A Mobile App collects and displays Daily COVID-19 and Flu Cases

Student: Xiangyang Xu
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, USA
xxu46@stevens.edu

Advisor: Professor Shucheng Yu
Department of Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, USA
syu19@stevens.ed

***Abstract*— Machine learning is widely used to do predictions and give suggestions. When enough data is available, supervised learning is a good way to build accuracy models. Covid-19 is one of the hottest topics this year and flu is also a problem that confused people for many years. This project is to develop an android app to predict the trend of flu and covid-19, also can give people suggestion whether they need to inject flu vaccine or not. All trend prediction and flu vaccine suggestions are based on their assemble maechine learning model. In this project, KNN, decision tree, and Multi-layer perceptron are mainly used to train models.**

Keywords— Covid-19, Flu, Android, Machine Learning, Deep learning

I. Problem Introduction, Challenges, and Related Work

A. Problem Introduction

From Dec.2019, the covid-19 spread from Wuhan to the world. Millions of people suffered from this new virus. At the same time, the economy of many regions stopped growing.

Machine learning and deep learning tech are widely used to help people to fight against this kind of new virus. For example, L. Mertz and his team developed AI-driven tools to quantify Lung Images.[1] At the same time, in people's daily life, many apps are developed for keeping people away from the Covid-19 virus. J. Berglund and his team developed software to track the virus.[2]

The epidemic has not been controlled so far and winter has come. Therefore, anticipation and early warning of Covid-19 are necessary. Use an app to help people away from Covid-19 seems necessary to work. If we can predict the possible outbreak of the epidemic, then forecast it to the public.

Currently, people use machine learning methods to predict the trend in the price of goods. Based on the history price, it is feasible to predict future prices. A. Yousefi and his team have used this tech to predict long-term electricity prices.[3] Therefore, it is also feasible to use machine learning to predict the future trend of covid-19 as well.

This technology can be also used to forecast flu. At the same time, machine learning models can be used to predict whether a person needs a flu vaccine shot or not.

In part I, this section introduces the project background, challenges meet in this project, and related work.

In part II, this section introduces the formal definition of the problem, including the framework of this project and the detailed parts of this problem.

In part III, this section introduces the algorithms which are used to make a machine learning model, android app (client-side) development, and server-side development.

In part IV, this section introduces the numerical solution of the machine learning part of this project and demonstrates the whole android app system including client-side and server-side.

In part V, this section summaries this paper.

B. Challenges

1) Data Collection and Procession

In the past, research project assignments are with datasets given by the professor. These datasets are all well designed for my projects, and no need to reprocess these datasets. This time one of the biggest problems is to get good datasets

2) Machine learning

Unsupervised learning has many kinds of algorithms, such as linear regression, support vector machines. however, different algorithms have different performance on a different task. Therefore, it's necessary to pick a good algorithm for the project.

3) Android App Development

A brief description of my project, this app should be a user-friendly manner. When this app is being designed, ease of use must be considered. So, there are many restrictions and requirements when designing apps.

C. Related Work

1) COVID-19 Future Forecasting Using Supervised Machine Learning Models[4]

Furqan Rustam and his team use the dataset provided by the Center for Systems Science and Engineering, Johns Hopkins University. This dataset includes features like latitude, longitude and the number of new inflect people.

They use 4 algorithms, Linear Regression, LASSO, Support Vector Machine, and Exponential Smoothing to do future covid-19 forecasting. They set 85% of data as training data and 15% as test data.

ES and LASSO method performs well in forecast new cases with a higher score and lower MSE. LR performs not bad and SVM cannot make a good prediction.

2) Prediction of Influenza and the Associated Pneumonia in Taiwan Using Machine Learning[5]

Sing-Ling Jhuo and his team use a different kind of dataset and algorithm to study the trend prediction of Influenza. In their dataset, there are a lot more kinds of features such as temperature, relative humidity, PM2.5, and CO. Then their study dataset has more features.

Then they use a deep learning method, Multilayer Perceptron (MLP), to make a model. Their study achieves 77.54% accuracy for the trend of influenza. The accuracy of the elder population is especially good. The accuracy gets 81.16%.

II. Formal Definition of the Problem

A. Overall Framework

An android can collect data input by the user, then a machine learning model can calculate the prediction of this data.

Then this app can display the prediction trend of covid-19 and flu or give app user suggestions whether he or she should take a flu vaccine shot or not.

