

Diabetes Prediction based on Machine Learning



Junyu PAN

2090693

Supervised by: Dr. Sujoy Sinha Roy

School of Computer Science

University of Birmingham

This dissertation is submitted for the degree of
MSc Advance Computer Science

04/09/20

Content

Acknowledgment	iii
Abstract	iv
List of Figures	v
List of Tables	vii
1.Introduction	1
1.1 Relevant Work and My Contributions	2
1.2 Machine Learning	3
1.2.1 Machine Learning in Medical.....	3
1.2.2 Supervised Learning	3
1.3 Preparation	4
1.4 OSEMN Principle	5
2.Data Pre-processing	7
2.1 Obtain the Data.....	7
2.1.1 The Information of Dataset.....	7
2.1.2 Data Description	7
2.2 Data Pre-processing Plans	7
2.3 Import Dependency Library / Package.....	8
2.4 Import and Observe Data.....	9
2.5 Count and Remove Null Values.....	10
2.5.1 Count Null Values	10
2.5.2 Select Replacement Method	11
2.5.3 Calculating the Pearson Product.....	11
2.5.4 Replace Null Values	12
2.6 Treatment of Outliers	13
2.6.1 Boxplot	13
2.6.2 Replacement of Outliers	14
2.7 Data Normalization	17
2.7.1 Necessity.....	17
2.7.2 Normalization Method.....	18

3. Modelling the Data	20
3.1 Dataset Split	20
3.2 KNN	20
3.3 SVM	21
3.3.1 RBF Kernel	21
3.3.2 Linear Kernel	21
3.4 Logistic Regression	22
3.5 Random Forest	22
3.6 ANN	23
3.7 Evaluation	24
3.7.1 Confusion matrix	24
3.7.2 F1-score	25
3.7.3 AUC/ROC	25
3.8 Model and Evaluation	26
4. Application and Optimization	31
4.1 Application	31
4.1.1 Model Packaging	31
4.1.2 Data interface	31
4.1.3 Data input standardization	32
4.1.4 Realization of prediction	33
4.2 Optimization	34
4.2.1 Random State	34
4.2.2 Grid Search	34
4.2.3 Ensemble	35
4.3 Optimized application	37
4.4 Conclusions	38
4.5 Future Work	39
References	40
Appendix	43
Appendix A: GitLab Repository	43
Appendix B: Code and Thought Reference Statement	44

Acknowledgment

Firstly, I would like to thank my supervisor, Dr. Sujoy Sinha Roy, for his continuous support for my paper. Then I would like to express my respect to the staffs of the National Institute of diabetes and digestive and kidney disease who collected the Pima dataset. Without their efforts and selfless publicity, I would not have obtained such valuable data. In addition, I also want to thank the pioneers in relevant fields on Kaggle and GitHub. Without their previous working ideas, I would not be able to complete this project in such a short time. I am grateful for disclosing their own views on this field and the basic process of solving such problems.

Abstract

With the improvement of living conditions, diabetes affects more and more people and brings a very serious burden to the patients in their life. Therefore, it is very important to check and prevent diabetes as soon as possible. In this paper, through the use of machine learning related knowledge and the work pipeline of OSEMN, I analysed and pre-processed the Pima diabetes data set, visualized the data by plots, discussed the function of normalization, built a number of prediction models (KNN,LR,RF,ANN,SVM), then used Grid Search to adjust the parameters and two ensemble method to optimize the models, and then imported the trained models into the prediction interface made by myself, and the diabetes prediction model with high accuracy is successfully realized. In addition, I concluded some existing problems in my project and the future work at the last of the paper.

Key words: Diabetes, Machine Learning, OSEMN, Classification Model, Model Ensemble

List of Figures

Figure 1: Process framework of the project.....	2
Figure 2: General process of supervised learning	4
Figure 3: The principle and process of OSEMN	6
Figure 4: Dependency libraries and packages (1)	8
Figure 5: Dependency libraries and packages (2)	8
Figure 6: Description of the dataset.....	9
Figure 7: Process of counting null values.....	10
Figure 8: Result of counting null values.....	10
Figure 9: Codes of Pearson correlation matrix	11
Figure 10: Pearson correlation matrix	12
Figure 11: Code of replacing null values.....	13
Figure 12: Result after replacing null values.....	13
Figure 13: Sample of boxplot	14
Figure 14: Boxplot of general features	15
Figure 15: Boxplot after outliers' removing.....	18
Figure 16: Train Test Split	20
Figure 17: Flow chart of random forest.....	22
Figure 18: The structure of ANN	24
Figure 19: Confusion matrix	24
Figure 20: ROC curve	26
Figure 21: Condition of ANN model.....	29
Figure 22: Code of saving models.....	31
Figure 23: Code of saving ANN model.....	31
Figure 24: Saved models	31
Figure 25: Prediction Interface	32
Figure 26: Definition of 'pre' button.....	32
Figure 27: Definition of 'standard' button	33
Figure 28: Prediction results (1).....	33

Figure 29: Prediction results (2).....	34
Figure 30: ROC curve of each model.....	36
Figure 31: Prediction rate	36
Figure 32: Prediction results after optimization	37
Figure 33: Feature importance.....	38

List of Tables

Table 1: Software and hardware used in this project.....	4
Table 2: The explanation of each prediction variable	7
Table 3: Explanation of Values	9
Table 4: Principles and conditions of replacing methods.....	11
Table 5: Median of values according to different outcomes	13
Table 6: Median of each feature	15
Table 7: Maximum and minimum of each feature	16
Table 8: Boxplot before and after outliers' removing	17
Table 9: The formulas and use conditions of normalization method	18
Table 10: Explanation of Outputs.....	25
Table 11: Results of KNN Model.....	26
Table 12: Results of logistic regression model.....	27
Table 13: Results of SVM(RBF) model	27
Table 14: Results of SVM(Linear) model	27
Table 15: Results of random forest model.....	28
Table 16: ROC curve of ANN	30
Table 17: Sample data	33
Table 18: Parameters of Models.....	35
Table 19: Comparison of accuracy before and after parameter adjustment	35
Table 20: Comparison of accuracy before and after mean ensemble	36
Table 21: Comparison before and after weighted ensemble	37
Table 22: Prediction results after optimization.....	37

Chapter 1

Introduction

Diabetes is a metabolic disease, the most obvious characteristic of patients with diabetes is hyperglycaemia, which is considered to be one of the most fatal and common diseases on the planet. Many people in the world have diabetes, and the number of patients is gradually increasing. Especially now people tend to have unbalanced eating habits, the lack of aerobic exercise also became one of the reasons for more and more diabetes. Of course, there are also genetic reasons, diabetes is hereditary [1]. According to the statistics of the World Diabetes Federation, by 2034, the number of people with diabetes will very likely to be more than 592 million before 2034, which is twice the current 382 million patients [2].

The blood sugar level in the patient's body will be unable to stabilize due to insufficient insulin secretion caused by imbalanced hormones and diseased pancreas. Type 1 diabetes is due to the absolute lack of insulin secretion in the body, which is more common in young people, they will become emaciated and needs insulin treatment. If the patient's blood sugar remains high due to insulin resistance or insufficient insulin secretion, it is likely to have another type of diabetes, type 2 diabetes, which is more common in middle-aged and obese people, Insulin is not needed for Type 2's early treatment [3].

Clinical medical research shows that diabetes can be reversed in the early stage, and the cost of treatment and inconvenience to patients in the early and late stage is completely incomparable. Therefore, if timely diagnosis and medical intervention can be carried out as soon as possible, it will play a vital role in the treatment of diabetes. The establishment of a rapid prediction system for diabetes has become the most urgent need in the future.

1.1 Relevant Work and My Contributions

This study refers to some authors who are engaged in this field, the following is a summary of some of the work and the improvements I have made based on them:

Nida Guler: have no Pre-processing and no normalization process.

Vincent Lugat: use median value to replace null values and outliers, 5-Folds Cross Validation.

Jason Li: use mean value to replace null values, but have no outliers' process, train: test=7:3, random_state controlled.

Chirag Samal: Dropped null values, have no outliers' process, train: test=8:2.

Riddhi Mehta: null values and outliers' processing are referred.

Tariq Mehmood and Harsh Mishra: use median for zeros and outliers.

My work: I combine the above framework, use median to replace null values and outliers, Z-Score normalization, train: test=8:2, random_state controlled, 5 models (KNN, LR, RBF-SVM, Linear-SVM, ANN, RF), grid search to optimize parameters, two ensemble functions, application to set the trained model. And for each step, the reasons and analysis for selecting this method are given in the report. It is worth mentioning that I am very much in favour of the learning framework constructed by Tariq Mehmood and Harsh Mishra, but there are some problems that I have modified: 1. They ignored the outliers of insulin; 2. Some outliers have not been cleared completely, which have been improved in my process.

Next is the details of my work:

At first, I customized the general process framework, as shown in Figure 1:

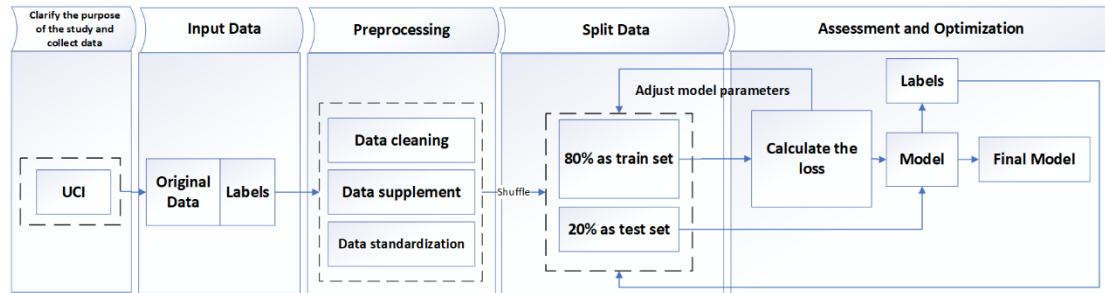


Figure 1: Process framework of the project

In chapter 1, I introduced various tools that need to be prepared, and then decided to choose OSEMN as the basic theoretical framework.

In chapter 2, the process of data pre-processing is introduced in detail, including the removal of null values, the removal of outliers and the visualization process of each important part. The formula is used to explain why this method is optimal for pre-processing.

In chapter 3, I explained the principle of each model, including some formula derivation, introduces how to evaluate the advantages and disadvantages of the model, and then use several tables to sort out the fitting situation of each model.

Finally, in the chapter 4, I introduced the program interface which can realize the prediction results by using the model, and showed the prediction results. After that, several optimization methods were introduced, and the effect comparison before and after optimization was displayed., the optimized model was imported into the interface, and the results were compared again. After that, I make my conclusion of this project according to the results, and I introduced the shortcomings in the project and what I goal need to continue to achieve in the future.

1.2 Machine Learning

In the medical field, machine learning has become an indispensable intelligent tool, which is very suitable for induction of diagnosis and prognosis rules, as well as for solving small and specialized diagnosis and prognosis problem [4].

1.2.1 Machine Learning in Medical

Firstly, in the diagnosis process, the corresponding data is obtained by interviewing the patient. After that, the doctor determines the diagnosis result based on the recorded patient data. This process is called the patient's initial diagnosis. The doctor will consider all the information provided by the patient to judge, and then will prescribe the treatment according to the diagnosis results the whole process can be repeated at the end. In the process of each repeated diagnosis, the doctor can deny, confirm or improve the diagnosis based on the actual medical problem [5].

The amount of prediction data stored in professional hospitals or clinics is increasing every day as the number of patients diagnoses. Finally, we need to import the data into the device, and transmit the obtained patient data in an appropriate form, then combine the algorithm to learn, and finally build the model framework. Whether the performance is qualified, whether the diagnosis knowledge is readable, whether the number of tests can be minimized, whether the prediction effect can be good in the case of missing values, etc., are all the factors we need to think about in the medical diagnosis process with machine learning model [6].

1.2.2 Supervised Learning

Machine learning has three kinds of learning methods: Unsupervised Learning, Supervised Learning and Semi Supervised Learning. According to the type of data used in this study, supervised learning is chosen, because the data in the study used are labelled, so it is more suitable for supervised learning to establish models. Figure 2 is the general process of supervised learning.

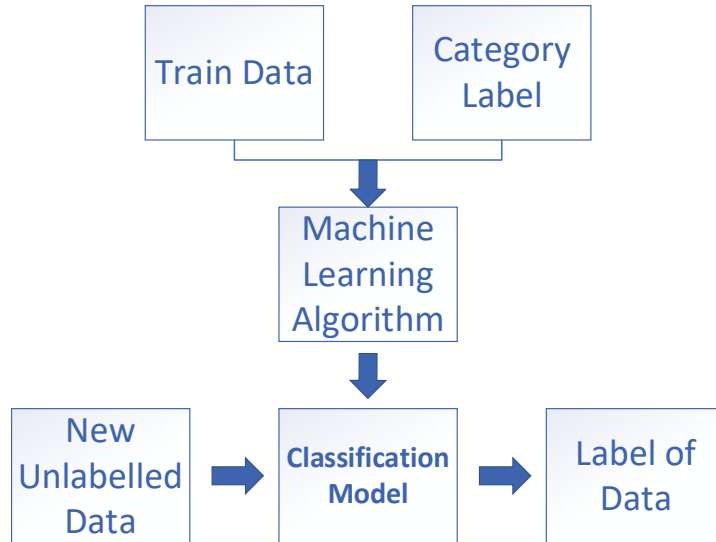


Figure 2: General process of supervised learning

Classification is one of the most common supervised learning methods. If there are only two categories of data - yes or no - then this kind of problem is called dichotomy problem. The common problems are to identify whether there is fraud, whether it is spam, whether it is a disease or not. Therefore, this experiment is very suitable for the classification with supervised learning to predict diabetes. The commonly used classification methods include neural network, k-nearest, random forest, deep learning and other algorithms.

1.3 Preparation

In order to study this project, we need to prepare for various aspects, such as environment configuration, hardware selection, software selection. The items needed for this project are listed in the Table 1.

Environment	Description	Reason to choose
Windows10	It is the last version released by Microsoft in the United States. It is the most popular system version installed in personal computers	Stable, fast and compatible
Software		
Pycharm	Pycharm is a powerful Python editor with cross platform features	Convenient to write code, powerful
Anaconda3	Open source python integrated compilation software	Integrated almost all the libraries for machine learning
Hardware		
Intel Core i7-9750	CPU developed by Intel Corporation	The operation of algorithm program depends on CPU operation

Table 1: Software and hardware used in this project

1.4 OSEM Principle

The OSEM process is a standardized process proposed in 2010 to describe the pattern of machine learning models and is widely used in the field of data science. The introduction of the process theory has greatly helped the solution of data analysis problems [7]. According to the theory, machine learning can be roughly divided into five processes: first, to obtain data, then to clean up the data; second, to visualize the data; the fourth step is to model the data; finally, the most important is to interpret the data. This workflow provides us with a clear idea and research route in the process of machine learning [8].

