香港中文大學
The Chinese University of Hong Kong

Multi-Disease Prediction Model Based on Deep Learning and U.S. Healthcare Data

Group 2
1155191644 Qi Zihan
1155215119 Zhao Chenran
1155215100 Jiang Yanze

Abstract

Non-communicable diseases (NCDs), such as angina, overweight, diabetes, and hypertension, place a heavy burden on global health and contribute to a significant portion of global mortality. This project developed a multi-disease prediction model based on US healthcare data (NHANES) designed to predict an individual's risk of developing multiple diseases . The model employs machine learning techniques including K-Nearest Neighbours (KNN), Random Forests (RF), Support Vector Machines (SVM) as well as deep learning and AutoML methods. The performance of these models is evaluated by various metrics and F1 scores. Additionally, to better demonstrate the prediction results, we also built a small web application based on the deep learning model using flask that has an average accuracy of roughly 78%. We believe that this model can be further upgraded in the open-source community and subsequent research to contribute to early disease prevention, precision medicine, and improved public health.

Keywords: Non-communicable diseases, multi-disease prediction, deep learning, machine learning, U.S. healthcare data, disease prevention, precision medicine, public health.

Contents

1 INTRODUCTION

Chronic non-communicable diseases (NCDs), often referred to as ”lifestyle diseases”, are medical conditions that tend to be long-lasting and typically progress slowly over time, which include angina, overweight, diabetes, and high blood pressure and so on (figure 1) (1). In recent years, non-communicable diseases (NCDs) have globally shown increasing impact on health status in the populations in the whole world (2), accounting for almost 60 % of global mortality (3). 17.9 million people died of high blood pressure in 2019 while angina were responsible for 6 % of global death (4). 463 million adults aged 20-79 years have diabetes globally in 2019 while more than 1.9 billion adults were overweight globally in 2016, among which 650 million met the criteria for obesity (4; 5). The data above all showcase the urgency and significance of solving this issue.

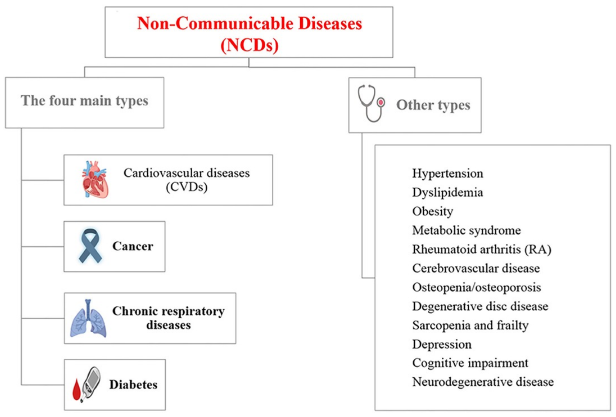


FIGURE 1: LIST OF NON-COMMUNICABLE DISEASES (NCDS)

Deep learning has shown great potential in the medical field, especially in predicting disease (6; 7; 8; 9). With the application of Deep learning, it is possible to overcome the limitations of the traditional single-disease prediction model and provides a more comprehensive assessment of an individual’s health status, beneficial for controlling chronic non-communicable diseases (NCDs) (10). Machine learning also performs well in structured data environments due to the clear organization and format of the data (11). Moreover, AutoML, a method that uses machine learning techniques to automate and simplify the construction and deployment of machine learning models,can save time and increase productivity, thus accelerating the development cycle of machine learning models (12). Therefore, our project aims to develop a multi-disease prediction model based on deep learning, Machine learning and AutoML method, which can utilize medical data (NHANES) to simultaneously predict the risk of an individual suffering from multiple diseases, thus realizing early prevention, precision medicine and improving public health. Our team also develop a web application using Flask to better showcase the predictive results and serve for more people.

2 DATA PREPROCESSING

We found U.S. Healthcare Data from kaggle: NHANES data for 06-14. But unfortunately there is no data description in there for the diseases associated with our predictions, and we will predict that the diseases are sampling data from 2015. So we decided to use python crawling technique to get the public data (13).

2.1 PYTHON CRAWLING TECHNIQUE

In this study, a simple crawling technique was employed. First the different categories of questions in the data on the four web pages were stored in a list. Then since all of the trivia questions including the description with the data source were stored in a table with id GridView1. Using a simple for loop to crawl one by one and download the question descriptions and data sources to the specified folder respectively, all the required data was obtained.

Web Scraping Declaration

Our use of web scraping technology to access Open Data is primarily aimed at expediting the data retrieval process. Manual downloading of data can be slow, whereas web scraping allows for the automated extraction of data from open data sources, resulting in the acquisition of large amounts of data within a shorter timeframe. We ensure compliance with relevant laws and regulations during the data scraping process, respect the rights of data providers, and commit to avoiding any adverse impact or infringement on data sources, including privacy concerns.

2.2 DATA FORMAT HARMONISATION AND MERGING

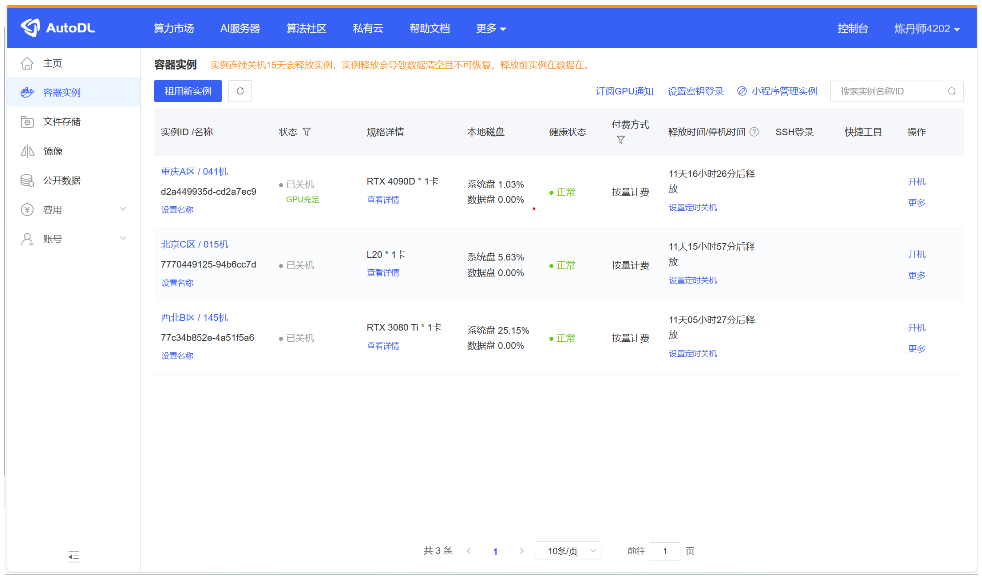
Convert each data csv to a uniform format ('utf-8') using pandas in python. Then the first row's id (SEQN) was used as the unique id for all data to be merged. Unfortunately, due to poor computer performance, the consolidation process lasted more than half an hour and resulted in 3.53G of data.

2.3 LARGE DATA FILE PROCESSING

When trying to use pandas to read the processed data, we encountered obstacles due to the large amount of data. It took more than 10 minutes just to read the data, which is very inefficient. Based on the SQL big data processing methods learnt in class, we converted the data to .bd format and tried to use mysql as a tool to pre-process the data. However, given the limitations of SQL syntax and performance issues, this method did not perform well in practice. Therefore, we turned to cloud services to deal with this problem.

2.3.1 FILTER OUT UNNECESSARY COLUMNS

Given that the data we have obtained by combining the row data set obtained previously has many vacancy value which are not good for analyzing, we need to delete the columns that are with especially high vacancy rate (here we take 50 percent as the standard). However, since the file is too big to deal with, our computer failed to do it. To over come the problem, we rent some cloud server from www.autodl.com (as the following picture shows).



容器实例 实例连续关机15天会释放实例，实例释放会导致数据清空且不可恢复，释放前实例在数据在。

实例ID /名称	状态	规格详情	本地磁盘	健康状态	付费方式	释放时间/停机时间	SSH登录	快捷工具	操作
重庆A区 / 041机 d2a449935d-cd2a7ec9 设置名称	● 已关机 GPU充足	RTX 4090D * 1卡 查看详情	系统盘 1.03% 数据盘 0.00%	● 正常	按量计费	11天16小时26分后释放 设置定时关机			开机 更多
北京C区 / 015机 7770449125-94b6cc7d 设置名称	● 已关机	L20 * 1卡 查看详情	系统盘 5.63% 数据盘 0.00%	● 正常	按量计费	11天15小时57分后释放 设置定时关机			开机 更多
西北B区 / 145机 77c34b852e-4a51f5a6 设置名称	● 已关机	RTX 3080 Ti * 1卡 查看详情	系统盘 25.15% 数据盘 0.00%	● 正常	按量计费	11天05小时27分后释放 设置定时关机			开机 更多

共 3 条 ‹ 1 › 10条/页 前往 1 页

FIGURE 2: CLOUD SERVER FROM AUTODL

2.3.2 FILTER OUT UNNECESSARY ROWS

After filting the columns, we find out that for the new dataset ”Cleaned File”, there are nearly 10000 training sample, among them nearly 2000 missed so many features which are quite hard to use, so we deleted them.

Apart from this, in the later part for model building (like Deep Learning and Machine Learning), to make sure the performance of the prediction model, we conduct data processing secondly (delete some unnecessary data, which you can see in our code).

3 FEATURE ENGINEERING AND DATA VISUALIZATION

3.1 DEFINITION OF FEATURE ENGINEERING

Feature engineering is the process of selecting, extracting, transforming, and creating features from raw data to enhance machine learning model performance and accuracy, the target of which is to optimize the representation of data for machine learning algorithms, leading to improved model interpretability and predictive power (14).

3.2 ANALYSIS OF TARGET VARIABLES

After data preprocessing, we found an imbalance in the distribution of angina and diabetes data through data analysis of the target variables and observation of the visualized picture.

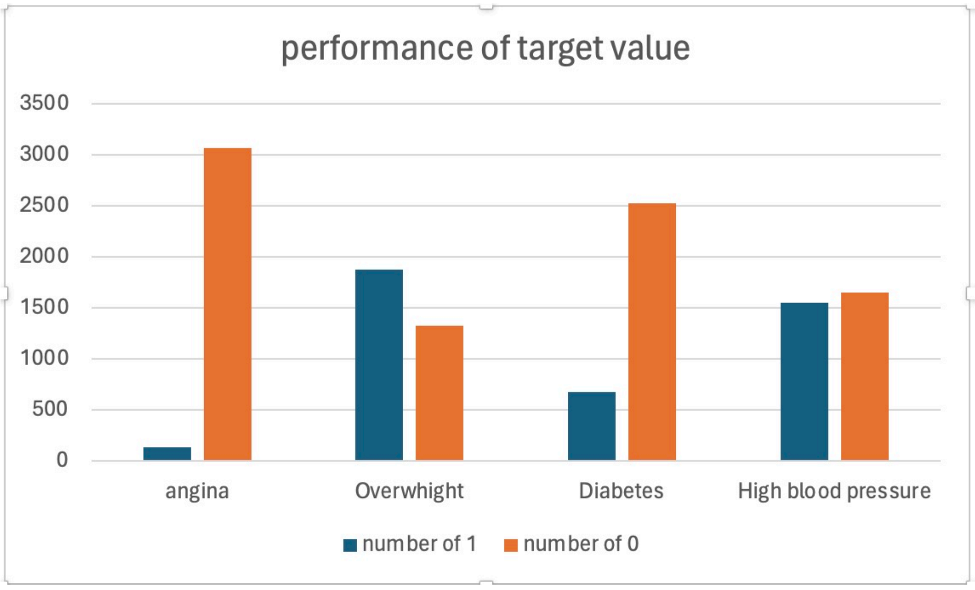


FIGURE 3: PERFORMANCE OF TARGET VARIABLES

3.3 ANALYSIS OF VARIABLE RELATIONSHIPS

3.3.1 LINE PLOT

Next, we looked for a random set of independent and dependent variables for data visualization to find the relationship between the two variables using line plot. By observing the visualization graphs, it was found that there was no relationship between the two variables (a random feature with overweight; one random feature with blood high pressure).

