Covid-19 Analysis & Predication with Multiple AI Methods

CS181 Artificial Intelligence I Course Project

1. Introduction

Today, the new coronavirus is still prevalent worldwide. In case of this, the issue of predicting the infection and severity rate according to the other health factors of people has been researched a lot. And this question of making prediction based on the related factors is a great task for realizing and testing multiple AI methods that we learned, making improvement and doing comparisons.

In our project, we adopted the a dataset from Mexico government and WHO, which contains the basic health attributes and covid infection information of 263008 People. This dataset not only covers the basic information, but also records most common types of underlying diseases and habits that affect health. It's very detailed meaningful true-world for operating our AI approaches.

Based on these knowledges, we cleaned and processed the dataset, making it more accurate. Then we do some learning-based observations on the correlation between thee 11 health factors and the covid infection, finding the 3 most related factors: pneumonia, age and diabetes.

According our observations, we designed the training and test set and carried out multiple AI methods in the prediction work on them. We accomplished our own version of Bayes Network, K nearest neighbors, Decision Tree, and Neural Network (with Multilayer Perceptron), and compared them with the scikit-learn versions in performance. This test and evaluation are based on ROC curve and accuracy rate. It turns out that our versions perform well and have a close outcome in accuracy compared with scikit-learn versions. In addition, some further analysis on the performance of the methods that we realized and iterating improvement are also done in order to achieve higher level completion.

2. Data Processing

Our data is from Mexico government's statistics on 263008 people in aspects of health factors and covid infections. It covers 3 types of basic health attributes that related to the probability of covid infection: age and sex, underlying diseases, habits. The age is a discrete variable that varies in a large range, from 0 to 120. And other 10 factors are 0-1 bool type, representing the person has it or not.

However, these datasets have some errors, like invalid age that out of the range and none bool attributes. Some cleaning and type transforms are done and we got a clean dataset. Below is a view of the mean, range and the distributions of the variables in the processed dataset.

	Age	Sex	Pneumonia	Diabetes	Ashma	Hypertension	CVDs	Obesity	CKDs	Tabacco	Result
count	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000	261409.000000
mean	42.551626	0.490006	0.161012	0.129361	0.036311	0.168472	0.025802	0.166528	0.021793	0.090368	0.389233
std	16.894625	0.499901	0.367543	0.335599	0.187063	0.374285	0.158546	0.372555	0.146009	0.286709	0.487577
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	31.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	41.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	53.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
max	120.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Fig1: data details after processing

The actual valid dataset size shrinks from 263008 to 261409. We visualized the distributions and 0-1 rates of the variables for further analysis.