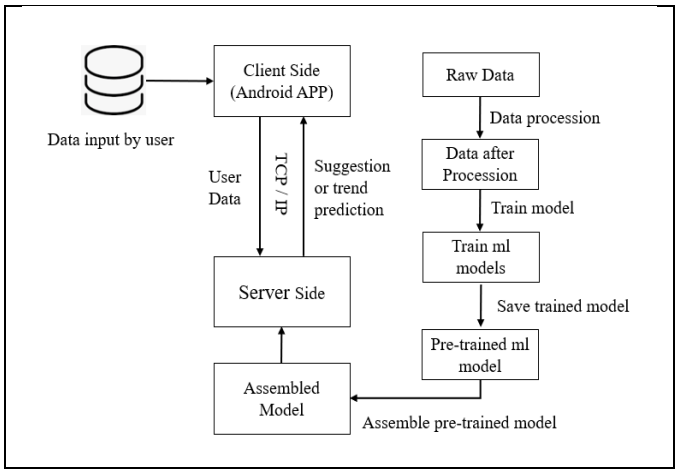


Fig. 1. Structure of Server Part and Client-Side

B. Data Procession Part

After collect datasets from the internet, make them become a suitable dataset for supervised learning. Make sure that they have suitable labels and their features.

For example, the raw data of flu trend prediction record flu cases happened in Australia. The focus labels are the numbers of new cases. Therefore, we need to process data, like adding cases that happen at the same time and same place to one row.

C. Machine Learning Part

Try different kinds of supervised machine learning algorithms to solve 3 problems, flu trend prediction, covid-19 trend prediction, and flu shoot suggestion.

Then, compare their accuracy, MSE, and other scores to choose a suitable model. Pick about 2~3 models as the final decision.

D. Andriod APP Development Part

The client part is an android app that collects the data input by the user and sends them to the server; finally, displays the solution given by Server.

The main part of the server part is assembling models based on the ml model picked at the machine learning part. The server part receives the data from the client-side and calculates them by machine learning models.

III. Description of the Solutions

A. Data Procession

1) Covid-19 trend prediction data procession

The raw datasets include us_counties_covid19_daily.csv (dataset 1 in the following part), us_county.csv (dataset 2 in following part), and Action.csv (dataset 3 in following part).

date	county	state	fips	cases	deaths
3/24/2020	Autauga	Alabama	1001	1	0
3/25/2020	Autauga	Alabama	1001	4	0
3/26/2020	Autauga	Alabama	1001	6	0
3/27/2020	Autauga	Alabama	1001	6	0
3/28/2020	Autauga	Alabama	1001	6	0

Fig. 2. Head of us_counties_covid19_daily.csv (dataset 1)

fips	county	state	state_cod	male	female	median_a	populatior	female_pe	lat	long
1001	Autauga C	Alabama	AL	26874	28326	37.8	55200	51.31522	32.53492	-86.6427
1003	Baldwin C	Alabama	AL	101188	106919	42.8	208107	51.37694	30.72748	-87.7226
1005	Barbour C	Alabama	AL	13697	12085	39.9	25782	46.87379	31.86958	-85.3932
1007	Bibb Coun	Alabama	AL	12152	10375	39.9	22527	46.05584	32.99863	-87.1265
1009	Blount Co	Alabama	AL	28434	29211	40.8	57645	50.67395	33.98087	-86.5674

Fig. 3. Head of us_county.csv (dataset 2)

State	stay at hor	>50 gather	>500 gath	public schc	restaurant	entertainn	Federal gu	foreign travel ban
AL	22-Mar	19-Mar	13-Mar	16-Mar	19-Mar		16-Mar	11-Mar
AK				19-Mar	17-Mar	17-Mar	16-Mar	11-Mar
AZ				16-Mar	20-Mar		16-Mar	11-Mar
AR				17-Mar	19-Mar	19-Mar	16-Mar	11-Mar
CA	19-Mar	19-Mar	19-Mar	19-Mar	15-Mar	15-Mar	16-Mar	11-Mar

Fig. 4. Head of Action.csv (dataset 3)

I set dataset 1 as the main dataset and I will merge necessary data from dataset 2 and dataset 3 to dataset 1. In dataset 1, there are two columns called fips and state, which will be used to merge dataset 2 and dataset 3. Fips is the code of county.

I use the dictionary method (which is a special kind of hash table) in python to merge data. In merging dataset 1 and dataset 2, I set fips in dataset 2 as key and other local information as value. Then we can add local data to dataset 1. For example, in the first row of dataset 1, fips is 1001, Then we can find the value of key 1001 in the dictionary including its state, population, and other elements. Then merge this to the first row of dataset 1.