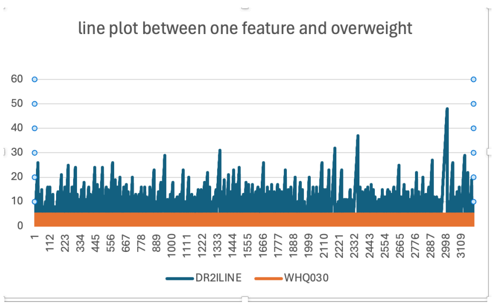


FIGURE 4: LINE PLOT OF OVERWEIGHT WITH ONE RANDOM FEATURE

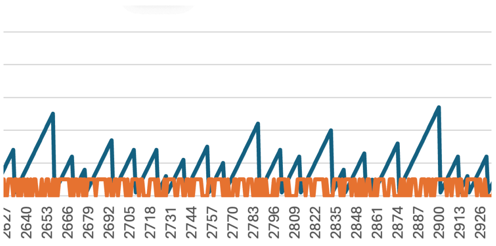


FIGURE 5: DETAIL$_{OVERWEIGHT}$

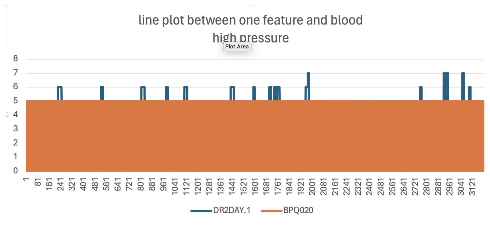


FIGURE 6: LINE PLOT OF BLOOD HIGH PRESSURE WITH ONE RANDOM FEATURE

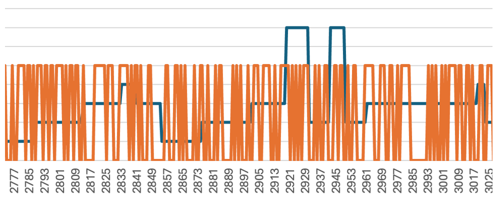


FIGURE 7: DETIAL$_{BLOODHIGHPRESSURE}$

3.3.2 CORR HEATMAP

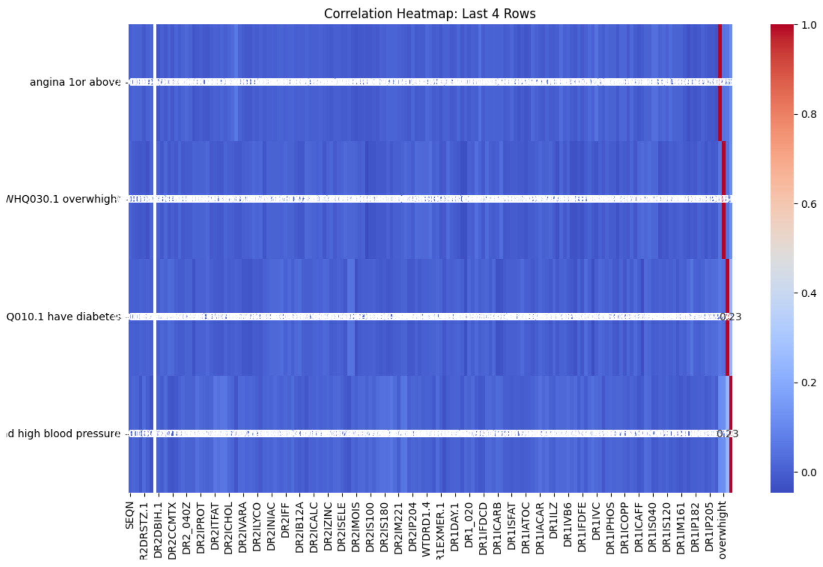


FIGURE 8: CORR HEATMAP

After that, we conducted a correlation analysis of all these four diseases of data. However, it turned out that they only had a strong correlation with themselves, without close connections with other variables. Thus, we want to find the hidden pattern of the variables and the diseases through Machine Learning, Deep Learning and AutoML.

4 MACHINE LEARNING

For this part, we are going to build 3 different models for each disease, i.e. K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM).

4.1 BRIEF INTRODUCTION

4.1.1 INTRODUCTION ON MODELS

K-Nearest Neighbors (KNN)

Principle

For a given test sample, calculate its distance from each training sample (usually using Euclidean distance). Then select a certain number (denoted as K) of training samples with the most smallest distance. Then count the number of categories in these k nearest neighbors. Finally, the test samples are classified into the category with the largest number of k nearest neighbors, that is, the prediction class of the test samples is obtained (15).

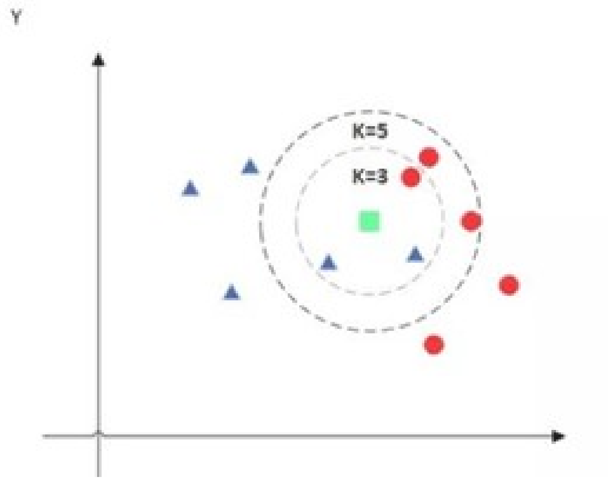


FIGURE 9: KNN

$$d(x,y) := \sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\cdots+(x_n-y_n)^2} = \sqrt{\sum_{i=1}^{n}(x_i-y_i)^2}.$$

FIGURE 10: EUCLIDEAN DISTANCE

Pros and Cons

The KNN model is less sensitive to unusual data and can effectively handle non-linear data. However, it requires a lot of storage space to store the training dataset. The classification accuracy of high-dimensional data can be low. For data with large value ranges, normalization is highly required, which makes it much more complex to implement (15; 16).

Method

We will reprocess the data before analyzing it with the KNN model. Samples with incomplete data will be removed, and different features will be normalized to better train the model. To increase the accuracy of the KNN model, a grid search is used, that is, within a certain range of values, to select a K value that makes the model most accurate (see Principle part for the explanation of the K value. In order to test and prevent overfitting, we will use the cross-validation method to evaluate the model and combine it with the confusion matrix to evaluate the model in more detail.

Random Forest (RF)

Principle

Based on decision trees, Random Forest randomly extracts multiple sample sets from the original data, and each sample set is trained to obtain a unique decision tree. (Since the samples are different, the decision trees are different.) When new data is to be processed, each tree gives a prediction. Based on the different trees, we can finally get the predicted class by voting or averaging (17; 18).

In the random forest part, we will adjust the performance of the model by adjusting the

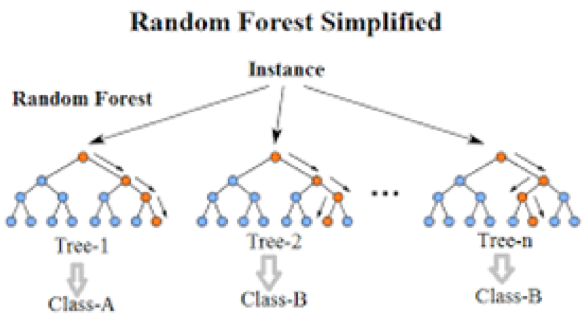


FIGURE 11: RF MODEL

three parameters i.e.'max depth', 'min samples split', and 'n estimators'. 'max depth' is the maximum depth limit of the decision tree, which controls the maximum number of layers that can be formed by the decision tree. Limiting the depth of the decision tree can alleviate the overfitting phenomenon but also decrease the accuracy of the model. 'min samples split' is the minimum number of samples required to perform node splitting, which controls the conditions for decision tree splitting. Increasing it can reduce the likelihood of overfitting, but also reduces the performance of the model. 'n estimators' is the number of decision trees in the random forest. Increasing the number of decision trees will increase the computational cost of random forests, but it can also improve the performance of the model to a certain extent.

Pros and Cons

Random Forests can have high accuracy, are not easy to overfit, are good at processing high-dimensional data, and run fast (17). However, Random Forests require a lot of decision trees, and when the dataset is large, it has high memory requirements and is very sensitive to small changes in the training data, which may affect accuracy (18; 19).

Method

Before analyzing with a random forest model, we will reprocess the data and delete incomplete samples. To increase the accuracy of the RF model, a grid search is used to find the 'max depth', 'min samples split' and 'n estimators' that make the most accurate (see Principle part for a detailed explanation of these three parameters). To test and prevent overfitting, we will use the cross-validation method and confusion matrix to evaluate the mode.

Support Vector Machine (SVM)

Principle

SVM is based on the principle of structural risk minimization in statistical learning theory, that is, through mathematical calculations, to find a beat hyperplane in the feature space where different data samples are separated so that the distance from each type of sample to the hyperplane is maximized (20; 21).

Pros and Cons

SVM can use different kernel functions to map the data to the high-dimensional space, which

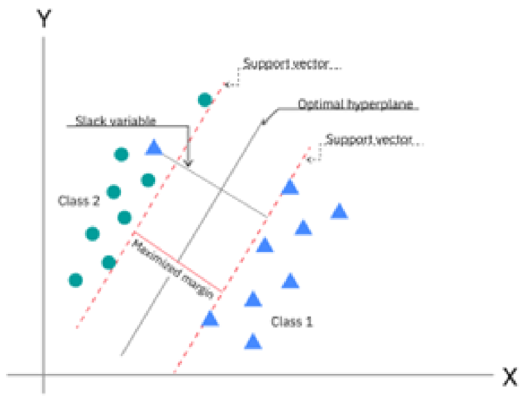


FIGURE 12: SVM MODEL

is conducive to solving the nonlinear classification problem. At the same time, the final model of SVM is determined by a small number of vectors, avoiding confusion caused by too many dimensions. However, it takes a lot for SV to train for large-scale samples and it is quite sensitive to data loss (20; 21).

Method

Before training with the SVM model, we will reprocess the data and delete incomplete samples. At the same time, considering the imbalance of the data, we will use category weights to improve this phenomenon. To test and prevent overfitting, we used a cross-validation method and confusion matrix to evaluate the model.

4.1.2 INTRODUCTION ON EVALUATION METHODS

Training Accuracy and Testing Accuracy

Training Accuracy indicates the accuracy of the model on the training data, that is, the classification accuracy of the model for training samples with known labels. Testing Accuracy indicates the accuracy of the model on test data, that is, the classification accuracy of the model for test samples with unknown labels.

Confusion Matrix

We will show two Confusion Matrix for each model, i.e. Training Confusion Matrix and Testing Confusion Matrix. Confusion Matrix is used to evaluate the performance of classification models based on the predicted result of the samples of different categories.

Classification Report

Classification Report provides more detailed model performance metrics, including precision, recall, and F1 values. In order to better evaluate the model performance, we will show 2 reports for each model, i.e. Testing Confusion Matrix and Training Classification Report. ”Precision” is the proportion of correctly predicted positive samples out of all samples predicted as positive by the model, while ”Recall” means the proportion of correctly predicted positive samples out of all actual positive samples. Meanwhile, ”F1” gives us a balanced measure of the model’s performance by considering both precision and recall (We will give a more detailed explanation of Precision, Recall and F1 in the deep learning part 4.4).

Precision, Recall, and F1 are calculated using the following formula:

$$Precision = (TruePositive)/(TruePositive + FalsePositive)$$

$$Recall = (TruePositive)/(TruePositive + FalseNegative)$$

$$F1 = 2 * (Precision * Recall)/(Precision + Recall)$$

4.2 ANALYSIS OF ANGINA

KNN

After the KNN model analysis, we get the optimal K which is 4, and the accuracy of the corresponding model is 0.958 in the training set (reserved to 3 decimal places, the same below) and 0.951 in the test set. This seems like a good model. However, through the confusion matrix, it is found that the high accuracy of this model is due to the high imbalance of the analysis data (too many samples belonging to the label “0” and fewer other classes). That is, the model predicts that the samples belonging to label “1” and label “2” are labeled “0”. In addition, according to the Classification Report, it can be seen that the model is accurate in classifying label “0”, but the prediction accuracy for label “1” and label “2” is extremely low. Therefore, even with a very high accuracy rate, the model is not ideal.

