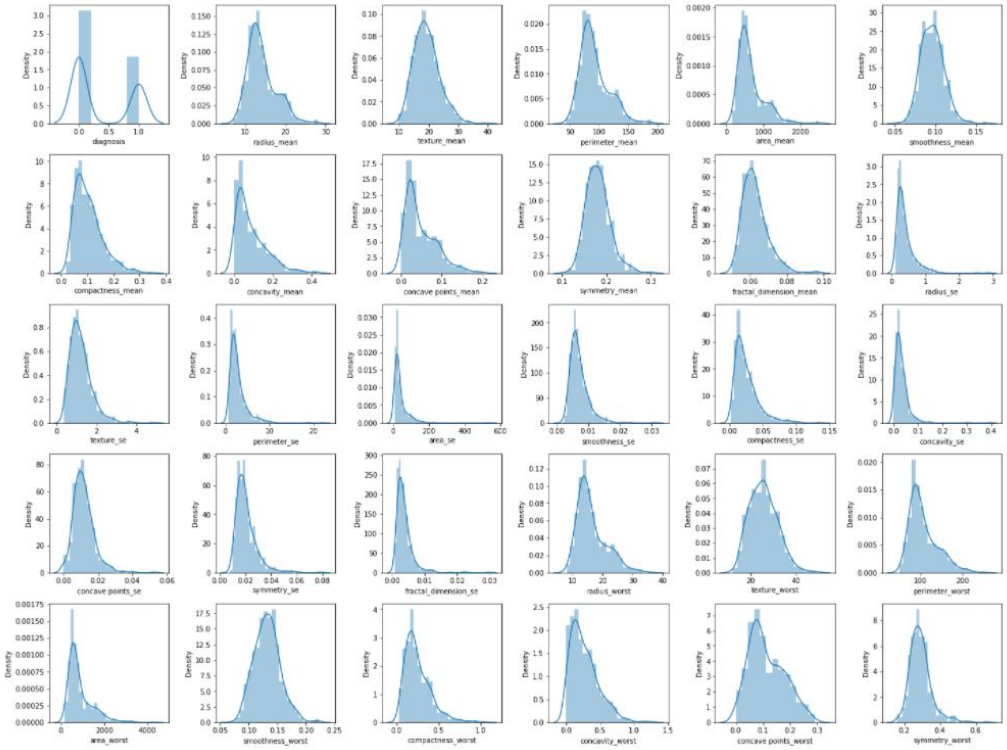
3.3 Construct the data

```
df.drop(['Unnamed: 32','id'], axis = 1 , inplace=True)
df.columns
```

```
Index(['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')
```

Plot 3.4

By deleting the outliers and deleting the ID column we get the following data.As can be seen from the figure, most of the data are normally distributed or left skew.This helped a lot with our predictions, and it showed that our data structure and data quality were very good.In line with our goal of data mining, these data may be sorted out or deleted later, but at present, this is the task of data preparation basically completed.The distribution of each variable in the final data set is as follows.



Plot3.5

Because all the data were related to breast tumor transection, we could not filter by category.And here we should be able to predict the benign and malignancy of the tumor model using just this data.Unlike other data where there might be other variables, like time or cancer rate or whatever, here we're just looking at the data from the cross-section of the tumor.

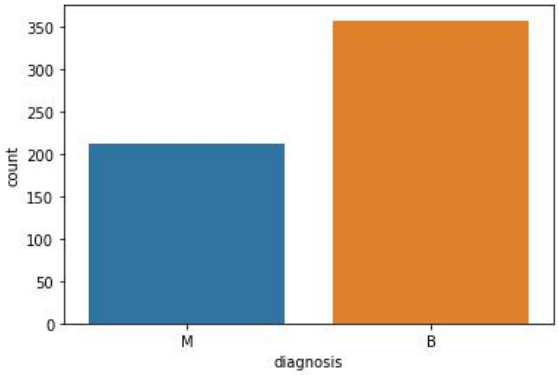
With this data set, there were 30 variables that predicted the risk of breast cancer.This

meets the requirements of data mining and data analysis.

3.4 Integrate various data sources

In the research of data mining, there are often multiple files of data. At this time, we need to connect each file with a key. This process is like connecting two tables through the primary key in the database. This process is often necessary because the data we collect often does not exist in the same file.

Because there is only one data set, which contains thirty variables, no consolidation is required. But in addition to integrating the data, we also have to separate the data to better study the whole process. Here I separated the flag variable from other variables and got the following results.



Plot 3.6

Through observation, we can know that most of the people in our set of data are benign tumors. So this meets the requirements we just started. Because malignant tumors can even affect people's lives, benign tumors can be cured if they are treated in time. In addition, if the results predict that there are too many malignant tumors, then we need to consider whether the results are wrong. If randomly selected from these people as a test set, then people with benign tumors also account for the majority.

3.5 Format the data as required

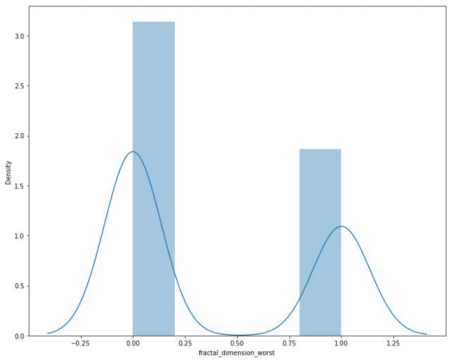
In the work of data mining, the formatting of data is very important. It will help us better analyze the data and improve the accuracy of the results. If we want to categorize the results binary, then we first need all the data to meet the requirements. In addition to the flag data, other data must be formatted. The result of formatting is float format or int format. In this way, the data can be better classified. This is also one of the necessary processes of machine learning algorithms.