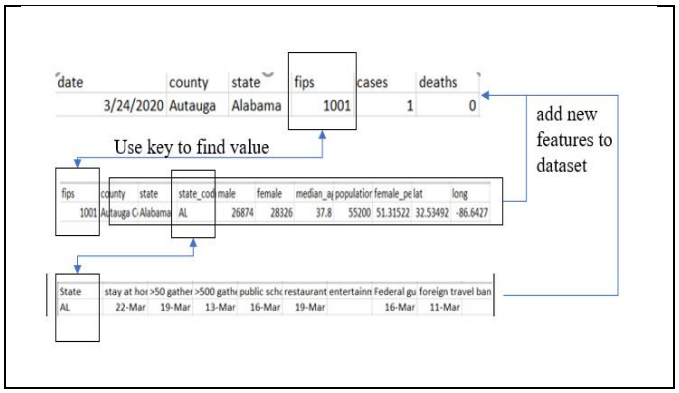


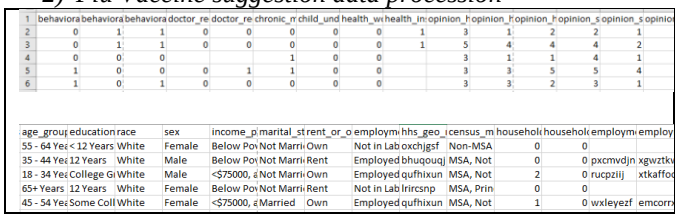
Fig. 5. Merge several datasets for the Covid-19 dataset

Use the dictionary method as before, I set state initial as key and other elements as value for merging dataset 1 and dataset 3.

Then, we need to make policy changes from date to yes or no. As we know the date and policy effective date. Therefore, it means the policy is effective that date is after the policy effective date, and I will record this as 1 and record policy no effective as 0.

Finally, drop the repeated data and unnecessary data to make the final dataset for study. Randomly set 70% of dataset as train data and other 30% of dataset as test data

2) Flu Vaccine suggestion data procession



	behaviora	behaviora	behaviora	doctor_re	doctor_re	chronic_m	child_und	health_wi	health_in	opinion_h	opinion_h	opinion_h	opinion_s	opinion_s	opinio
2	0	1	1	0	0	0	0	0	1	3	1	2	2	1	
3	0	1	1	0	0	0	0	0	1	5	4	4	4	2	
4	0	0	0			1	0	0		3	1	1	4	1	
5	1	0	0	0	1	1	0	0		3	3	5	5	4	
6	1	0	1	0	0	0	0	0		3	3	2	3	1	

age_grou	education	race	sex	income_p	marital_st	rent_or_o	employm	hhs_geo_i	census_m	household	household	employm	employ
55 - 64 Yea	< 12 Years	White	Female	Below Pov	Not Marri	Own	Not in Lab	oxchjgsf	Non-MSA	0	0		
35 - 44 Yea	12 Years	White	Male	Below Pov	Not Marri	Rent	Employed	bhuqouqj	MSA, Not	0	0	pxcmvdjn	xgwztku
18 - 34 Yea	College Gi	White	Male	<$75000, a	Not Marri	Own	Employed	qufhixun	MSA, Not	2	0	rucpziij	xtkaffoo
65+ Years	12 Years	White	Female	Below Pov	Not Marri	Rent	Not in Lab	lrircsnp	MSA, Prin	0	0		
45 - 54 Yea	Some Coll	White	Female	<$75000, a	Married	Own	Employed	qufhixun	MSA, Not	1	0	wxleyezf	emcorr

Fig. 6. Head of Flu Vaccine suggestion dataset (dataset 2)

Flu Vaccine dataset is a relatively mature dataset, no need to merge more data. However, there is some mysterious data in the dataset. For example, as in Fig.6. the data in the hhs_geo_region column is 'oxchjgsf' and other data cannot be understood. Therefore, I need to drop these mysterious data for further study.

Finally, after drop the mysterious data to make the final dataset for study. Randomly set 70% of dataset as train data and other 30% of dataset as test data

3) Flu trend prediction data procession

As is shown in Fig.7., the dataset record flu cases in Australia, but not how many new cases happen in a place. As I want to have a dataset to train the machine learning model to predict the trend of flu in a place, I need to change the dataset.

I set age 0-9 as group 1, 10-19 as group 2, 20-39 as group 3, 40~54 as group 4, and 55+ as group 5.

Then calculate the number of new cases in the same place, at the same time, and in the same age group to form the new dataset.

Week Ending (Friday)	State	Age group	Sex	Indigenous status	Type/Subtype
1/5/2018	NSW	00-04	Female	not available	B
1/5/2018	NSW	00-04	Female	not available	B
1/5/2018	NSW	00-04	Male	not available	A(unsubtyped)
1/5/2018	NSW	20-24	Male	not available	B
1/5/2018	NSW	25-29	Male	not available	A(unsubtyped)

Fig. 7. Head of Influenza (laboratory-confirmed) Public dataset 2008 to 2017 - Copy

Then as lack local information also no specific policy for flu. I search for the local population on the internet. Add the population to the dataset.