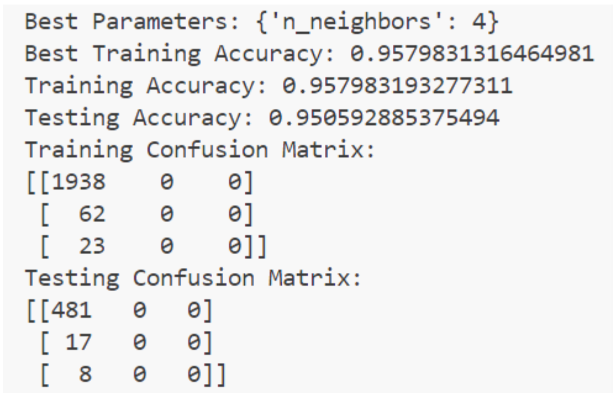
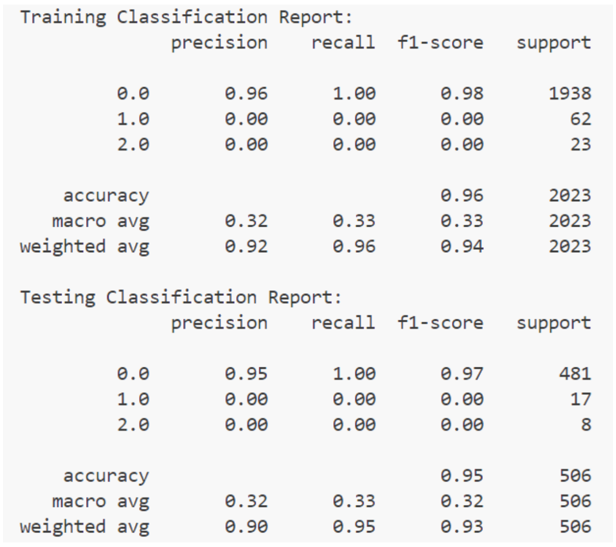

```
Best Parameters: {'n_neighbors': 4}
Best Training Accuracy: 0.9579831316464981
Training Accuracy: 0.957983193277311
Testing Accuracy: 0.950592885375494
Training Confusion Matrix:
[[1938    0    0]
 [  62    0    0]
 [  23    0    0]]
Testing Confusion Matrix:
[[481   0   0]
 [ 17   0   0]
 [  8   0   0]]
```

FIGURE 13: 3.2 ACCURACY AND CONFUSION MATRIX FOR KNN MODEL

RF

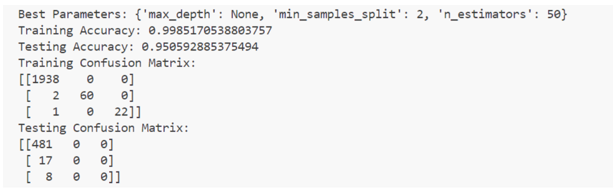
After the RF model analysis, the optimal parameter combination is ’max depth’: None, ’min samples split’: 2, ’n estimators’: 50, and the accuracy of the training set and the test set are 0.999 and 0.951 respectively. However, considering the imbalance of the data, we found that the three



```
Training Classification Report:
              precision    recall  f1-score   support

         0.0       0.96      1.00      0.98      1938
         1.0       0.00      0.00      0.00        62
         2.0       0.00      0.00      0.00        23

    accuracy                           0.96      2023
   macro avg       0.32      0.33      0.33      2023
weighted avg       0.92      0.96      0.94      2023

Testing Classification Report:
              precision    recall  f1-score   support

         0.0       0.95      1.00      0.97       481
         1.0       0.00      0.00      0.00        17
         2.0       0.00      0.00      0.00         8

    accuracy                           0.95       506
   macro avg       0.32      0.33      0.32       506
weighted avg       0.90      0.95      0.93       506
```

FIGURE 14: 3.2 CLASSIFICATION REPORT FOR KNN MODEL

labels in the training set had more accurate predictions, while the test set still failed to predict labels ”1” and ”2”, so there seems to be an overfitting. According to the classification report, the random forest model is better than the KNN model, but it is still not ideal.



```
Best Parameters: {'max_depth': None, 'min_samples_split': 2, 'n_estimators': 50}
Training Accuracy: 0.9985170538803757
Testing Accuracy: 0.950592885375494
Training Confusion Matrix:
[[1938    0    0]
 [   2   60    0]
 [   1    0   22]]
Testing Confusion Matrix:
[[481   0   0]
 [ 17   0   0]
 [  8   0   0]]
```

FIGURE 15: 3.2 ACCURACY AND CONFUSION MATRIX FOR RF MODEL

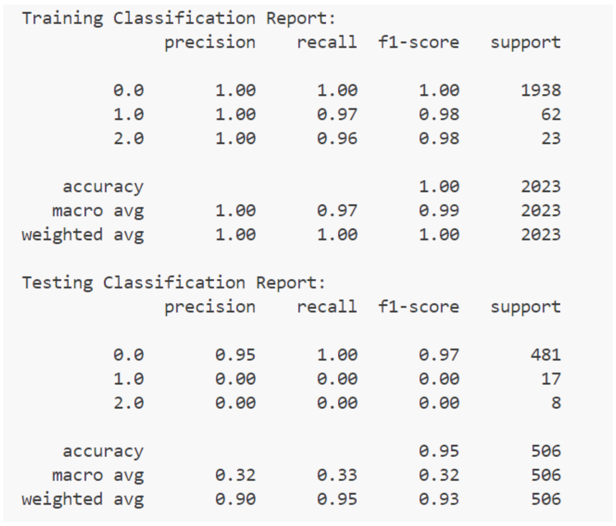
SVM

After SVM model training, the accuracy of the training set is 0.800, and the accuracy of the test set is 0.702. Although the accuracy decreased, we found that the SVM model improved the accuracy of the prediction of the Label ”1” and Label ”2” samples through confusion. Judging from the classification report, although the accuracy of the SVM is lower than that of both the KNN and RF models, the performance in the test group has been greatly improved.

4.3 ANALYSIS FOR OVERWEIGHT

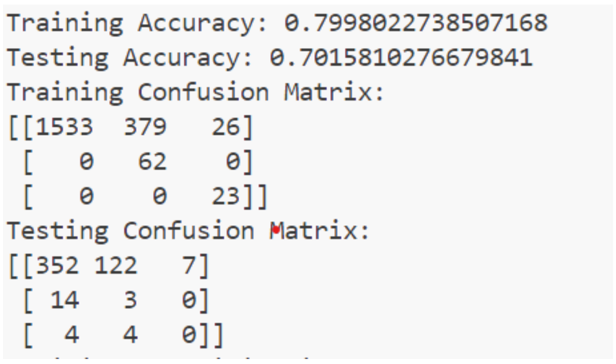
KNN

After the KNN model analysis, we get the optimal K value of 14, and the accuracy of the corresponding model is 0.589 in the training set (to 3 decimal places, the same below) and 0.543 in the test set. Through the confusion matrix, it is found that the model can deal with two types of labels, but cannot predict the other three types of samples (because the number of these three



```
Training Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      1938
         1.0       1.00      0.97      0.98        62
         2.0       1.00      0.96      0.98        23

    accuracy                           1.00      2023
   macro avg       1.00      0.97      0.99      2023
weighted avg       1.00      1.00      1.00      2023

Testing Classification Report:
              precision    recall  f1-score   support

         0.0       0.95      1.00      0.97       481
         1.0       0.00      0.00      0.00        17
         2.0       0.00      0.00      0.00         8

    accuracy                           0.95       506
   macro avg       0.32      0.33      0.32       506
weighted avg       0.90      0.95      0.93       506
```

FIGURE 16: 3.2 CLASSIFICATION REPORT FOR RF MODEL



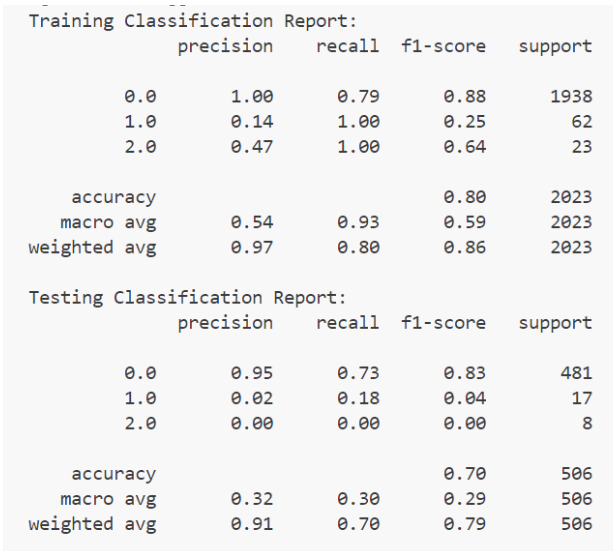
```
Training Accuracy: 0.7998022738507168
Testing Accuracy: 0.7015810276679841
Training Confusion Matrix:
[[1533  379   26]
 [   0   62    0]
 [   0    0   23]]
Testing Confusion Matrix:
[[352 122   7]
 [ 14   3   0]
 [  4   4   0]]
```

FIGURE 17: 3.2 ACCURACY AND CONFUSION MATRIX FOR SVM MODEL

types of samples is too small). In addition, according to the classification report, it can be seen that the classification of labels “1” and “3” can be basically completed, but the accuracy is low, and the other labels cannot be predicted, so the model is not very ideal.

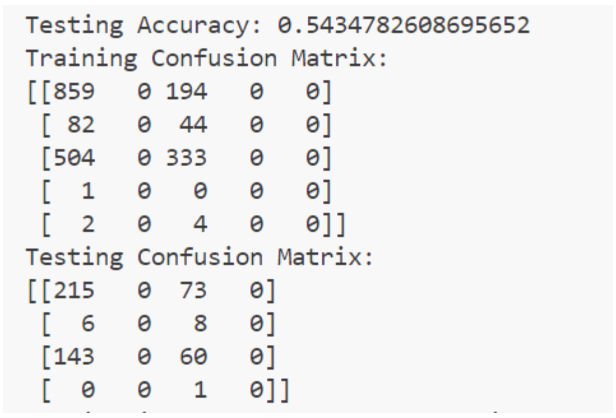
RF

After the RF model analysis, the best parameter combination is ’max depth’: 5, ’min samples split’: 5, ’n estimators’: 100, and the accuracy of the training set is 0.615 and the test set is 0.565. Considering the imbalance of the data, we found that the prediction of label “1” was more accurate in the training set and less accurate for the other samples. In addition, according to the classification report, it can be seen that the classification of labels “1” and “3” can be basically completed, but the accuracy is low, and it cannot be predicted for other labels, so the model does not have many advantages compared with KNN.