Data acquisition is the first step in the whole machine learning research. In any machine learning process, without data, it is impossible to continue to do the following steps. We can usually collect data ourselves or use data collected by others for experiments.

Before processing and analysing the obtained data, there are usually several operations to be performed:

1. Merge each data column into a table (because this study uses the data collected and integrated by predecessors, so this operation is not needed)
2. Clearing data from invalid values
3. Data normalization [9]

The visualization of data is very helpful for the subsequent processing of data, not only because the optimal method is often obtained by the visualized data, but also for the special elements such as outliers, we can also observe them clearly by this way [10].

Data modelling refers to the use of mathematical expressions to predict or classify each feature, abstracting each feature into a normal form that humans can't understand. Before modelling, we often need to standardize the data and other pre-operations [11].

Interpreting data means that the questions that lead to the data modelling process and requirements must be fully or partially answered. In this step, we often have to evaluate the data, and then draw corresponding conclusions according to the evaluation report, so as to obtain information beyond the data itself [8]. The specific process of OSEM is shown in Figure 3:

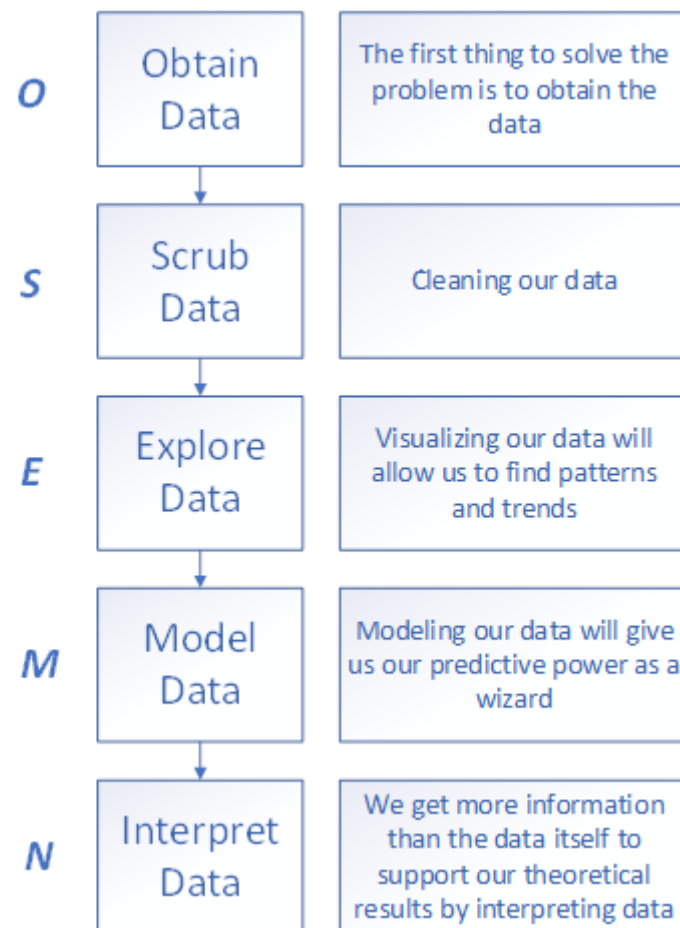


Figure 3: The principle and process of OSEMN

Chapter 2

Data Pre-processing

2.1 Obtain the Data

In this project, the Pima data set was collected from a diabetes research centre in the United States for several characteristics of diabetic patients. The original intention of making this data set is to help patients with corresponding diseases by using data science methods.

2.1.1 The Information of Dataset

The data set collected information of Pima women over the age of 21, and the data set consists of multiple medical prediction variables and a target variable. The predictive variables included the number of pregnancies, BMI, insulin level and age.

2.1.2 Data Description

The data set contains medical records of Pimas and whether they had diabetes in the past five years. All the data are numbers. The question is (whether there is diabetes is 1 or 0), it's a dichotomous problem. Data has 8 attributes and 2 categories. The meanings of each variable are shown in Table 2.

Prediction variable	Explanation
Pregnancies	Number of pregnancies
Glucose	Glucose test value
Bloodpressure	Blood pressure (mm Hg)
SkinThickness	Cortical thickness (mm)
Insulin	Serum insulin within 2 hours(mu U / ml)
BMI	Body mass index (weight / height) ^ 2
DiabetesPedigreeFunction	Diabetes genetic function
Age	Age (years)
Outcome	Class label variable (0 or 1)

Table 2: The explanation of each prediction variable

2.2 Data Pre-processing Plans

- First of all, it is necessary to make clear how many features there are, which are continuous and which are categories.

- Check whether there are missing values, and select the appropriate way to make up for the true features, so as to make the data complete.
- Standardization of continuous numerical features.
- The classification results are output by one hot coding.
- Binarize continuous data and convert it into category data.
- To prevent over fitting or other reasons, choose whether to regularize the data.
- After a preliminary study of the data, it is found that the effect is not good. We can try to use the polynomial method to find the nonlinear relationship.
- According to the actual problems, it is analysed whether the corresponding function transformation is needed.

2.3 Import Dependency Library / Package

First of all , we should install and import the dependent packages required by the notebook in this project. The specific imported packages are shown in Figure 4 and Figure 5.

```
# Classic, data manipulation and linear algebra
import numpy as np # Data science computing tools
import pandas as pd # Visualization

# Data visualization
import matplotlib.pyplot as plt # Numerical tools
import seaborn as sns # Advanced API of Matplotlib
import plotly.graph_objs as go
import plotly.offline as py
from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.tools as tls
import plotly.figure_factory as ff
py.init_notebook_mode(connected=True)
import squarify

%matplotlib inline
# Drawing plot in notebook

from sklearn.model_selection import train_test_split, GridSearchCV, KFold # Divide data test and train set/Automatic parameter adjustment
from sklearn.preprocessing import StandardScaler # De mean and variance normalization
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, accuracy_score, auc, precision_recall_curve, roc_auc_score
# Confusion matrix/Show main classification indicators/Receiver operating characteristic/Classification accuracy score/Area Under Curve
from sklearn.linear_model import LogisticRegression # Logistic regression classification
from sklearn.svm import SVC #svm. SVC classification
from sklearn.neighbors import KNeighborsClassifier #KNN classification
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier #Randomforest classification
from keras.utils import np_utils # Be similar to tf.one_hot()
from keras.models import Sequential, load_model # sequential model/load model
from keras.layers import Dense, Dropout, Activation, Conv2D, MaxPooling2D, Flatten
# Fully connected neural network layer/Dropout layer to prevent over fitting/Convolution neural network/MaxPooling layer/Flatten layer
import pickle # ackaging Model
```

Figure 4: Dependency libraries and packages (1)

```
# Stats
import scipy.stats as ss
from scipy import interp
from scipy.stats import randint as sp_randint
from scipy.stats import uniform as sp_uniform

# correlation matrix
from mlens.visualization import corrrmat
# interface
from tkinter import *
import numpy as np
import pandas as pd
import pickle
from sklearn.preprocessing import StandardScaler

#ignore warning messages
import warnings
warnings.filterwarnings('ignore')
#define the seed
random_seed=2008
```

Figure 5: Dependency libraries and packages (2)

2.4 Import and Observe Data

DataFrame.describe(): this method descriptive statistical information was generated to summarize the central trend, dispersion and shape of data set distribution, excluding null value. The meaning of each value in the table generated by this method is shown in Table 3, and the result generated by this method is shown in Figure 6.

Value	Explanation
count	the number of Non-empty rows in each feature.
mean	the mean value of each feature
std	the Standard Deviation Value of each feature
min	the minimum value of each feature
25%, 50%, 75%	the percentile/quartile of each feature.
max	the maximum value of each feature

Table 3: Explanation of Values

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

Figure 6: Description of the dataset

From the values of the items listed in the table, we can find that there are 768 rows and 9 columns of data from the count row, and there is no null value, Min can get the average, the highest and the lowest level of each characteristic in the sample population, and can preliminarily judge which items may have abnormal values; from the STD item, we can preliminarily judge the dispersion degree of the data, except that the distribution of the insulin term is relatively scattered, other data distribution sets can be more intuitive from 25%, 50%, 75%, and can be compared with Max and min It can be concluded that each feature has a certain number of outliers, which need to be removed later. The most important thing is that although we can't get null feedback from the count column, it's practical for a normal person to have a pregnancy record of 0, but it is completely illogical for life indicators such as blood sugar and blood pressure to be 0. I am sure that these values are the null values that were previously ignored by the null value test and we need find a way to replace them.

2.5 Count and Remove Null Values

There are many reasons for the existence of null values in the data, mainly including the following:

- Some information couldn't be obtained temporarily, resulting in a part of attribute value vacancy.
- Some information is lost due to some human factors.
- Some properties of some objects are not available, like the name of the spouse of an unmarried person.
- The cost of obtaining the information is too high to obtain the data.

The effects of null values on the experimental results include:

- Loss of valid information
- Data stability becomes weak
- The excavation process will become chaotic, resulting in loss of reliability

Removing null value is the first step of data pre-processing, and it is particularly important for subsequent data mining

2.5.1 Count Null Values

df_copy.isnull().sum(): This method can check the null value in the data, but according to the situation described in the previous data, the null value exists here, but it cannot be sniffed by the method. Therefore, the value of 0 needs to be replaced with the Nan value that the method can detect. The specific process and result of counting null values is shown in Figure 7 and Figure 8:

```
df_copy = df.copy(deep = True) # Create a copy to avoid affecting the original data
df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] = \
df_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']].replace(0, np.NaN)
# Replace the 0 value with a null value
print(df_copy.isnull().sum()) # showing the count of NaNs
```

Figure 7: Process of counting null values

Pregnancies	0
Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11
DiabetesPedigreeFunction	0
Age	0
Outcome	0
dtype:	int64

Figure 8: Result of counting null values

After the above codes are replaced, the null values of 5, 35, 227, 374 and 11 exist in the features of glucose, bloodpressure, skinhickness, insulin and BMI, which need further replacement. It is worth noting that although the previous null value judgment proves that there is also null value in pregnancies feature, considering that female patients are not pregnant in reality, the null value here can be considered as not to be processed.

2.5.2 Select Replacement Method

Generally, there are two options for the above-mentioned null values:

1. Directly delete the items with null values;
2. Replace the null values with another generally representative value

The size of the data set we used is not very sufficient, and the proportion of null value of single insulin has reached nearly 50% of the total sample size. Therefore, we choose the second method, and replace it with specific values. The method, principle and conditions of replacement are shown in Table 4:

Method	Principle	Condition
Mode/Mean	Maximum probability possible value	Not Num/Num
Median	Similar to Mean	Data has outliers
Hot Deck	Closest data	Not complex data
Clustering	Closest data with Euclidean distance	data correlation is weak

Table 4: Principles and conditions of replacing methods

2.5.3 Calculating the Pearson Product

Here, in order to study the correlation and complexity of data, we introduce the concept of **Pearson correlation coefficient**. The correlation coefficient is proposed to measure the linear correlation between two variables.

Pearson correlation coefficient between two variables is defined as the quotient of covariance and standard deviation between two variables:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

$$\rho_{X,Y} = \frac{1}{n} \sum_{i=1}^n \left(\frac{X_i - \bar{X}}{\sigma_X} \right) \left(\frac{Y_i - \bar{Y}}{\sigma_Y} \right)$$

In the formula $\left(\frac{X_i - \bar{X}}{\sigma_X} \right)$, \bar{X} and σ_X are the standard score, average value and standard deviation of the sample respectively. If the value of $\rho_{X,Y}$ is closer to -1 or 1, the negative correlation between them is greater, and vice versa. The code and result of Pearson correlation matrix is shown in Figure 9 and Figure 10:

```
datacor=np.corrcoef(df,rowvar=0)# Pairwise calculation of Pearson product
datacor=pd.DataFrame(data=datacor,columns=df.columns,index=df.columns)
plt.figure(figsize=(8,8))# Set the size of the image
ax=sns.heatmap(datacor,square=True,annot=True,
                linewidth=0.5,cmap='YlGnBu',
                cbar_kws={'fraction':0.046,'pad':0.03}) # Heat mapping
ax.set_title('correlation')
plt.show()
```

Figure 9: Codes of Pearson correlation matrix

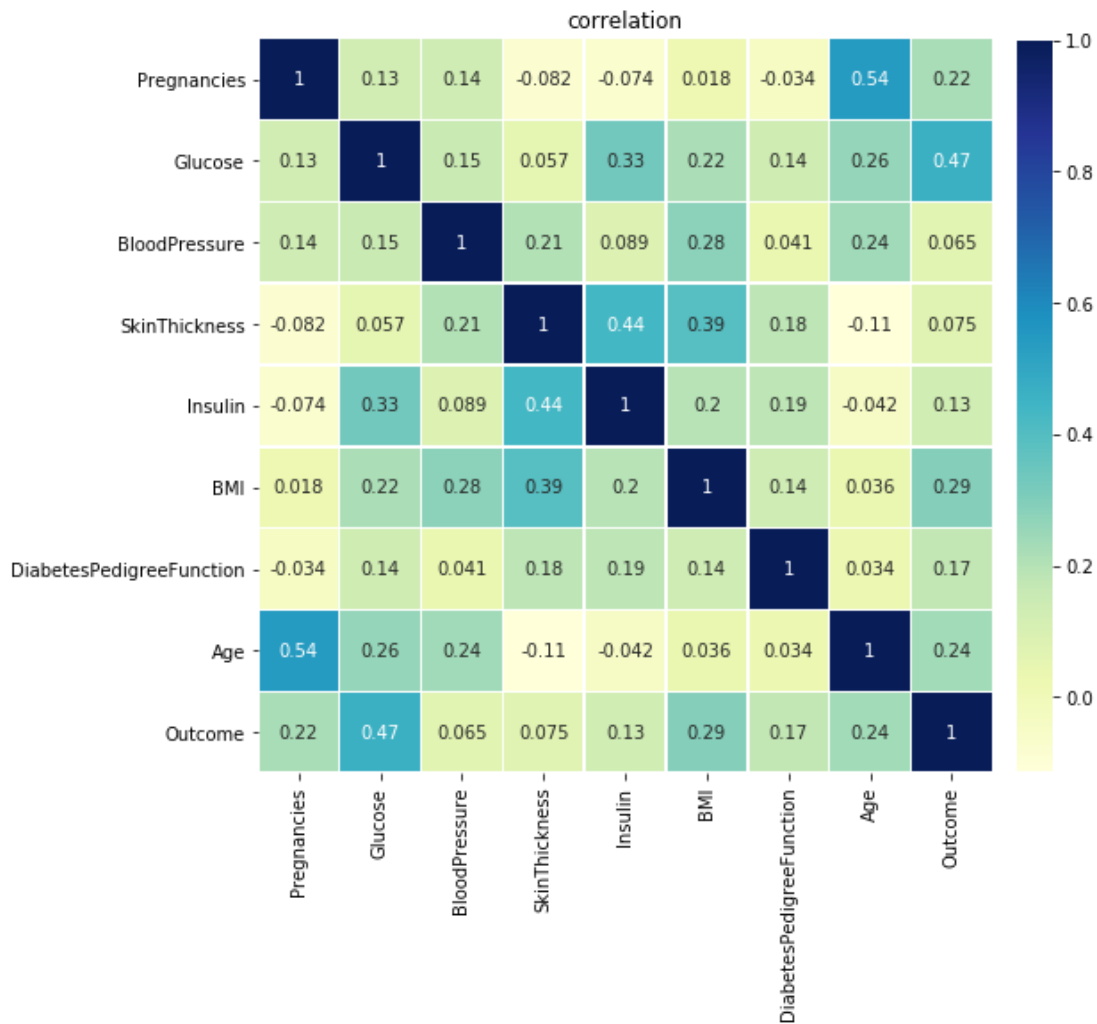


Figure 10: Pearson correlation matrix

According to the Pearson product heat map drawn, it can be seen that the relationship between the data features is relatively complex, and there is a certain degree of autocorrelation. For example, the Pearson product between Age and Pregnancies is 0.54, and there is a strong positive correlation. The correlation coefficients between insulin, BMI and SkinThickness are 0.44 and 0.39 respectively, and there is a certain positive correlation, so hot cannot be selected Deck and clustering are used to deal with the missing values. Combined with the situation of abnormal values in the previous judgment, the median is used to replace the missing values.