Finally, drop some unnecessary data. Randomly set 70% of dataset as train data and other 30% of dataset as test data.

B. Machine Learning Part

1) Instruction of Algorithms used in this Project

a) Multiple Linear Regression & Lasso Regression

Linear regression is a linear approach model the relationship between value (y_i) and one or more variables (x_{ij}). Their relation formulation i.e.

$$y_i = X_i^T w + \beta \tag{1}$$

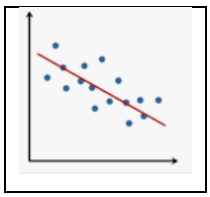


Fig. 8. Linear Regression Sample

MSE cost function of linear regression the average of the square distance between real value y and predict the value $\hat{y}$ i.e.

$$MSE = \frac{\sum_{i=1}^{n}(y-\hat{y})^2}{n} \tag{2}$$

The difference between Lasso regression and linear regression is their cost function. The cost function of Lasso regression adds a penalty equal to the absolute value of coefficients i.e.

$$Cost = \sum_{i=1}^{n}(y_i - \sum_j x_{ij} w_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3}$$

b) K-Nearest Neighbors Algorithm

Different from Linear regression, KNN is a non-parameter method machine learning algorithm. KNN algorithm is that if

most of the K nearest samples in the feature space of a sample belong to a certain category, the sample also belongs to this category and has the characteristics of the samples in this category.

c) Decision Tree Algorithm

Decision tree algorithm is a kind of predictive model. The decision tree algorithm uses a tree structure and uses layered inference to achieve the final classification.

d) Multi-layer Perceptron

A multilayer perceptron (MLP) is a kind of neural network, a mlp at least consist of 3 layers of nodes, an input layer, a hidden layer, and an output layer

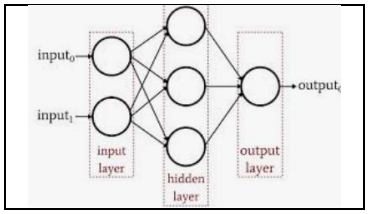


Fig. 9. Multi-Layer Perceptron Network Sample

Like other neural networks as CNN, MLP also needs activation function. In this project, the classification model use relu and thah as the activation function in hidden layers, and use the sigmoid activation function in the last layer and the regressive model uses linear activation function in the last layer.

e) Score Method

In this project, the F1 score and R2 score are mainly used for judging a model is good or not. F1 score is used for the classification model and R2 score is used for the regression model.

F1 score is a measure of a test's accuracy. It is calculated from the precision and recall of the test. It's more reasonable than an accuracy score for judging a classification model is good or not. For example, in a dataset that 10 people are suffering cancel and 90 people are healthy. A model that classifies all people healthy has a 0.9 accuracy score, it's unreasonable.to think that this model is good.

$$F1 = \frac{2*(precision*recall)}{(precision*recall)} = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \quad (4)$$

TP – Number of True positives
FP – Number of False positives
FN – Number of False negatives

R^2 score is the coefficient of determination, which is used for how good reflects the fluctuation of the dependent variable y (labels) what percentage can be used as the independent variable x (features). The higher the R^2 score, the better the model explains the dataset.

R2 score is equal to 1 - the sum of squares of residuals/total sum of squares. i.e.

$$R^2 = 1 - \frac{RSS}{TSS} \quad (5)$$

RSS - the sum of squares of residuals
TSS - the total sum of squares

$$RSS = \sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2 \quad (6)$$

$$TSS = \sum_{i=1}^{n}(y_i - \bar{y})^2 \quad (7)$$

n - number of samples
$f(x_i)$ - predicted value of y_i;
y_i - ith value of the variable to be predicted;
$\bar{y}$ - mean value of a sample

2) Covid-19 Trend Prediction

In Covid-19 trend prediction, I try linear regression, Lasso regression, decision tree, KNN, and multilayer perceptron.

At the first try of linear regression and Lasso regression, the R2 score is relatively low. So, I don't use these two models to solve this problem. Decision tree, KNN, and multilayer perceptron are promising models for solving this problem.

I use k-folder cross-validation to decide the max depth of the decision tree model; change the hyperparameter K to decide the neighbor number of the KNN model; build MLP neural network based on my experience, set epoch stop when loss almost not change.

Based on their R2 score, I set the weighted average to predict new cases of these three models as the final assemble model. i.e.

$$C_a = \frac{C_D*S_D+C_K*S_K+C_M*S_M}{S_D+S_K+S_M} \quad (8)$$

C_a – New cases predicted by assemble model
C_D - New cases predicted by the decision tree model
C_K - New cases predicted by the KNN model
C_M- New cases predicted by the MLP model
S_D – R2 score of the decision tree model
S_K - R2 score of the KNN model
S_M- R2 score of the MLP model

3) Flu Vaccine Suggestion

As linear regression and Lasso regression cannot be used to solve a classification problem, in Flu Vaccine suggestion models, I try KNN, Decision tree, and MLP algorithm. As the first try of these three models makes a good solution with a high F1 score, I try to study and improve all these three models.