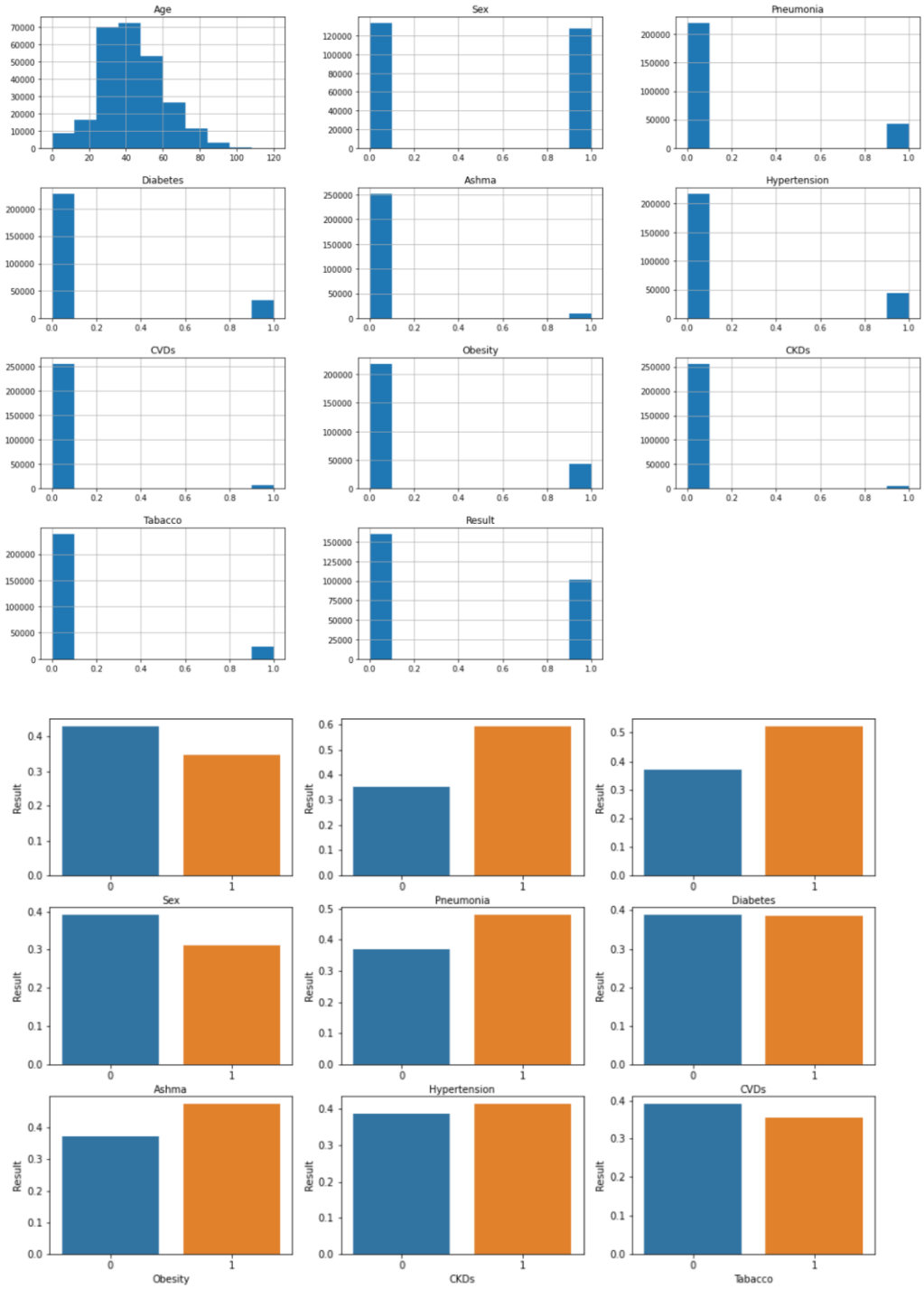


Fig2. distributions of the variables

3. Correlation

After data processing, we want to zoom in to the core of the task. In order for this, we need to know which factors have the closest relation with covid infection result. With Calculating pairwise correlation, we build the correlation map between the variables and the covid result.

From the correlation map, we find that the Age, Pneumonia, Diabetes are the top3 related factors to the covid infection. According to this, we divide the dataset into train set and test set and keep only these 3 factors in them to reduce the dataset's complexity.

Especially, for the age variable, which is not bool type but varies in a large range, we visualized the covid infection result based on this distribution. This is a direct and powerful tool in our next part's model design and customized work, since it gives a quick overview of the outcomes.