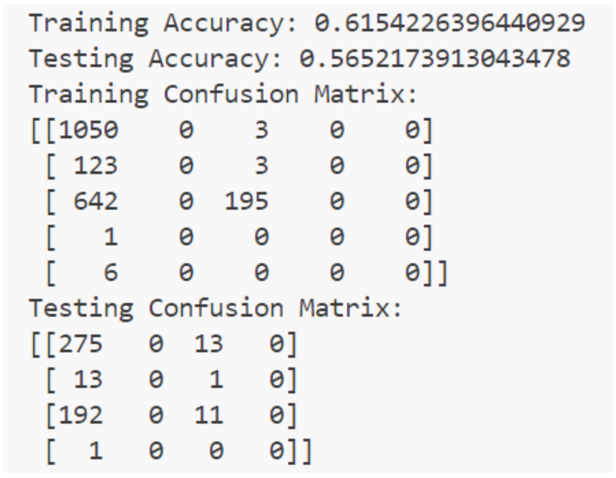
```
Training Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.79      0.88      1938
         1.0       0.14      1.00      0.25        62
         2.0       0.47      1.00      0.64        23

    accuracy                           0.80      2023
   macro avg       0.54      0.93      0.59      2023
weighted avg       0.97      0.80      0.86      2023

Testing Classification Report:
              precision    recall  f1-score   support

         0.0       0.95      0.73      0.83       481
         1.0       0.02      0.18      0.04        17
         2.0       0.00      0.00      0.00         8

    accuracy                           0.70       506
   macro avg       0.32      0.30      0.29       506
weighted avg       0.91      0.70      0.79       506
```

FIGURE 18: 3.2 CLASSIFICATION REPORT FOR SVM MODEL



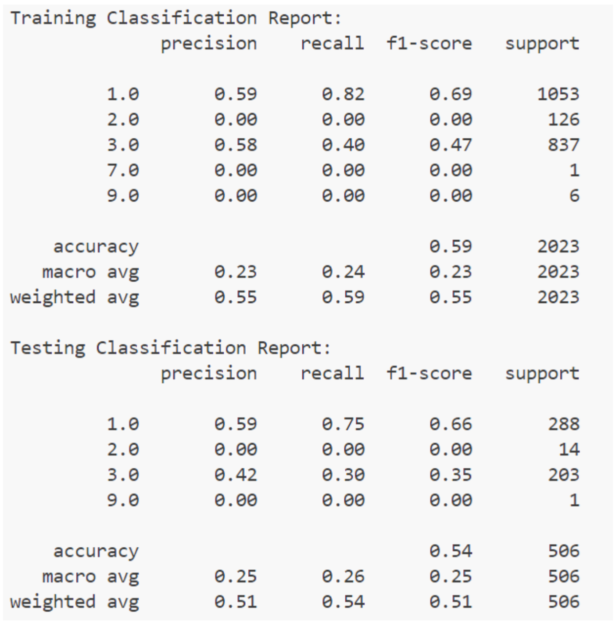
```
Testing Accuracy: 0.5434782608695652
Training Confusion Matrix:
[[859   0 194   0   0]
 [ 82   0  44   0   0]
 [504   0 333   0   0]
 [  1   0   0   0   0]
 [  2   0   4   0   0]]
Testing Confusion Matrix:
[[215   0  73   0]
 [  6   0   8   0]
 [143   0  60   0]
 [  0   0   1   0]]
```

FIGURE 19: 3.3 ACCURACY AND CONFUSION MATRIX FOR KNN MODEL



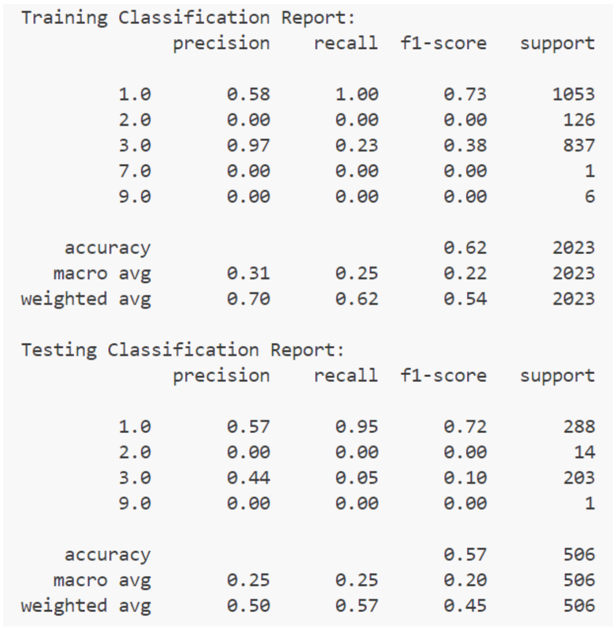
```
Training Accuracy: 0.6154226396440929
Testing Accuracy: 0.5652173913043478
Training Confusion Matrix:
[[1050    0    3    0    0]
 [ 123    0    3    0    0]
 [ 642    0  195    0    0]
 [   1    0    0    0    0]
 [   6    0    0    0    0]]
Testing Confusion Matrix:
[[275   0  13   0]
 [ 13   0   1   0]
 [192   0  11   0]
 [  1   0   0   0]]
```

FIGURE 21: 3.3 ACCURACY AND CONFUSION MATRIX FOR RF MODEL



```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.59      0.82      0.69      1053
         2.0       0.00      0.00      0.00       126
         3.0       0.58      0.40      0.47       837
         7.0       0.00      0.00      0.00         1
         9.0       0.00      0.00      0.00         6

    accuracy                           0.59      2023
   macro avg       0.23      0.24      0.23      2023
weighted avg       0.55      0.59      0.55      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.59      0.75      0.66       288
         2.0       0.00      0.00      0.00        14
         3.0       0.42      0.30      0.35       203
         9.0       0.00      0.00      0.00         1

    accuracy                           0.54       506
   macro avg       0.25      0.26      0.25       506
weighted avg       0.51      0.54      0.51       506
```

FIGURE 20: 3.3 CLASSIFICATION REPORT FOR KNN MODEL

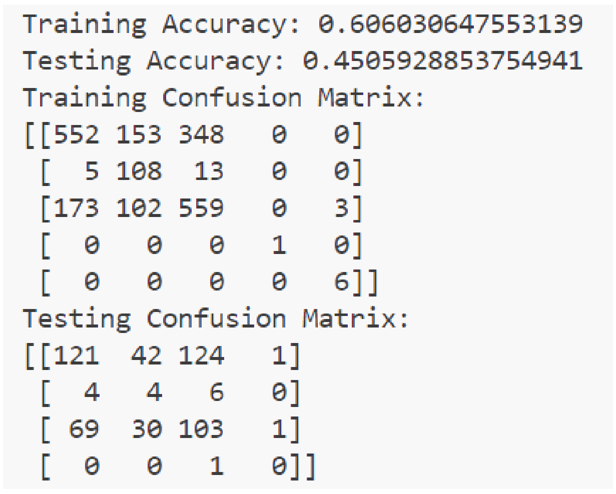


```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.58      1.00      0.73      1053
         2.0       0.00      0.00      0.00       126
         3.0       0.97      0.23      0.38       837
         7.0       0.00      0.00      0.00         1
         9.0       0.00      0.00      0.00         6

    accuracy                           0.62      2023
   macro avg       0.31      0.25      0.22      2023
weighted avg       0.70      0.62      0.54      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.57      0.95      0.72       288
         2.0       0.00      0.00      0.00        14
         3.0       0.44      0.05      0.10       203
         9.0       0.00      0.00      0.00         1

    accuracy                           0.57       506
   macro avg       0.25      0.25      0.20       506
weighted avg       0.50      0.57      0.45       506
```

FIGURE 22: 3.3 CLASSIFICATION REPORT FOR RF MODEL

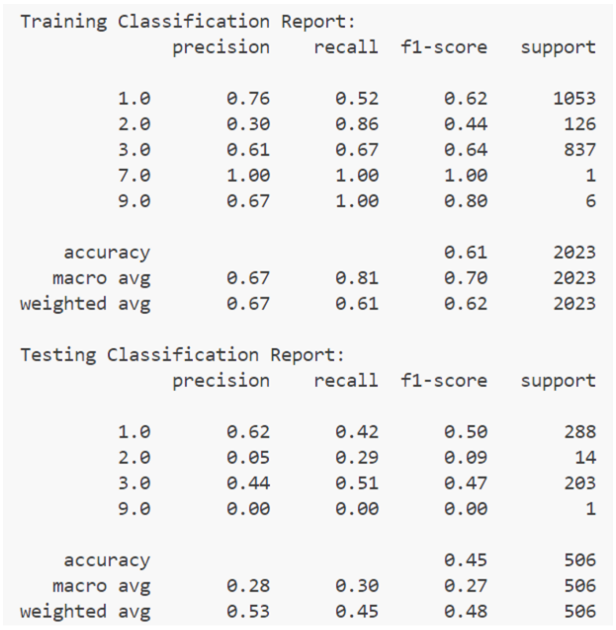
SVM

After SVM model training, the accuracy of the training set is 0.606, and the accuracy of the test set is 0.451. Through the confusion matrix, we find that the prediction of label “1” decreases, but the prediction of label “2” and label “3” is enhanced.



```
Training Accuracy: 0.606030647553139
Testing Accuracy: 0.4505928853754941
Training Confusion Matrix:
[[552 153 348   0   0]
 [  5 108  13   0   0]
 [173 102 559   0   3]
 [  0   0   0   1   0]
 [  0   0   0   0   6]]
Testing Confusion Matrix:
[[121  42 124   1]
 [  4   4   6   0]
 [ 69  30 103   1]
 [  0   0   1   0]]
```

FIGURE 23: 3.3 ACCURACY AND CONFUSION MATRIX FOR SVM MODEL



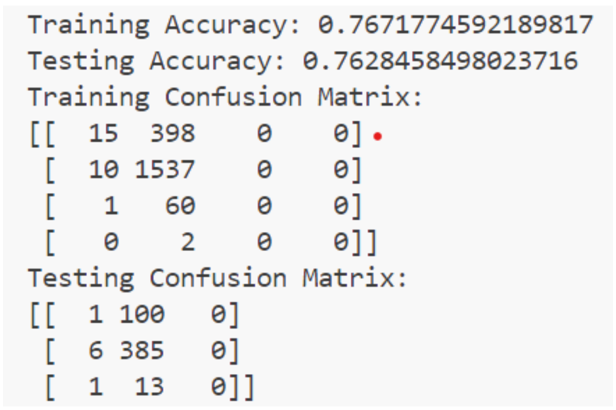
```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.76      0.52      0.62      1053
         2.0       0.30      0.86      0.44       126
         3.0       0.61      0.67      0.64       837
         7.0       1.00      1.00      1.00         1
         9.0       0.67      1.00      0.80         6

    accuracy                           0.61      2023
   macro avg       0.67      0.81      0.70      2023
weighted avg       0.67      0.61      0.62      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.62      0.42      0.50       288
         2.0       0.05      0.29      0.09        14
         3.0       0.44      0.51      0.47       203
         9.0       0.00      0.00      0.00         1

    accuracy                           0.45       506
   macro avg       0.28      0.30      0.27       506
weighted avg       0.53      0.45      0.48       506
```

FIGURE 24: 3.3 CLASSIFICATION REPORT FOR SVM MODEL

4.4 ANALYSIS FOR DIABETES

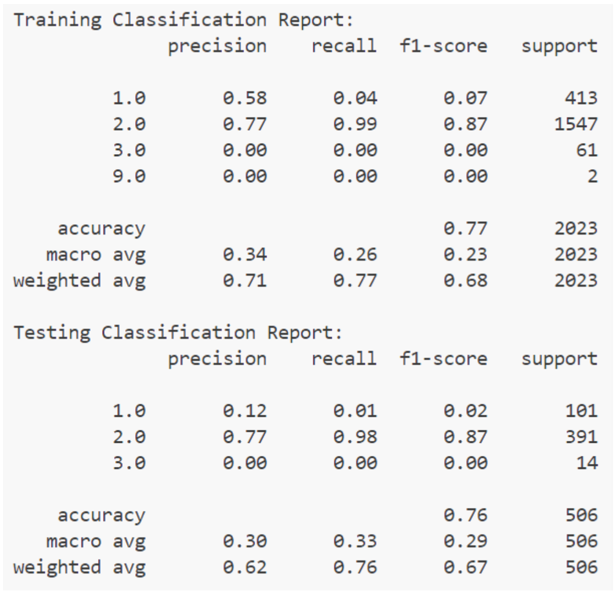
KNN

After the KNN model analysis, we get the best K value of 11, and the accuracy of the corresponding model is 0.767 in the training set (reserved to 3 decimal places, the same below) and 0.763 in the test set. Through the confusion matrix, it is found that the model treats two types of labels, but cannot predict the other two types of samples (because the number of these two types of samples is too small). In addition, according to the classification report, it can be seen that only label "2" can be classified, but the accuracy is low, and the accuracy is unpredictable or low for other labels, so the model is not very ideal.



```
Training Accuracy: 0.7671774592189817
Testing Accuracy: 0.7628458498023716
Training Confusion Matrix:
[[  15  398    0    0]
 [  10 1537    0    0]
 [   1   60    0    0]
 [   0    2    0    0]]
Testing Confusion Matrix:
[[  1 100   0]
 [  6 385   0]
 [  1  13   0]]
```

FIGURE 25: 3.4 ACCURACY AND CONFUSION MATRIX FOR KNN MODEL



```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.58      0.04      0.07       413
         2.0       0.77      0.99      0.87      1547
         3.0       0.00      0.00      0.00        61
         9.0       0.00      0.00      0.00         2

    accuracy                           0.77      2023
   macro avg       0.34      0.26      0.23      2023
weighted avg       0.71      0.77      0.68      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.12      0.01      0.02       101
         2.0       0.77      0.98      0.87       391
         3.0       0.00      0.00      0.00        14

    accuracy                           0.76       506
   macro avg       0.30      0.33      0.29       506
weighted avg       0.62      0.76      0.67       506
```

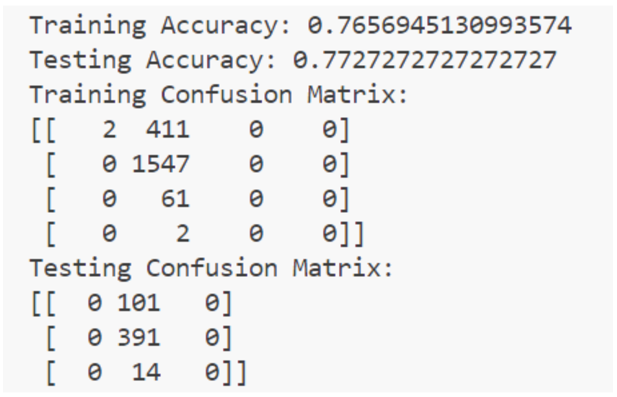
FIGURE 26: 3.4 CLASSIFICATION REPORT FOR KNN MODEL

RF

After the RF model analysis, it can be found that the optimal parameter combination is 'max depth': 10, 'min samples split': 5, 'n estimators': 300, and the accuracy of the training set and the accuracy of the test set are 0.766 and 0.733. However, through the confusion matrix, it can be found that the data imbalance is high, and the model predicts all samples as label "2", so the classification effect is even worse than that of the KNN model.