After previous data cleaning and dividing variables into 0 and 1, we get the following plot:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 31 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   diagnosis                569 non-null    int64
 1   radius_mean              569 non-null    float64
 2   texture_mean             569 non-null    float64
 3   perimeter_mean           569 non-null    float64
 4   area_mean                569 non-null    float64
 5   smoothness_mean          569 non-null    float64
 6   compactness_mean         569 non-null    float64
 7   concavity_mean           569 non-null    float64
 8   concave points_mean      569 non-null    float64
 9   symmetry_mean            569 non-null    float64
 10  fractal_dimension_mean   569 non-null    float64
 11  radius_se                569 non-null    float64
 12  texture_se               569 non-null    float64
 13  perimeter_se             569 non-null    float64
 14  area_se                  569 non-null    float64
 15  smoothness_se            569 non-null    float64
 16  compactness_se           569 non-null    float64
 17  concavity_se             569 non-null    float64
 18  concave points_se        569 non-null    float64
 19  symmetry_se              569 non-null    float64
 20  fractal_dimension_se     569 non-null    float64
 21  radius_worst             569 non-null    float64
 22  texture_worst            569 non-null    float64
 23  perimeter_worst          569 non-null    float64
 24  area_worst               569 non-null    float64
 25  smoothness_worst         569 non-null    float64
 26  compactness_worst        569 non-null    float64
 27  concavity_worst          569 non-null    float64
 28  concave points_worst     569 non-null    float64
 29  symmetry_worst           569 non-null    float64
 30  fractal_dimension_worst  569 non-null    float64
dtypes: float64(30), int64(1)
memory usage: 137.9 KB
```

Plot3.7

All of our data is floating-point. DiagnoISI is a class variable and is converted to 0 and 1 for our modeling purposes. The data format here is correct so we just need to change the object variables of diagnosis to 0 and 1.



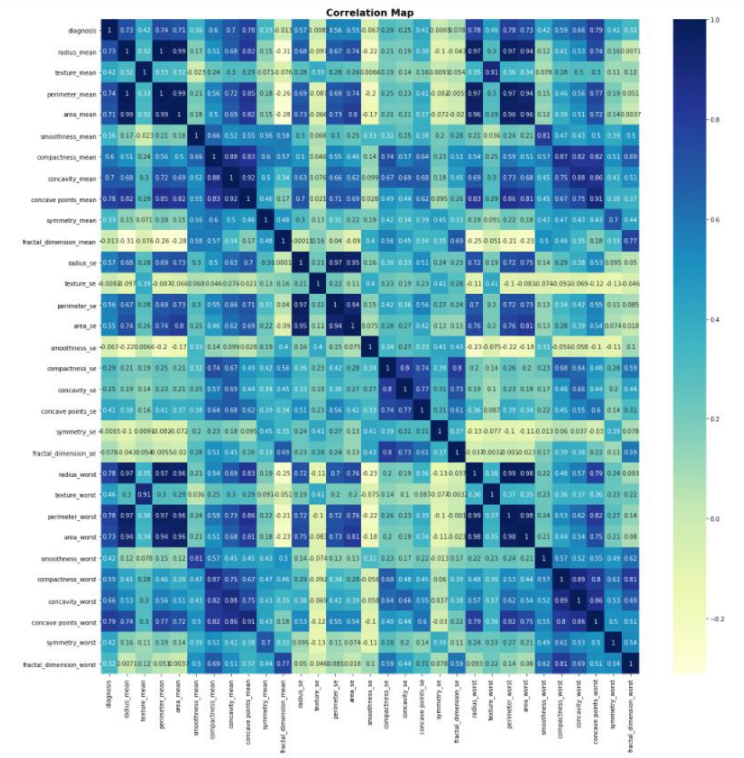
Plot 3.8

After all the processing we have 29 feature variables and one FLAG feature variable. These converted variables and data meet the requirements of our data mining, so we can proceed with the following work.

4. Data transformation

4.1 Reduce the data

In some special analysis, it is unnecessary to have a large data set (especially many features). Therefore, in order to analyze accurately, we need to select the features related to the prediction variables. I chose to use feature selection to achieve this. There are many factors that need to be taken into account in data mining.The magnitude of their impact on the outcome will invalidate or validate the data.



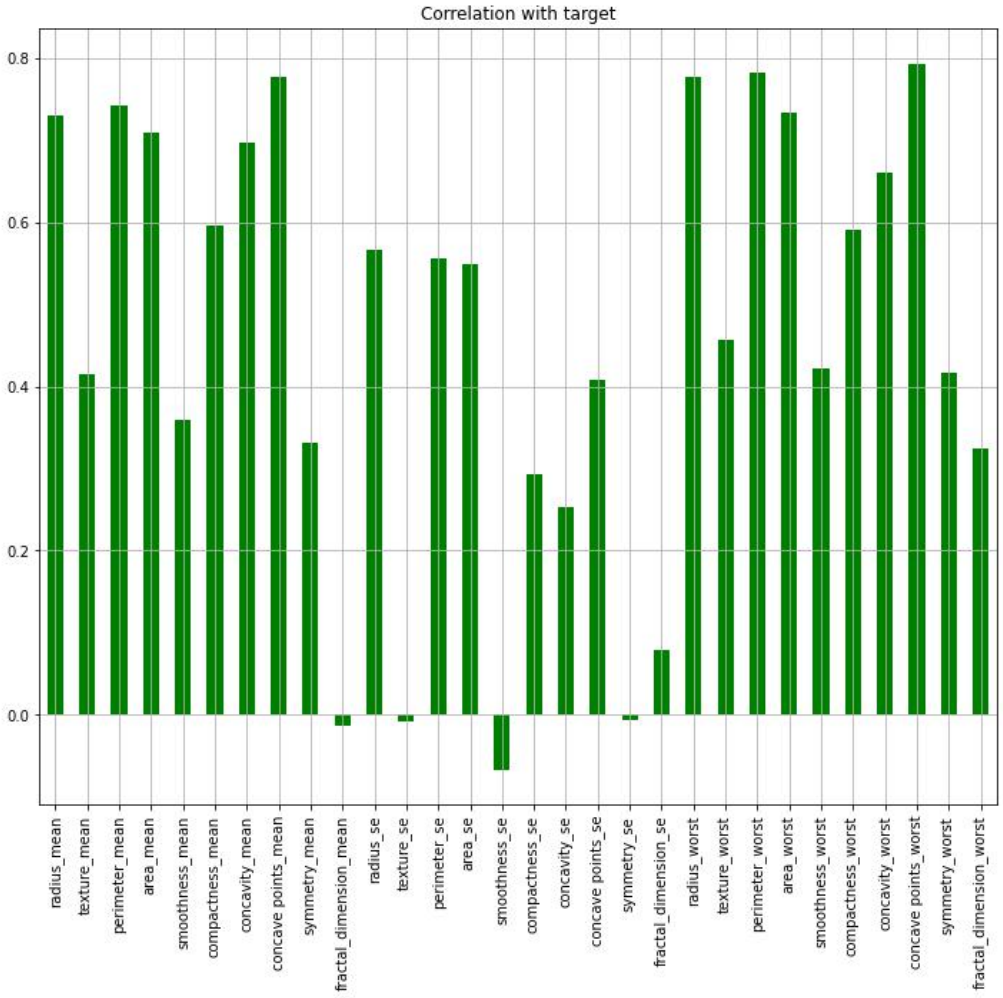
Plot4.1

From the figure we can see that in addition to some variables that are highly correlated, there are also some variables that are not very correlated.Then these variables will be removed, as you can see from the figure.The result of this can increase the accuracy of the prediction to some extent.I picked out columns with a correlation greater than 0.59.Because you can pick out variables that are more than 60% correlated, which I think is reasonable for my forecasting model.