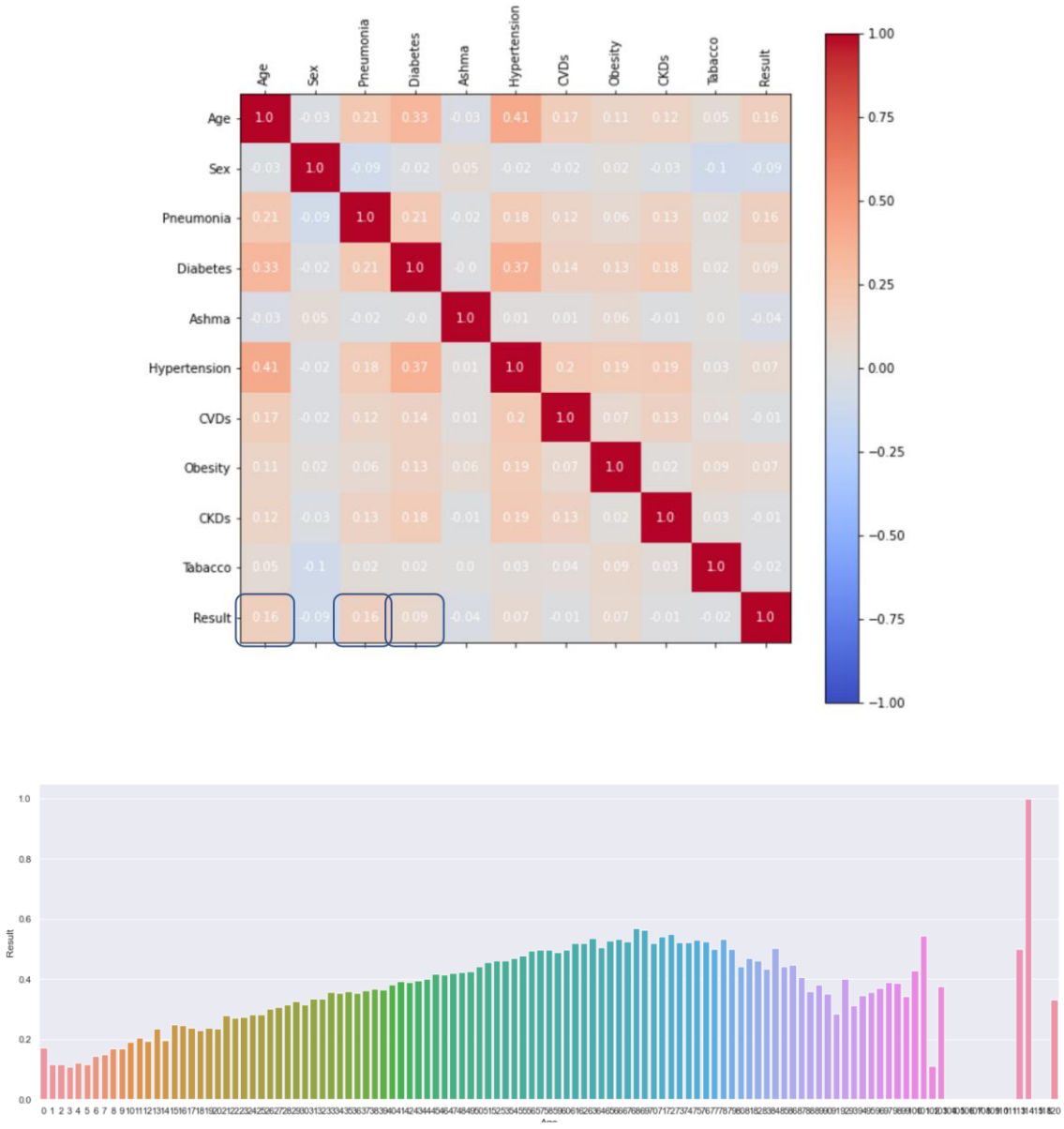


Fig3. Correlation map and aged-based covid distribution

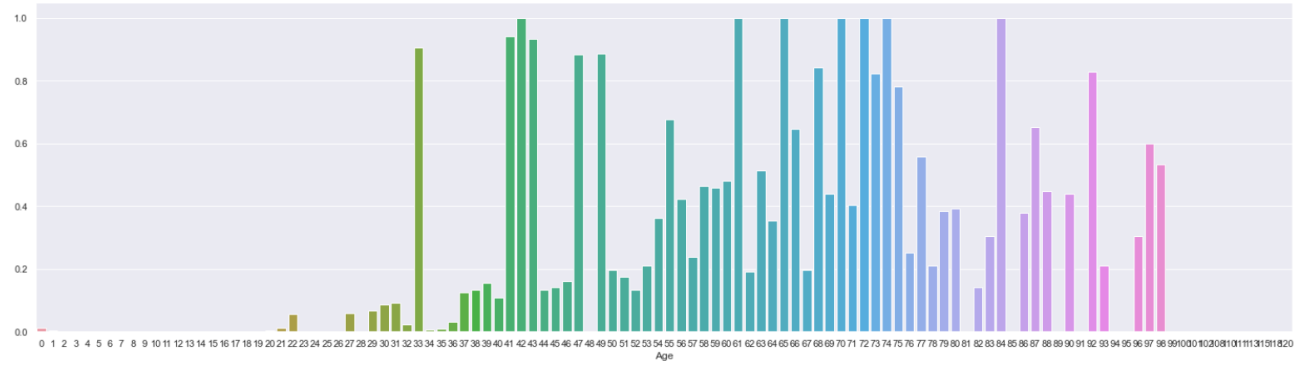
4. Algorithm: Multiple AI Methods

We realized these algorithms on our own, made some improvements and tested them on the dataset and accomplished evaluation based on ROC curve and accuracy rate. The performance of our versions are compared with sk-learn's standard versions.

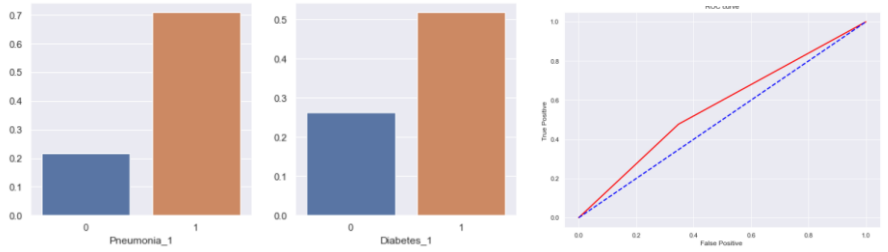
4.1 KNN

KNN is an algorithm that picks the k nearest neighbors of the sample to be predicted. The distance can be Euclidean distance, Manhattan distance, Minkowski distance, Hamming distance, etc. We implemented KNN from scratch and also from sklearn. Euclidean distance and Manhattan distance is enabled in our implementation.

Distribution of age on the proportion of results:



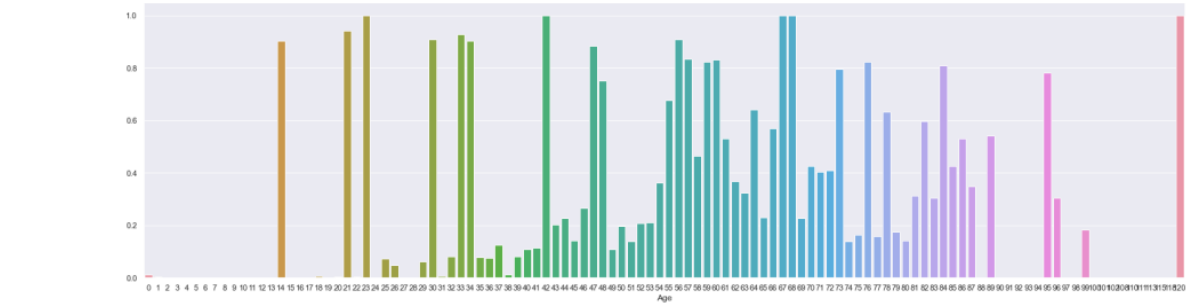
Distribution of Pneumonia, Diabetes on the proportion of results:




```
[[34805 12899]
 [18992 11727]]
              precision    recall  f1-score   support

           0       0.65      0.73      0.69     47704
           1       0.48      0.38      0.42     30719

    accuracy                           0.59     78423
   macro avg       0.56      0.56      0.55     78423
weighted avg       0.58      0.59      0.58     78423
```

We also tried to measure the strength of the algorithm using different metrics. We plot the ROC curve, which horizonal coordinates is FPR(False Positive Rate), and vertical coordinates is TPR(True Positive Rate). Also wo plotted the confusion matrix, precision and recall which are commonly used in machine learning prediction.

We also introduced the AUC, the area under the ROC curve, who can quantitatively describe the output and the correctness of the actual results. The distribution of the results with respect to age shows that the KNN is a poor fit, but the accuracy is good as can be seen by the ROC plot. The prediction of our own implementation is quite similar to sk-learn:

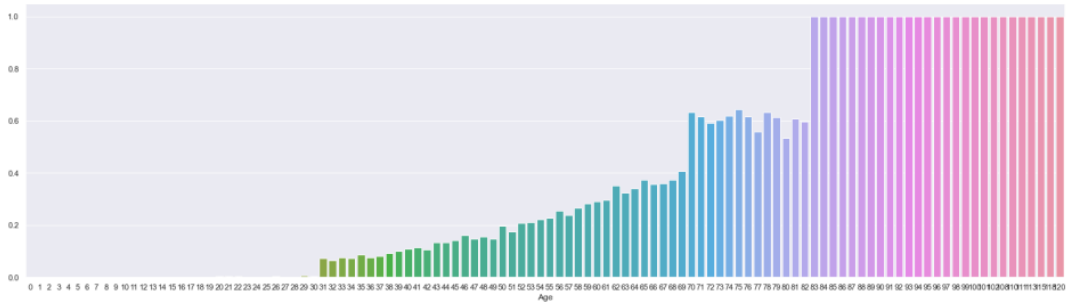



```
[[33329 14375]
 [18595 12124]]
              precision    recall  f1-score   support

           0       0.64      0.70      0.67     47704
           1       0.46      0.39      0.42     30719

    accuracy                           0.58     78423
   macro avg       0.55      0.55      0.55     78423
weighted avg       0.57      0.58      0.57     78423

Precision Score :  0.5795876209785394
Recall Score :  0.5795876209785394
0.5795876209785394
```

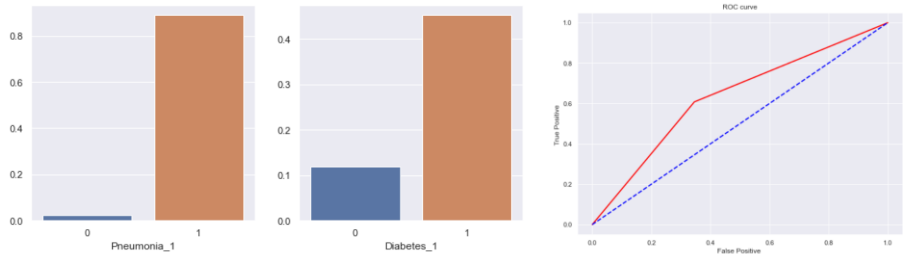
Fig4. Our KNN Performance: accuracy rate, ROC and AUC

4.2 Logistic Regression

Logistic regression is a combination of linear regression and activation function. In this case, we choose sigmoid function. Loss function is cross entropy, and use gradient descent to get optimal solution.



The distribution of the results with respect to age is a linear distribution, however, it is quite different to the original distribution in our training dataset. We prefer to evaluate it as a poor fit. It performs better than KNN in the accuracy test and also has a better ROC curve.




```
[[42601  5103]
 [23079  7640]]
              precision    recall  f1-score   support

           0       0.65      0.89      0.75     47704
           1       0.60      0.25      0.35     30719

    accuracy                           0.64     78423
   macro avg       0.62      0.57      0.55     78423
weighted avg       0.63      0.64      0.59     78423
```