2.5.4 Replace Null Values

Calculate the median of each feature according to different outcomes, the result is in Table 5 and replace it with the code in Figure 11. After replacement, execute the null value statistics method again, and the results are shown in the Figure 12.

Outcome	Median	
	0	1
Glucose	107.0	140.0
BloodPressure	70.0	74.5
SkinThickness	27.0	32.0

Insulin	102.5	169.5
BMI	30.1	34.3

Table 5: Median of values according to different outcomes

```
#Fill in the null value with the median according to the property and the outcome
df.loc[(df['Outcome']==0)&(df['Glucose'].isnull()),'Glucose']=107.0
df.loc[(df['Outcome']==1)&(df['Glucose'].isnull()),'Glucose']=140.0
df.loc[(df['Outcome']==0)&(df['BP'].isnull()),'BP']=70.0
df.loc[(df['Outcome']==1)&(df['BP'].isnull()),'BP']=74.5
df.loc[(df['Outcome']==0)&(df['Insulin'].isnull()),'Insulin']=102.5
df.loc[(df['Outcome']==1)&(df['Insulin'].isnull()),'Insulin']=169.5
df.loc[(df['Outcome']==0)&(df['BMI'].isnull()),'BMI']=30.1
df.loc[(df['Outcome']==1)&(df['BMI'].isnull()),'BMI']=34.3
df.loc[(df['Outcome']==0)&(df['Skin'].isnull()),'Skin']=27.0
df.loc[(df['Outcome']==1)&(df['Skin'].isnull()),'Skin']=32.0
```

Figure 11: Code of replacing null values

```
Pregnant      0
Glucose       0
BP            0
Skin          0
Insulin       0
BMI           0
Pedigree_Function  0
Age           0
Outcome       0
dtype: int64
```

Figure 12:Result after replacing null values

2.6 Treatment of Outliers

The existence of outliers will affect the accuracy of data. Unreasonable values in the data are called outliers, and the removal of outliers helps to standardize and generalize the model.

2.6.1 Boxplot

Boxplot is a statistical chart used to show the dispersion of a set of data, It can roughly estimate the skewness and tail weight of data, and roughly understand the shape of data distribution. The most important thing is that box diagram can help us identify outliers. First, the box diagram is drawn according to the actual data, and it is not necessary to assume the distribution of data in advance. In other words, the limit of log data is relatively small. Secondly, the index for judging the abnormal value of box diagram is based on quartile and quartile for example, data larger than 25% will not affect the quartile, and the outliers obtained are more objective.

The size of the box, IQR (Interquartile Range) depends on the quartile distance of the data:

$$IQR = Q3 - Q1$$

In the formula, $Q3$ is 75% quantile, $Q1$ is 25% quantile, $Q3$ and $Q1$ are quartiles. The horizontal line in the box is the median 50% of the data is concentrated in the box. he distribution is discontinuous, and unstable data shows that the frame is very large. On the contrary, the frame of centralized data tends to be small. The maximum and minimum values of the non-abnormal points of the data are represented by the upper and lower edges of the box, which is called the upper and lower adjacent value. If the

data value $> Q3 + 1.5 * IQR$ (upper limit value) or data value $< Q1 - 1.5 * IQR$ (lower limit value), they are regarded as abnormal values. Data value $> Q3 + 3 * IQR$ or data value $< Q1 - 3 * IQR$ are regarded as extreme values. In practical application, the boundary between outliers and extremum will not be displayed, and they are generally referred to as outliers. Figure 13 is an example of a boxplot.

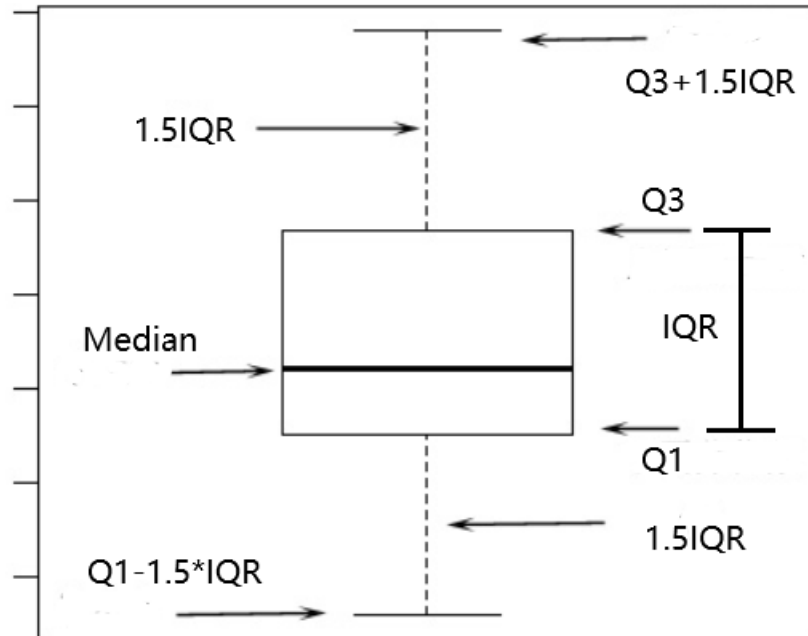


Figure 13: Sample of boxplot

2.6.2 Replacement of Outliers

First of all, we use Seaborn to realize data visualization.

Seaborn.boxplot() method draws the boxplot of each feature to roughly judge whether there is abnormal value in the data to consider whether to do further processing. The results are shown in Figure 14:

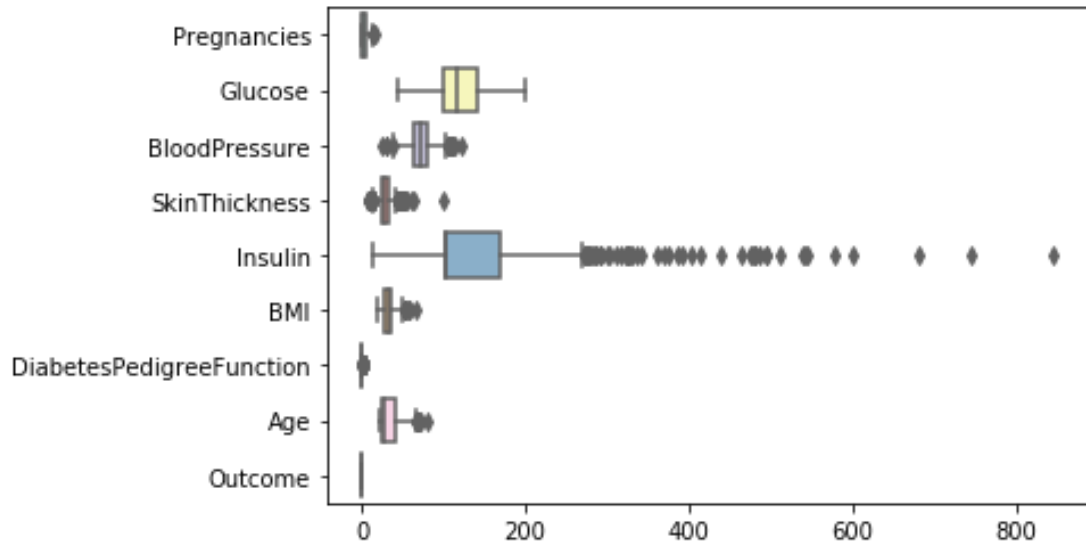


Figure 14: Boxplot of general features

From the boxplot, we can find that there are some outliers in other features except Glucose. Therefore, we need to draw a separate boxplot for each feature, observe and record the maximum and minimum non outlier values on the upper edge and the lower edge. Similar to the null value replacement rule mentioned before, the value in each feature exceeding the maximum value and the value less than the maximum value will be replaced with the median value. Table 6 shows the median of each feature, the maximum and minimum value of non-outlier, and the individual boxplot of each data before and after outliers processing.

Outcome	0	1
Median		
Pregnancies	2	4
Glucose	107.0	140.0
BloodPressure	70.0	74.5
SkinThickness	27.0	32.0
Insulin	102.5	169.5
BMI	30.1	34.3
DiabetesPedigreeFunction	0.336	0.449
Age	27	36

Table 6: Median of each feature

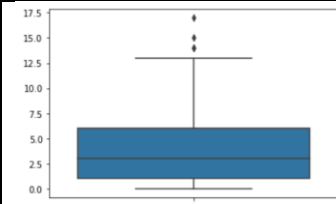
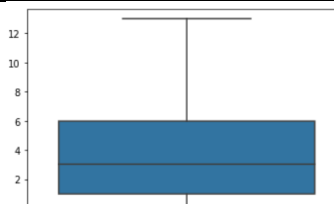
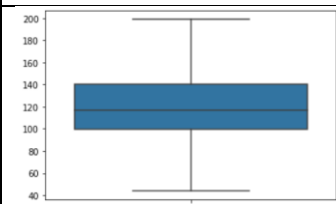
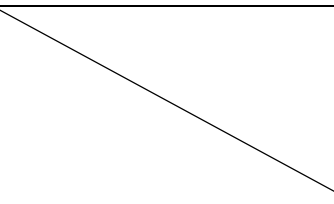
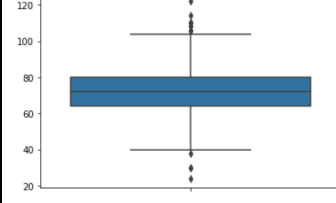
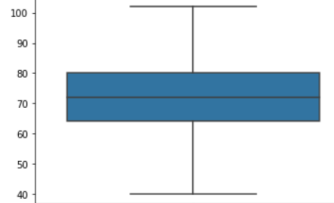
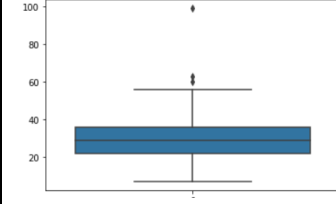
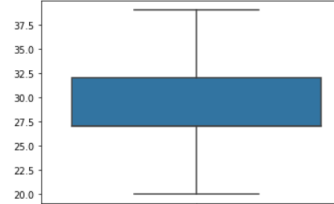
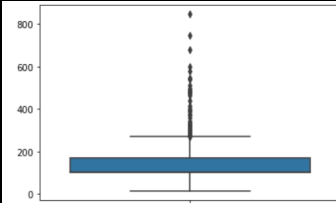

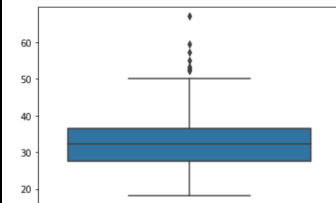
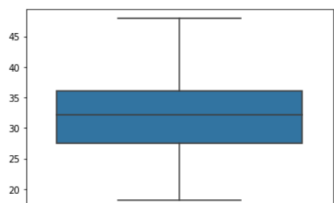
First of all, I observed the boxplot of each feature and listed the values of min and max in Table 7. I find that the values of the features Pregnancies, BMI and DiabetesPedigreeFunction are larger than the upper limit, which are 13, 48 and 1, respectively. The features of BloodPressure and SkinThickness have both values greater than the upper limit and values less than the lower limit, which are 40 and 103, 20 and 39, respectively. The boundary with abnormal values in the table has been marked in bold.

Outlier	Minimum	Maximum
Feature		
Pregnancies	0	13

Glucose	44	200
BloodPressure	40	103
SkinThickness	20	39
Insulin	14	270
BMI	18.2	48
DiabetesPedigreeFunction	0.078	1
Age	21	61

Table 7: Maximum and minimum of each feature

Table 8 shows the boxplot of each feature before and after removing outliers and the comparison of data distribution.

