

**Using Statistical Regression and Classification Methods to Predict
Human Body Temperature and Presence of Fever**

Ziqi Zhang

Supervisor: Dr. Hong Gu

Data courtesy of: Phil Munz

Department of Mathematics and Statistics, Dalhousie University

1 Introduction

Due to the COVID-19 pandemic, more and more people prefer to keep a safe social distance. Taking people's temperature is an important thing to do when they enter public places. This is mainly because one of the COVID-19 symptoms is the presence of a fever. Even with a non-contact thermometer, the staff on-site need to be close to many people for reading temperatures every day. In order to improve the efficiency and safety of the measurement, we conduct a research here to see if a thermal camera can replace the thermometer. The thermal camera Axis Q2901-E can detect temperatures from -40°C to 550°C . It is useful to check for temperatures in this range or for excessive temperature changes. It is not intended for checking body temperatures; however it could be used as a tool when combined with Machine Learning or Artificial Intelligence techniques for automating temperature differences between people. We will use the images to find some potential variables to create a suitable model.

Thermal imaging technology has application potential. It allows for very safe non-contact measurements. This is one of the main advantages, with the possibility to evaluate the average skin surface temperature as well (Bernard et al., 2012). The glasses, hat, and long hair in the forehead area will influence the thermal camera temperature measurement (Ladislav & Veronika, 2021). Under these circumstances, we will filter out the images with these factors.

Consequently, our goal is to build the model from two aspects, one is a single prediction of whether a person has a fever or not, and the other is trying to predict a

specific temperature value. We will use python to convert the photos into numerical format and then use R to create and predict and compare logistic regression, linear regression, random forest classification and random forest regression models.

2 Data

2.1 Data Collection

The data was collected by a third-party resource called rProcess in collaboration with local hospitals. Data were collected during the pre-screening of hospital visitors before they enter the hospital, and visitors can choose whether or not to have their data collected as part of the data collection. Simultaneously, rProcess company used a thermal camera Axis Q2901-E model with a 9mm lens to obtain thermal imagery and video. The camera is connected to a laptop, and the live video stream from the thermal camera is transmitted to the laptop directly. For simplicity, Patriot One company created a python script in-house with a GUI to take snapshots of visitors and record temperatures. After collecting data through the day, rProcess provide metadata about the location of the face within the frame for each image and then sent to Patriot One company for quality check. The series of images in the data set is now obtained by Patriot One company after quality checking and cleaning abnormal data. Since wearing glasses and hats would obstruct the upper part of a person's face, any images including such objects were filtered out in the quality check.

2.2 Data Contents

The image data were obtained through the thermal camera. The data contained 1404 images for people with normal temperature (≤ 37 degrees Celsius) and 106 images for people with higher temperature (> 37 degrees Celsius).

As a result of the limitation of our data collection, with only 3 people with an actual fever, in our analysis we put the threshold of greater than 37.0 degree Celsius as fever,

Table 1. We labeled normal temperature 37.0 degrees Celsius and below.

Thermal Temperature Reading	number of people	label
34.1-35°C	7	Normal
35.1-36°C	155	Normal
36.1-37°C	1242	Normal
37.1-38°C	103	Fever
38°C and above	3	Fever

Table 1: Number of personnels and labels per temperature band

Due to COVID-19, all data were limited to those people who wore masks.

Temperatures were measured using non-contact thermometer readings taken from the middle of the forehead by field medical personnel. For each image, Patriot One company provided a corresponding text file, which include their body temperature and potential coordinates of the face area in the image. Body temperature used for later prediction. The coordinates of the face are prepared for the picture processing.

We used the “*OpenCV*” package in Python to process the data.