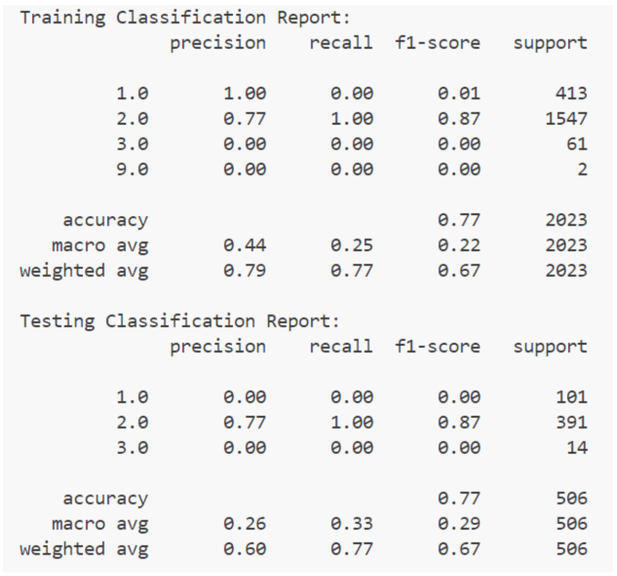
SVM

After SVM model training, the accuracy of the training set is 0.677, and the accuracy of the test set is 0.522. Through the confusion matrix, we find that the prediction ability of the model



```
Training Accuracy: 0.7656945130993574
Testing Accuracy: 0.7727272727272727
Training Confusion Matrix:
[[   2  411    0    0]
 [   0 1547    0    0]
 [   0   61    0    0]
 [   0    2    0    0]]
Testing Confusion Matrix:
[[  0 101   0]
 [  0 391   0]
 [  0  14   0]]
```

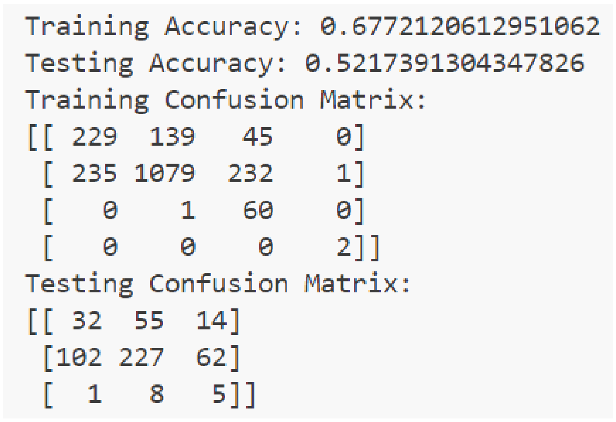
FIGURE 27: 3.4 ACCURACY AND CONFUSION MATRIX FOR RF MODEL



```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       1.00      0.00      0.01       413
         2.0       0.77      1.00      0.87      1547
         3.0       0.00      0.00      0.00        61
         9.0       0.00      0.00      0.00         2

    accuracy                           0.77      2023
   macro avg       0.44      0.25      0.22      2023
weighted avg       0.79      0.77      0.67      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.00      0.00      0.00       101
         2.0       0.77      1.00      0.87       391
         3.0       0.00      0.00      0.00        14

    accuracy                           0.77       506
   macro avg       0.26      0.33      0.29       506
weighted avg       0.60      0.77      0.67       506
```

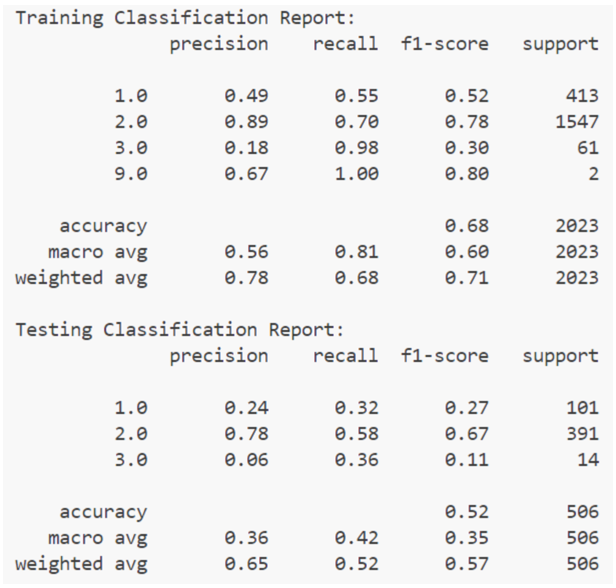
FIGURE 28: 3.4 CLASSIFICATION REPORT FOR RF MODEL

for labels “1”, “3” and “2” has increased, but by comparing the accuracy of the training set and the test set, we can see that there is a certain overfitting problem in the SVM. Compared to the other two models, although the accuracy of the SVM has decreased, the predictions for different categories have improved.



```
Training Accuracy: 0.6772120612951062
Testing Accuracy: 0.5217391304347826
Training Confusion Matrix:
[[ 229  139   45    0]
 [ 235 1079  232    1]
 [   0    1   60    0]
 [   0    0    0    2]]
Testing Confusion Matrix:
[[ 32  55  14]
 [102 227  62]
 [  1   8   5]]
```

FIGURE 29: 3.4 ACCURACY AND CONFUSION MATRIX FOR SVM MODEL



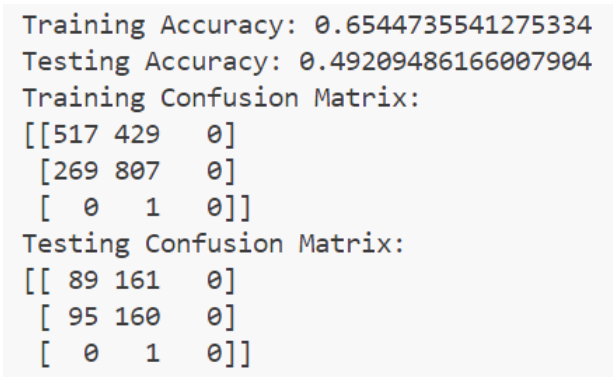
```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.49      0.55      0.52       413
         2.0       0.89      0.70      0.78      1547
         3.0       0.18      0.98      0.30        61
         9.0       0.67      1.00      0.80         2

    accuracy                           0.68      2023
   macro avg       0.56      0.81      0.60      2023
weighted avg       0.78      0.68      0.71      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.24      0.32      0.27       101
         2.0       0.78      0.58      0.67       391
         3.0       0.06      0.36      0.11        14

    accuracy                           0.52       506
   macro avg       0.36      0.42      0.35       506
weighted avg       0.65      0.52      0.57       506
```

FIGURE 30: 3.4 CLASSIFICATION REPORT FOR SVM MODEL

4.5 ANALYSIS FOR HIGH BLOOD PRESSURE

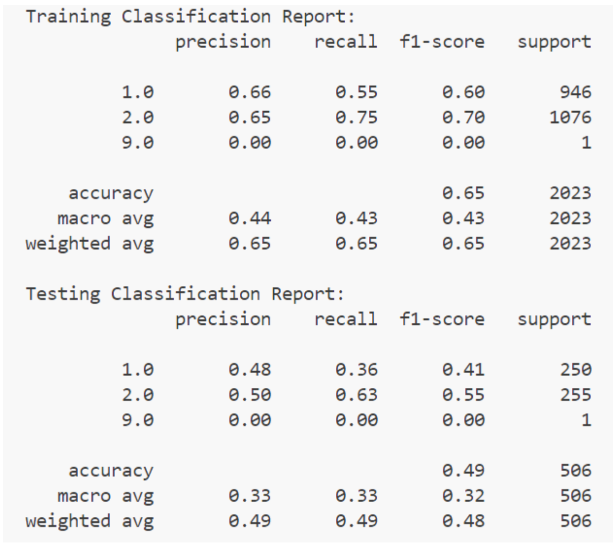
KNN

After the analysis of the KNN model, the optimal K value is 7, and the accuracy of the corresponding model is 0.511 in the training set (reserved to 3 decimal places, the same below) and 0.654 in the test set. From the perspective of the confusion matrix, the trained KNN model can basically complete the prediction of label “1” and label “2”, although the effect is very ideal. Since the number of label “9” is too small (just one) to achieve prediction, we can ignore it. There is a significant difference between the accuracy of the training set and the test set, so there is an overfitting phenomenon in the KNN model.



```
Training Accuracy: 0.6544735541275334
Testing Accuracy: 0.49209486166007904
Training Confusion Matrix:
[[517 429   0]
 [269 807   0]
 [  0   1   0]]
Testing Confusion Matrix:
[[ 89 161   0]
 [ 95 160   0]
 [  0   1   0]]
```

FIGURE 31: 3.5 ACCURACY AND CONFUSION MATRIX FOR KNN MODEL



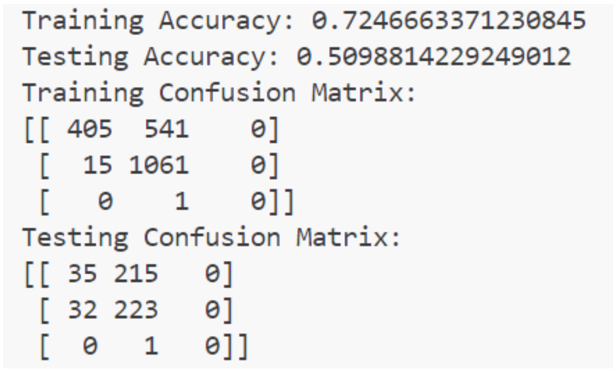
```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.66      0.55      0.60       946
         2.0       0.65      0.75      0.70      1076
         9.0       0.00      0.00      0.00         1

    accuracy                           0.65      2023
   macro avg       0.44      0.43      0.43      2023
weighted avg       0.65      0.65      0.65      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.48      0.36      0.41       250
         2.0       0.50      0.63      0.55       255
         9.0       0.00      0.00      0.00         1

    accuracy                           0.49       506
   macro avg       0.33      0.33      0.32       506
weighted avg       0.49      0.49      0.48       506
```

FIGURE 32: 3.5 CLASSIFICATION REPORT FOR KNN MODEL

RF

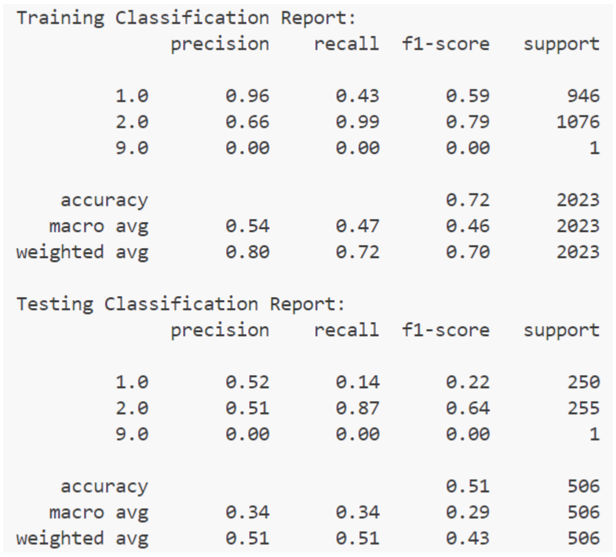
After the RF model analysis, it can be found that the optimal parameter combination is ’max depth’: 10, ’min samples split’: 5, ’n estimators’: 300, and the accuracy of the training set is 0.723 and the accuracy of the test set is 0.511. Through the Confusion Matrix, it can be found that the Random Forest model has a higher prediction accuracy for label “2”. According to the comparison of the accuracy of the classification report with the training set and the test set, it can be seen that the random forest model also has the problem of overfitting.

