Analysis on Covid-19 Diagnosis by Machine Learning

Lizhen Qiao
Department of Electrical and Computer Engineering
University of Toronto
lizhen.qiao@mail.utoronto.ca

Tianyu Hu
Department of Computer Science
University of Toronto
hutianyu@cs.toronto.edu

Yi Zhou
Department of Electrical and Computer Engineering
University of Toronto
alexy.zhou@mail.utoronto.ca

Abstract

In this report, we optimized six representative machine learning models to predict whether patients get SARS-Cov-2. As for the dataset, We chose the data which was released by the Israrli Ministry of Health which contained data of residents who were tested for SARS-CoV-2. In this research, we firstly implemented and optimized classifiers, then compared their performances from various perspectives. During the research we had several interesting findings and made several suggestions on diagnosing and predicting SARS-Cov-2, hoping our research could make some contribution to this epidemic and provide some basic ideas for other researches related to diseases diagnosis.

1 Introduction

The COVID-19 disease caused by SARS-CoV-2 has been seriously impacting people's lives and the global economy since the end of 2019. There have been more than 250 million confirmed cases worldwide and more than 5 million people died from it. Since the beginning of the pandemic, lots of efforts have been made to better diagnose, treat and prevent this disease. But still, in some countries or areas, it is hard to manage the spread of the disease.

While machine learning is an approach that has extensive applications in prediction[1-3]. It can be used to understand the nature of this virus and further predict the issues. We plan to study how machine learning algorithms can be deployed to fight the pandemic, implement mainstream machine learning algorithms and compare their prediction results.

2 Background and Related Work

There have been many prediction models developed to estimate the risk of infection based on various symptoms such as computer tomography (CT) scans[4] and laboratory tests[5], they can be helpful for effective screening and reducing the stress on healthcare systems. However, these models didn't perform well when patients were not hospitalized.

A recent study, with the Gradient Boosting based method[6], has shown that using simple binary criteria like sex, whether age is greater or equal to 60, and other 5 features can achieve relatively high accuracy on COVID-19 test results prediction, which further improves the effectivity in screening for SARS-CoV-2 in the general population.

3 Data

3.1 Data

The Israrli Ministry of Health released data of residents who were tested for SARS-CoV-2. Other than their test results, other information such as sex, age, various clinical symptoms was also available. Eight binary features were selected to develop models: 1. Sex (male/female), 2. Known contact with infected individuals (true/false), 3. Age above 60 (true/false), 4. Fever (true/false), 5. Headache (true/false), 6. Sore throat (true/false), 7. Cough (true/false) 8. Shortness of breath (true/false).

122000 out of 278849 records consist of completed information for all features we have chosen. 70% of the data was used as the training set, the rest was splitted in half as the validation set and the test set.

4 Model

4.1 K-Nearest Neighbours

4.1.1 Setup

K-Nearest Neighbours (K-NN) with L2 regularization was used. We implemented K-NN using Scikit-learn and used cross-validation to tune the K value.

4.1.2 Determining K

Cross-validation is a popular method for tuning hyperparameters and is mainly used in estimating how accurately a predictive model will perform in practice.

In addition, for k greater than 1, K-NN might encounter ties that need to be broken. However, since our output is binary (i.e. either yes or no), we decided to use odd number of K to avoid number of neighbours being tied.

We adopted the k-fold cross-validation (K = 10) approach. First, the whole dataset was split into 10 groups. Then each group was used once as a validation set while the 9 remaining groups formed as the training set. After iterating this process for K from 1 to 31 increment by 2, the validation accuracy for each K was averaged and the K with the highest accuracy was selected and used to predict the test set.

4.1.3 Result

We can observe from Figure 1 that K = 11 neighbours produced the best average accuracy across folds with 96.01%. The accuracies range is from 94.47% to 96.01% which clearly shows that K-NN is a good technique for this dataset.

4.2 Naive Bayes

4.2.1 Setup

There are three types of Naive Bayes models: Gaussian, Multinomial, and Bernoulli. We adopted Gaussian Naive Bayes which requires each of data point is mutually independent.

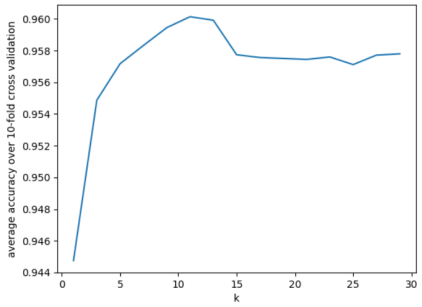

Figure 1: **Performances of KNN with different K value**

4.2.2 Result

The resulting accuracy of the Naive Bayes approach reached 93% with the Bernoulli distributions assumption.

4.3 Support-vector machine

4.3.1 Setup

Support-vector-machines are popular supervised learning models that perform well on classification and regression problems. They are able to find a hyperplane that separates two classes and maximises the gap between the two classes. Other than performing linear classification, we also tried different kernel types that can express input data into higher dimensional feature spaces.