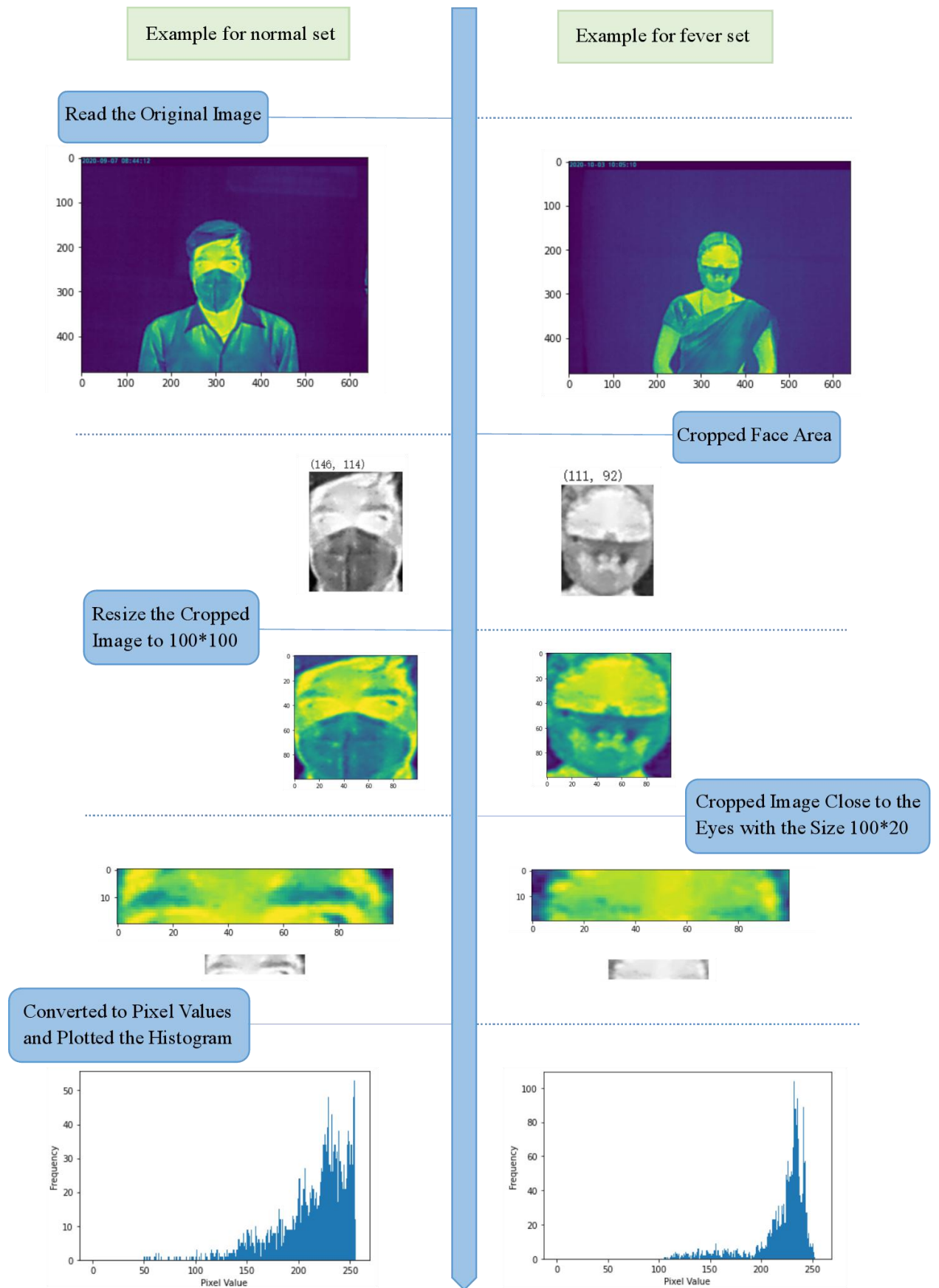


Figure 1. An example of processing images for normal set and fever set

The processing steps as shown in *Figure 1*. Firstly, we read in images and corresponding text files. Then we used coordinates to crop the subsections of the image, using only the face that shows skin (i.e. not the bottom part of the face that's wearing the mask). In order to facilitate automatic cropping and subsequent steps, we reset the size of all cropped images to 100*100. Our cropped image is to crop the line close to the eyes. Each cropped image is 100*20, with a total of 2000 pixel values. Next, we converted each image to the greyscale numerical values. Greyscale values are also called pixel values. The range of pixel values is always from 0 to 255. If the area in the image is brighter, then the temperature of that area is higher. Its pixel number is positively correlated to its temperature. When the area comes black, the pixel number is approximately 0. Finally, we generated the output “.csv” file. We extracted each pixel value of each cropped image and the temperature they correspond to.