Features	Boxplot & Part of Data Outliers Distribution																																																																																																			
	Before Outliers Replacement		After Outliers Replacement																																																																																																	
Pregnancies		<table><tr><td>1</td><td>135</td></tr><tr><td>0</td><td>111</td></tr><tr><td>2</td><td>103</td></tr><tr><td>3</td><td>75</td></tr><tr><td>4</td><td>68</td></tr><tr><td>5</td><td>57</td></tr><tr><td>6</td><td>50</td></tr><tr><td>7</td><td>45</td></tr><tr><td>8</td><td>38</td></tr><tr><td>9</td><td>28</td></tr><tr><td>10</td><td>24</td></tr><tr><td>11</td><td>11</td></tr><tr><td>13</td><td>10</td></tr><tr><td>12</td><td>9</td></tr><tr><td>14</td><td>2</td></tr><tr><td>15</td><td>1</td></tr><tr><td>17</td><td>1</td></tr></table>	1	135	0	111	2	103	3	75	4	68	5	57	6	50	7	45	8	38	9	28	10	24	11	11	13	10	12	9	14	2	15	1	17	1		<table><tr><td>1</td><td>135</td></tr><tr><td>0</td><td>111</td></tr><tr><td>2</td><td>103</td></tr><tr><td>3</td><td>75</td></tr><tr><td>4</td><td>72</td></tr><tr><td>5</td><td>57</td></tr><tr><td>6</td><td>50</td></tr><tr><td>7</td><td>45</td></tr><tr><td>8</td><td>38</td></tr><tr><td>9</td><td>28</td></tr><tr><td>10</td><td>24</td></tr><tr><td>11</td><td>11</td></tr><tr><td>13</td><td>10</td></tr><tr><td>12</td><td>9</td></tr></table>	1	135	0	111	2	103	3	75	4	72	5	57	6	50	7	45	8	38	9	28	10	24	11	11	13	10	12	9																																		
1	135																																																																																																			
0	111																																																																																																			
2	103																																																																																																			
3	75																																																																																																			
4	68																																																																																																			
5	57																																																																																																			
6	50																																																																																																			
7	45																																																																																																			
8	38																																																																																																			
9	28																																																																																																			
10	24																																																																																																			
11	11																																																																																																			
13	10																																																																																																			
12	9																																																																																																			
14	2																																																																																																			
15	1																																																																																																			
17	1																																																																																																			
1	135																																																																																																			
0	111																																																																																																			
2	103																																																																																																			
3	75																																																																																																			
4	72																																																																																																			
5	57																																																																																																			
6	50																																																																																																			
7	45																																																																																																			
8	38																																																																																																			
9	28																																																																																																			
10	24																																																																																																			
11	11																																																																																																			
13	10																																																																																																			
12	9																																																																																																			
Glucose		<table><tr><td>99.0</td><td>17</td></tr><tr><td>100.0</td><td>17</td></tr><tr><td>129.0</td><td>14</td></tr><tr><td>106.0</td><td>14</td></tr><tr><td>111.0</td><td>14</td></tr><tr><td>..</td><td>..</td></tr><tr><td>182.0</td><td>1</td></tr><tr><td>169.0</td><td>1</td></tr><tr><td>160.0</td><td>1</td></tr><tr><td>62.0</td><td>1</td></tr><tr><td>149.0</td><td>1</td></tr></table>	99.0	17	100.0	17	129.0	14	106.0	14	111.0	14	182.0	1	169.0	1	160.0	1	62.0	1	149.0	1		<table><tr><td>92.0</td><td>8</td></tr><tr><td>75.0</td><td>8</td></tr><tr><td>65.0</td><td>7</td></tr><tr><td>85.0</td><td>6</td></tr><tr><td>94.0</td><td>6</td></tr><tr><td>48.0</td><td>5</td></tr><tr><td>96.0</td><td>4</td></tr><tr><td>44.0</td><td>4</td></tr><tr><td>100.0</td><td>3</td></tr><tr><td>98.0</td><td>3</td></tr><tr><td>110.0</td><td>3</td></tr><tr><td>50.0</td><td>2</td></tr><tr><td>46.0</td><td>2</td></tr><tr><td>108.0</td><td>2</td></tr><tr><td>55.0</td><td>2</td></tr><tr><td>104.0</td><td>2</td></tr><tr><td>102.0</td><td>1</td></tr><tr><td>98.0</td><td>1</td></tr><tr><td>114.0</td><td>1</td></tr><tr><td>122.0</td><td>1</td></tr><tr><td>61.0</td><td>1</td></tr><tr><td>24.0</td><td>1</td></tr><tr><td>40.0</td><td>1</td></tr><tr><td>58.0</td><td>1</td></tr></table>	92.0	8	75.0	8	65.0	7	85.0	6	94.0	6	48.0	5	96.0	4	44.0	4	100.0	3	98.0	3	110.0	3	50.0	2	46.0	2	108.0	2	55.0	2	104.0	2	102.0	1	98.0	1	114.0	1	122.0	1	61.0	1	24.0	1	40.0	1	58.0	1																										
99.0	17																																																																																																			
100.0	17																																																																																																			
129.0	14																																																																																																			
106.0	14																																																																																																			
111.0	14																																																																																																			
..	..																																																																																																			
182.0	1																																																																																																			
169.0	1																																																																																																			
160.0	1																																																																																																			
62.0	1																																																																																																			
149.0	1																																																																																																			
92.0	8																																																																																																			
75.0	8																																																																																																			
65.0	7																																																																																																			
85.0	6																																																																																																			
94.0	6																																																																																																			
48.0	5																																																																																																			
96.0	4																																																																																																			
44.0	4																																																																																																			
100.0	3																																																																																																			
98.0	3																																																																																																			
110.0	3																																																																																																			
50.0	2																																																																																																			
46.0	2																																																																																																			
108.0	2																																																																																																			
55.0	2																																																																																																			
104.0	2																																																																																																			
102.0	1																																																																																																			
98.0	1																																																																																																			
114.0	1																																																																																																			
122.0	1																																																																																																			
61.0	1																																																																																																			
24.0	1																																																																																																			
40.0	1																																																																																																			
58.0	1																																																																																																			
BloodPressure		<table><tr><td>92.0</td><td>8</td></tr><tr><td>75.0</td><td>8</td></tr><tr><td>65.0</td><td>7</td></tr><tr><td>85.0</td><td>6</td></tr><tr><td>94.0</td><td>6</td></tr><tr><td>48.0</td><td>5</td></tr><tr><td>96.0</td><td>4</td></tr><tr><td>44.0</td><td>4</td></tr><tr><td>100.0</td><td>3</td></tr><tr><td>98.0</td><td>3</td></tr><tr><td>110.0</td><td>3</td></tr><tr><td>50.0</td><td>2</td></tr><tr><td>46.0</td><td>2</td></tr><tr><td>108.0</td><td>2</td></tr><tr><td>55.0</td><td>2</td></tr><tr><td>104.0</td><td>2</td></tr><tr><td>102.0</td><td>1</td></tr><tr><td>98.0</td><td>1</td></tr><tr><td>114.0</td><td>1</td></tr><tr><td>122.0</td><td>1</td></tr><tr><td>61.0</td><td>1</td></tr><tr><td>24.0</td><td>1</td></tr><tr><td>40.0</td><td>1</td></tr><tr><td>58.0</td><td>1</td></tr></table>	92.0	8	75.0	8	65.0	7	85.0	6	94.0	6	48.0	5	96.0	4	44.0	4	100.0	3	98.0	3	110.0	3	50.0	2	46.0	2	108.0	2	55.0	2	104.0	2	102.0	1	98.0	1	114.0	1	122.0	1	61.0	1	24.0	1	40.0	1	58.0	1		<table><tr><td>92.0</td><td>8</td></tr><tr><td>75.0</td><td>8</td></tr><tr><td>65.0</td><td>7</td></tr><tr><td>85.0</td><td>6</td></tr><tr><td>94.0</td><td>6</td></tr><tr><td>48.0</td><td>5</td></tr><tr><td>96.0</td><td>4</td></tr><tr><td>44.0</td><td>4</td></tr><tr><td>100.0</td><td>3</td></tr><tr><td>98.0</td><td>3</td></tr><tr><td>110.0</td><td>3</td></tr><tr><td>50.0</td><td>2</td></tr><tr><td>46.0</td><td>2</td></tr><tr><td>108.0</td><td>2</td></tr><tr><td>55.0</td><td>2</td></tr><tr><td>104.0</td><td>2</td></tr><tr><td>102.0</td><td>1</td></tr><tr><td>98.0</td><td>1</td></tr><tr><td>114.0</td><td>1</td></tr><tr><td>122.0</td><td>1</td></tr><tr><td>61.0</td><td>1</td></tr><tr><td>24.0</td><td>1</td></tr><tr><td>40.0</td><td>1</td></tr><tr><td>58.0</td><td>1</td></tr></table>	92.0	8	75.0	8	65.0	7	85.0	6	94.0	6	48.0	5	96.0	4	44.0	4	100.0	3	98.0	3	110.0	3	50.0	2	46.0	2	108.0	2	55.0	2	104.0	2	102.0	1	98.0	1	114.0	1	122.0	1	61.0	1	24.0	1	40.0	1	58.0	1
92.0	8																																																																																																			
75.0	8																																																																																																			
65.0	7																																																																																																			
85.0	6																																																																																																			
94.0	6																																																																																																			
48.0	5																																																																																																			
96.0	4																																																																																																			
44.0	4																																																																																																			
100.0	3																																																																																																			
98.0	3																																																																																																			
110.0	3																																																																																																			
50.0	2																																																																																																			
46.0	2																																																																																																			
108.0	2																																																																																																			
55.0	2																																																																																																			
104.0	2																																																																																																			
102.0	1																																																																																																			
98.0	1																																																																																																			
114.0	1																																																																																																			
122.0	1																																																																																																			
61.0	1																																																																																																			
24.0	1																																																																																																			
40.0	1																																																																																																			
58.0	1																																																																																																			
92.0	8																																																																																																			
75.0	8																																																																																																			
65.0	7																																																																																																			
85.0	6																																																																																																			
94.0	6																																																																																																			
48.0	5																																																																																																			
96.0	4																																																																																																			
44.0	4																																																																																																			
100.0	3																																																																																																			
98.0	3																																																																																																			
110.0	3																																																																																																			
50.0	2																																																																																																			
46.0	2																																																																																																			
108.0	2																																																																																																			
55.0	2																																																																																																			
104.0	2																																																																																																			
102.0	1																																																																																																			
98.0	1																																																																																																			
114.0	1																																																																																																			
122.0	1																																																																																																			
61.0	1																																																																																																			
24.0	1																																																																																																			
40.0	1																																																																																																			
58.0	1																																																																																																			
SkinThickness		<table><tr><td>38.0</td><td>7</td></tr><tr><td>12.0</td><td>7</td></tr><tr><td>11.0</td><td>6</td></tr><tr><td>16.0</td><td>6</td></tr><tr><td>43.0</td><td>6</td></tr><tr><td>45.0</td><td>6</td></tr><tr><td>14.0</td><td>6</td></tr><tr><td>10.0</td><td>5</td></tr><tr><td>44.0</td><td>5</td></tr><tr><td>48.0</td><td>4</td></tr><tr><td>47.0</td><td>4</td></tr><tr><td>60.0</td><td>3</td></tr><tr><td>40.0</td><td>3</td></tr><tr><td>8.0</td><td>2</td></tr><tr><td>54.0</td><td>2</td></tr><tr><td>7.0</td><td>2</td></tr><tr><td>62.0</td><td>2</td></tr><tr><td>63.0</td><td>1</td></tr><tr><td>96.0</td><td>1</td></tr><tr><td>51.0</td><td>1</td></tr><tr><td>60.0</td><td>1</td></tr><tr><td>99.0</td><td>1</td></tr></table>	38.0	7	12.0	7	11.0	6	16.0	6	43.0	6	45.0	6	14.0	6	10.0	5	44.0	5	48.0	4	47.0	4	60.0	3	40.0	3	8.0	2	54.0	2	7.0	2	62.0	2	63.0	1	96.0	1	51.0	1	60.0	1	99.0	1		<table><tr><td>27.0</td><td>306</td></tr><tr><td>32.0</td><td>176</td></tr><tr><td>30.0</td><td>27</td></tr><tr><td>23.0</td><td>22</td></tr><tr><td>28.0</td><td>20</td></tr><tr><td>33.0</td><td>20</td></tr><tr><td>31.0</td><td>19</td></tr><tr><td>39.0</td><td>18</td></tr><tr><td>29.0</td><td>17</td></tr><tr><td>25.0</td><td>16</td></tr><tr><td>37.0</td><td>16</td></tr><tr><td>22.0</td><td>16</td></tr><tr><td>26.0</td><td>16</td></tr><tr><td>35.0</td><td>15</td></tr><tr><td>36.0</td><td>14</td></tr><tr><td>20.0</td><td>13</td></tr><tr><td>24.0</td><td>12</td></tr><tr><td>21.0</td><td>10</td></tr><tr><td>34.0</td><td>8</td></tr><tr><td>38.0</td><td>7</td></tr></table>	27.0	306	32.0	176	30.0	27	23.0	22	28.0	20	33.0	20	31.0	19	39.0	18	29.0	17	25.0	16	37.0	16	22.0	16	26.0	16	35.0	15	36.0	14	20.0	13	24.0	12	21.0	10	34.0	8	38.0	7												
38.0	7																																																																																																			
12.0	7																																																																																																			
11.0	6																																																																																																			
16.0	6																																																																																																			
43.0	6																																																																																																			
45.0	6																																																																																																			
14.0	6																																																																																																			
10.0	5																																																																																																			
44.0	5																																																																																																			
48.0	4																																																																																																			
47.0	4																																																																																																			
60.0	3																																																																																																			
40.0	3																																																																																																			
8.0	2																																																																																																			
54.0	2																																																																																																			
7.0	2																																																																																																			
62.0	2																																																																																																			
63.0	1																																																																																																			
96.0	1																																																																																																			
51.0	1																																																																																																			
60.0	1																																																																																																			
99.0	1																																																																																																			
27.0	306																																																																																																			
32.0	176																																																																																																			
30.0	27																																																																																																			
23.0	22																																																																																																			
28.0	20																																																																																																			
33.0	20																																																																																																			
31.0	19																																																																																																			
39.0	18																																																																																																			
29.0	17																																																																																																			
25.0	16																																																																																																			
37.0	16																																																																																																			
22.0	16																																																																																																			
26.0	16																																																																																																			
35.0	15																																																																																																			
36.0	14																																																																																																			
20.0	13																																																																																																			
24.0	12																																																																																																			
21.0	10																																																																																																			
34.0	8																																																																																																			
38.0	7																																																																																																			
Insulin		<table><tr><td>0</td><td>374</td></tr><tr><td>105</td><td>11</td></tr><tr><td>140</td><td>9</td></tr><tr><td>130</td><td>9</td></tr><tr><td>120</td><td>8</td></tr><tr><td>...</td><td>...</td></tr><tr><td>271</td><td>1</td></tr><tr><td>270</td><td>1</td></tr><tr><td>108</td><td>1</td></tr><tr><td>112</td><td>1</td></tr><tr><td>846</td><td>1</td></tr></table>	0	374	105	11	140	9	130	9	120	8	271	1	270	1	108	1	112	1	846	1		<table><tr><td>102.5</td><td>259</td></tr><tr><td>169.5</td><td>166</td></tr><tr><td>105.0</td><td>11</td></tr><tr><td>140.0</td><td>9</td></tr><tr><td>130.0</td><td>9</td></tr><tr><td>...</td><td>...</td></tr><tr><td>14.0</td><td>1</td></tr><tr><td>250.0</td><td>1</td></tr><tr><td>86.0</td><td>1</td></tr><tr><td>188.0</td><td>1</td></tr><tr><td>42.0</td><td>1</td></tr></table>	102.5	259	169.5	166	105.0	11	140.0	9	130.0	9	14.0	1	250.0	1	86.0	1	188.0	1	42.0	1																																																				
0	374																																																																																																			
105	11																																																																																																			
140	9																																																																																																			
130	9																																																																																																			
120	8																																																																																																			
...	...																																																																																																			
271	1																																																																																																			
270	1																																																																																																			
108	1																																																																																																			
112	1																																																																																																			
846	1																																																																																																			
102.5	259																																																																																																			
169.5	166																																																																																																			
105.0	11																																																																																																			
140.0	9																																																																																																			
130.0	9																																																																																																			
...	...																																																																																																			
14.0	1																																																																																																			
250.0	1																																																																																																			
86.0	1																																																																																																			
188.0	1																																																																																																			
42.0	1																																																																																																			
BMI		<table><tr><td>30.1</td><td>18</td></tr><tr><td>32.0</td><td>13</td></tr><tr><td>31.6</td><td>12</td></tr><tr><td>31.2</td><td>12</td></tr><tr><td>32.4</td><td>10</td></tr><tr><td>...</td><td>...</td></tr><tr><td>53.2</td><td>1</td></tr><tr><td>46.3</td><td>1</td></tr><tr><td>36.2</td><td>1</td></tr><tr><td>32.1</td><td>1</td></tr><tr><td>41.8</td><td>1</td></tr></table>	30.1	18	32.0	13	31.6	12	31.2	12	32.4	10	53.2	1	46.3	1	36.2	1	32.1	1	41.8	1		<table><tr><td>30.1</td><td>20</td></tr><tr><td>34.3</td><td>20</td></tr><tr><td>32.0</td><td>13</td></tr><tr><td>31.2</td><td>12</td></tr><tr><td>31.6</td><td>12</td></tr><tr><td>...</td><td>...</td></tr><tr><td>19.4</td><td>1</td></tr><tr><td>21.2</td><td>1</td></tr><tr><td>19.9</td><td>1</td></tr><tr><td>32.6</td><td>1</td></tr><tr><td>45.7</td><td>1</td></tr></table>	30.1	20	34.3	20	32.0	13	31.2	12	31.6	12	19.4	1	21.2	1	19.9	1	32.6	1	45.7	1																																																				
30.1	18																																																																																																			
32.0	13																																																																																																			
31.6	12																																																																																																			
31.2	12																																																																																																			
32.4	10																																																																																																			
...	...																																																																																																			
53.2	1																																																																																																			
46.3	1																																																																																																			
36.2	1																																																																																																			
32.1	1																																																																																																			
41.8	1																																																																																																			
30.1	20																																																																																																			
34.3	20																																																																																																			
32.0	13																																																																																																			
31.2	12																																																																																																			
31.6	12																																																																																																			
...	...																																																																																																			
19.4	1																																																																																																			
21.2	1																																																																																																			
19.9	1																																																																																																			
32.6	1																																																																																																			
45.7	1																																																																																																			