Fig5. Our Logistic Regression Performance

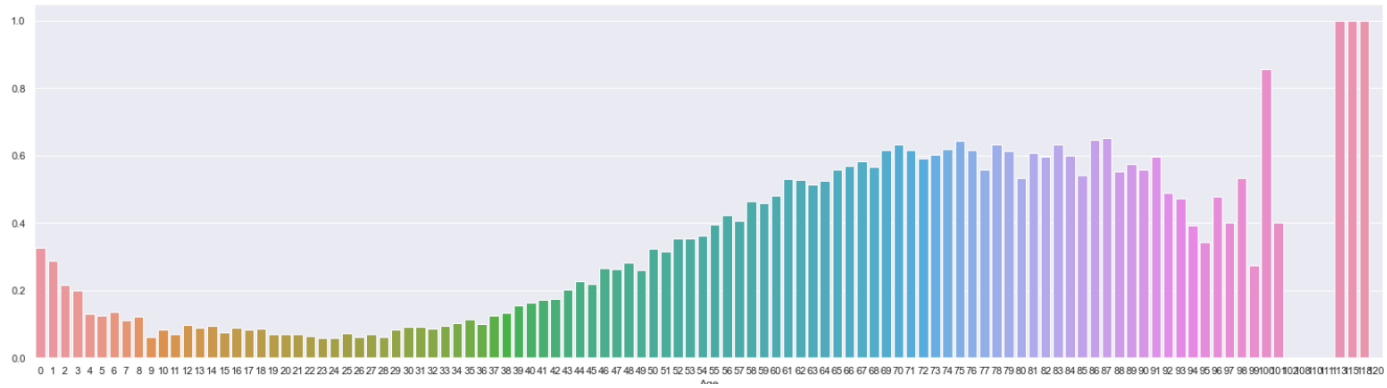
4.3 Naïve Bayes

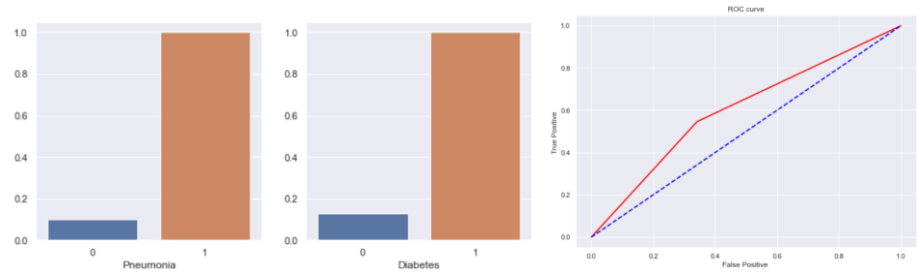
Naïve Bayes is based on Bayes Rule and Conditional Independence, and also, because of there is continuous variable in our dataset, so we decided to choose Gaussian Distribution to approximate the probability of a particular value. The below is core basis for our model realization:

$$P(Y=c_k|X=x)=\frac{P(Y=c_k)P(X=x|Y=c_k)}{\sum_k P(X=x|Y=c_k)P(Y=c_k)} \quad =\frac{P(Y=c_k)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=c_k)}{\sum_k P(Y=c_k)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=c_k)}$$

$$N(x|\mu,\sigma^2)=\frac{1}{\sqrt{2\pi}\sigma}exp\{-\frac{(x-\mu)^2}{2\sigma^2}\} \qquad f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

The distribution of the results with respect to age is shown below. So far, this distribution best fits the performance of the original data set, and naive Bayes also has a good performance in accuracy.