SVM



```
Training Accuracy: 0.7246663371230845
Testing Accuracy: 0.5098814229249012
Training Confusion Matrix:
[[ 405  541    0]
 [  15 1061    0]
 [   0    1    0]]
Testing Confusion Matrix:
[[ 35 215   0]
 [ 32 223   0]
 [  0   1   0]]
```

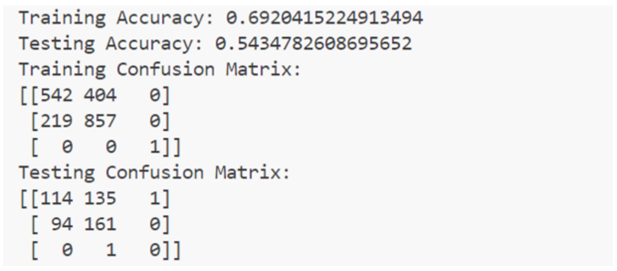
FIGURE 33: 3.5 ACCURACY AND CONFUSION MATRIX FOR RF MODEL



```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.96      0.43      0.59       946
         2.0       0.66      0.99      0.79      1076
         9.0       0.00      0.00      0.00         1

    accuracy                           0.72      2023
   macro avg       0.54      0.47      0.46      2023
weighted avg       0.80      0.72      0.70      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.52      0.14      0.22       250
         2.0       0.51      0.87      0.64       255
         9.0       0.00      0.00      0.00         1

    accuracy                           0.51       506
   macro avg       0.34      0.34      0.29       506
weighted avg       0.51      0.51      0.43       506
```

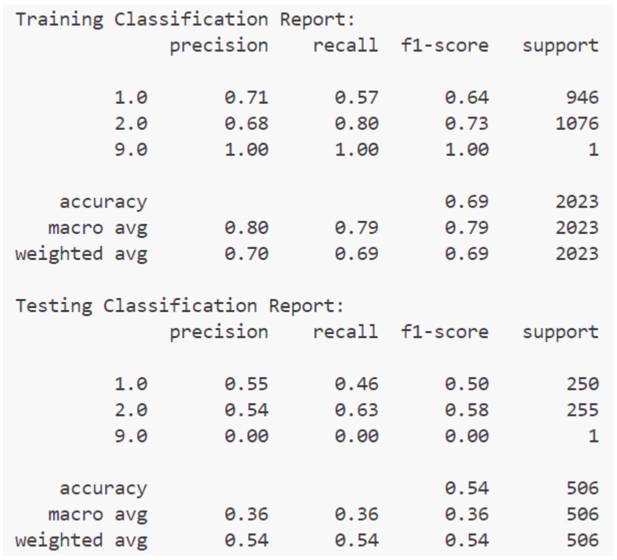
FIGURE 34: 3.5 CLASSIFICATION REPORT FOR RF MODEL

After the SVM model training, the accuracy of the training set is 0.692, and the accuracy of the test set is 0.543. Through the Confusion Matrix, we find that the prediction ability of the proposed model for labels “1” and “2” is higher than that of the KNN model, but the prediction ability for label “2” is worse than that of label “1”. According to the comparison of the accuracy of the classification report with the training set and the test set, it can be seen that the random forest model also has the problem of overfitting.



```
Training Accuracy: 0.6920415224913494
Testing Accuracy: 0.5434782608695652
Training Confusion Matrix:
[[542 404   0]
 [219 857   0]
 [  0   0   1]]
Testing Confusion Matrix:
[[114 135   1]
 [ 94 161   0]
 [  0   1   0]]
```

FIGURE 35: 3.5 ACCURACY AND CONFUSION MATRIX FOR SVM MODEL



```
Training Classification Report:
              precision    recall  f1-score   support

         1.0       0.71      0.57      0.64       946
         2.0       0.68      0.80      0.73      1076
         9.0       1.00      1.00      1.00         1

    accuracy                           0.69      2023
   macro avg       0.80      0.79      0.79      2023
weighted avg       0.70      0.69      0.69      2023

Testing Classification Report:
              precision    recall  f1-score   support

         1.0       0.55      0.46      0.50       250
         2.0       0.54      0.63      0.58       255
         9.0       0.00      0.00      0.00         1

    accuracy                           0.54       506
   macro avg       0.36      0.36      0.36       506
weighted avg       0.54      0.54      0.54       506
```

FIGURE 36: 3.5 CLASSIFICATION REPORT FOR SVM MODEL

4.6 CONCLUSION FOR MACHINE LEARNING

For Angina, a more ideal model is SVM, with Training Accuracy 0.800 and Testing Accuracy 0.702.

For Overweight, the random forest model predicts better for label ”1” with Training Accuracy 0.615 and Testing Accuracy 0.565, and the SVM model predicts better for label ”2” and label ”3”, with Training Accuracy 0.606 and Testing Accuracy 0.451 (Label ”7” and Label ”9” are too small in number to be taken into account).

For Diabetes, SVM is more desirable considering the prediction of different kinds, with Training Accuracy 0.677 and Testing Accuracy 0.521.

For High Blood Pressure, SVM is more desirable considering different kinds of predictions, with Training Accuracy 0.692 and Testing Accuracy 0.543.

5 SUPERVISED DEEP LEARNING

Supervised learning is a deep learning model that can predict diseases and can employ more neurons and more complex models to learn hidden features compared to machine learning.

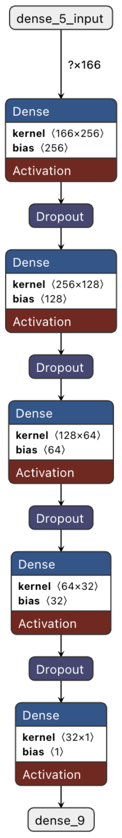


FIGURE 37: DEEP LEARNING NETWORK ARCHITECTURE

5.1 DATA PRESENTATION

The data consisted of 166 characteristics on four aspects of the questionnaire, dietary structure, biochemical indicators, demographics, and predictive data on four diseases (Angina, Overwhight, Diabetes, High blood pressure) denoted as Disease 1,2,3,4. and transformed the predictive data with the presence of multiple data, retaining only 1: Presence of Disease, and 0:no disease present. and discard all NA values.

5.2 MODEL SELECTION

Consider that we have 166 features and four types of diseases to be predicted. One can choose 166input-4output or four 1output models. After trying to have 4output the model performance results are extremely poor and accuracy is only around 50

5.3 MODEL LAYER SELECTION

The tensorflow library provides easy code for building deep learning models. However, considering that our data is structured, we are not sure that long and short memory models, RNN, CNN kind of layers perform better. After trial and error, our best model uses 5 fully linked layers with

$$2^7, \quad 2^6, \quad 2^5, \quad 2^4, \quad 2^0$$

neurons respectively.

5.4 OVERFITTING

In order to reduce the occurrence of overfitting, we add a Dropout(0.4) layer in every two layers and perform L1 and L0 regularisation technique for each layer. And use early stopping technique during training.

5.5 SELECTION OF TRAINING METRICS

The F1 metric is one of the common evaluation metrics in both academic and deep learning fields to measure the performance of classification models. It combines the Precision and Recall of the model and aims to combine the accuracy and comprehensiveness of the model.The F1 score takes values between 0 and 1, with the closer to 1 indicating that the model performs better in terms of Precision and Recall. The F1 score is calculated as.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

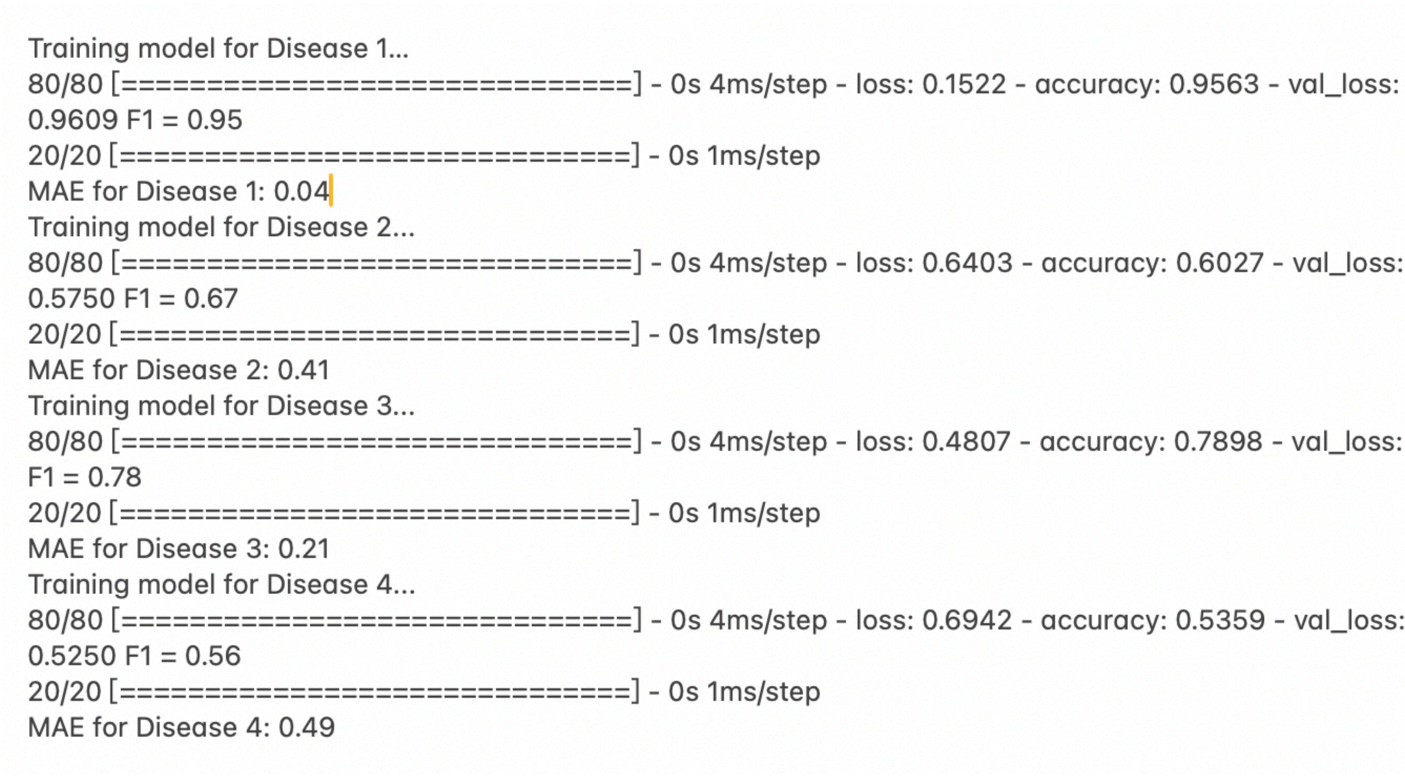
Where Precision and Recall are defined as.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

One consideration for using F1 as an evaluation metric is that the metric can be applied to machine learning, deep learning, and automated machine learning, facilitating direct comparisons of these models by researchers. Another consideration is that this metric is extremely friendly to our unbalanced dataset.