```
corr[abs(corr['diagnosis']) > 0.59].index
```

```
Index(['diagnosis', 'radius_mean', 'perimeter_mean', 'area_mean',
       'compactness_mean', 'concavity_mean', 'concave points_mean',
       'radius_worst', 'perimeter_worst', 'area_worst', 'compactness_worst',
       'concavity_worst', 'concave points_worst'],
      dtype='object')
```

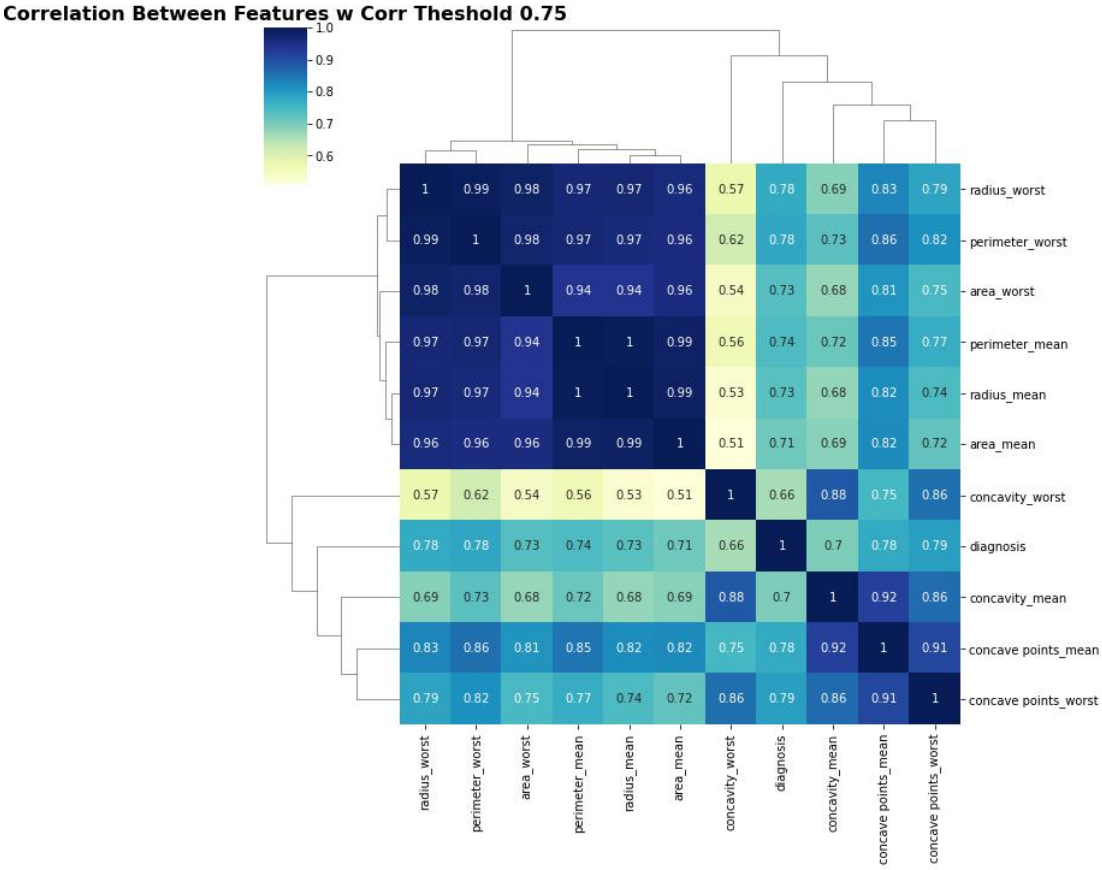
Plot 4.2



Plot 4.3

As can be seen from the figure above, there are about eleven values over 0.59, so we chose these values to predict.The other variables are discarded.Since we have too many dimensions and 32 variables, we should choose the variables with stronger correlations rather than those with weaker correlations.

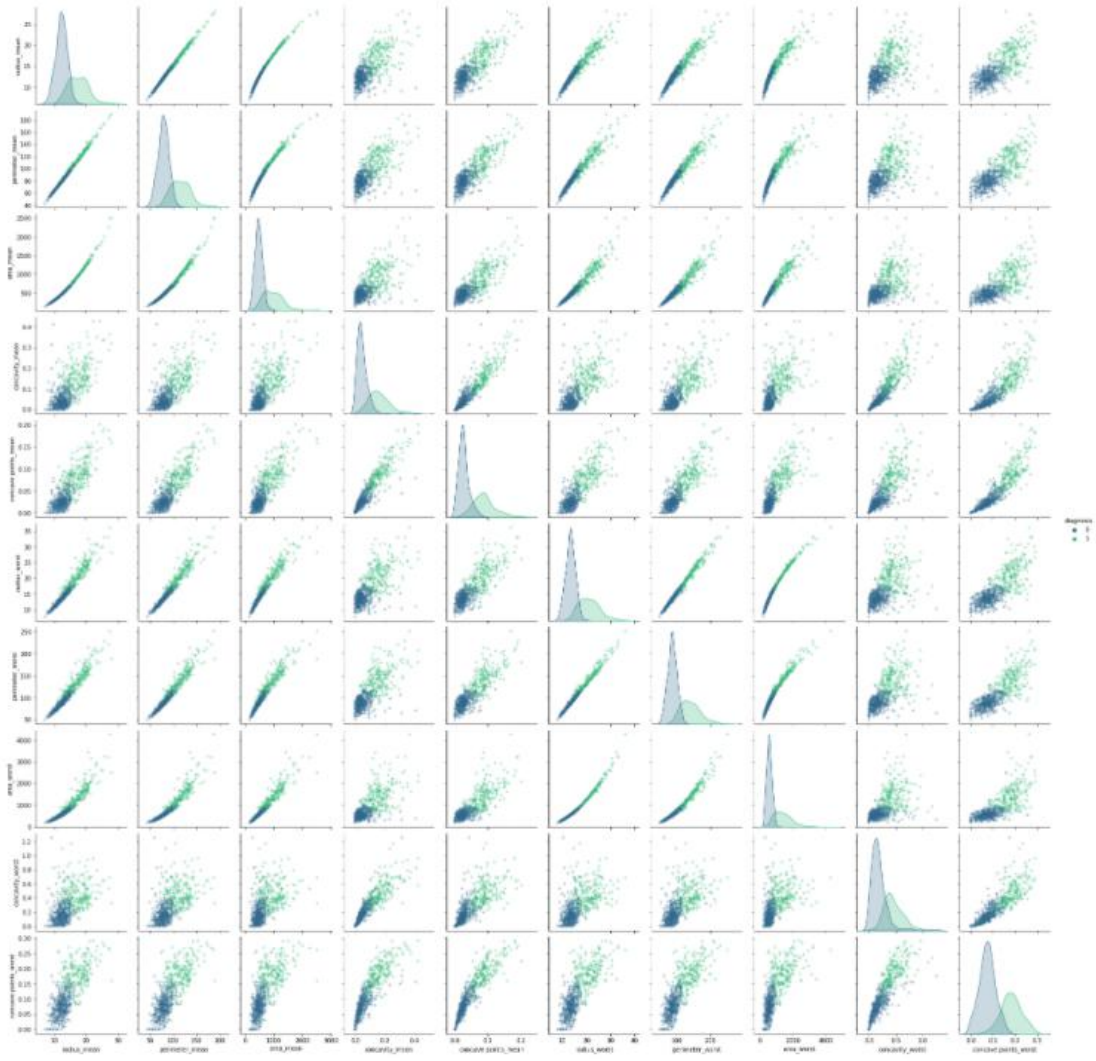
4.2 Project the data



Plot4.4

We have processed the data in the dimension before, and now we can visualize the data to see the results.We conducted a visual correlation analysis of these 10 variables, and found that all variables were basically related and the ratio of the relationship was basically above 60.I think this data is very reasonable, so that the prediction result will be more accurate.

The plot 4.5 shows that almost all of the variables tend to be linear, and some of them are quite obvious.Does that mean that our model would be better done with linear regression?This inspired our subsequent modeling.We construct a suitable data set through data cleaning and screening.An appropriate data set not only makes our modeling prediction rate higher, but also provides us with more ideas in the future.In these scatter plots we can see that all the graphs are almost straight, and although some of them are not obvious, they have some sort of fixed direction.All of these indicate that the data preparation has been basically completed and can be modeled.



Plot4.5

5. Data-mining method selection

5.1 Match and discuss the objectives of data mining

Supervised learning:

When the form of the class conditional probability density function is unknown, there are various non-parametric methods to estimate the class conditional probability density function with learning samples. In the general case that the classification decision rule is expressed by a discriminant function, a learning goal can be determined, for example, the result of the classifier's classification of the given sample is as consistent as possible with the category given by the "teacher", and then an iterative optimization algorithm is used to obtain The parameter value in the discriminant function.

Unsupervised Learning:

A typical example in unsupervised learning is clustering. The purpose of clustering is to group similar things together, and we don't care what this category is. Therefore, a clustering algorithm usually only needs to know how to calculate the similarity before it can start working.

There are generally five methods for clustering algorithms, the most important ones are the division method and the hierarchical method. The partition clustering algorithm divides the data set into K parts by optimizing the evaluation function, and it needs K as the input parameter. Typical segmentation clustering algorithms include K-means algorithm, K-medoids algorithm, and CLARANS algorithm. Hierarchical clustering consists of segmented clusters at different levels, and the segmentation between levels has a nested relationship. It does not require input parameters, which is an obvious advantage over segmentation clustering algorithms. Its disadvantage is that the termination conditions must be specified specifically. Typical hierarchical clustering algorithms include BIRCH algorithm, DBSCAN algorithm and CURE algorithm.

For this research:

The ultimate goal of the project is to predict the risk of breast cancer.And then this is a way to remind women to get screened for breast cancer early and reduce the risk of breast cancer.The project has 10 sets of variables (filtered) worth considering in the model.Of course, the only predictor is whether the tumor is benign or malignant.Obviously this is a binary classification problem, because we only care about whether the outcome is good or bad.Machine learning, which can be divided into supervised learning and unsupervised learning, is a good choice because of the large amount of data.It can effectively avoid the problem of over-fitting. Machine learning approaches will learn from large amounts of data.And then the model is built and then the prediction is made through unsupervised learning or supervised learning.

The construction method of decision tree is very suitable for the research of this project.Because the decision tree approach is a way of classifying outcomes by prediction.The purpose of the decision tree in this project is to determine whether a tumor is benign or malignant using these variables.Each of these variables has an impact on the outcome.In addition, which variable will affect the judgment more will also be the main issue to be discussed.If the decision tree is used well, it can predict whether a tumor is benign or malignant nearly 100 percent of the time.

5.2 Select the appropriate data-mining method(s)

Classification is to find out the common characteristics of a group of data objects in the database and divide them into different classes according to the classification mode. Its purpose is to map the data items in the database to a given class through the classification model, which is used to predict the discrete class of data objects.Classification technology has been applied in many fields, it can be applied to customer classification, customer attributes and characteristics analysis, customer satisfaction analysis, customer buying trend prediction and so on.Classification is also applicable to the model, and decision tree is one of the classification methods.Classification can be used to classify the predicted outcomes into benign and malignant ones and then make predictions.

Clustering is similar to classification, but different from the purpose of classification, it is to divide a group of data into several categories according to the similarity and difference of data.The data belonging to the same category are very similar, but the data between different

categories are very little similar, and the data correlation across classes is very low.However, the clustering method can help us to better classify the tumor and judge the general model of the data first.This is a necessary step before proceeding to any other method.As a complete data mining process, clustering method can be said to be indispensable.