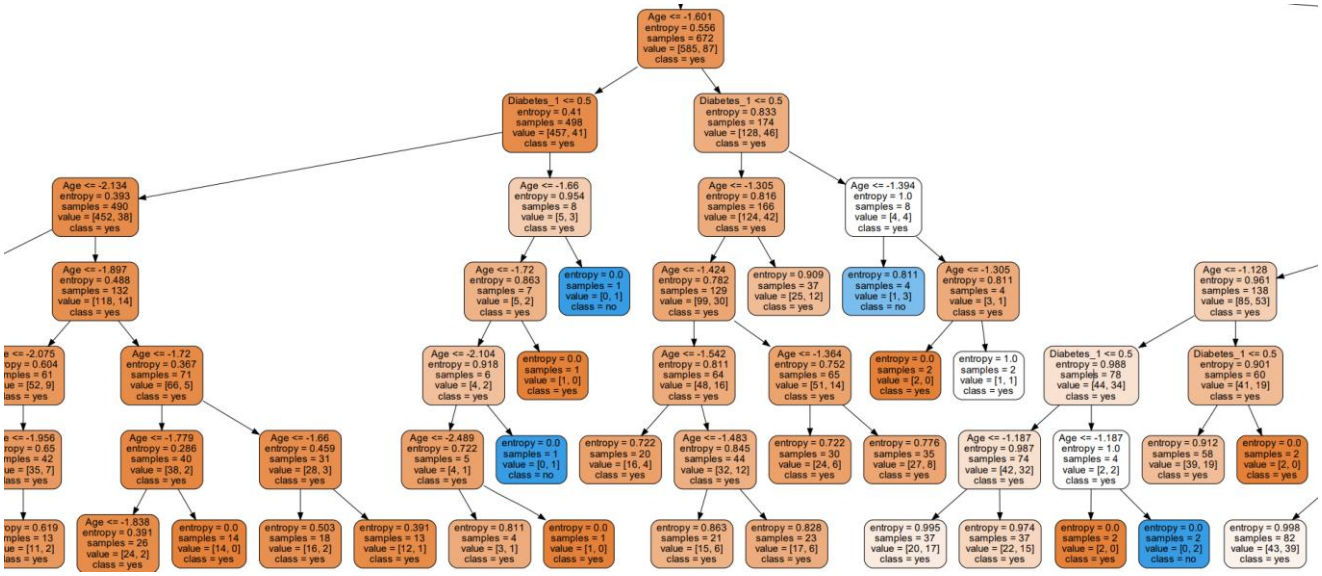


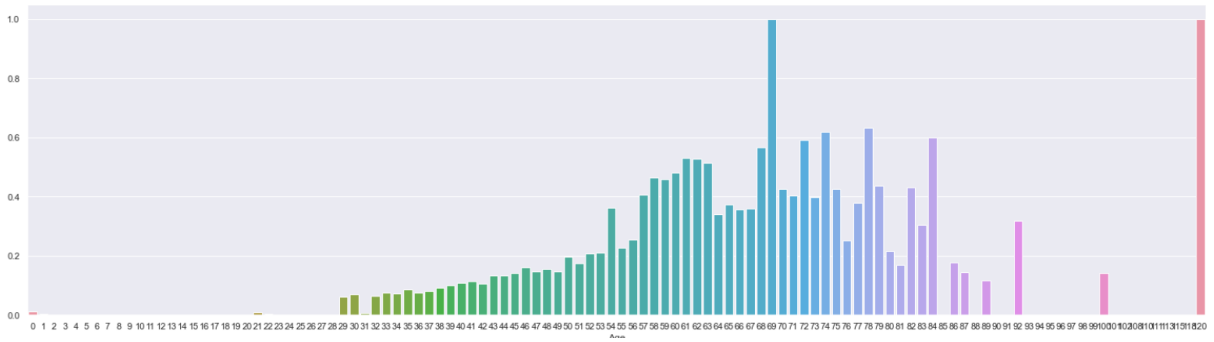

```
[[38998  8706]
 [20227 10492]]
              precision    recall  f1-score   support

           0       0.66      0.82      0.73     47704
           1       0.55      0.34      0.42     30719

    accuracy                           0.63     78423
   macro avg       0.60      0.58      0.57     78423
weighted avg       0.61      0.63      0.61     78423
```

Fig6. Our Naïve Bayes from scratch, performs as well as the sk-learn version.

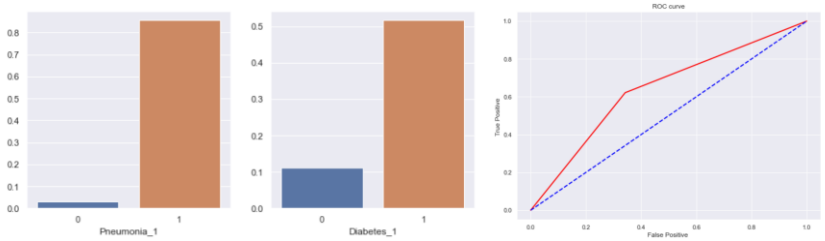
4.4 Decision Tree

For decision tree, we also provide 2 versions, from scratch and from sklearn. Decision tree is a machine learning algorithm that divides dataset into a tree. In ID3 and C4.5, It branches according to the information entropy of each tag. In CART, it branches according to Gini impurity.





This distribution looks good because of the upward and downward trends is similar to our dataset.




```
[[42662  5042]
 [22912  7807]]
              precision    recall  f1-score   support

           0       0.65      0.89      0.75     47704
           1       0.61      0.25      0.36     30719

    accuracy                           0.64     78423
   macro avg       0.63      0.57      0.56     78423
weighted avg       0.63      0.64      0.60     78423
```

Fig7. Our Decision Tree Performance

The scratch version implements a C4.5-liked algorithm, but it does not includes pre-pruning(except the limitation of the max depth) and post-pruning. Because of this, the scratch version has a poor behavior on our test dataset.

4.5 SVM

The SVM algorithm is similar to the logistic regression, that is to find a hyperplane in the 3-dimension space(in our dataset).