As I have done in covid-19 trend prediction, k-folder cross-validation is used to decide the max depth of the decision tree model; try hyperparameter K to decide the best neighbor number of KNN model; build MLP neural network are built based on my experience.

As their average F1 score and accuracy score are very close, I use all of them to make flu vaccine suggestion.

The final score is equal to the sum of the prediction. 0 meant to suggest this person not to take the flu vaccine, 1 means suggest to this person take the flu vaccine. A

suggestion based on the final score. The higher the final score, the stronger suggestion will be given to the user.

TABLE I.

FLU VACCINE SUGGESTION

Final score	***Suggestion***
3	Very strongly suggest
2	Strongly suggest
1	Suggest
0	Not suggest

4) Flu Trend Prediction

In flu trend prediction, I try linear regression, Lasso regression, decision tree, KNN, and multilayer perceptron.

At the first try of linear regression and Lasso regression, the R2 score is very low. I will not use them to assemble the final model. I try MLP, however, the score is still not enough high. Only the KNN and Decision tree models are relatively high. These models are not so good as the former two functions, in my opinion mainly because data is very limited for machine learning.

Similar to covid-19 trend prediction, I use k-folder cross-validation to decide the max depth of the decision tree model; and try different hyperparameter K to decide the neighbor number of the KNN model.

Based on their R2 score, I set the weighted average to predict new cases of these three models as the final assemble model. i.e.

$$C_a = \frac{C_D * S_D + C_K * S_K}{S_D + S_K} \quad (9)$$

C_a – New cases predicted by assemble model
C_D - New cases predicted by the decision tree model
C_K - New cases predicted by the KNN model
S_D – R2 score of the decision tree model
S_K- R2 score of the KNN model

C. Andriod APP Development Part

1) Client Part

The main function of the client part is to let the user input data such as his/her age and other easy-to-get information. Save this information to let the app autofill if the user uses this app next time.

On the first page of this app, let the user choose the function he or she wants to use, including flu trend prediction, covid-19 trend prediction, and flu vaccine suggestion. On the next page, the user will input his or her data.

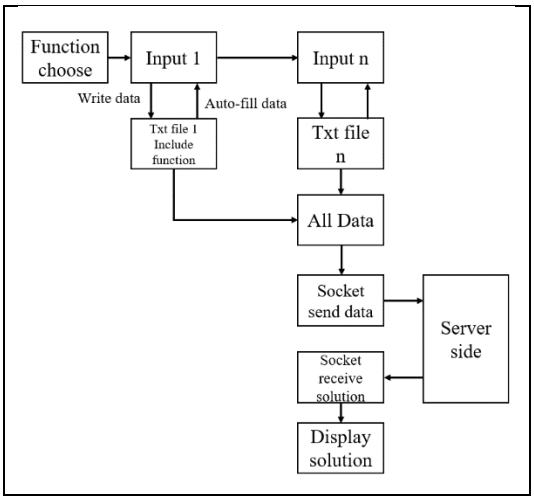


Fig. 10. Function diagram of client-side

As the requirement of this app is to let the user be easy to use. I delete some data columns which may let users hard to use and this leads the score of models to decrease a little. After the user input all data, this client will send all data to the server-side with the first element of data is the function name. Other elements are data that will be used to make the prediction. The final part of the app is to display the received suggestion from the server-side.

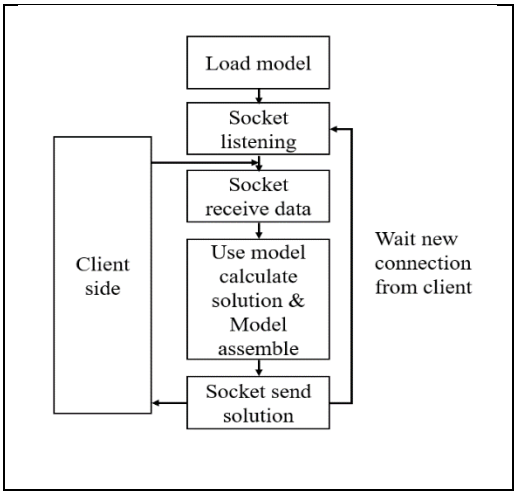


Fig. 11. Function diagram of server-side

2) Server Part

Before the server-side runs, it loads all 8 models first. Then the client will judge which models will be used to predict. After it gets the prediction and assembles the prediction of 2~3 models, a final suggestion will be sent to the client-side.