4.3.2 Finding Optimal Parameters

We used GridSearchCV from scikit-learn to find the optimal parameters for the estimator. This approach checks exhaustively all possible combinations of parameters and outputs the parameters that maximize the score. 15% of the data was used as the validation set for this process. The result showed that using a radial basis function kernel with a classifier margin of 0.1 and gamma of 3.0 could yield the best accuracy score.

4.4 Logistic Regression

4.4.1 Setup

Decision Tree is a widely used supervised learning method used for classification and regression. It is simple to interpret and understand and the cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. In this chapter, we built a decision tree with chosen parameters and applied it with boosting and bagging to further optimize.

4.4.2 Result

Accuracy of logistic regression reached 95.38%. Furthermore, we believed that different symptoms and features should have different weights for diagnosing Covid-19. Therefore, after accomplished logistic regression, we normalized all eight features, printed out the weights of each feature for our trained model and got the importance sequence of all eight features in the dataset as below. According to the result, sore throat, shortness of breath and headache had the top3 biggest influences on whether a patient got Covid-19. Based on the statistical result of our logistic regression, we would strongly recommend hospitals to pay more attention to these three symptoms.

feature name	weight
cough	-0.62
fever	2.22
sore throat	5.70
shortness of breath	6.17
head ache	6.89
age 60 and above	-0.41
gender	-0.53

Table 1: feature weights of Logistic Regression Model

4.5 Decision Tree

4.5.1 Setup

Decision Tree is a widely used supervised learning method used for classification and regression. It is simple to interpret and understand and the cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree. In this chapter, we built a decision tree with chosen parameters and applied it with boosting and bagging to further optimize.

4.5.2 Optimization

We scanned different split criteria, tree depth and other main parameters of our decision tree model and dataset, eventually got optimal parameters where split criteria was chosen to be "Gini impurity" and depth was optimized to be 11. Furthermore, we employed our decision tree model to Boosting and Bagging to see if we could get even better performance. However, the result showed that both Boosting and Bagging did not bring a prominent improvement for our model.

4.5.3 Result

With the optimized parameters chosen above, our decision tree model's accuracy reached 96.19%.

4.6 Multi-Layer Perceptron Neural Network

4.6.1 Setup

We designed and implemented a fully connected network with two output classes. The input layer has one unit for each feature we selected. Our model was built using the PyTorch Neural Network library. Only linear transformation and Rectified Linear Unit were applied in each layer, and mean square error was used to evaluate training loss at each epoch.

4.6.2 Finding Optimal Hyperparameters

A threshold of 0.01% was set to check whether the training loss has converged. Since the input dimension of our data is not large, one hidden layer is sufficient to generate accurate predictions. The size of the hidden layer was set to be the squared root of the product of the input and output neurons [7], which is 4 in our case. A learning rate of 0.131 was selected so that the training loss can converge quickly and avoid overfitting.

4.6.3 Result

Our MLP model reached 96.11% accuracy within 100 epochs.

5 Model Comparison

So far we have accomplished and optimized six classification models. Because of the differences on mathematical implementation and the distribution of our dataset, they have very different performances on our dataset. In this section, we compared these models and tried to explain part of the reasons of their performances on our dataset.

We firstly calculated accuracy rate, precision rate and recall rate. Please note that output of our classification is binary, so the accuracy rate would always be equal to recall rate.

Method	accuracy	precision	recall
bayes	93.79%	94.74%	93.79%
decision tree	96.14%	95.87%	96.14%
knn	95.73%	95.40%	95.73%
logistic regression	95.56%	95.21%	95.56%
neural network	96.03%	95.76%	96.03%
svm	96.27%	96.02%	96.27%

Table 2: performance of each model

Apart from accuracy rates, we also evaluated classification models from aspects of time complexity.

Method	Train	Test	Total
bayes	0.0176	0.005	0.0226
decision tree	0.036	0.003	0.039
knn	7.611	16.685	24.297
logistic regression	0.181	0.002	0.183
neural network	14.67	1.55	16.22
svm	26.21	9.58	35.79

Table 3: time costing of each model (s)

From the result , we found that our SVM model got the highest accuracy rate and worst time cost. The Neural Network model also had relatively good accuracy but bad time cost. While the Decision Tree models have both good accuracy rates and time cost. Bayes, Logistic Regression and KNN models either had bad time cost or accuracy rates. The result mostly matched our expectations. Considering that even with this big number of samples, the test time is still less than 10 seconds, we believed that the time complexity of the models could be ignored. In this case, our SVM model was considered to be the best classifier among our six classification models.

Our SVM model had the highest accuracy and worst time cost on this dataset. This is reasonable considering the mathematical implementation of our SVM model and our dataset. In general, SVM is one of the most powerful classifiers when there is a clear margin of separation between classes which clearly fits our dataset. However, the disadvantages of our SVM model were also reflected in the result for costing the longest time. Though Support Vector Machines are powerful tools, their compute and storage requirements increase rapidly with the number of training vectors. The core of an SVM is a quadratic programming problem , separating support vectors from the rest of the training data. The QP solver scales between O(n$_{features} \times n^2_{samples}) and O(n_{features} \times n^3_{samples})$.