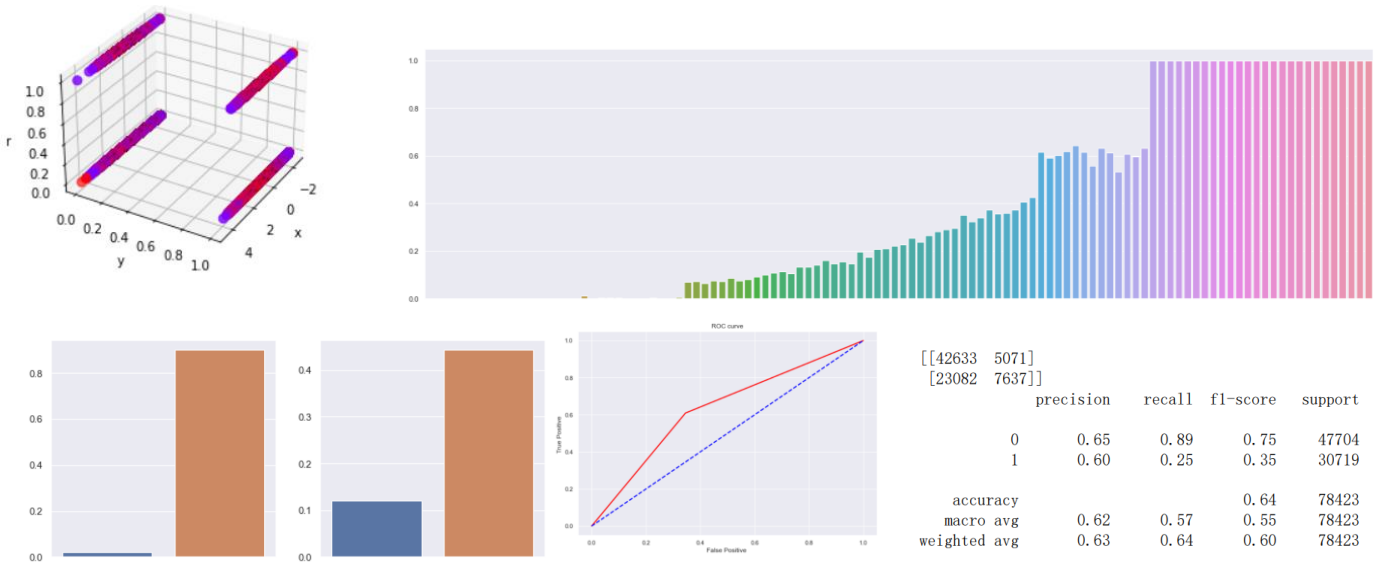


Fig8. SVM Performance

4.6 Neural Network

To implement this neural network, we uses Tensorflow and Keras to implement a sequential network with 2 hidden dense layers with ReLu and 1 output layer with sigmoid function.