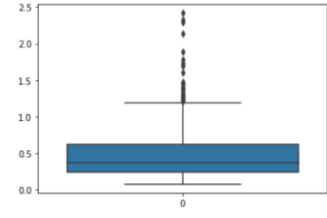
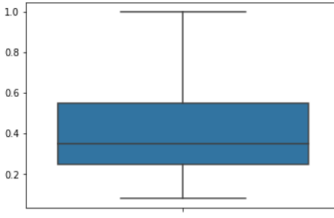
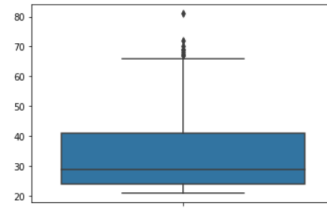
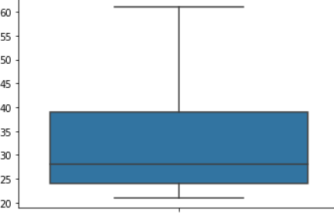
Diabetes Pedigree Function		0.254 6 0.258 6 0.259 5 0.238 5 0.207 5 .. 0.886 1 0.804 1 1.251 1 0.382 1 0.375 1		0.449 29 0.336 24 0.254 6 0.258 6 0.238 5 .. 0.773 1 0.804 1 0.382 1 0.743 1 0.375 1
Age		58 7 47 6 54 6 57 5 60 5 48 5 49 5 53 5 55 4 62 4 63 4 66 4 66 3 59 3 65 3 67 3 61 3 69 2 72 1 64 1 68 1 70 1 51 1		50 8 44 8 51 8 52 8 58 7 54 6 47 6 57 5 49 5 53 5 48 5 60 5 55 4 56 3 59 3 61 2

Table 8: Boxplot before and after outliers' removing

The right side of the table shows the boxplot and data distribution of each feature after the abnormal value replacement. It can be seen that there is no abnormal value in each feature, and the next processing can be carried out

Although I preliminarily judge that there is no abnormal value in the Glucose feature, the possibility of ignoring the abnormal value of Glucose due to too many features in the same image leading to incomplete display of image details. Therefore, I still consider Glucose as a part of the outliers processing, although it does show that there is no abnormal value in the Glucose feature.

2.7 Data Normalization

In the field of data mining, we must first standardize the data in order to eliminate the analysis errors caused by the different dimensions of the data. Only when the data is in the same dimension, we can have further evaluate and process the data.

2.7.1 Necessity

According to the principle of machine learning, we assume that the threshold value of outcome is 0 or 1 is μ , and we call the multiplication of the value and weight of each feature as "the importance of this feature", which means the influence of the feature on the final result. We assume the value of each feature is x_1, x_2, \dots, x_n , and the weights are $\theta_1, \theta_2, \dots, \theta_n$, and y is the difference between the sum of judgment and the threshold value. We can get the formula:

$$y = \sum_{i=1}^n \theta_i x_i - \mu$$

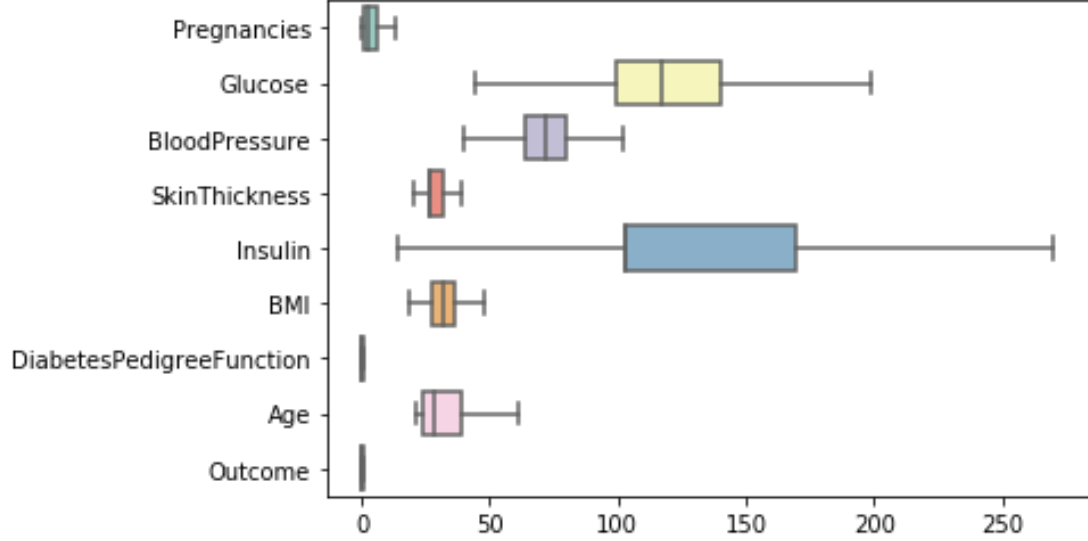


Figure 15: Boxplot after outliers' removing

We can know that the value of x in a certain feature will affect the value of y and the importance of this feature at the same time when the weight of each weight is fixed. Combining with the boxplot in Figure 15, it is obvious that the dimensions of each data are not the same, that is, Insulin and Glucose, whose values are generally large, assume that the weight of these two features is small. If their values are large, the importance of these two features will obviously occupy a relatively large proportion, but in fact, this is not reasonable, because the importance of features in machine learning should depend on the weight, so we must normalize the data to eliminate such errors.

2.7.2 Normalization Method

According to the characteristics of the data, we should choose the appropriate normalization method. Min - Max Normalization and Z-score Standardization are the two most commonly used standardization methods in the data standardization process. Their conversion formula is shown in Table 9, μ is the mean and σ is the standard deviation:

Name	Formula	Use
Min- Max Normalization	$z = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$	bring all values into the range [0,1]
Z-score Standardization.	$z = \frac{x_i - \mu}{\sigma}$	Data for normal distribution

Table 9: The formulas and use conditions of normalization method

In this study, we think that Z-score standardization is more appropriate for the following reasons, If we assume that in two-dimensional variables (x, y) , x' , y' are the converted values, σ is the covariance of data:

$$x' = x - \bar{x}$$

$$y' = y - \bar{y}$$

$$\sigma_{xy}' = \frac{1}{n} \sum_{i=1}^n (x_i' - \bar{x}')(y_i' - \bar{y}')$$

$$\sigma_{xy} = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

After the mean return to zero processing, $\bar{x} = 0, \bar{y} = 0$, So we have, $\sigma_{xy} = \sigma_{xy}'$,

After variance normalization, we have:

$$x' = \frac{x - \bar{x}}{\sigma_x}$$

$$y' = \frac{y - \bar{y}}{\sigma_y}$$

$$\sigma_{xy}' = \frac{1}{n} \sum_{i=1}^n (x_i' - \bar{x}')(y_i' - \bar{y}') = \frac{1}{n\sigma_x\sigma_y} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) = \frac{\sigma_{xy}}{\sigma_x\sigma_y}$$

We can see that after Z-score Standardization the dimension of each dimension is equivalent due to the normalization of square difference. Each dimension presents a normal distribution, where the variance is 1 and the average is 0. When calculating distance, each dimension is de dimensioned, which avoids the huge impact of different dimension selection on distance calculation.

Then turn to the Min-Max Normalization, assume that k is the rate of change of the transformation:

$$x' = k_x x$$

$$y' = k_y y$$

$$\sigma_{xy}' = \frac{1}{n} \sum_{i=1}^n (k_x x_i - k_x \bar{x})(k_y y_i - k_y \bar{y}) = k_x k_y \sigma_{xy}$$

After using Max-Min Normalization, its covariance has a multiple, which is worth scaling. Therefore, this method can't eliminate the influence of dimension on square deviation and covariance. At the same time, due to the existence of dimensions, the calculation results of distance will be different if different dimensions are used.

In general, if distance measurement (cluster analysis) or covariance analysis (PCA, LDA, etc.) are involved in the subsequent calculation of the algorithm, and the data distribution can be approximately the state distribution, the normalization method of 0-means should be used.

So in this study, we should choose Z-score Standardization as the method of data standardization.

Chapter 3

Modelling the Data

In machine learning, modelling means that we extract the features we need as the key factor to predict the expected value. Generally speaking, there are two steps:

- Training
- Prediction

Training means that our model is constantly learning how to judge the expected value, while prediction refers to the evaluation of the prediction effect after the training of our model.

3.1 Dataset Split

Train Test Split: To have unknown datapoints to test the data rather than testing with the same points with which the model was trained. This helps capture the model performance much better. The diagram of data segmentation is shown in Figure 16:

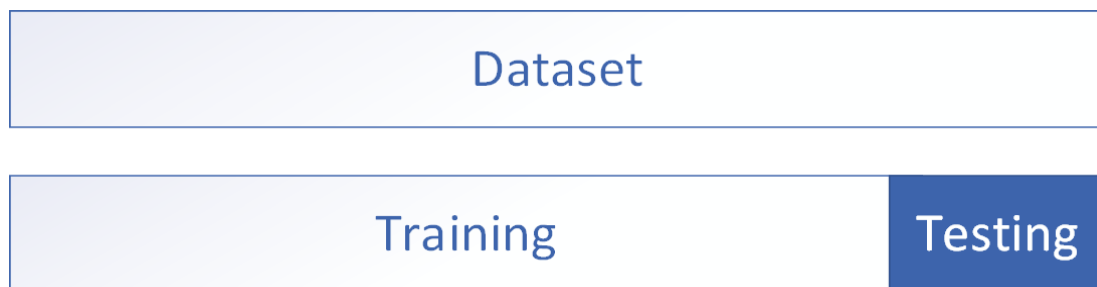


Figure 16: Train Test Split

3.2 KNN

KNN (K-Nearest Neighbour) algorithm is one of the most common and basic regression and classification algorithms. The classification principle of the algorithm is to determine which category it belongs to through the nearest K points of an object, and the final classification result is a cluster.

KNN is a case-based learning or local approximation. It is an inert learning that delays all calculations to the classification. The advantage of KNN is that it is relatively simple, but the disadvantage is that it is very sensitive to the local structure of data, and the selection of K and distance formula will affect its accuracy.

The basic process of the KNN algorithm is as follows:

Firstly, calculate the Euclidean distance between two points or tuple to find the nearest k points:

$$X1 = (x_{11}, x_{12}, \dots, x_{1n})$$

$$X2 = (x_{21}, x_{22}, \dots, x_{2n})$$

$$dist(X1, X2) = \sqrt{\sum_{i=1}^n (x_{i1} - x_{i2})^2}$$

Secondly, according to the classification of k proximity points:

$$C_x = \arg_{j \in l} \max \sum_{y \in X_k} I(C_y = j)$$

X_k is the k-nearest neighbours including y , C is the label; the return value is 1 when the label of the y is equal to j [12].

3.3 SVM

Support Vector Machine (SVM) is a data classification technology. The basic idea of SVM learning is to construct a hyperplane with the largest geometric separation distance. This plane may be linear, but it is often incomprehensible to humans. The data is divided to achieve classification. In other words, we should not only separate the positive and negative instances, but also ensure that the points closest to the hyperplane should have enough confidence to separate them. Only in this way can the hyperplane have better classification and prediction ability for unknown data sets.

For the data set which can't be classified well by using a linear plane, the nonlinear support vector machine needs to be solved, and then the kernel function is used. In order to study how to measure the similarity between two vectors, the kernel function often provides us with specific standards for measurement.

Support vector machine algorithm can have a variety of kernel functions, like linear, polynomial, RBF, sigmoid based. in this paper, we mainly use RBF and linear function [13].