In addition, logistic regression model is also a very appropriate model.It is suitable for modeling probabilities of specific categories or events.The basic form of logistic regression is to model binary dependent variables with logical functions.In regression analysis, logistic regression is estimating the parameters of the logical model.In mathematics, binary logic models have dependent variables with one or two possible values such as pass or fail, which in this model are benign and malignant.

Due to the large amount of data and variables of this project, the Twostep algorithm of unsupervised learning was first used for cluster analysis.The algorithm can process the data twice to compress the data into a manageable subgroup.This pattern is ideal for managing different field types and large data sets.And the method can explore unfamiliar data.Then supervised learning is carried out using random forest, SVM or KNN algorithm to predict whether the tumor is malignant or benign.

6. Data-mining algorithm(s) selection

6.1 Conduct exploratory analysis and discuss

The goal of the project was to predict the risk of developing aggressive breast cancer.After the screening of this data and the conversion of variables, there are finally 10 variables that are very important.Of course the prediction is whether the tumor is malignant or benign, which is obviously a binary classification problem.We only care if the outcome is malignant or benign.

For this problem, supervised learning may be a better algorithm.Supervised learning is a machine learning task that inferences a function from labeled training data.The training data includes a set of training examples.In supervised learning, each instance consists of an input object (usually a vector) and an expected output value (also known as a supervised signal). The supervised learning algorithm is to analyze the training data and produce an inference function that can be used to map out new instances. Decision tree, Random Forest algorithm and Logistic Regression algorithm are all supervised learning algorithms. Of course, there are some other algorithms such as AdaBoosting or GradientBoosting classification algorithm. These algorithms are all very useful, and different classification algorithms are very different for different data. No one knows which way is better before processing it. Both classification and regression are methods of supervised learning. The supervised learning algorithm has two main tasks one is regression and the other is classification.In this model we basically classified the tumor by these variables.So the supervised learning algorithm works well.

6.2 Select data-mining algorithms based on discussion

Therefore, the algorithm of this model adopts uses supervised learning method to classify and predict. The supervised learning methods used include Logistic Regression, Random Forest, Decision Tree, Gradient Boosting, Ada Boost, SVC and KNN algorithm.The best algorithm with the highest accuracy should be selected from the three algorithms as the final algorithm of the model.

Random forest is composed of many decision trees, and there is no correlation between different decision trees.When we carry out the classification task, new input samples come in, and each decision tree in the forest will be judged and classified separately. Each decision tree will get its own classification result. Which one of the classification results of the decision tree has the most classification will be regarded as the final result by the random forest.It can produce data of very high dimensions (many features) without dimensionality reduction and feature selection.It can determine how important a feature is and the interplay between different characteristics.

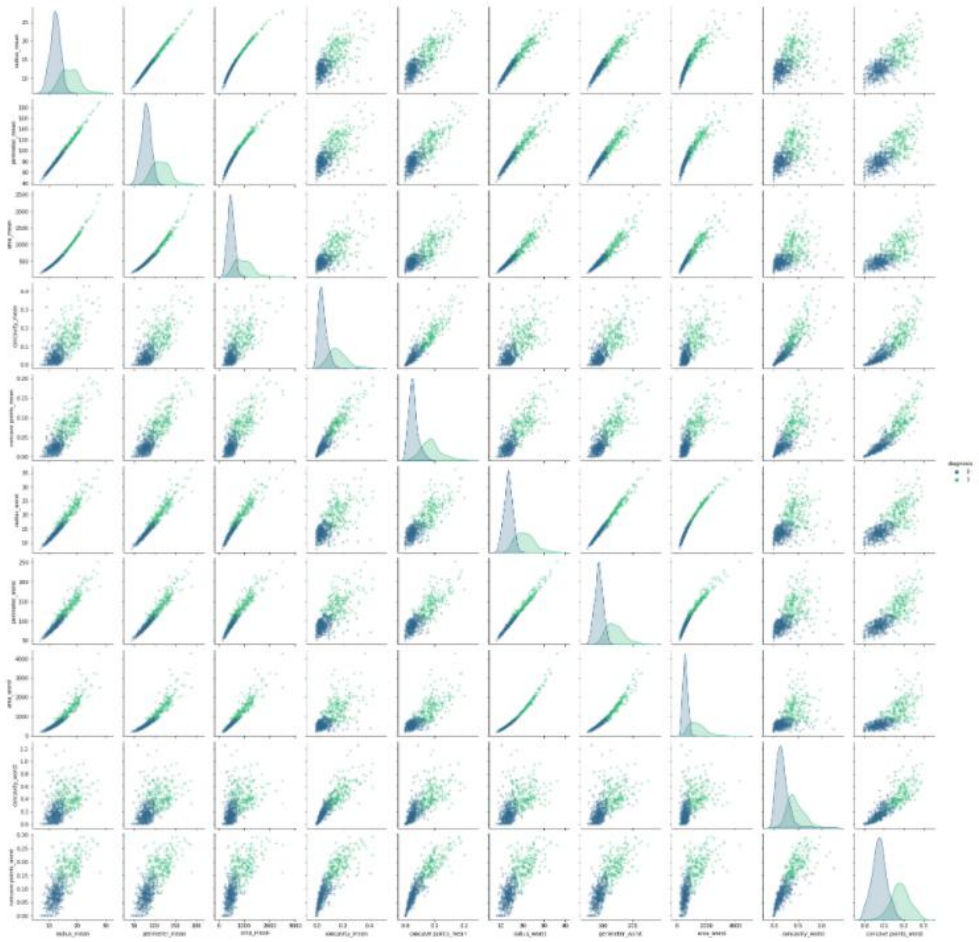
SVM algorithm is a supervised learning model and related learning algorithm for analyzing data in classification and regression analysis.Given a set of training instances, each of which is labeled as belonging to one or the other of two categories, the SVM training algorithm creates a model that assigns the new instance to one of the two categories, making it an improbabilistic binary linear classifier.

Logistic regression model is also a very appropriate model.It is suitable for modeling probabilities of specific categories or events.The basic form of logistic regression is to model binary dependent variables with logical functions.In regression analysis, logistic regression is estimating the parameters of the logical model.In mathematics, binary logic models have dependent variables with one or two possible values such as pass or fail, which in this model are benign and malignant.

The KNN algorithm is a sample that is most similar to K samples in the data set. If most of the K samples belong to a certain category, then the sample also belongs to this category. K-nearest neighbor algorithm is a basic classification and regression method.In this paper, we only discuss the k-nearest neighbor method for classification problems.KNN is given a training data set. For a new input instance, K instances closest to the instance are found in the training data set. Most of these K instances belong to a certain class, and the input instance is classified into this class

There are some other algorithms that I haven't covered, but the basic principle is the same and we'll talk more about it later on if it's more accurate.

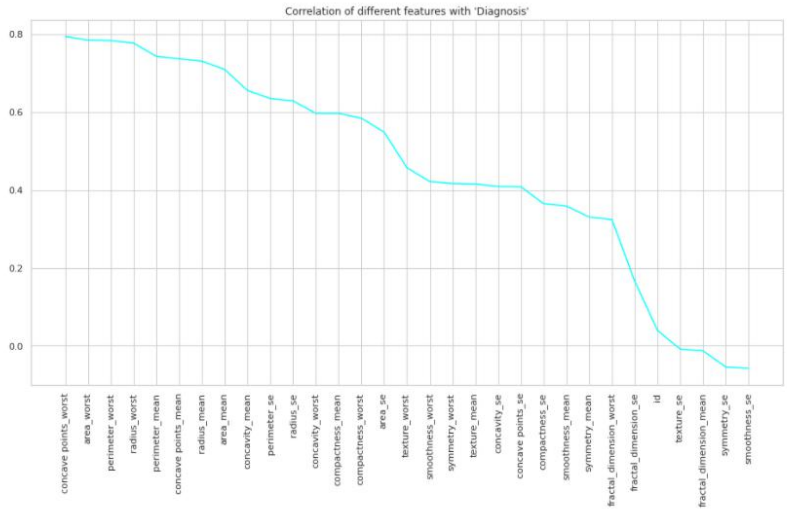
6.3 Build/Select appropriate model(s) and choose relevant parameter(s)



Plot 6.1

As for which model to choose, my first priority is logistic regression model.The first thing I noticed was that each of the data, taken separately, was almost a straight line, which meant that probably all of the data had linear regressions.At least it's a regression model.So second, this is a binary judgment model, and it's also a categorical predictor.I think logistic regression is a great way to do that especially if all the data is normally distributed and it's linear.

Of course, I've listed a lot of methods of supervised learning, besides regression and classification.I do not rule out that other classifiers have better prediction accuracy, especially random forests and decision trees.I think these two methods are very helpful for this binary classification and will make it more accurate.Adaboost belongs to Decesion Stump, and I think the classification may not be as good as the former.So the other two are based on how accurate the model is at that time.I still like logistic regression models and classifiers in random forests and decision trees.



Plot 6.2

After selecting the model, we selected the corresponding data. Previously, I selected the data with more than 60% correlation from this set of graphs.We can see that 60 is the dividing line, and we take the first few variables that are correlated and we don't take the other variables, which makes our model more accurate, because the latter variables are more unrelated.

Then we build the model using PySpark.