2.3 Data Partition and Information Extraction

There are 1404 observations for normal people set (called normal set) and 106 observations for fever people set (called fever set). Since the number of the fever set and the normal set is unbalanced, our direct random sampling ratio is not accurate. To avoid bias, we separated the fever set and the normal set first. In each of these two groups, 80% of the people were randomly selected as the training dataset and the other 20% as the testing dataset.

We wanted to determine if we could predict both the true body temperature and detect whether the person had a fever. We first drew the survival curve for the pixel values of

each data point. It was observed that the difference of pixel value distribution between data exists when the pixel value is greater than 50. For this reason, we selected a pixel point range of 50 to 250 for information extraction (*Figure 2*), with 25 pixel value per interval (e.g., SX1 means the survival curve value at 50 pixel value, SX2 means the survival curve value at 75 pixel value, and SX8 means survival curve value at 250 pixel value). Since the points from SX1 to SX8 are very irregular, we have used the Gaussian inverse method to transform each survival curve point. The transformed variables are named TSX1 to TSX8.

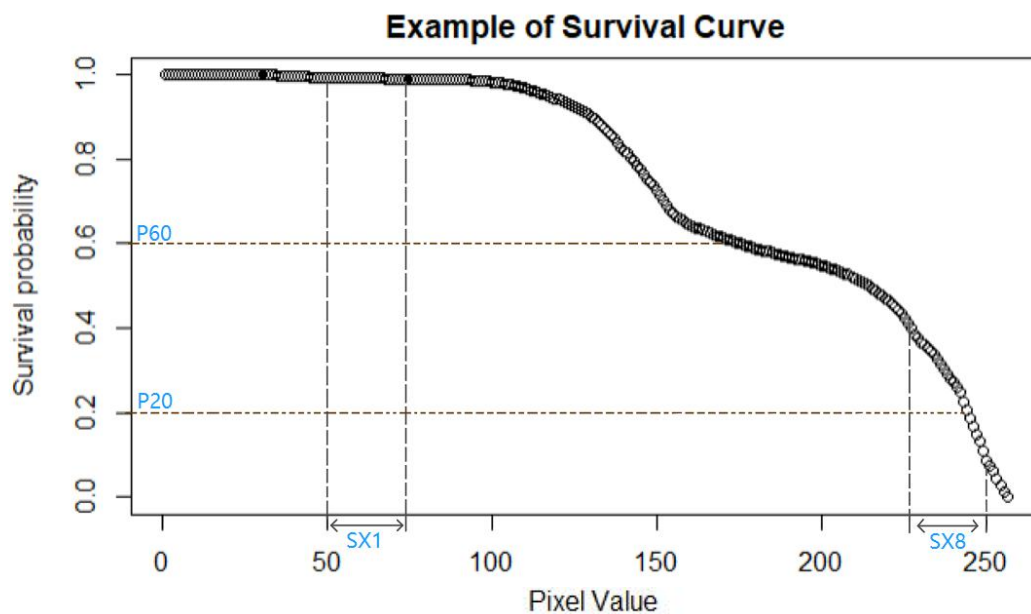


Figure 2. An example of how to set variables for each image

In addition, we also extracted different percentile values as potential useful variables.

We extract a variable every 10th percentile. For example, P10 means there are 10% of pixel values less than this value. We found that the 90th percentile shows more diversity in the data, so we added more variables from 98% to 80% at 2% intervals (i.e., P98, P96, P94). In total, there are 25 variables and two response variables in our project, detailed in *Table 2*.

Variables Name	Actual Meaning
temp	Real temperature value (unit: “degree Celsius”)
fever	“F” for fever set, “N” for normal set
P10 to P98	The percentile of the brightness (possible value from 0 to 255)
TSX1 to TSX8	After Gaussian inverse for the height of survival curve

Table 2. Variables used in the project and their actual meaning

3 Method

We used statistical learning methods to predict the temperatures in two ways. One is based on the real temperature and the other one is based on whether or not a fever is present. We used logistic regression method and random forest for classification technique. Since our sample set is unbalanced, we need to add weights to the minority set (fever set) when doing classification analysis.