3.3.1 RBF Kernel

Radial Basis Function (RBF) kernel, is defined as:

$$K(x, x') = \exp \left(-\frac{\|x - x'\|^2}{2\sigma^2} \right)$$

$\|x - x'\|^2$ can be regarded as the square Euclidean distance between two eigenvectors, σ is a free parameter, we assume that $\gamma = \frac{1}{2\sigma^2}$, we have,

$$K(x, x') = \exp (-\gamma \|x - x'\|^2)$$

Because the value of RBF kernel decreases with distance and is between 0 and 1, it is a ready-made representation of similarity measure.

3.3.2 Linear Kernel

For relatively large and complex data sets, linear SVM will perform better [14]. The basic formula of this function is:

$$K(x, x') = x^T x'$$

The advantages are: less possibility of over fitting; fast; good interpretability; we can directly get x^T , x' , which directly correspond to the weight of each feature.

The disadvantage is: not suitable for linear inseparable environment.

3.4 Logistic Regression

Logistic regression model is one of the commonly used classification methods for multi-level and multi factor classification. Logistic regression takes probability as dependent variable, and its expression is as follows [15]:

$$\ln\left(\frac{P}{1-p}\right) = a + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

In the formula, P is the probability that the event is true, and the value range is (0, 1), β_i is the Partial regression coefficient. [16]

For multi-class logistic regression models, the expression is:

$$\ln\left[\frac{\sum_{i=1}^j P_i}{1 - \sum_{i=1}^j P_i}\right] = \alpha_j + \sum_{i=1}^n \beta_i x_i$$

In the formula, $j = 1, 2, \dots, K - 1$, α_j is the intercept, β_i is the same as in the previous formula.

3.5 Random Forest

Because of its extensive application, Random Forest Algorithm is more and more frequently used in various fields to classify or predict scenarios [17]. Its central idea is to build a basic prediction or classification model by randomly integrating and voting multiple decision tree models. The specific flow chart is as Figure 17:

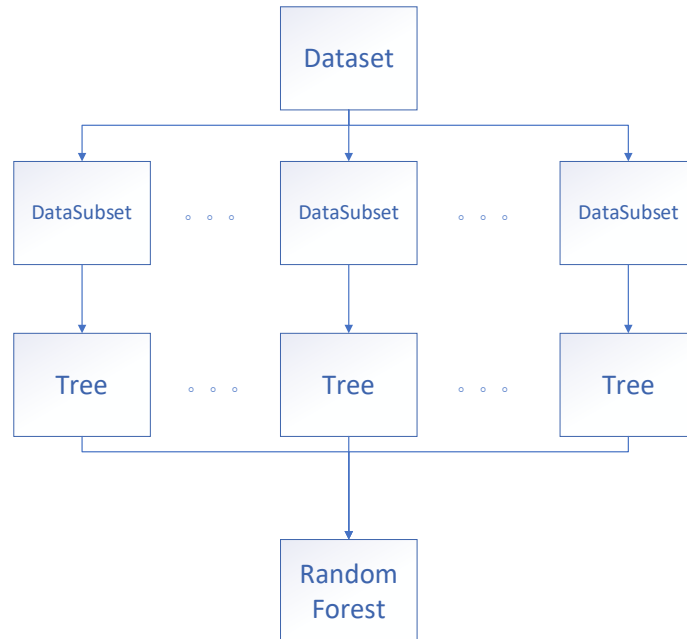


Figure 17: Flow chart of random forest

The random forest algorithm first generates multiple decision trees, each of which is equivalent to a classifier, and then assigns specific weights to each classifier after voting according to the situation. After that, each decision tree will randomly select the data in the training set to perform random forest fitting and node segmentation. For example, two small decision trees are likely to be combined to generate new branches that help continue classification. The above is the specific process of the random forest algorithm. For the decision tree based on the Gini value, the smallest Gini value will be selected as the feature segment. The Gini coefficient is calculated as follows [18]:

$$Gini(S) = 1 - \sum_{i=1}^m P_i^2$$

$$Gini_{split}(S) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2)$$

In the formula, P_i is the probability of category in the sample S . $|S|$, $|S_1|$ and $|S_2|$ is the number of S , S_1 and S_2 respectively.

The excellent robustness and fault tolerance of the random forest algorithm benefit from its own data selection mechanism, in other words, it has strong generalization.

3.6 ANN

ANN (Artificial Neural Network) is a kind of soft computing used to gradually analyse a certain complex phenomenon through modelling and self-learning [19]. Neural network can obtain external information and gradually improve the structure of the model according to these changes. Just like the construction of neural synapses when people recognize new things, neural network is also an adaptive system that uses a large number of artificial neurons to implement algorithms and models.

In the artificial neural network, one of the neural networks with relatively simple connection method is the fully connected neural network, and its structure usually has three layers: input, output and hidden, as shown in Figure 18:

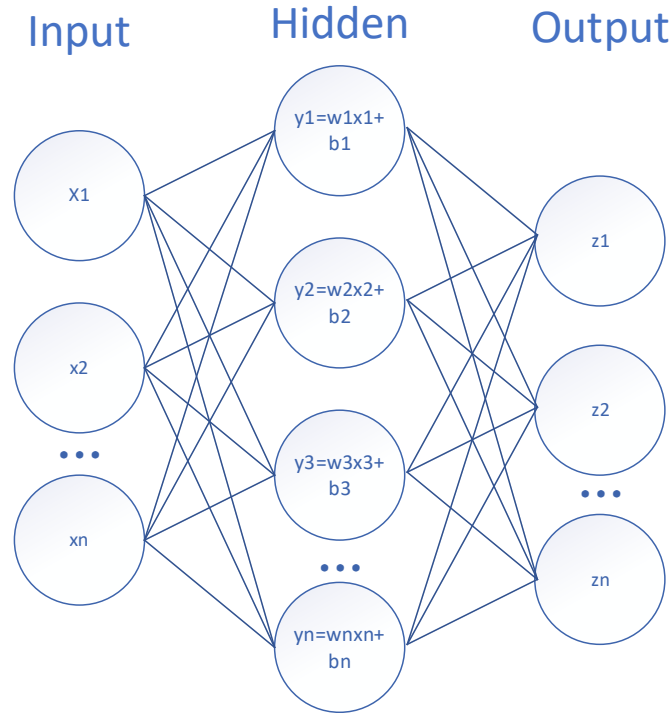


Figure 18: The structure of ANN

3.7 Evaluation

In the machine learning system, how to evaluate the model and how to judge the effect of the model is a very important process, because it directly affects whether we can analyse the shortcomings of the model according to the evaluation results, and prepare for the improvement of the model later.

3.7.1 Confusion matrix

Confusion matrix can realize the distribution of correct and wrong prediction results [20], it is based on the combination of the predicted results and the actual results, which is a matrix to analyse and summarize the prediction results. Let's take the dichotomy as an example to see the matrix representation, as Figure 19 shows:

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Figure 19: Confusion matrix

The attributes in the table are explained as follows:

True Positive: the probability that the predicted sample is positive and the real result is also positive, usually abbreviated as TP.

False Negative: the frequency that the predicted sample is negative but the real result is positive, usually abbreviated as FN.

False Positive: the frequency that the predicted sample is positive but the real result is negative, usually abbreviated as FP

True Negative: the frequency that the predicted sample is negative and the real result is negative, usually abbreviated as TN.

3.7.2 F1-score

`metrics.classification_report()`: will get a report containing the results of the main evaluation methods, The meanings of the output values are shown in Table 10:

OUTPUT	Explanation
Precision	$precision = TP / (TP + FP)$
Recall	$recall = TP / (TP + FN)$
F1-Score	$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$
Accuracy	$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
Support	It refers to how many samples are correctly classified in the corresponding class

Table 10: Explanation of Outputs

3.7.3 AUC/ROC

In order to evaluate the generalization and accuracy of the model algorithm by some specific index, we introduce the concept of Receiver Operating Characteristic (ROC) curve [21]. The horizontal axis and the vertical axis respectively represent the true positive rate and false positive rate. We usually abbreviate them as TPR and FPR respectively. The area enclosed by the ROC image on the horizontal axis is called AUC, which is an indicator to measure the quality of the model. The larger the AUC is, the more accurate the model and the stronger the generalization. Figure 20 is the sample of ROC curve:

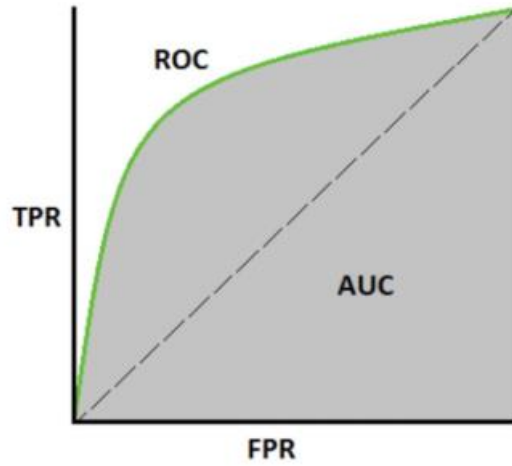


Figure 20: ROC curve

3.8 Model and Evaluation

The next step is the model fitting and evaluation. In this experiment, I fit six models, KNN, SVM (RBF), SVM (linear), logistic regression, random forest and ANN respectively, the results of the first four models are shown in Table 11-15:

KNN	precision		recall	f1-score	support
	0	0.95	0.93	0.94	100
	1	0.88	0.91	0.89	54
	accuracy			0.92	154
	macro avg		0.91	0.92	154
	weighted avg		0.92	0.92	154
Confusion Matrix	array([[93, 7], [5, 49]], dtype=int64)		AUC - ROC	 Knn(n_neighbors=8) ROC curve ROC curve(area=0.96)	

Table 11: Results of KNN Model

Logistic Regression n	precision		recall	f1-score	support
	0	0.91	0.95	0.93	100
	1	0.90	0.83	0.87	54
	accuracy			0.91	154
	macro avg	0.91	0.89	0.90	154
	weighted avg	0.91	0.91	0.91	154

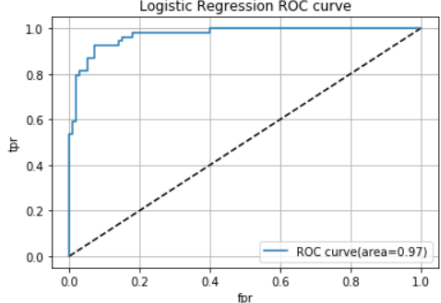
Confusion Matrix	<code>array([[95, 5], [9, 45]], dtype=int64)</code>	AUC - ROC	 <p>Logistic Regression ROC curve</p> <p>ROC curve(area=0.97)</p>
-------------------------	--	------------------	---

Table 12: Results of logistic regression model

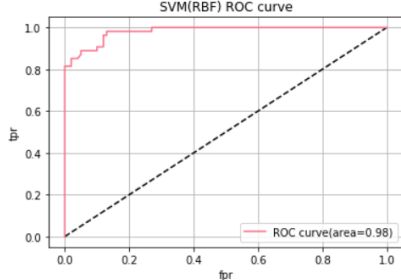
SVM(RBF)	precision		recall	f1-score	support	
	0	0.94	0.95	0.95	100	
	1	0.91	0.89	0.90	54	
	accuracy			0.93	154	
	macro avg	0.92	0.92	0.92	154	
	weighted avg		0.93	0.93	0.93	154
Confusion Matrix	array([[95, 5], [6, 48]], dtype=int64)		AUC - ROC			

Table 13: Results of SVM(RBF) model

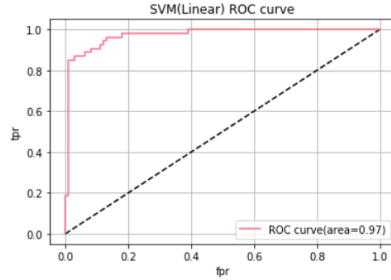
SVM(Linear)	precision		recall	f1-score	support
	0	0.93	0.96	0.95	100
	1	0.92	0.87	0.90	54
	accuracy			0.93	154
	macro avg		0.93	0.92	154
	weighted avg		0.93	0.93	154
Confusion Matrix	array([[96, 4], [7, 47]], dtype=int64)		AUC - ROC		

Table 14: Results of SVM(Linear) model

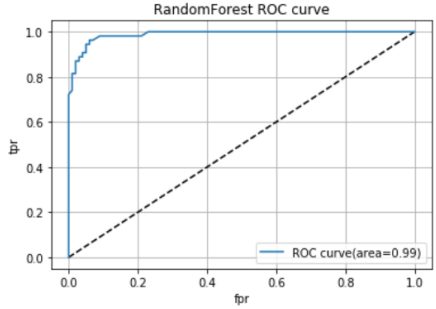
		precision	recall	f1-score	support
Random Forest	0	0.95	0.96	0.96	100
	1	0.92	0.91	0.92	54
	accuracy			0.94	154
	macro avg	0.94	0.93	0.94	154
	weighted avg	0.94	0.94	0.94	154
Confusion Matrix	array([[96, 4], [5, 49]], dtype=int64)		AUC - ROC		

Table 15: Results of random forest model

According to the above four charts, it can be found that the accuracy rate of the five models is 92.86%, 90.90%, 92.86%, 92.86% and 94.16% respectively. Although the accuracy of the two models of KNN and SVM are the same, in fact, the confusion matrix shows that the classification results are not the same, and the principle is also quite different. The reason for this result is that the test set data is too small, which leads to the same accuracy coincidence.

Before fitting ANN model, it is necessary to mention two activation functions I used in modelling: ReLU, Softmax and what are activation functions.

The change rule of neuron excitation value is defined by activation function, just like defining a rule that will change due to certain circumstances, and the rule will change due to the change of weight in the network.

The complexity of activation function execution is often a part of machine learning algorithm that can't be ignored [22]. Relu function is different from the usual relatively dense activation function, and its calculation process is relatively simple. Because the training speed of ReLU function is very fast, the period is short, the problem is less and the sample size is less, so it is often recognized by the pioneers in related fields, which is also the reason why we choose this function as the activation function of this model [23]. Moreover, in the actual model fitting process, there is no defect similar to the information explosion which may appear in the ReLU function.

The formula of ReLU is:

$$f(x) = \max(0, x)$$

In the neural network, the Softmax layer is an essential neural network layer. The function of this layer is mainly to classify the scene and optimize the neural network coefficient in the loss function [24] - [25]. Suppose an array V , where V_i represents the i th element in V , the Softmax regression formula is as follows [27]:

$$S_i = \frac{e^i}{\sum_j e^j}$$

In this experiment, we design an artificial neural network model with seven hidden layers. Take the first layer as an example: the parameter 32 indicates that the input data in the input layer will output 32 feature maps after being processed in this layer, and the second parameter 'relu' represents the activation function after processing using the linear correction unit function. The same principle applies to other layer and the Softmax layer's parameter interpretation.