```
data = data['diagnosis', 'radius_mean', 'perimeter_mean', 'area_mean',
       'compactness_mean', 'concavity_mean', 'concave points_mean',
       'radius_worst', 'perimeter_worst', 'area_worst', 'compactness_worst',
       'concavity_worst', 'concave points_worst']
```

```
data.columns
```

```
['diagnosis',
 'radius_mean',
 'perimeter_mean',
 'area_mean',
 'compactness_mean',
 'concavity_mean',
 'concave points_mean',
 'radius_worst',
 'perimeter_worst',
 'area_worst',
 'compactness_worst',
 'concavity_worst',
 'concave points_worst']
```

Plot 6.3

So first we pick all the variables that have a correlation greater than 60% and there are 12 variables. So we're going to store these variables in this set called 'data'. Then I will divide the set to 70% of training set and 30% of test set to get the accuracy.

Then we use the 'diagnosis' as the response variable, which is the flag variable, and the other variables as the characteristic variables. We will use other variables to predict the value of the variable of FLAG, and then calculate the accuracy rate.

```
dtc = DecisionTreeClassifier(labelCol='diagnosisIndex',featuresCol='features')
rfc = RandomForestClassifier(labelCol='diagnosisIndex',featuresCol='features')
gbt = GBTClassifier(labelCol='diagnosisIndex',featuresCol='features')
```

Plot 6.4

The three models are decision tree classification, random forest classification and GBT classification algorithm.We will first model with the default parameters and then explore the best parameters to optimize the model in the subsequent optimization. For the three models, the default depth parameter for the tree is MAXDEPTH of 5. For the random forest and GBT algorithm, the maximum value of the generated tree is 20.

Let's look at the output of our model:

```
print("DTC")
print(my_binary_eval.evaluate(dtc_predictions))

print("RFC")
print(my_binary_eval.evaluate(rfc_predictions))

my_binary_gbt_eval = BinaryClassificationEvaluator(labelCol='diagnosisIndex', rawPredictionCol='prediction')
print("GBT")
print(my_binary_gbt_eval.evaluate(gbt_predictions))
```

```
DTC
0.9250700280112045
RFC
0.9859243697478992
GBT
0.9415966386554621
```

Plot 6.5

We found that no matter how many times it was run, the accuracy of random forest prediction was always higher than that of the other two algorithms. Then can we think that the random forest algorithm is more suitable for this data mining? The results may change if we use a multivariate classification algorithm instead of just analyzing the two categories.

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

```
acc_evaluator = MulticlassClassificationEvaluator(labelCol="diagnosisIndex", predictionCol="prediction", metricName="accuracy")
```

```
dtc_acc = acc_evaluator.evaluate(dtc_predictions)
rfc_acc = acc_evaluator.evaluate(rfc_predictions)
gbt_acc = acc_evaluator.evaluate(gbt_predictions)
```

```
print("Here are the results!")
print('-'*40)
print('A single decision tree has an accuracy of: {0:2.2f}%'.format(dtc_acc*100))
print('-'*40)
print('A random forest ensemble has an accuracy of: {0:2.2f}%'.format(rfc_acc*100))
print('-'*40)
print('An ensemble using GBT has an accuracy of: {0:2.2f}%'.format(gbt_acc*100))
```

```
Here are the results!
----------------------------------------
A single decision tree has an accuracy of: 94.22%
----------------------------------------
A random forest ensemble has an accuracy of: 96.53%
----------------------------------------
An ensemble using GBT has an accuracy of: 94.80%
```

Plot 6.6

We used the multi-set classification algorithm in PySpark and found that the accuracy of the decision tree and GBT algorithm increased a bit, while the accuracy of the random forest algorithm decreased a bit. But in general, the random forest model has a higher accuracy rate.

Parameter selection:

```
dtc = DecisionTreeClassifier(labelCol='diagnosisIndex',featuresCol='features',maxDepth=10)
rfc = RandomForestClassifier(labelCol='diagnosisIndex',featuresCol='features',numTrees=30,maxDepth=10)
gbt = GBTClassifier(labelCol='diagnosisIndex',featuresCol='features',maxDepth=10)
```

Plot 6.7

Now to change the parameters of each model, change all MAXDEPTH parameters to 10 and number trees in the random forest to 30 and model.

```
dtc_model
```

```
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_4f309fd779c3f9a3ddbb) of depth 8 with 37 nodes
```

```
rfc_model
```

```
RandomForestClassificationModel (uid=rfc_e12721c441c0) with 30 trees
```

```
gbt_model
```

```
GBTClassificationModel (uid=GBTClassifier_4022a54253bba2f764ce) with 20 trees
```

Plot 6.8

We found through three models that the depth of the decision tree model was 8 with 37 nodes, the random forest generated 30 trees, and the GBT algorithm generated 20 trees. Let's take a look at the results:

```
print("DTC")
print(my_binary_eval.evaluate(dtc_predictions))

print("RFC")
print(my_binary_eval.evaluate(rfc_predictions))

my_binary_gbt_eval = BinaryClassificationEvaluator(labelCol='diagnosisIndex', rawPredictionCol='prediction')
print("GBT")
print(my_binary_gbt_eval.evaluate(gbt_predictions))
```

```
DTC
0.9334880561530623
RFC
0.969489414694894
GBT
0.9245443224272614
```

Plot 6.9

Through these results, we found that the prediction accuracy of the three models all decreased a little, but the random forest algorithm still had the highest accuracy, as high as 96.9%. That's a pretty good solution, but the old parameter is a little bit better. Now let's run the multivariate classification algorithm and see if the results change.

```
print("Here are the results!")
print('-'*40)
print('A single decision tree has an accuracy of: {0:2.2f}%'.format(dtc_acc*100))
print('-'*40)
print('A random forest ensemble has an accuracy of: {0:2.2f}%'.format(rfc_acc*100))
print('-'*40)
print('An ensemble using GBT has an accuracy of: {0:2.2f}%'.format(gbt_acc*100))
```

```
Here are the results!
----------------------------------------
A single decision tree has an accuracy of: 93.30%
----------------------------------------
A random forest ensemble has an accuracy of: 93.81%
----------------------------------------
An ensemble using GBT has an accuracy of: 93.30%
```

Plot 6.10

We found that the predictive rate of these outcomes was lower, which should be a problem of parameter selection. More accuracy may be achieved by using the default parameters, so in the final result, the parameters will be set as the default parameters for the time being.

7. Data Mining

7.1 Create and justify test designs

In the model classification task, I randomly divide the training set and the test set according to a random ratio of 7:3 to test the model. You need to create a logical test and divide it into a 70% training set and a 30% validation set. The purpose of this is to make more effective use of data, because machine learning will learn from existing data sets. If the learning results continue to be used in the training set, even if the accuracy rate reaches 100%, this is meaningless, so an additional verification set is required to verify, which is a good comparison with 7-3, because it can ensure that the test set is on The test results are valid and can ensure training on a larger data set to avoid insufficient fitting.

7.2 Conduct data mining

We'll use Python and PySpark in the next steps。