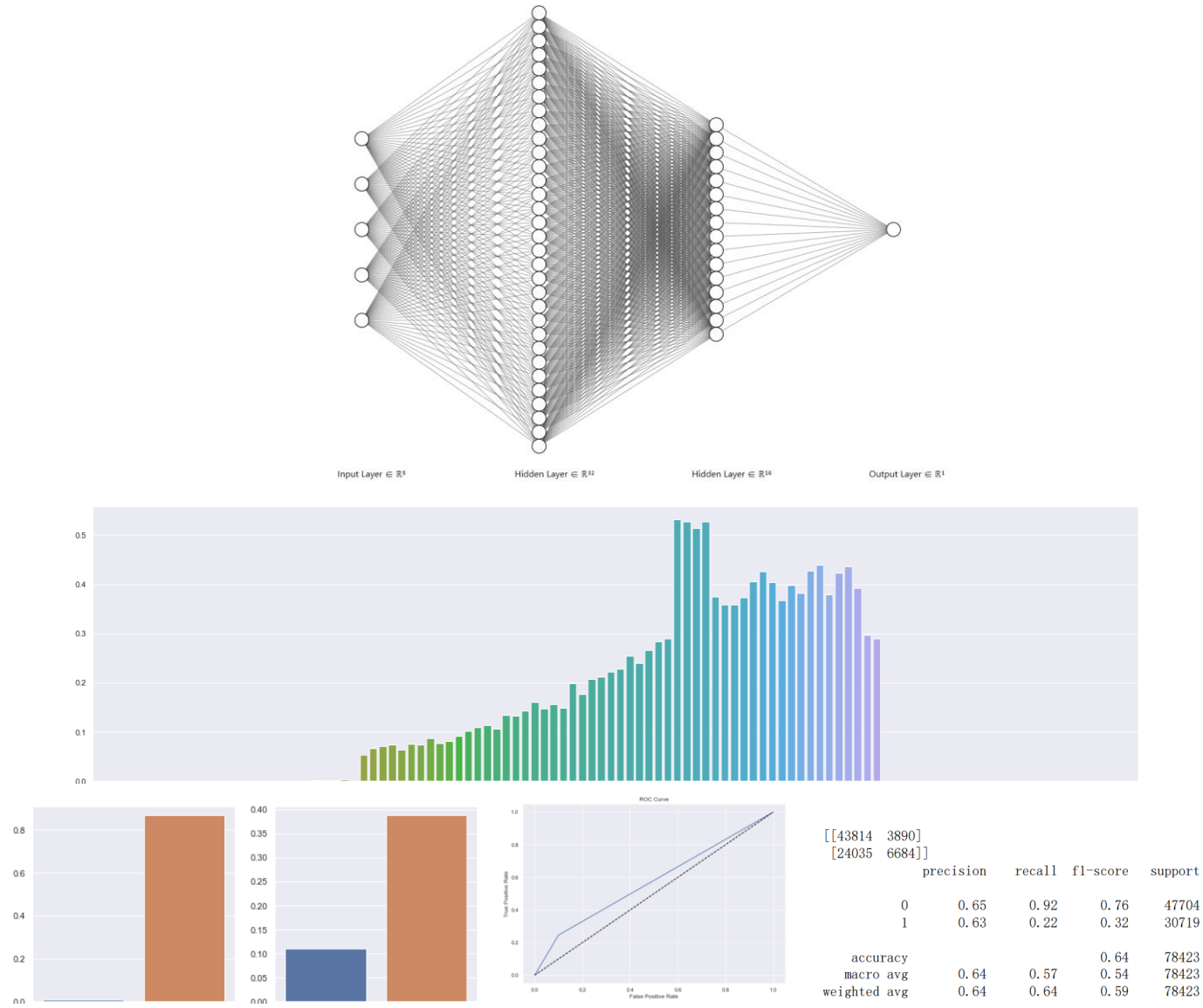


Fig9. neural network method's performance & network layer

5 Comparsion and Summary:

The list of the performance of every model algorithm in terms of accuracy score:

	Algorithms	Accuracy (%)
0	KNN(our)	59.031152
1	KNN(sklearn)	58.190000
2	Logistic Regression(Modified)	64.060000
3	Decision Tree(sklearn)	64.350000
4	Bayes Network(our)	63.097561
5	Bayes Network(sklearn)	63.100000
6	Support Vector Machine(sklearn)	64.100000
7	Neural Network(our)	64.398199

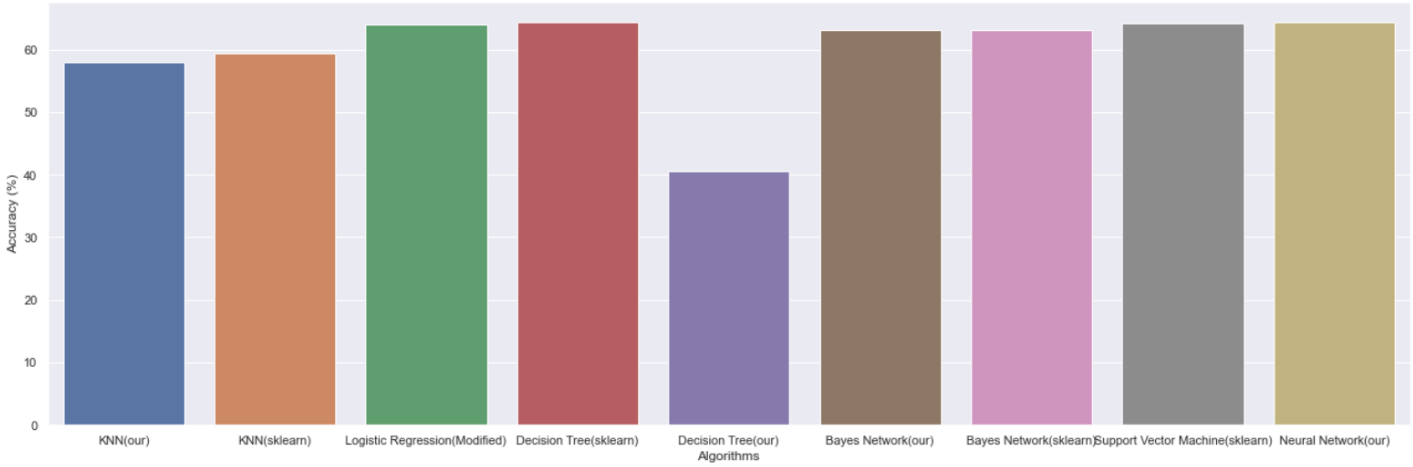


Fig11. A comparison of the performance of all the methods

After comprehensive consideration of accuracy and probability distribution accuracy, there are 3 well-bahaved algorithms: Naïve Bayes, Decision Tree(sklearn version) and Neural Network. Our Decision Tree is not satifying, comapered with sklearn's. However, our version of KNN and Beyes network implementation show high performance as well as the sk-learns, which is delightful for us.

The overall outcome of our work reached our initial plan and expectations. We enjoyed this journey. Thanks for reading!

Appendix

Related links about the dataset:

1.Mexico government's covid-19 dataset source:
https://www.gob.mx/salud/documentos/datos-abiertos-152127

2.WHO's medical instructions on this topic:
https://www.who.int/emergencies/diseases/novel-coronavirus-2019?adgroupsurvey={adgroupsurvey}&gclid=CjwKCAjwtIaVBhBkEiwAsr7-c7X93JRLY1MfND3xJKiTkiHkoslSXpbX-Ndem6Vk3XlRlwZjoyu0ghoCm94QAvD_BwE