The specific conditions of the model are shown in Figure 21. There are 130,772 parameters to be trained [26].

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	288
activation (Activation)	(None, 32)	0
dense_1 (Dense)	(None, 64)	2112
activation_1 (Activation)	(None, 64)	0
dense_2 (Dense)	(None, 64)	4160
activation_2 (Activation)	(None, 64)	0
dense_3 (Dense)	(None, 128)	8320
activation_3 (Activation)	(None, 128)	0
dense_4 (Dense)	(None, 128)	16512
activation_4 (Activation)	(None, 128)	0
dense_5 (Dense)	(None, 256)	33024
activation_5 (Activation)	(None, 256)	0
dense_6 (Dense)	(None, 256)	65792
activation_6 (Activation)	(None, 256)	0
dense_7 (Dense)	(None, 2)	514
activation_7 (Activation)	(None, 2)	0
Total params: 130,722		
Trainable params: 130,722		

Figure 21: Condition of ANN model

After 100 layers of iteration, the information of the model is shown in Table 16:

Loss	0.46
Accuracy	92.86%

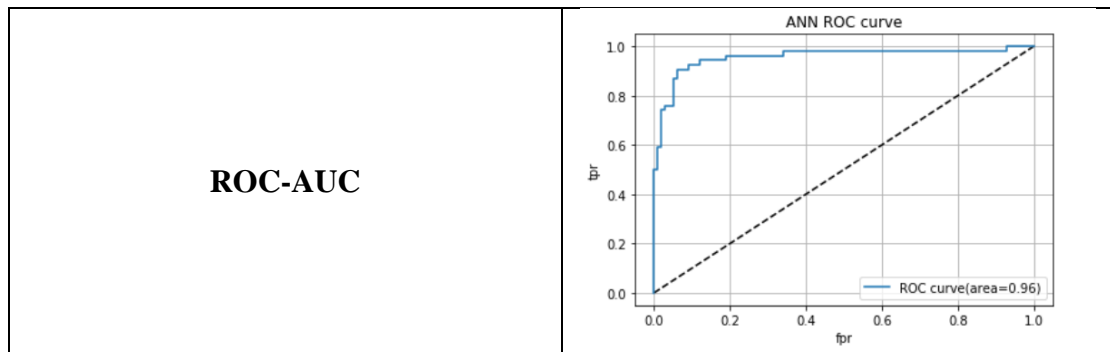


Table 16: ROC curve of ANN

The results show that the loss is basically stable between 0.3 and 0.5, the accuracy is about 92%, and the ROC value is about 0.96.

To sum up, the accuracy of the test set of the trained model is above 90%, which is within the acceptable range. The establishment of the model is satisfactory and the prediction result is good.

Chapter 4

Application and Optimization

In this study, in addition to the theoretical analysis and construction of the model, in order to achieve its prediction function, we also made an interface carrying the model to realize its actual function. I also consider the optimization of the model. In order to achieve better results, we need to consider whether we can optimize the model to achieve better results

4.1 Application

In order to put the constructed model into practical application, several important steps are preliminarily planned: 1. Model packaging; 2. Data interface; 3. Data input standardization; 4. Realization of prediction

4.1.1 Model Packaging

The first five models can use the function in the pickle module to package the fitted models, codes are shown in Figure 22:

```
f = open('saved_model/knn.pickle', 'wb')
pickle.dump(knn, f)
f.close()
```

Figure 22: Code of saving models

The ANN model can be saved with the save () method as shown in Figure 23:

```
model.save('saved_model/ANN.h5')
```

Figure 23: Code of saving ANN model

The saved model is shown as Figure 24:

ANN.h5	37 分钟前	568 kB
ANN.pickle	38 分钟前	0 B
knn.pickle	1 小时前	92.4 kB
LR.pickle	1 小时前	916 B
RF.pickle	43 分钟前	921 kB
SVM(Linear).pickle	43 分钟前	21.1 kB
SVM(RBF).pickle	44 分钟前	23.3 kB

Figure 24: Saved models

4.1.2 Data interface

About the interface making, we use Python's own **Tkinter** interface editing library to edit GUI easily. The design of the interface is that there are nine lines of input box from top to bottom, which are the **Name** of the tester, **Pregnant**, **Glucose**, **BloodPressure**, **SkinThickness**, **Insulin**, **BMI**, **DiabetesPedigreeFunction** and **Age** attributes. There is a prediction button below these input boxes. When the tester enters name and attributes, the program can automatically analyse and predict whether the tester is suffering from diabetes or not. The specific interface is shown in Figure 25:

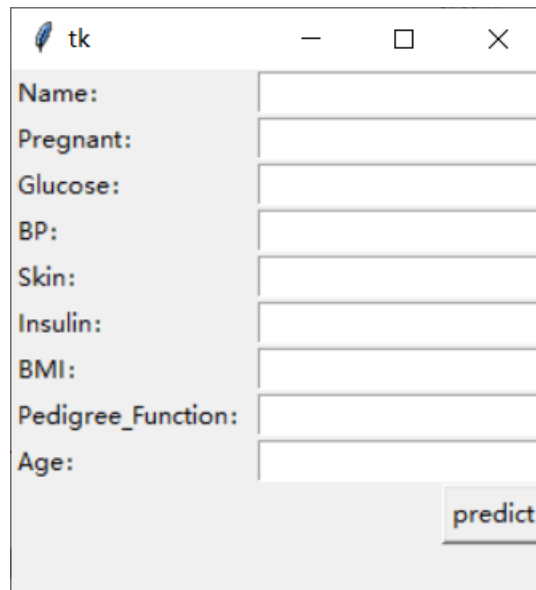


Figure 25: Prediction Interface

Giving the predict button method **pre** can obtain the value in each input box and then make prediction (taking KNN model as an example), the codes are shown in Figure 26:

```
def pre():
    Name = e_name.get()
    Preg = float(e_preg.get())
    Glu = float(e_glu.get())
    Blo = float(e_blo.get())
    Ski = float(e_ski.get())
    Ins = float(e_ins.get())
    Bmi = float(e_bmi.get())
    Dia = float(e_dia.get())
    Age = float(e_age.get())
    DATA={'Pregnant':Preg,'Glucose':Glu,'BP':Blo,'Skin':Ski,'Insulin':Ins,'BMI':\
        Bmi,'Pedigree_Function':Dia,'Age':Age}
    df=pd.DataFrame(DATA,index=[Name])
    df2=df.append(X)
    with open('knn.pickle', 'rb') as fr:
        knn = pickle.load(fr)
    predict=knn.predict(Standard(df2,Name)[0].reshape(1,-1))
    if predict == [1]:
        l_msg['text'] =Name+' has diabete'
    if predict == [0]:
        l_msg['text'] =Name+' are healthy'
```

Figure 26: Definition of 'pre' button

4.1.3 Data input standardization

As mentioned before, the model is normalized before fitting, so we need to normalize the input data with the same standard after obtaining the accurate prediction results. Therefore, we need to define a method, standard, to put the input data into the original

data, standardize the same standard, and then make the prediction. After that, we also consider the data default situation. For example, if the tester does not have the data of a certain item, the system will directly give the average data to replace. The method is shown in Figure 27:

```
def Standard(df,name):
    if df.loc[name,'Glucose']==0:
        df.loc[name,'Glucose']=121.6
    if df.loc[name,'BP']==0:
        df.loc[name,'BP']=72
    if df.loc[name,'Skin']==0:
        df.loc[name,'Skin']=29
    if df.loc[name,'Insulin']==0:
        df.loc[name,'Insulin']=141.7
    if df.loc[name,'BMI']==0:
        df.loc[name,'BMI']=32
    if df.loc[name,'Pedigree_Function']==0:
        df.loc[name,'Pedigree_Function']=0.408
    if df.loc[name,'Age']==0:
        df.loc[name,'Age']=32
    std=StandardScaler()
    df=std.fit_transform(df)
    return df
```

Figure 27:Definition of 'standard' button

4.1.4 Realization of prediction

Finally, we input several groups of data in the experimental data set to predict. The information of these sample are shown in Table 17, the results are shown in Figure 28 and 29:

Name	Pregnan t	Glucos e	B P	Ski n	Insuli n	BM I	DPF	Ag e	Outcom e
Alice	6	148	72	35	0	33.6	0.627	50	1
Rose	1	85	66	29	0	26.6	0.351	31	0
Reimu	8	183	64	0	0	23.3	0.672	32	1
Unknown	0	137	40	35	168	43.1	2.288	33	1

Table 17: Sample data

tk

Name:

Alice

Pregnant:

6

Glucose:

148

BP:

72

Skin:

35

Insulin:

0

BMI:

33.6

Pedigree_Function:

0.627

Age:

50

predict

Alice has diabete

tk

Name:

Rose

Pregnant:

1

Glucose:

85

BP:

66

Skin:

29

Insulin:

0

BMI:

26.6

Pedigree_Function:

0.351

Age:

31

predict

Rose are healthy

tk

Name:

Reimu

Pregnant:

8

Glucose:

183

BP:

64

Skin:

0

Insulin:

0

BMI:

23.3

Pedigree_Function:

0.672

Age:

32

predict

Reimu has diabete

Figure 28:Prediction results (1)

Name:	Unknown
Pregnant:	0
Glucose:	137
BP:	40
Skin:	35
Insulin:	168
BMI:	43.1
Pedigree_Function:	2.288
Age:	33

Unknown are healthy

predict

Figure 29: Prediction results (2)

We can see that the prediction results are basically accurate, but there are still some wrong judgment data, which may be caused by the input data is too extreme, the classifier can't correctly distinguish.

4.2 Optimization

Although we have trained the model, but in order to get better results, we need to optimize. There are many optimization angles. In this experiment, we mainly use the idea of tuning parameter and ensemble.

4.2.1 Random State

Random_State is a random seed, which is used as a parameter to control the random pattern in any class or function with randomness. When **Random_State** takes a certain value, it determines a rule.

Before optimizing, we need to change random_ The state is fixed, otherwise each time a model is created, it will produce different results, in this experiment, we will set a **Random_Seed** = 2008 as the **Random_State** parameter.

4.2.2 Grid Search

Grid Search is a method of parameter adjustment. Through the exhaustive method, traverse all the listed candidate parameters, and then evaluate the model, and finally get the best result and the corresponding parameter settings. The main disadvantage of this method is that it is time-consuming [28]. The optimal parameters of each model are arranged in Table 18 after automatic adjustment of Grid Search parameters:

Models	Params
KNN	n_neighbors:5
Logistic Regression	C:100
SVM_RBF	C:100,gamma:0.01
SVM_Linear	C:1,gamma:1
Random Forest	max_features:2,n_estimators:100

Table 18: Parameters of Models

After adjusting the parameters of each model, we can see that the scores of most models have improved in Table 19:

Model	ROC_AUC	
	Before Adjusting Parameters	After Adjusting Parameters
KNN	0.948	0.948
Logistic Regression	0.972	0.972
SVM_RBF	0.981	0.971
SVM_Linear	0.973	0.973
Random Forest	0.985	0.986

Table 19: Comparison of accuracy before and after parameter adjustment

We can see that the effect is not very obvious after the grid search parameter adjustment, and even some individual phenomenon of attenuation. This phenomenon may be due to the random function of SVM can 't be changed from **Random_State** control.

4.2.3 Ensemble

The basic concept of integration is to combine the prediction of multiple models and average the specific errors, so as to obtain better overall prediction results. The basic concept of integration is to combine the prediction of multiple models and average the specific errors, so as to obtain better overall prediction results. How to combine the prediction results is the most important part of our optimization. We choose mean ensemble and weighted ensemble.

A. Mean Ensemble

the specific formula of mean ensemble is:

$$e(x) = \frac{1}{n} \sum_{i=1}^n f_i(x)$$

Table 20 and Figure 30 is used for comparing the ensemble model score with the previous model score.

Model	ROC_AUC	
	Before Adjusting Parameters	After Adjusting Parameters
KNN	0.948	0.948
Logistic Regression	0.972	0.972
SVM_RBF	0.981	0.971
SVM_Linear	0.973	0.973
Random Forest	0.985	0.986
Mean Ensemble	0.986	

Table 20: Comparison of accuracy before and after mean ensemble

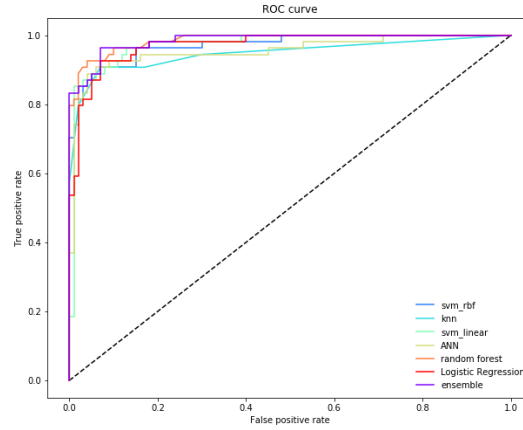


Figure 30: ROC curve of each model

It can be seen that the score of the ensemble model is indeed the highest. However, we need to check whether any model has lowered the score. If so, we need to remove it. So, we have Figure 31 to see the prediction rate of each model.

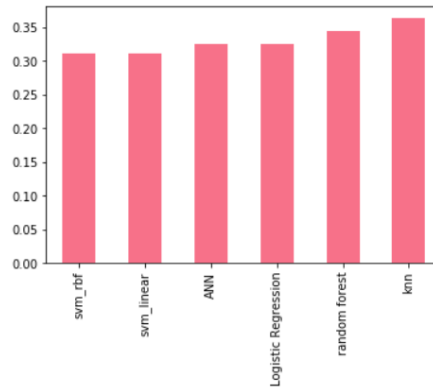


Figure 31: Prediction rate

We can see that each model basically captures the same parts, so we don't need to remove any.

B. Weighted Ensemble

In the weighted ensemble model, each trained $f_i(x)$ assigned a corresponding weight parameter $w_i \in (0,1)$. Generally, a good model has a larger weight, and the sum of each weight should be equal to 1. The specific ensemble formula is as follows:

$$e(x) = \sum_{i=1}^n w_i f_i(x)$$

Once the model generates the prediction $p_i=f_i(x)$, then the learning weight is equivalent to fitting linear regression based on prediction:

$$e(p_1, \dots, p_n) = w_1 p_1 + \dots + w_n p_n$$

Table 21 is a comparison of our weighted ensemble scores with all previous models.