```
data = data['diagnosis', 'radius_mean', 'perimeter_mean', 'area_mean',
       'compactness_mean', 'concavity_mean', 'concave points_mean',
       'radius_worst', 'perimeter_worst', 'area_worst', 'compactness_worst',
       'concavity_worst', 'concave points_worst']
```

```
data.columns
```

```
['diagnosis',
 'radius_mean',
 'perimeter_mean',
 'area_mean',
 'compactness_mean',
 'concavity_mean',
 'concave points_mean',
 'radius_worst',
 'perimeter_worst',
 'area_worst',
 'compactness_worst',
 'concavity_worst',
 'concave points_worst']
```

Plot 7.1

So first we pick all the variables that have a correlation greater than 60% and there are 12 variables. So we're going to store these variables in this set called 'data'. Then I will divide the set to 70% of training set and 30% of test set to get the accuracy.

Then we use the 'diagnosis' as the response variable, which is the flag variable, and the other variables as the characteristic variables. We will use other variables to predict the value of the variable of FLAG, and then calculate the accuracy rate.

```
dtc = DecisionTreeClassifier(labelCol='diagnosisIndex',featuresCol='features')
rfc = RandomForestClassifier(labelCol='diagnosisIndex',featuresCol='features')
gbt = GBTClassifier(labelCol='diagnosisIndex',featuresCol='features')
```

Plot 7.2

The three models are decision tree classification, random forest classification and GBT classification algorithm.We will first model with the default parameters and then explore the best parameters to optimize the model in the subsequent optimization. For the three models, the default depth parameter for the tree is MAXDEPTH of 5. For the random forest and GBT algorithm, the maximum value of the generated tree is 20.

Let's look at the output of our model:

```
print("DTC")
print(my_binary_eval.evaluate(dtc_predictions))

print("RFC")
print(my_binary_eval.evaluate(rfc_predictions))

my_binary_gbt_eval = BinaryClassificationEvaluator(labelCol='diagnosisIndex', rawPredictionCol='prediction')
print("GBT")
print(my_binary_gbt_eval.evaluate(gbt_predictions))
```

```
DTC
0.9250700280112045
RFC
0.9859243697478992
GBT
0.9415966386554621
```

Plot 7.3

We found that no matter how many times it was run, the accuracy of random forest prediction was always higher than that of the other two algorithms. Then can we think that the random forest algorithm is more suitable for this data mining? The results may change if we use a multivariate classification algorithm instead of just analyzing the two categories.

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
```

```
acc_evaluator = MulticlassClassificationEvaluator(labelCol="diagnosisIndex", predictionCol="prediction", metricName="accuracy")
```

```
dtc_acc = acc_evaluator.evaluate(dtc_predictions)
rfc_acc = acc_evaluator.evaluate(rfc_predictions)
gbt_acc = acc_evaluator.evaluate(gbt_predictions)
```

```
print("Here are the results!")
print('-'*40)
print('A single decision tree has an accuracy of: {0:2.2f}%'.format(dtc_acc*100))
print('-'*40)
print('A random forest ensemble has an accuracy of: {0:2.2f}%'.format(rfc_acc*100))
print('-'*40)
print('An ensemble using GBT has an accuracy of: {0:2.2f}%'.format(gbt_acc*100))
```

```
Here are the results!
----------------------------------------
A single decision tree has an accuracy of: 94.22%
----------------------------------------
A random forest ensemble has an accuracy of: 96.53%
----------------------------------------
An ensemble using GBT has an accuracy of: 94.80%
```

Plot 7.4

We used the multi-set classification algorithm in PySpark and found that the accuracy of the decision tree and GBT algorithm increased a bit, while the accuracy of the random forest algorithm decreased a bit. But in general, the random forest model has a higher accuracy rate.

In addition, I also implemented the Skrlearn algorithm in Python.(Note here that this will be the data mining process for the convenience of visualization in the following results)：

```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import NeighborhoodComponentsAnalysis
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.3, random_state=42)

# Dont fit the scaler while standardisate X_test !
scaler = StandardScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
```

Plot7.5

First we call the sklearn package to split the training set and the test set.

```
key = ['LogisticRegression','KNeighborsClassifier','SVC','DecisionTreeClassifier','RandomForestClassifier','GradientBoostingClassifier','AdaBoo
value = [LogisticRegression(), KNeighborsClassifier(n_neighbors = 2, weights ='uniform'), SVC(kernel='rbf',random_state=15), DecisionTreeClass
models = dict(zip(key,value))
models
```

Plot7.6

Then we modeled each model with Sklearn.

```
{'LogisticRegression': LogisticRegression(),
 'KNeighborsClassifier': KNeighborsClassifier(n_neighbors=2),
 'SVC': SVC(random_state=15),
 'DecisionTreeClassifier': DecisionTreeClassifier(random_state=10),
 'RandomForestClassifier': RandomForestClassifier(n_estimators=60, random_state=0),
 'GradientBoostingClassifier': GradientBoostingClassifier(random_state=20),
 'AdaBoostClassifier': AdaBoostClassifier()}
```

Plot7.7

The above diagram shows the modeling process and parameters.

```
predicted =[]
for name,algo in models.items():
    model=algo
    model.fit(X_train,y_train)
    predict = model.predict(X_test)
    acc = accuracy_score(y_test, predict)
    predicted.append(acc)
    print(name,acc)
```

```
LogisticRegression 0.9824561403508771
KNeighborsClassifier 0.9532163742690059
SVC 0.9649122807017544
DecisionTreeClassifier 0.9005847953216374
RandomForestClassifier 0.9649122807017544
GradientBoostingClassifier 0.9766081871345029
AdaBoostClassifier 0.9590643274853801
```

Plot 7.8

We then fed the data into the model to calculate the predictive accuracy of each model.

The above is all the data mining model and the predicted accuracy rate, the process is relatively simple, mainly by calling the skrlearn package.

7.3 Search for patterns

Our results using PySpark are as follows:

```
print("DTC")
print(my_binary_eval.evaluate(dtc_predictions))

print("RFC")
print(my_binary_eval.evaluate(rfc_predictions))

my_binary_gbt_eval = BinaryClassificationEvaluator(labelCol='diagnosisIndex', rawPredictionCol='prediction')
print("GBT")
print(my_binary_gbt_eval.evaluate(gbt_predictions))
```

```
DTC
0.9250700280112045
RFC
0.9859243697478992
GBT
0.9415966386554621
```

Plot 7.9

Because we are performing binary classification, I think the binary classifier can classify more accurately and has a higher accuracy rate. So I did not show the results of the multi-classifier algorithm. From the point of view of the choice of classifier, the random forest algorithm is more representative, its correct rate is higher, and it is more significant than the other two results. The random forest algorithm has a correct rate of 98.6%, the decision tree calculation has a correct rate of 92.5%, and the GBT algorithm has a correct rate of 94.2%. So I think the performance of random forest may be better in these three classifiers.

Here are the results of data mining using Python's SkrLearn extension package:

```
predicted =[]
for name,algo in models.items():
    model=algo
    model.fit(X_train,y_train)
    predict = model.predict(X_test)
    acc = accuracy_score(y_test, predict)
    predicted.append(acc)
    print(name,acc)
```

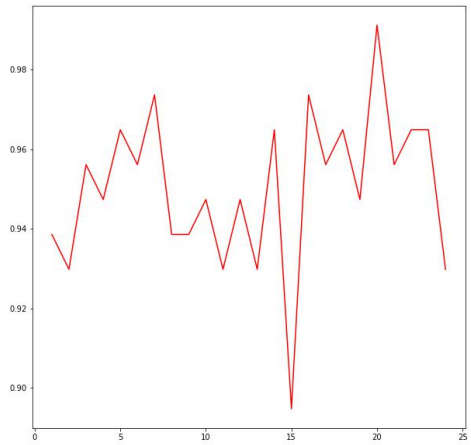
```
LogisticRegression 0.9824561403508771
KNeighborsClassifier 0.9532163742690059
SVC 0.9649122807017544
DecisionTreeClassifier 0.9005847953216374
RandomForestClassifier 0.9649122807017544
GradientBoostingClassifier 0.9766081871345029
AdaBoostClassifier 0.9590643274853801
```

Plot 7.10

Logistic Regression, KNN, SVC, and Adaboost are added to the Python Skrlearn package.They are also supervised learning algorithms. These algorithms are also used because they are typical and have high accuracy.Here we only care that the lowest accuracy is 90% for the decision tree algorithm and 98% for the logistic regression algorithm. Because of the parameters, we are not sure whether this model will be more accurate if the parameters change.So let's try to find a better accuracy.