3) TCP/IP (Socket Part)

Server-side and Client part connect by TCP/IP (socket), server-side will be started first to listen and accept socket from client-side. Next, the server-side receives the data from the client-side. After machine learning models make a prediction,

the server-side sends a prediction or suggestion to the client-side.

IV. Numerical Results and Analysis

1) Covid-19 Trend Prediction Model

I use k-fold cross-validation to decide the max depth of the Decision tree model for covid-19 prediction. As Fig.13 shown, the mistake is smallest when the max tree size is 6, therefore I choose 6 as the max depth of decision tree models

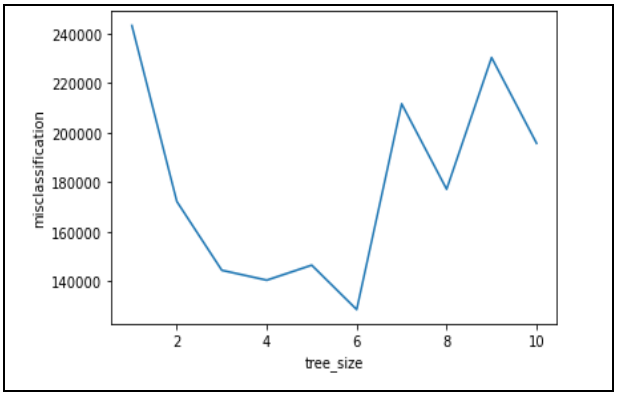


Fig. 12. k-fold cross-validation of Covid-19 decision tree model

To the KNN model, I change the hyperparameter k from low to high. In the case of the test score as high as possible, choose the most appropriate K neighbor number. In the prediction covid-19 model, I pick 7 as the number of neighbors.

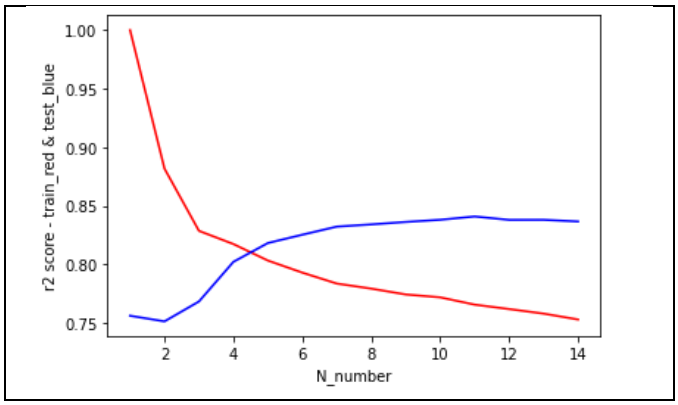


Fig. 13. Change Hyperparameter Neighbors Number and score

As table II shows, Linear regression & Lasso regression have 0.76 R2 scores, lower than decision tree 0.78, KNN 0.83, and MLP 0.78 R2 score

TABLE II.

Covid Trend Prediction Score

Algorithm	**Score**	
	R2 Score	***MSE Score***
Linear Regression	0.763	1034.628
Lasso Regression	0.763	1037.829
Decision Tree	0.782	953.66
KNN	0.832	734.83
MLP	0.789	922.74

Also, Decision tree, KNN, and MLP have better performance at MSE score.

The final model is assembled by 2~3 models, so I picked KNN, MLP & Decision tree to assemble the model on the server-side.

2) Flu Prediction Model

K-fold cross-validation is used for the flu trend model as well. As the mistake of flu trend is almost the same when max depth more than 21; therefore, to let server-side calculate quicker, I choose 21 as tree size. Max depth of decision tree here is 21

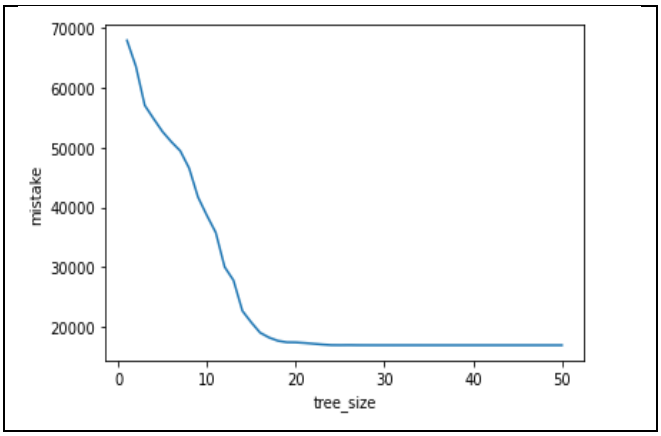


Fig. 14. k-fold cross-validation of flu trend decision tree model

I use changing the hyperparameter k from low to high in this model as well. In the case of the test score as high as possible. As the score of test prediction become lower after N number is more than 4, I set N number is 4 for this function