6 Conclusion

In conclusion, based on the dataset, we have developed six different machine learning models, including K-NN, Naive Bayes, Decision Tree, SVM, Neuron Network, and Logistic Regression, to predict COVID-19 diagnosis with eight different binary features.

As demonstrated in the model comparison section, It is clear that all our models work very well with the dataset since all their accuracy has achieved at least 93%, indicating the features we chose were suitable for the prediction. Additionally, SVM does seem to have the best performances on accuracy, precision, and recall.

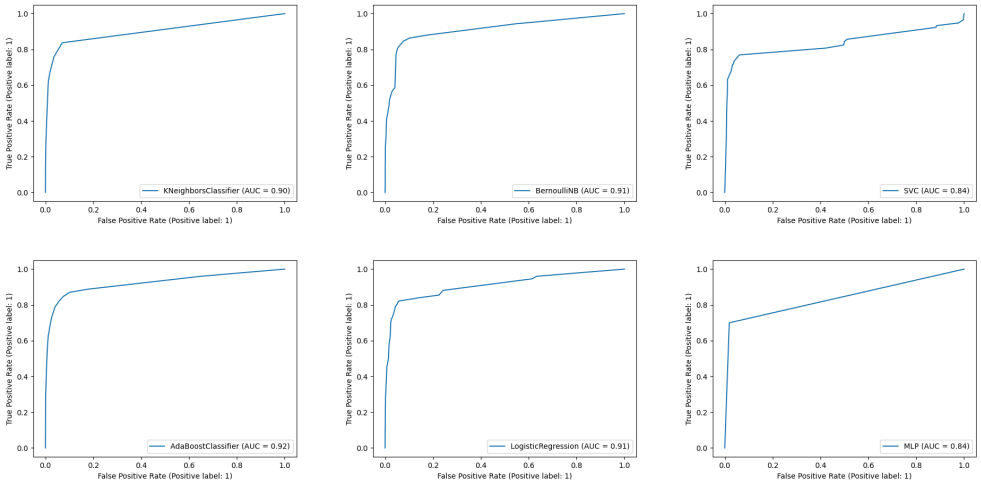

Figure 2: **ROC curve of models**

However, the weaknesses in SVM's computational complexity can not be ignored when it comes to a much larger data set, especially considering that our dataset only consisted of early patients of Israel, it would be reasonable to expect a much bigger dataset among the world later, plus the time complexity of SVM is not linear or logarithmic linear, this means our SVM model would not suffer dataset with that much data. Considering all indications in section 5, we believe that Decision Tree might be the best overall model when dealing with this dataset.

7 References

[1] Haleem A, Javaid M, Vaishya. Effects of COVID 19 pandemic in daily life. Curr Med Res Pract 2020. https://doi.org/10.1016/j.cmrp.2020.03.011.

[2] Bai HX, Hsieh B, Xiong Z, Halsey K, Choi JW, Tran TM, Pan I, Shi LB, Wang DC, Mei J, Jiang XL. Performance of radiologists in differentiating COVID-19 from viral pneumonia on chest CT. Radiology 2020. https://doi.org/10.1148/ radiol.2020200823.

[3] Hu Z, Ge Q, Jin L, Xiong M. Artificial intelligence forecasting of COVID-19 in China. arXiv preprint arXiv:2002.07112. 2020 Feb 17

[4]Wang, S. et al. A deep learning algorithm using CT images to screen for CoronaVirus Disease (COVID-19). medRxiv, https://doi.org/10.1101/2020.02.14.20023028 (2020).

[5]Feng, C. et al. A novel triage tool of artificial intelligence assisted diagnosis aid system for suspected COVID-19 pneumonia in fever clinics. medRxiv, https://doi.org/10.1101/2020.03.19.20039099 (2020).

[6]Zoabi, Y., Deri-Rozov, S. Shomron, N. Machine learning-based prediction of COVID-19 diagnosis based on symptoms. npj Digit. Med. 4, 3 (2021).

[7] K. Shibata and Y. Ikeda, "Effect of the number of hidden neurons on learning in large-scale layered neural networks," in Proceedings of the ICROS-SICE International Joint Conference 2009 (ICCAS-SICE '09), pp. –, August

8 Contributions

Lizhen Qiao	Logistic Regression, Decision Tree, MLP, Performance Evaluation of models
Tianyu Hu	SVM, Data-preprocessing
Yi Zhou	KNN, Bayes Classifier, ROC curve of models

Table 4: Contribution of Coding

Abstract	Lizhen Qiao
Introduction	Lizhen Qiao, Tianyu Hu, Yi Zhou
Prior Work	Lizhen Qiao, Tianyu Hu, Yi Zhou
KNN, Bayes Classifier	Yi Zhou
SVM, MLP	Tianyu Hu
Logistic Regression, Decision Tree	Lizhen Qiao
Model Comparison	Lizhen Qiao
Conclusion	Tianyu Hu, Yi Zhou

Table 5: Contribution of Report