5.6 RESULT



```
Training model for Disease 1...
80/80 [==============================] - 0s 4ms/step - loss: 0.1522 - accuracy: 0.9563 - val_loss:
0.9609 F1 = 0.95
20/20 [==============================] - 0s 1ms/step
MAE for Disease 1: 0.04
Training model for Disease 2...
80/80 [==============================] - 0s 4ms/step - loss: 0.6403 - accuracy: 0.6027 - val_loss:
0.5750 F1 = 0.67
20/20 [==============================] - 0s 1ms/step
MAE for Disease 2: 0.41
Training model for Disease 3...
80/80 [==============================] - 0s 4ms/step - loss: 0.4807 - accuracy: 0.7898 - val_loss:
F1 = 0.78
20/20 [==============================] - 0s 1ms/step
MAE for Disease 3: 0.21
Training model for Disease 4...
80/80 [==============================] - 0s 4ms/step - loss: 0.6942 - accuracy: 0.5359 - val_loss:
0.5250 F1 = 0.56
20/20 [==============================] - 0s 1ms/step
MAE for Disease 4: 0.49
```

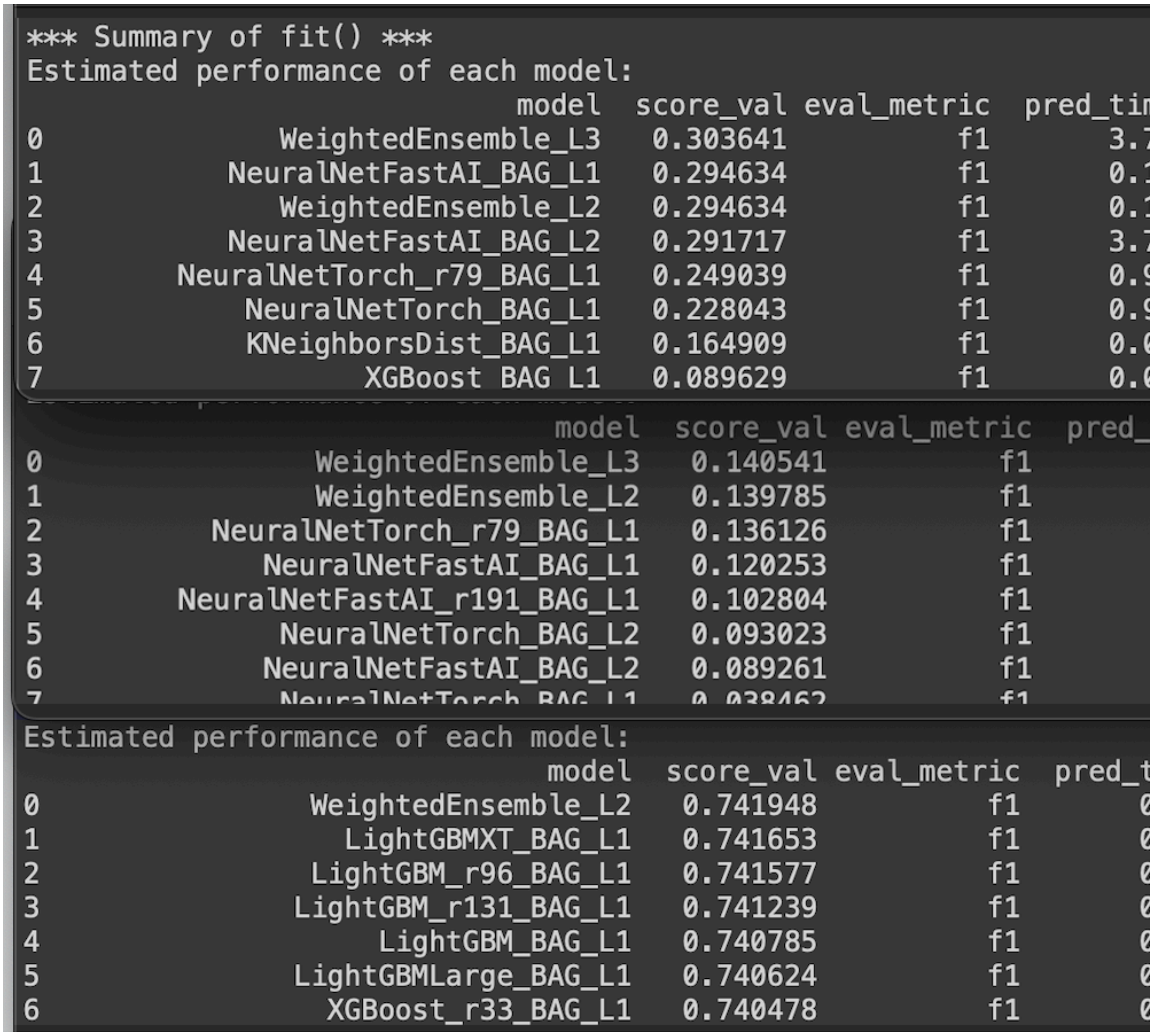
FIGURE 38: DEEP LEARNING NETWORK EFFECTS

For the model of disease 1, although it achieved the highest accuracy of 0.9609 and the best F1 score of 0.95 on the validation set, it was higher than the performance on the training set or there was some risk of overfitting. The models for Disease 2 and Disease 3, on the other hand, show relatively satisfactory generalization ability, with the former having an F1 score of 0.67 and an accuracy of 0.5750, and the latter having an F1 score of 0.78 and an accuracy of 0.7875, and both of them reach an acceptable level in terms of prediction performance. However, the model of Disease 4 performed the most poorly, with a validation set accuracy of only 0.5250 and an F1 score of 0.56, and the prediction effect needs to be further optimized and improved.

5.7 AUTOGLUON

After using machine learning and deep learning with manually tuned parameters, we tried Autogluon, an automated neural network training tool that helps developers train models with less code, more efficient computational resources, and faster. Here are the results.

We can notice that this result is not satisfactory, only the third model has an F1 above 0.7, the others are below 0.5, so for our data, manual adjustment of the parameters could give better results



```
*** Summary of fit() ***
Estimated performance of each model:
                              model  score_val eval_metric  pred_tim
0               WeightedEnsemble_L3   0.303641          f1       3.7
1            NeuralNetFastAI_BAG_L1   0.294634          f1       0.1
2               WeightedEnsemble_L2   0.294634          f1       0.1
3            NeuralNetFastAI_BAG_L2   0.291717          f1       3.7
4         NeuralNetTorch_r79_BAG_L1   0.249039          f1       0.9
5             NeuralNetTorch_BAG_L1   0.228043          f1       0.9
6            KNeighborsDist_BAG_L1   0.164909          f1       0.0
7                    XGBoost BAG L1   0.089629          f1       0.0
```

```
                              model  score_val eval_metric  pred_
0               WeightedEnsemble_L3   0.140541          f1
1               WeightedEnsemble_L2   0.139785          f1
2         NeuralNetTorch_r79_BAG_L1   0.136126          f1
3            NeuralNetFastAI_BAG_L1   0.120253          f1
4       NeuralNetFastAI_r191_BAG_L1   0.102804          f1
5             NeuralNetTorch_BAG_L2   0.093023          f1
6            NeuralNetFastAI_BAG_L2   0.089261          f1
7             NeuralNetTorch_BAG_L1   0.038462          f1
```

```
Estimated performance of each model:
                              model  score_val eval_metric  pred_t
0               WeightedEnsemble_L2   0.741948          f1      0
1                 LightGBMXT_BAG_L1   0.741653          f1      0
2               LightGBM_r96_BAG_L1   0.741577          f1      0
3              LightGBM_r131_BAG_L1   0.741239          f1      0
4                   LightGBM_BAG_L1   0.740785          f1      0
5              LightGBMLarge_BAG_L1   0.740624          f1      0
6                XGBoost_r33_BAG_L1   0.740478          f1      0
```

FIGURE 39: DEEP LEARNING NETWORK EFFECTS

5.8 SUMMARY AND PERSPECTIVE

Using simple deep learning with autoML can get good results. However, take the logarithmic processing of the oversized data, add textual information about the introduction of the features, extract the features with a transformer using multimodal, do multi-model fusion, and then do multi-layer model ensemble to get better accuracy. However, due to the limitation of the equipment, if we add the above additional optimisation, it may consume one to two days to train the model, and debugging may take even longer, so we did not make the above optimisation, but we expect that there are other research teams that can obtain more accurate multi-disease prediction models with suitable equipment.

6 WEB APPLICATION

6.1 INTRODUCTION

To better display the results of multi-disease prediction, we will use the Flask establishing a small web application to display predictive outcomes. Flask is a lightweight web application framework written in Python, which utilizes the Werkzeug WSGI toolkit and Jinja2 template engine, licensed under BSD (22), which allows users to add application functionality as if they were built into the framework itself (23; 24). The picture below shows the basic file structures of a developed web application which contains 3 main parts.

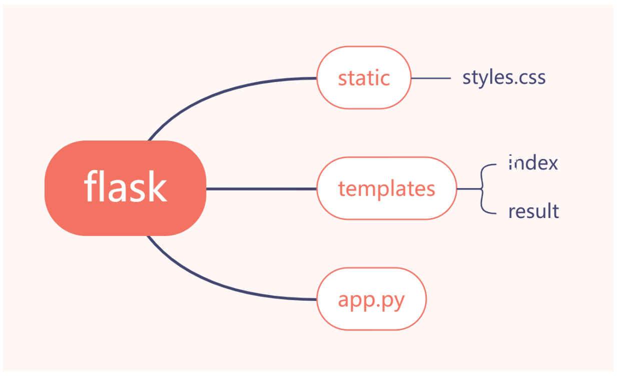


FIGURE 40: BASIC FILE STRUCTURE

6.2 BASIC STRUCTURE

app.py: This python file includes Flask APIs which can receive information of multi-diseases through GUI or API calls, compute the predictive value using our model and return the results.

Templates: There are two HTML files in the templates. The index.html in this folder allows the user to enter essential information and the result.html will showcase the expected outcome.

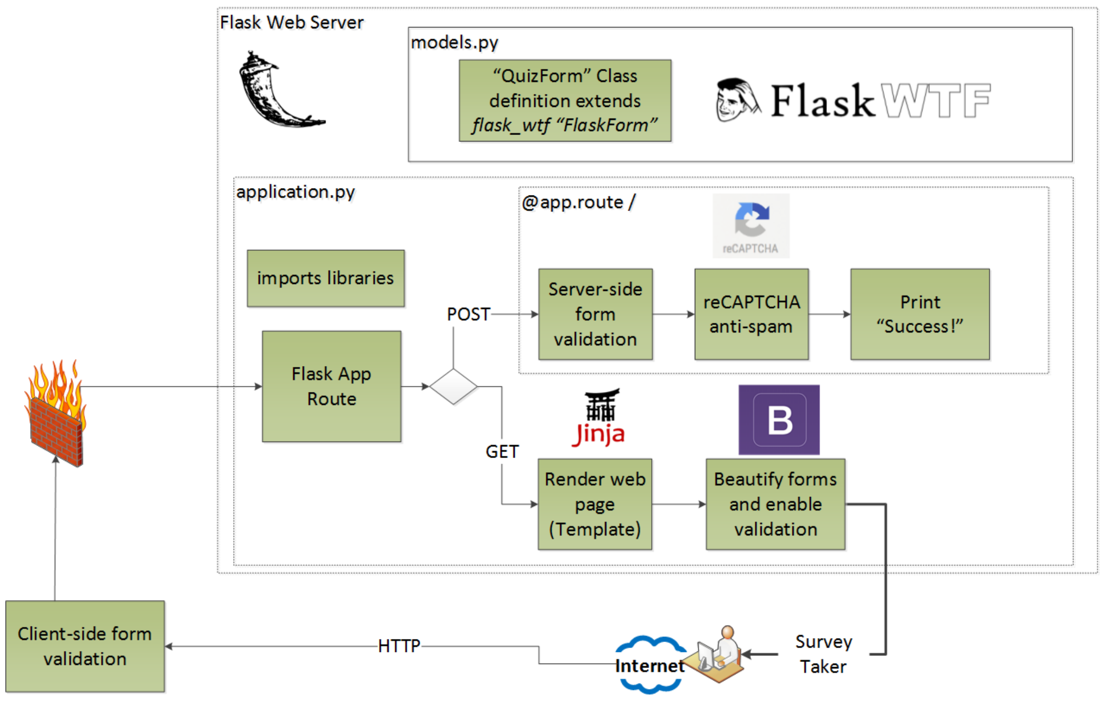
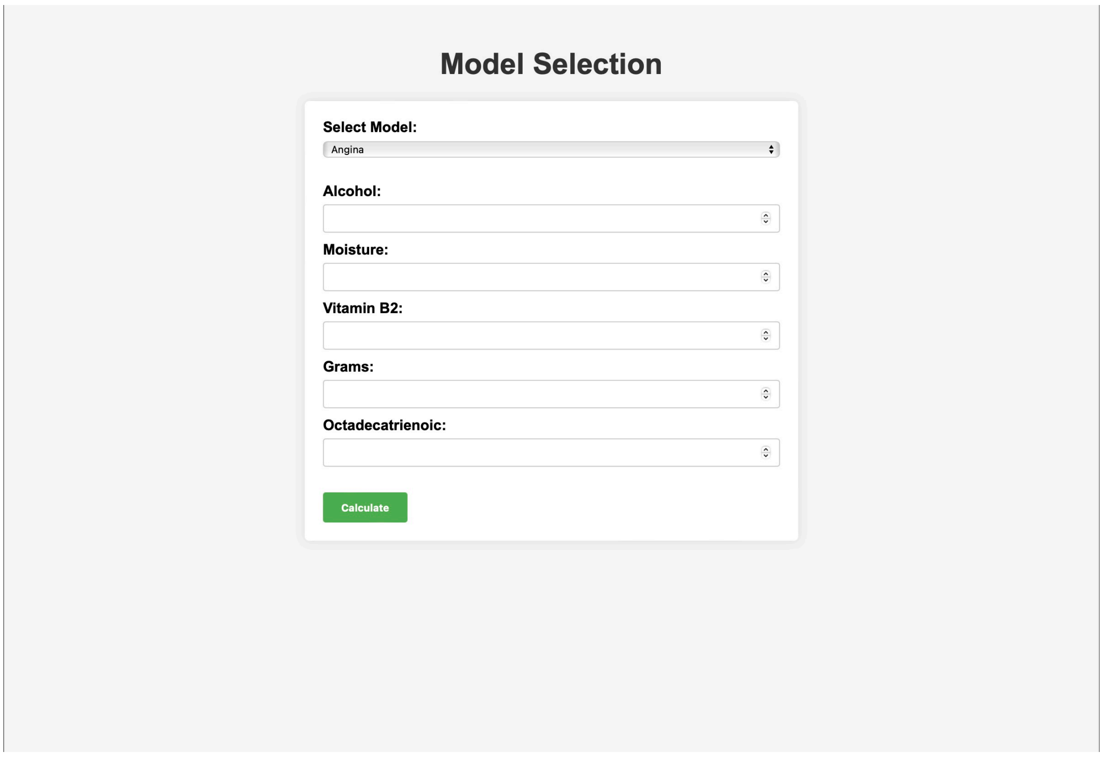