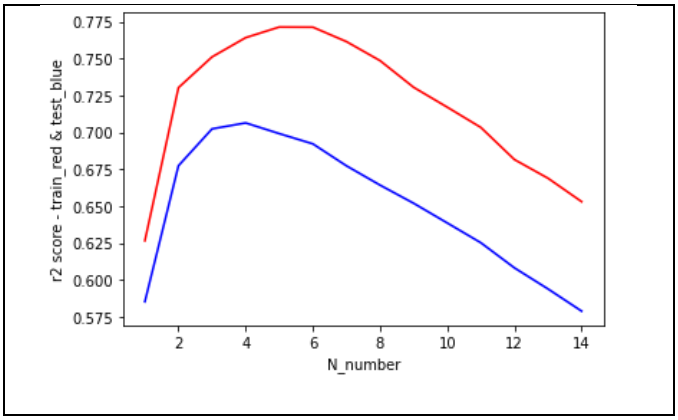


Fig. 15. Change Hyperparameter Neighbors Number and score

TABLE III.

FLU TREND PREDICTION SCORE

Algorithm	**Score**	
	R2 Score	***MSE Score***
Linear Regression	0.046	21423.66
Lasso Regression	0.046	21423.57
Decision Tree	0.763	5318.29
KNN	0.706	7250.93
MLP	0.257	16689.06

As table III shown, Linear regression & Lasso regression have 0.04 R2 score, MLP 0.25 R2 score which is lower than decision tree 0.74 and KNN 0.70

Also, Decision tree and KNN have better performance at MSE score.

The final model is assembled by 2~3 models, so I picked KNN & Decision tree to assemble the model on the server-side.

3) Flu Vaccine Suggestion Model

I use k-fold cross-validation to decide the max depth of the decision tree model for flu vaccine shot suggestion as well. the mistake is smallest when the max tree size is 5, therefore I choose 5 as the max depth of decision tree models

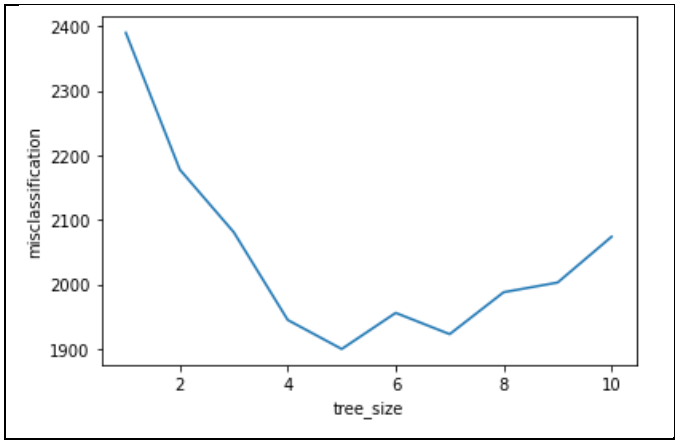


Fig. 16. K-fold Cross-validation of flu vaccine suggestion decision tree model

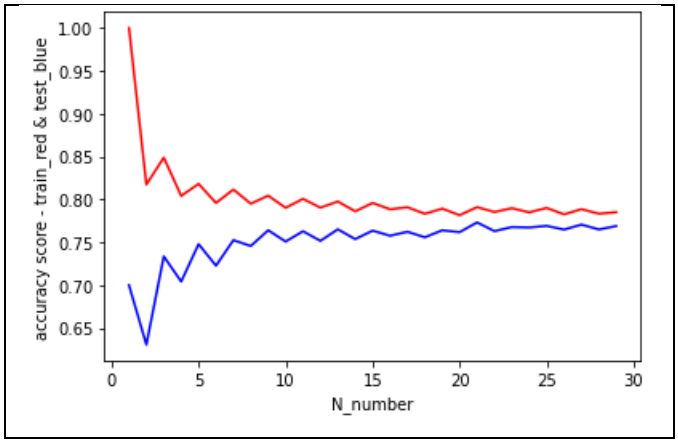


Fig. 17. Change Hyperparameter Neighbors Number and score

As the test score does almost not change when the N number is more than 18, I pick 20 as the number of neighbors.

TABLE IV.

FLU SUGGESTION PREDICTION SCORE

Algorithm	**Score**	
	F1 Score	***Accuracy Score***
Decision Tree	0.765	0.766
KNN	0.785	0.771
MLP	0.796	0.796

As table IV shows, Decision tree classification, KNN classification, and MLP classification are all have similar F1 score and Accuracy Score.

Therefore, I pick all of them to assemble the final models.