So here I adjusted the parameters of the Logistic regression to find a higher accuracy rate：

We changed the parameters of the logistic regression algorithm, and then obtained the following graph, which shows how many times the model's accuracy rate changes. We can find that the maximum value appears around the twentieth time in the figure, and this maximum value is the highest prediction accuracy of the model we are looking for.




```
for i in range(0,24):
    if acc[i] > 0.98:
        print(acc[i])
```

```
0.9912280701754386
```

Plot 7.11

We then calculated that the highest accuracy rate was an astonishing 99.1%. This is in line with our initial accuracy requirements, and to ensure the accuracy of more than 99%. So I think the best performance of the Skrlearn package in Python is the logistic regression algorithm.It has the highest accuracy of more than 99 percent.

8. Interpretation

8.1 Study and discuss the mined patterns

The whole data mining project is basically over. We used PySpark and the machine learning algorithm in the Python program package Skrlearn to build the model, and finally achieved an accurate prediction rate of 99.1%. This result is very exciting, because we can almost 100% predict whether breast cancer tumors are benign or malignant. This fits the result we originally wanted. And we found 12 important eigenvalue variables from 32 eigenvalues. These feature values can help us better predict the results. In addition, these characteristic values also help us reduce the required memory size. So that our results can be run faster.

In this prediction model, we mainly explored 12 variables to predict whether the tumor is benign or malignant. In this experiment, we set the goal from the beginning, so it went quite smoothly. We just started by cleaning the data to remove extreme values and outliers, making our final results very accurate. In the random forest, an astonishing 99.1% accuracy rate was achieved. This is the ultimate goal of machine learning to be as close to 100% as possible. Of course, it may be because of the selection of parameters or the ratio of training set to test set not being right, and other algorithm did not achieve high prediction accuracy. But they also have an accuracy rate over 90%. But through this, we can find that the key to solving the problem lies in the supervised learning method.

The difficulty with this dataset is that, in general, it is easy to classify a dataset as a problem or a regression as a problem, but at this time, we can easily see a dataset as a problem or a regression as a problem in the process of using a dataset and it is hard to see at a glance its mining purpose.

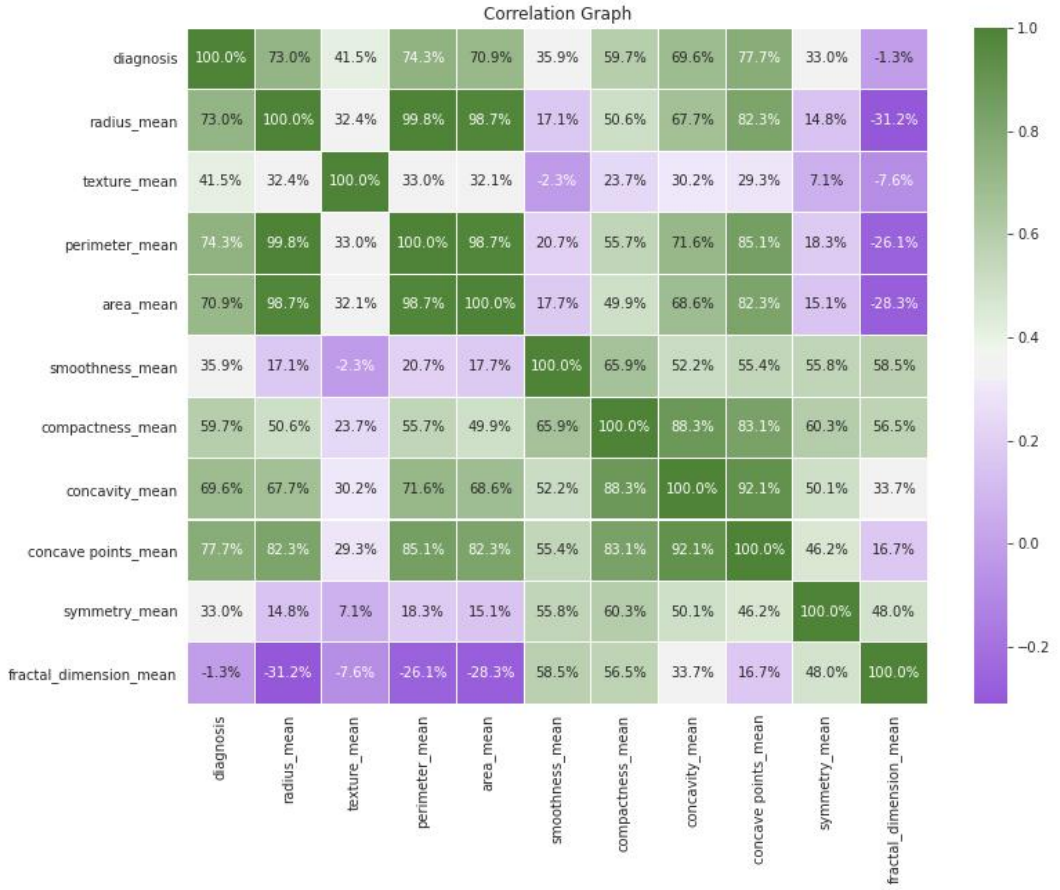
We selected 12 important variables from the 32 variables at the beginning. Then filter the

data step by step, integrate and transform the data, and finally get the desired result. This process is full of challenges, but the end result is exciting. We can achieve 99.1% accuracy through the logistic regression model, thus achieving the goal of this experiment. However, it takes a lot of time to discover the logistic regression in the process, so a certain degree of optimization can be carried out to achieve a faster time. Of course, the other algorithms are relatively fast, but the results are not more accurate than random forest.

In short, the logistic regression method under the supervised learning method performed best, which also reached the goal of this prediction. Of course, the random forest model has an acceptable 98% accuracy rate in PySpark.And that's a very high accuracy rate. For machine learning, there may be more data to help random forests achieve higher accuracy.

8.2 Visualize the data, results, models, and patterns

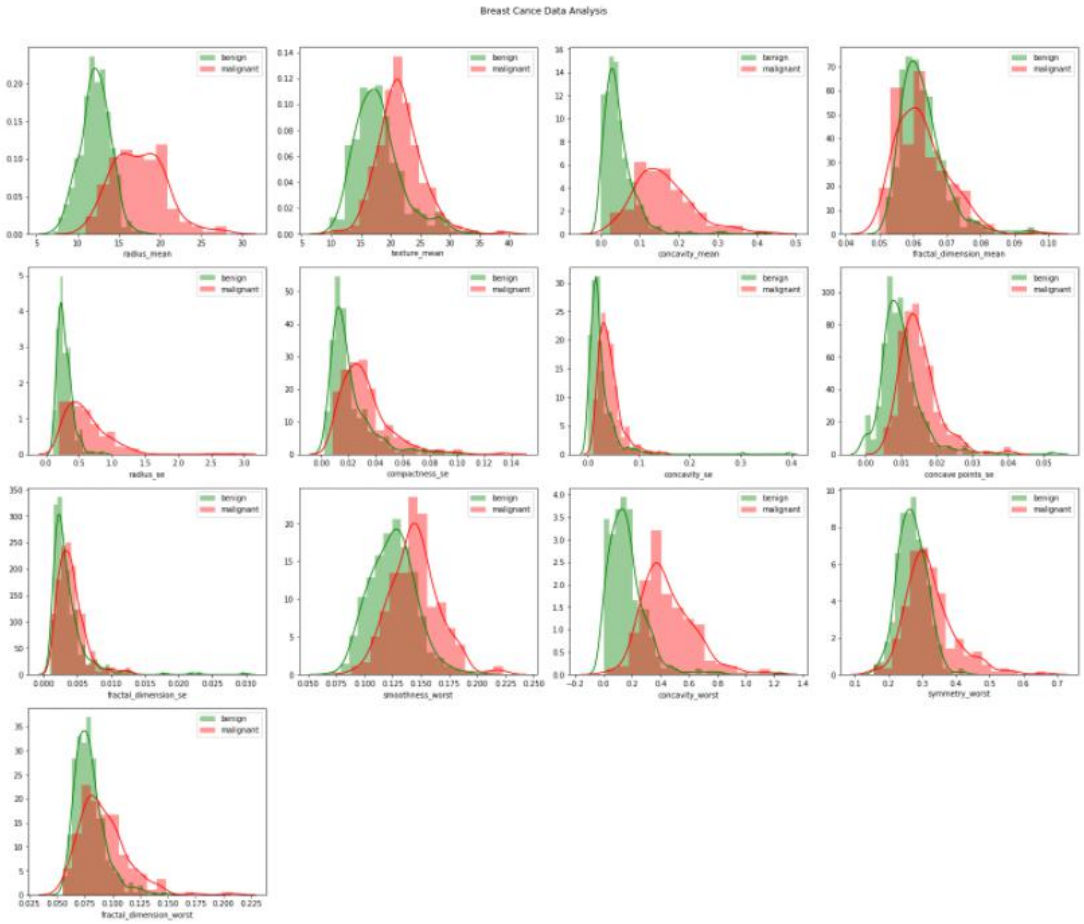
This is a visualization of the correlation:



Plot 8.1

From the figure, we can see that the correlation of the features we selected is extremely high, so it makes a great contribution to the accuracy of the subsequent prediction.

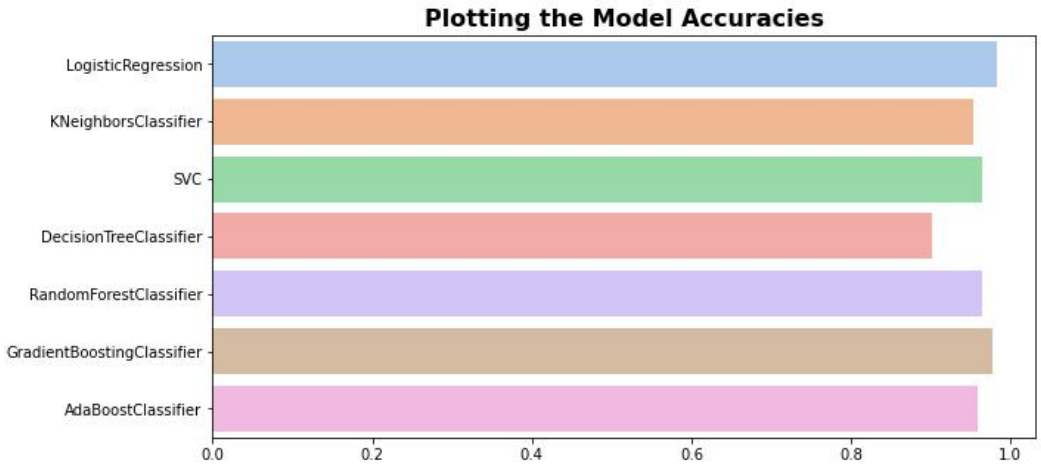
So this is all the eigenvalues that we took out, and we visualized it:



Plot 8.2

Removal of different features from the data set will have different effects on the p-value for the data set. We can remove different features and measure the p-value in each case. These measured p-values can be used to decide whether to keep a feature or not.

This is a visualization of all the results:



Plot 8.3

It can be found from the results that the logistic regression algorithm has the highest

accuracy and we have also verified that the accuracy of this algorithm is as high as 99.1% if the appropriate regression times are selected.

8.3 Interpret the results, models, and patterns

This model is regression and classifier models that divides our final results into two categories. There are two reasons why this model is used. The reason is that our prediction goal is to judge whether the tumor is benign or malignant. This is enough to prove that the classification is feasible. Logical regression models are probably better because our variables are more linear, so the models are more likely to predict things correctly.

So in supervised learning, we use the other algorithms, why their accuracy is not as high as the logistic regression ?

Because logistic regression can be used to judge data with many features without dimensionality reduction, let alone feature selection. It can judge the importance of each feature and then perform modeling analysis. It can also determine the mutual influence between different features. The most important thing is that the logistic regression will not overfit and the training speed is relatively fast. And it is easy to make parallel algorithms. For unbalanced data sets, it can also balance errors, and for miss data in the data, it can maintain accuracy.

For the algorithm SVC, if the feature dimension is much larger than the number of samples, the performance is average. Moreover, it is difficult to choose a suitable kernel function, and it is very sensitive to missing data. In this data, the dimension of the feature variable is much larger than the variable we want to predict, so this algorithm is not suitable.

For the algorithm knn, when the dimensionality is very high, the amount of calculation is abnormally large, and the prediction accuracy of rare categories is low. Modeling requires a lot of memory. It is a lazy learning method that basically does not learn, so the predicted results are not very accurate. Compared with decision trees, the interpretability of the knn model is not strong. So there is no high accuracy rate of random forest.

For the other algorithms, they're all classifier algorithms, they don't have logistic regression and they do well in this model because all the data tends to be linear regression distributed.So you can see the problem from the first observation data.

8.4 Assess and evaluate results, models, and patterns

The best way to evaluate the result is to look at the accuracy of the result. Therefore, the random forest model with the highest accuracy is selected among the these models. Moreover, the data is almost the same as the actual value. The ratio reached 99.1%. Of course, the evaluation methods for different models have different effects. Although the logistic regression has a relatively long running time, it has a higher accuracy rate and higher interpretability for the other models. This is why the logistic regression model was chosen.

Logistic regression is easier to implement, interpret, and very efficient to train. It makes no assumptions about distributions of classes in feature space. It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions. It not only provides a measure of how appropriate a predictor(coefficient size)is, but also its direction of

association (positive or negative). Good accuracy for many simple data sets and it performs well when the dataset is linearly separable. We know that our data set is not very complex and it's mostly linear, as we talked about when we were working with the data.So I think there's a huge advantage to using logistic regression if the data set is linear. It can interpret model coefficients as indicators of feature importance. Logistic regression is less inclined to over-fitting but it can overfit in high dimensional datasets. So that's why logistic regression works so well in this model, so logistic regression works best in this model.

8.5 Iterate prior steps (1 – 7) as required

In our previous experiments, we found that in addition to logistic regression algorithm, random forest is also an algorithm with high accuracy.So whether we can find the right parameters for going back to the forest or whether we can get better predictions.So I changed the parameters in the random forest and iterated the data mining process again.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state = 0)
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Plot 8.4

I'm still using the Python Skrlearn package here, because it works really well.I'm going to divide all the data into 70% of the training set and 30% of the test set just like before.Then use StandardScaler to scal the data.

```
from sklearn.ensemble import RandomForestClassifier

rand_clf = RandomForestClassifier(criterion = 'entropy', max_depth = 11, max_features = 'auto',
                                  min_samples_leaf = 2, min_samples_split = 3, n_estimators = 130)
rand_clf.fit(X_train, y_train)
```

```
RandomForestClassifier(criterion='entropy', max_depth=11, min_samples_leaf=2,
                       min_samples_split=3, n_estimators=130)
```

Plot 8.5

Combined with the random forest classifier in the previous data mining process, I set the maximum depth to 11, the maximum feature to be 'auto', the minimum leaf tree to be 2, the minimum samples split to be 3, and the estimators to be 130.Then the model is trained to get the results.

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
print(accuracy_score(y_train, rand_clf.predict(X_train)))

ran_clf_acc = accuracy_score(y_test, y_pred)
print(ran_clf_acc)
```

```
0.9924433249370277
0.9590643274853801
```

Plot 8.6

Through this training, we can find that the accuracy of random forest on the training set is very high, as high as 99.2%, but the training accuracy on the test set was not so high, only 95.9%.Maybe our parameters need to be improved, but compared to logistic regression

algorithm, this accuracy is really not high.

Reference

SRK Of Kaggle.(2016).*Breast Cancer Wisconsin (Diagnostic) Data Set* from https://www.kaggle.com/uciml/breast-cancer-wisconsin-data