Flask Web Server
models.py
"QuizForm" Class definition extends flask_wtf "FlaskForm"
Flask WTF
application.py
@app.route /
reCAPTCHA
imports libraries
POST
Server-side form validation
reCAPTCHA anti-spam
Print "Success!"
Flask App Route
GET
Jinja
B
Render web page (Template)
Beautify forms and enable validation
Client-side form validation
HTTP
Internet
Survey Taker

FIGURE 41: WORKFLOW OF WEB APPLICATION

Static: It contains the Styles.css file which has the styling required for our HTML form, like size, margin, color and so on.

6.3 WORKFLOW OF WEB APPLICATION

In brief, the workflow of flask has four steps: First, The user sends the necessary information like alcohol, moisture, Vitamin B2, Grams and Octadecatrienoic, which are required by the application in a Webpage. Then, the important information is sent to the back-end. Next, the flask server, adopted with the prepared DL algorithm models, predicts the results. Finally, the predicted result is shown in the webpage.



Model Selection

Select Model: Angina

Alcohol:

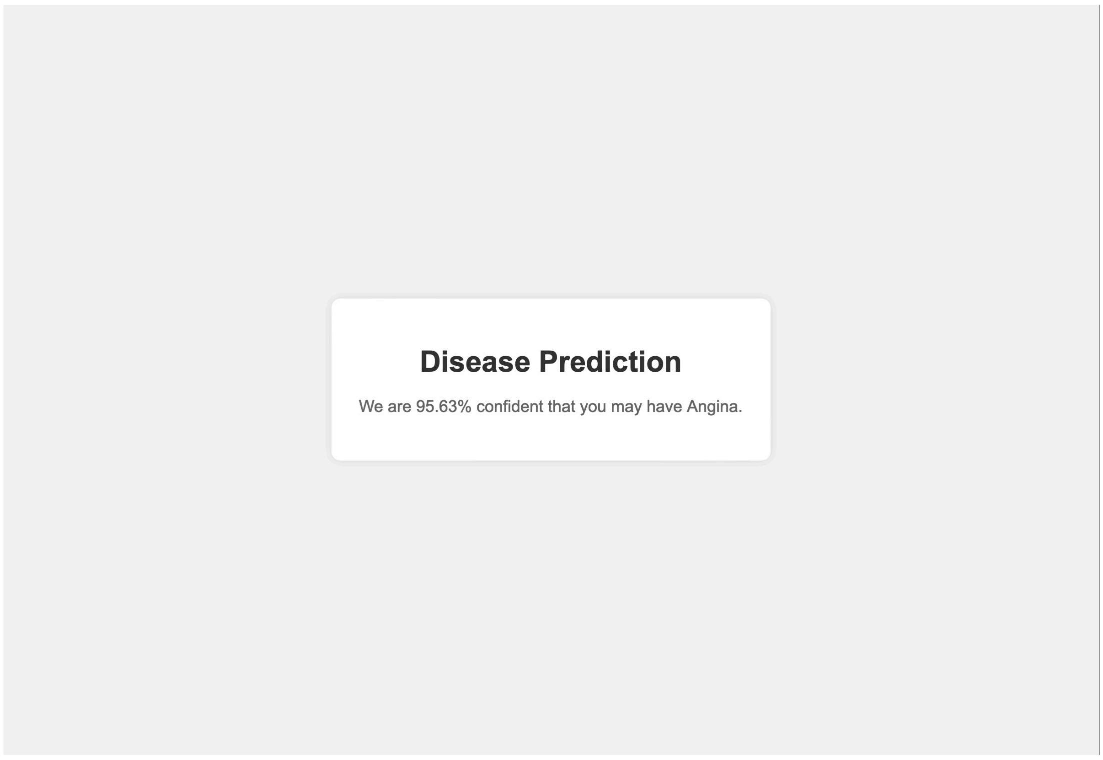
Moisture:

Vitamin B2:

Grams:

Octadecatrienoic:

FIGURE 42: WEBPAGE



Disease Prediction

We are 95.63% confident that you may have Angina.

FIGURE 43: POTENTIAL RESULT

7 CONCLUSION

7.1 RESULT

In the experiment, as mentioned in the introduction before, we used Machine Learning, Deep Learning and AutoML (Automatic Machine Learning). In terms of model training, we found that deep learning has better accuracy and applicability than machine learning, given the complexity of the data set.

Considering the accuracy of model training and the convenience of model export, we decided to use the model generated by deep learning to design the Python-based web application which can make preliminary predictions for these four diseases. Although there is still room for improvement, yet at least it can make basic judgments about people's health conditions based on some key indicators with the accuracy rate 0.9563, 0.6027, 0.7898 and 0.5359 respectively for the 4 diseases.

7.2 LIMITATIONS AND FUTURE WORK

Using simple deep learning model gives better results, but taking logarithms optimizes the structure of data that changes quickly and has large values. And adding textual information to each feature, extracting features using transformers and performing multi-modal fusion can give better results.

Our flask application also has high accuracy, but the UI design of the user interface is not quite reasonable. Besides, there are only 5 features with the highest correlation coefficients for the input values, which is not sufficient to output a more comprehensive result. Optimization of the program can be done by processing methods like importing the medical physical examination CSV file to increase the correctness and accuracy of the application.

In addition, we open the source code of the file, hoping that there will be more research on multi-disease prediction models in the future, making full use of data resources in the era of big data for the benefit of all mankind.

8 CODE AND MODEL ACQUISITION

8.1 CODE AVAILABILITY:

The source code used in this project is open source on GitHub (`https://github.com/normal743/Multi-Disease-Prediction-Model-Based-on-Deep-Learning-and-U.S.-Healthcare-Data`), under the MIT license.

8.2 MODEL AVAILABILITY:

The automated neural network trained model is too large to upload to the repository. It can be accessed from this Google Drive link: `https://drive.google.com/drive/folders/1h-VdLZPPfOlQ8nA2Kmd6-NZiFXuZZPQm?usp=sharing`

References

[1] A. Budreviciute, S. Damiati, D. K. Sabir, and R. Kodzius, “Management and prevention strategies for non-communicable diseases (ncds) and their risk factors,” *Frontiers in public health*, vol. 8, p. 574111, 2020.

[2] S. M. S. Islam, T. D. Purnat, N. T. A. Phuong, U. Mwingira, K. Schacht, and G. Fr”oschl, “Non-communicable diseases (ncds) in developing countries: a symposium report,” *Global Health*, vol. 10, p. 81, 2014.

[3] N. Unwin and K. G. M. M. Alberti, “Chronic non-communicable diseases,” *Annals of Tropical Medicine & Parasitology*, vol. 100, no. 5-6, pp. 455–464, 2006.

[4] World Health Organization, “World health statistics 2021: Monitoring health for the sdgs, sustainable development goals.” [Online]. Available: `https://www.who.int/data/gho/publications/world-health-statistics`, 2021.

[5] I. Csige, D. Ujv’arosy, Z. Szab’o, I. Lőrincz, G. Paragh, M. Harangi, and S. Somodi, “The impact of obesity on the cardiovascular system,” *Journal of Diabetes Research*, vol. 2018, pp. 1–12, 2018.

[6] Z. Hossain, K. A. Ahmed, M. R. Hasan, T. Gedeon, and Z. Zhang, “A deep learning approach to diabetes diagnosis,” *arXiv*, Mar. 2024.

[7] J. Weiß and et al., “Deep learning to estimate cardiovascular risk from chest radiographs,” *Annals of Internal Medicine*, Mar. 2024.

[8] K. W. DeGregory and et al., “A review of machine learning in obesity,” *Obesity Reviews (Print)*, vol. 19, pp. 668–685, Feb. 2018.

[9] P. Khan and et al., “Machine learning and deep learning approaches for brain disease diagnosis: Principles and recent advances,” *IEEE Access*, vol. 9, pp. 37622–37655, Jan. 2021.

[10] C. Anilkumar, A. Kishore, B. K. Sasapu, and K. Seepana, “Multi chronic disease prediction system using cnn and random forest,” *SN Computer Science*, vol. 5, Jan. 2024.

[11] M. Harrison, *Machine Learning Pocket Reference: Working with Structured Data in Python*. O’Reilly Media, 2019.

[12] T. Nagarajah and G. Poravi, “A review on automated machine learning (automl) systems,” in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1–6, IEEE, March 2019.

[13] BURYBURYZYMON, "Us healthcare data." https://www.kaggle.com/datasets/maheshdadhich/us-healthcare-data/discussion/373146#2193184, 2022. Accessed: 2024-05-13.

[14] C. R. Turner, A. Fuggetta, L. Lavazza, and A. L. Wolf, "A conceptual basis for feature engineering," *Journal of Systems and Software*, vol. 49, no. 1, pp. 3–15, 1999.

[15] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *Lecture notes in computer science*, pp. 986–996, 2003.

[16] M.-L. Zhang and Z. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, pp. 2038–2048, July 2007.

[17] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, pp. 197–227, Apr. 2016.

[18] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues*, vol. 9, pp. 272–278, Sept. 2012.

[19] L. Langsetmo and et al., "Advantages and disadvantages of random forest models for prediction of hip fracture risk versus mortality risk in the oldest old," *JBMR Plus*, vol. 7, July 2023.

[20] H. Yu and S. Kim, *SVM Tutorial — Classification, Regression and ranking*, pp. 479–506. 2012.

[21] L. Auria and R. A. Moro, "Support vector machines (svm) as a technique for solvency analysis," *Social Science Research Network*, Jan. 2008.

[22] M. Grinberg, *Flask Web Development*. O'Reilly Media, Inc., 2018.

[23] N. Ahmed, R. Ahammed, M. M. Islam, M. A. Uddin, A. Akhter, M. A. Talukder, and B. K. Paul, "Machine learning based diabetes prediction and development of smart web application," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 229–241, 2021.

[24] A. Yaganteeswarudu, "Multi disease prediction model by using machine learning and flask api," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, (Coimbatore, India), pp. 1242–1246, IEEE, 2020.