The suggestion will be given based on the total score of the model solution.

B. Client Part (Andriod APP)

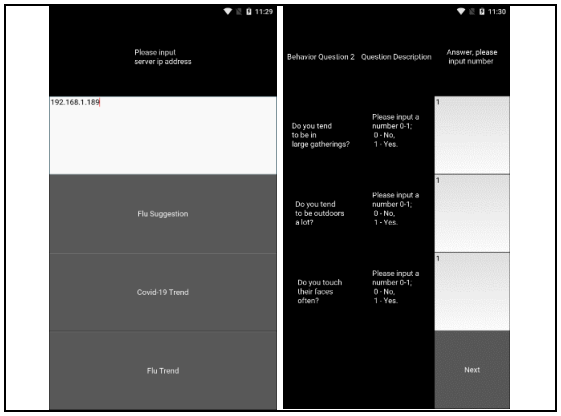


Fig. 18. Function choose Page and one of the Data input Pages

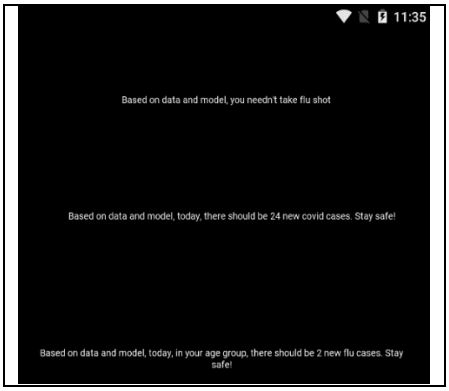


Fig. 19. Merged received Suggestion from Server-side.

Users pick a function first, then input their data in one or more data input pages. These data will be saved as one or more txt files for auto-input next time. When the user completes the last input page, and tap the 'Next' button, data will be sent to the server-side. After the client-side receive the suggestion or

trend prediction from the server-side, the client-side will display it.

C. Server Part

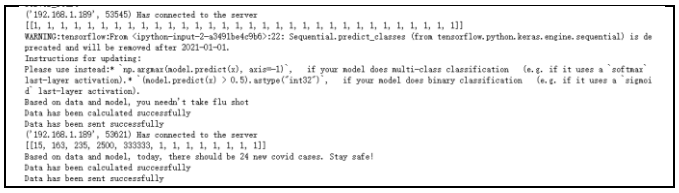


Fig. 20. The Server received Data from the Andriod APP

After the client-side is linked with the server-side, the server-side will judge the function client-side ask, then calculate based on machine learning and give the final solution. Finally, send a message to the client-side.

V. Conclusion

In this paper, I design 3 functions of an android app, forecast covid-19 trend, forecast flu trend, and give flu vaccine shot suggestion.

The user of this android app just needs to do very easy steps, choose the function they want to use, input data that are very easy to get by them, and read the suggestion of flu vaccine, flu or covid-19 trend forecast from the server-side. All calculation work is done at server-side, which means it's easier for the developer to update better models. Unless the dataset has more data columns to learn, the client-side no need to be updated.

The machine learning parts mainly use decision tree and KNN algorithms as their score is higher than linear regression and Lasso regression. MLP is also a good deep learning method to analyze a good dataset.

As the covid-19 trend forecast score is relatively higher than the other two functions, the performance of the covid-19 trend forecast is the best in this app. Performance of flu trend prediction is the worst, mainly because of the dataset of flu it the worst among these three functions.

References

[1] L. Mertz, "AI-Driven COVID-19 Tools to Interpret, Quantify Lung Images," in IEEE Pulse, vol. 11, no. 4, pp. 2-7, July-Aug. 2020, doi: 10.1109/MPULS.2020.3008354.

[2] [J. Berglund, "Tracking COVID-19: There's an App for That," in IEEE Pulse, vol. 11, no. 4, pp. 14-17, July-Aug. 2020, doi: 10.1109/MPULS.2020.3008356.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] A. Yousefi, O. A. Sianaki and D. Sharafi, "Long-Term Electricity Price Forecast Using Machine Learning Techniques," 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), Chengdu, China, 2019, pp. 2909-2913, doi: 10.1109/ISGT-Asia.2019.8881604.

[4] F. Rustam et al., "COVID-19 Future Forecasting Using Supervised Machine Learning Models," in IEEE Access, vol. 8, pp. 101489-101499, 2020, doi: 10.1109/ACCESS.2020.2997311.

[5] S. Jhuo, M. -. Hsieh, T. Weng, M. Chen, C. Yang and C. Yeh, "Trend Prediction of Influenza and the Associated Pneumonia in Taiwan Using Machine Learning," 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Taipei, Taiwan, 2019, pp. 1-2, doi: 10.1109/ISPACS48206.2019.8986244.