Model	ROC_AUC	
	Before Adjusting Parameters	After Adjusting Parameters
KNN	0.948	0.948
Logistic Regression	0.972	0.972
SVM_RBF	0.981	0.971
SVM_Linear	0.973	0.973
Random Forest	0.985	0.986
Ensemble	0.986	
Weighted Ensemble	0.995	

Table 21:Comparison before and after weighted ensemble

According to the evaluation results of the model, the performance of weighted integration model is the best, so the optimization result is good.

4.3 Optimized application

We import the optimized model into the previous interface. The comparison with the previous results is shown as Table 22 and Figure 32:

Name	Pregnant	Glucose	B P	Skin	Insulin	BMI	DPF	Age	Outcome
Alice	6	148	72	35	0	33.6	0.627	50	1
Rose	1	85	66	29	0	26.6	0.351	31	0
Reimu	8	183	64	0	0	23.3	0.672	32	1
Unknown	0	137	40	35	168	43.1	2.288	33	1

Table 22: Prediction results after optimization

Figure 32: Prediction results after optimization

We can see that the results predicted correctly before have not changed, and those that have not been predicted before are also predicted successfully this time. The optimization results are very good.

4.4 Conclusions

In this project, we successfully used the Machine Learning algorithm knowledge to build six machine learning models, used the known information to predict the diabetes patients successfully, and used the integrated method to significantly improve the model. However, there are still many problems in the process. For example, the data sets used in this experiment are not many, only nearly 700 objects. The training results may be localized and can't be extended to most levels. Moreover, the processing of outliers in this study is relatively rough, which may lead to the incompleteness of the prediction results. This experiment shows that machine learning method is very helpful for the prediction of diabetes mellitus and medical diseases, but each step should be analysed carefully to get a satisfactory result. And even if we can't get a good prediction result, we can also analyse the qualitative relationship between the disease and some features. For example, in this prediction process, I actually analysed the importance of each feature, as shown in Figure 33:

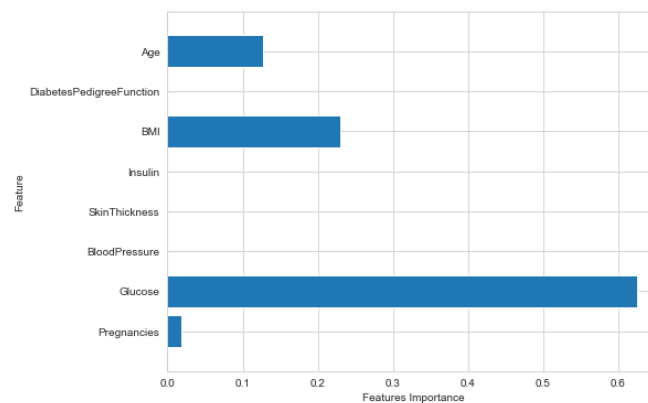


Figure 33: Feature importance

It can be seen that among the data, glucose is the most influential factor, followed by BMI, Age and pregnancies, these are the main, that is, the decisive factors. The others do not mean that they do not affect, but the degree of influence is relatively minor. Through these, we can also analyse which groups of people need to pay attention to the risk of diabetes, and which indicators are the decisive factors of diabetes. These are actually the results obtained in the process of research. These are the results of our research the topics are of great help, although they are not very relevant to our final predictions.

In addition, the evaluation of the prediction interface can be said to be very simple, because it has no front-end basis, and in the process of experiment, it is found that the effect of the trained and optimized model on the verification set is very good. However, when I use the interface to carry out loading and make prediction, the discovery accuracy is reduced. I guess it may be due to the processing of 0 value Not in place, and the data standardization mode is different from the standardized benchmark in previous training.

To sum up, this experiment successfully analysed the experimental data and discussed the processing method, eliminated the interference of null value and abnormal value to the experiment, carried out and discussed the mode of data standardization, completed the data segmentation and random number control. After the preliminary construction of the model, a comprehensive evaluation of the model is carried out. The model is optimized by adjusting parameters and ensemble, and the prediction interface made by myself is carried with the model before and after optimization to verify the remarkable effect of optimization.

In this project, I started from the basic theory of machine learning, based on the achievements achieved by predecessors in related fields, through my own research and exploration, improved and gradually optimized, and finally realized the actual application and completed a leap from theory to practice.

4.5 Future Work

First of all, we need to solve the problem of data set supplement. The number of data sets is too small to make the result convincing enough. On this issue, several medical institutions have expressed clear interest in my project, and they may provide me with relevant help in the future. Secondly, it is the selection and optimization of the model. What we choose in this experiment is more basic and popular, which is relatively simple, time-saving and labour-saving, but this does not mean that this is the best and most appropriate. As for optimization, I learned that k-fold cross-validation will play a better role in the subsequent optimization, but in the experiment, how to combine k-fold cross validation with model ensemble encountered great resistance, so I only stayed in ensemble at last, and I will continue to try to learn more ensemble methods in the future. Finally, and most importantly, the road of machine learning is very long. The knowledge I used in this experiment is only a small part of this huge resource pool. If human beings want to continue to master this powerful tool, we need to continue to work hard in the future.

References

- [1] S. M. JACOB, K. RAIMOND and D. KANMANI, "Associated Machine Learning Techniques based On Diabetes Based Predictions," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 1445-1450, doi: 10.1109/ICCS45141.2019.9065411
- [2] Nongyao Nai-arun, Rungruttikarn Moungrmai, "Comparison of Classifiers for the Risk of Diabetes Prediction", *Procedia Computer Science* 69, pg 132 – 142, 2015
- [3] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, Nicos Maglaveras, Ioannis Vlahavas, Ioanna Chouvarda, "Machine Learning and Data Mining Methods in Diabetes Research", *Computational and Structural Biotechnology Journal* 15, pg 104–116, 2017
- [4] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective.", *Artificial Intelligence in medicine*, Aug. 2001, 23(1), pp.89-109.
- [5] T.G.Dietterich, J.W.Shavlik (eds.) *Readings in machine learning*, Morgan Kaufmann, 1990.
- [6] Huiying Li, Dechang Li, Changhai Zhang and Shubin Nie, "An Application of Machine Learning in the Criterion Updating of Diagnosis Cancer," *2005 International Conference on Neural Networks and Brain*, Beijing, 2005, pp. 187-190, doi: 10.1109/ICNNB.2005.1614594.
- [7] Byrne, C,"Development workflows for Data Scientists", O'Reilly,2017, pp.28.
- [8] Dineva, Kristina, and Tatiana Atanasova. "OSEMN process for working over data acquired by IoT devices mounted in beehives." *Current Trends in Natural Sciences Vol 7.13* (2018): 47-53.
- [9] Kantardzic, Mehmed. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.
- [10] Borcard, Daniel, François Gillet, and Pierre Legendre. *Numerical ecology with R*. Springer, 2018.
- [11] Hackeling, Gavin. *Mastering Machine Learning with scikit-learn*. Packt Publishing Ltd, 2017.
- [12] J. Huang, Y. Wei, J. Yi and M. Liu, "An Improved kNN Based on Class Contribution and Feature Weighting," *2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, Changsha, 2018, pp. 313-316, doi: 10.1109/ICMTMA.2018.00083.
- [13] CoW. Hsu, C-C Chang, and C-I. Lin, "A Practical Guide to Support Vector Classification," Taipei, Apr. 2010.
- [14] B. Sanjaa and E. Chuluun, "Malware detection using linear SVM," *Ifost*, Ulaanbaatar, 2013, pp. 136-138, doi: 10.1109/IFOST.2013.6616872.

- [15] Y. Wang, Y. Ou, X. Deng, L. Zhao and C. Zhang, "The Ship Collision Accidents Based on Logistic Regression and Big Data," *2019 Chinese Control And Decision Conference (CCDC)*, Nanchang, China, 2019, pp. 4438-4440, doi: 10.1109/CCDC.2019.8832686.
- [16] X.M Ding, L.H. Zhang, Y.D. Zhao, X.J. Zhang, G.Z. Bao, J.C. Ma. Application of Logistic regression analysis in evaluation of experimental teaching effect and Its Realization on SPSS 19.0[J].*Heilongjiang Animal Husbandry and Veterinary Medicine*, 2017(9): 261-265.
- [17] T.K. Ho, "random decision forests,"*J. Economic Problem* ,vol.8,pp.278-282,1995.
- [18] H. Lan and Y. Pan, "A Crowdsourcing Quality Prediction Model Based on Random Forests," *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, Beijing, China, 2019, pp. 315-319, doi: 10.1109/ICIS46139.2019.8940306.
- [19] A. Muhtar, I. W. Mustika and Suharyanto, "The comparison of ANN-BP and ANN-PSO as learning algorithm to track MPP in PVSsystem," *2017 7th International Annual Engineering Seminar (InAES)*, Yogyakarta, 2017, pp. 1-6, doi: 10.1109/INAES.2017.8068573.
- [20] D. E. Zomahoun, "A Semantic Collaborative Clustering Approach Based on Confusion Matrix," *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Sorrento, Italy, 2019, pp. 688-692, doi: 10.1109/SITIS.2019.00112.
- [21] S. S. Khalid and S. Abrar, "Area under the ROC Curve of Enhanced Energy Detector," *2013 11th International Conference on Frontiers of Information Technology*, Islamabad, 2013, pp. 131-135, doi: 10.1109/FIT.2013.31.
- [22] J. Si, S. L. Harris and E. Yfantis, "A Dynamic ReLU on Neural Network," *2018 IEEE 13th Dallas Circuits and Systems Conference (DCAS)*, Dallas, TX, 2018, pp. 1-6, doi: 10.1109/DCAS.2018.8620116.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012.
- [24] Y. Wu, J. Li, Y. Kong, and Y. Fu, "Deep convolutional neural network with independent softmax for large scale face recognition," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1063–1067.
- [25] K. Sanni, G. Garreau, J. L. Molin, and A. G. Andreou, "Fpga implementation of a deep belief network architecture for character recognition using stochastic computation," in *Information Sciences and Systems (CISS)*, 2015 49th Annual Conference on. IEEE, 2015, pp. 1–5.
- [26] Tariq Mehmood, Harsh Mishra,kaggles,' PIMA India Diabetes Prediction with 6 Algorithms',<https://www.kaggle.com/tariqmhmd5/pima-india-diabetes-prediction-with-6-algorithms/comments>

- [27] L. Zhang, H. Huang and X. Jing, "A modified cyclostationary spectrum sensing based on softmax regression model," *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*, Qingdao, 2016, pp. 620-623, doi: 10.1109/ISCIT.2016.7751707.
- [28] Qiujun Huang, Jingli Mao and Yong Liu, "An improved grid search algorithm of SVR parameters optimization," *2012 IEEE 14th International Conference on Communication Technology*, Chengdu, 2012, pp. 1022-1026, doi: 10.1109/ICCT.2012.6511415.

Appendix

Appendix A: GitLab Repository

Description:

In my GitLab, you'll see a folder called 'data' and a file called 'Diabetes_Predict.ipynb'. If you open the 'data' folder, you will see the dataset I used in this project, called 'diabetes 1.csv'. 'Diabetes_Predict.ipynb' file contains all the code used in this project.

The **code reuse statement** is also in the file.

SSH: <git@git-teaching.cs.bham.ac.uk:mod-msc-proj-2019/jxp924.git>

HTTP: <https://git-teaching.cs.bham.ac.uk/mod-msc-proj-2019/jxp924.git>

How to run the code:

Please run under anaconda3 Python version 3.7, where additional packages need to be installed may but not be limited to **keras**, **TensorFlow**, **iploty**, **mlens**, et

After that, the code block can be executed in turn. The input method of the prediction interface is to input the name and its specific indicators, and then click the predict button to get the result.

Appendix B: Code and Thought Reference Statement

In the process of studying this topic, I can say that I "stand on the shoulders of giants to see the world". Without the experience and ideas provided by these pioneers, I will not be able to complete my work. Next, I will briefly state the source of the code I reused and ideas I quoted to separate what they have done and what I have done on their basis.

The source of reference inspiration for the workflow ideas of the model is :

Nida Guler <https://www.kaggle.com/nidaguler/eda-and-prediction-indians-diabetes/notebook> :NO Preprocessing, No normalization

Vincent Lugat <https://www.kaggle.com/vincentlugat/pima-indians-diabetes-eda-prediction-0-906/notebook>: Median for zeros,Median for outliers,5-Folds Cross Validation

Jason Li <https://www.kaggle.com/dbsnail/diabetes-prediction-over-0-86-accuracy/data>: Mean for zeros,But no outliers' processing,train:test=7:3,randomstate controlled

Chirag Samal <https://www.kaggle.com/chirag9073/diabetes-using-deep-learning>: Drop zero,But no outliers' processing,train:test=8:2,

Riddhi Mehta <https://www.kaggle.com/rnmehta5/pima-indian-diabetes-binary-classification> :zero and outliers' processing are referred

Tariq Mehmood and Harsh Mishra <https://www.kaggle.com/tariqmehmd5/pima-india-diabetes-prediction-with-6-algorithms/comments>: Median for zeros

My work: The combination of the above framework, Median for zero, ,Median for outliers, Z-Score Normalization, train:test=8:2,random_state controlled, 5 models, gridsearch to optimize parameters, two ensemble functions, application to set the trained model. And for each step, the reasons and analysis for selecting this method are given in the report. It is worth mentioning that I am very much in favor of the learning framework constructed by Tariq Mehmood and harsh Mishra, but there are some problems that I have modified: 1. They ignored the outliers of insulin; 2. Some outliers have not been cleared completely, which have been improved in my process.

The source of reference inspiration for the ensemble ideas of the model is :

Fares Sayah <https://www.kaggle.com/faressayah/ensemble-ml-algorithms-bagging-boosting-voting/notebook#5.-Voting-Ensemble>

His work: Bagging, Boosting, Voting ensemble algorithms

My work: The method of determining the GBM to determine the weight of each model

Code Reuse:

1.code block[2]: Tariq Mehmood and Harsh Mishra <https://www.kaggle.com/tariqmhmd5/pima-india-diabetes-prediction-with-6-algorithms/comments> in block[8]

Reason: Reference to obtain the median of each method, simplify the code.

2.code block[7]:Vincent Lugat <https://www.kaggle.com/vincentlugat/pima-indians-diabetes-eda-prediction-0-906/notebook> in block [86]

Reason: It is considered that the interface given by him is more intuitive, beautiful and clear. I made some changes to his code

3.code block[74]:Tariq Mehmood and Harsh Mishra <https://www.kaggle.com/tariqmhmd5/pima-india-diabetes-prediction-with-6-algorithms/comments> in block[73].

Reason: I don't know how to choose the layer number of ANN model, so we decide to select the model that has been built and then fine tune it