3.1 Logistic Regression

The logistic regression model involves a multiple regression relationship between multiple independent variables and a dependent variable to predict the probability of events occurring. The outcome of logistic regression fell between 0 to 1, which is a discrete response value. Logistic regression can demonstrate which of the various variables to be evaluated has the strongest relationship with the outcome and provides a way to measure the degree of potential impact. It also provides both the coefficient size and direction of the association. Y is the estimated probability of a binary category outcome (0 or 1). The probability of Y given experimental data is:

$$P(Y | X_i) = \begin{cases} p, & Y = 1 \\ 1 - p, & Y = 0 \end{cases}$$

Logistic regression function is defined as follows:

$$P_{Y=1} = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i}} = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i)}}, i \leq 25$$

The components in this function are as follows:

- 1) X_1, X_2, \dots, X_{25} is the independent variables (i.e., P10-P98, TSX1-TSX8);
- 2) $\beta_0, \beta_1, \dots, \beta_i \in \mathbb{R}$ is the regression coefficient, which β_0 is the intercept;
- 3) $e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i}$ represents the variable expressed in the logic scale by using a linear regression equation, and it must be positive;
- 4) The theoretical model $P_{Y=1}$ has a slender sigmoidal shape with asymptotes between 0 and 1.

In order to improve the accuracy, we mainly looked at the AIC(Akaike Information Criterion) values and ROC(Receiver Operating Characteristic) curve for the logistic regression.

AIC (Akaike, 1974) is used to measure the goodness of fit for the statistical model to avoid over-fitting. Lower AIC value indicates a better fit. Therefore, AIC provides a means to compare possible models. The basic formula is defined as:

$$AIC = -2 \log(L) + 2K$$

Where K is denoted as the number of model parameters, and L means Log-likelihood function for measure of model fit. The fit of the model is proportional to the magnitude of L .

The ROC curves are graphical plots used for binary classifier to show the diagnostic capability of the binary classifier. The ROC curve plots the sensitivity and specificity of the model for each possible decision rule threshold between 0 and 1. The Area

Under the Curve (AUC) provides useful statistical information. The larger the area, the better the model. This provides information on the effectiveness of the model predictions and the best cut-off point to find the model. In R, use “*pROC*” package to draw ROC curve and get AUC value.

3.2 Linear Regression

Linear regression is used to build a linear model from historical data to predict future results. Linear regression modeling is fast and computationally uncomplicated. It will predict the value of a dependent variable Y based on multiple independent variables X_i . The function of the linear model is:

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_i X_i + \varepsilon, i \leq 25$$

Where, β_0 is the intercept and $\beta_1, \beta_2, \dots, \beta_i$ is the slopes, and ε is the residual.

We used value of AIC, Mean Square Error (MSE), Root Mean Squared Error (RMSE), and Adjusted R-square to find best linear regression model.

MSE measure how close the predicted value is to fitted line. The closer the value is to 0, the better the model is. Our goal is to find the appropriate $\beta_1, \beta_2, \dots, \beta_i$ to minimize the squared error. The function for MSE is:

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Where Y_i is the observed value, and \hat{Y}_i is predicted value. RMSE is the square root of MSE.

The adjusted R-squared is a parameter used to explain the power of the regression model. The adjusted R-squared value is in the interval between 0 and 1. The larger the number of adjusted R-squared, the better the model. The adjusted R-squared takes

into consideration the effect of sample size and the number of independent variables in the regression model, so the value of the adjusted R-squared does not get closer to 1 as the number of independent variables in the regression increases. The adjusted R-squared increases only when new predictor variables improve the model beyond what would be expected by chance.

3.3 Random Forest

Random forest uses decision trees as the base classifier, which was first proposed by Ho in 1995. Multiple decision trees are constructed using the Bagging algorithm, and the decision trees are constructed by randomly selecting the training set only. The random forest is easy to implement and can avoid over-fitting. In R, the “*randomForest*” function implements Breiman’s random forest algorithm.

3.3.1 Random Forest Classification

Random forest classification is a combination of many decision tree classification models, and each decision tree classification model has one vote to choose the best classification result. The basic steps of random forest classification (*Figure 3*):

- 1) Using bootstrap sampling to draw k samples from the original training set, and each sample has the same sample size as the original training set;
- 2) k decision tree models are built for each of the k samples to obtain k classification results;
- 3) Each record is voted on according to the k classification results to decide its final classification.

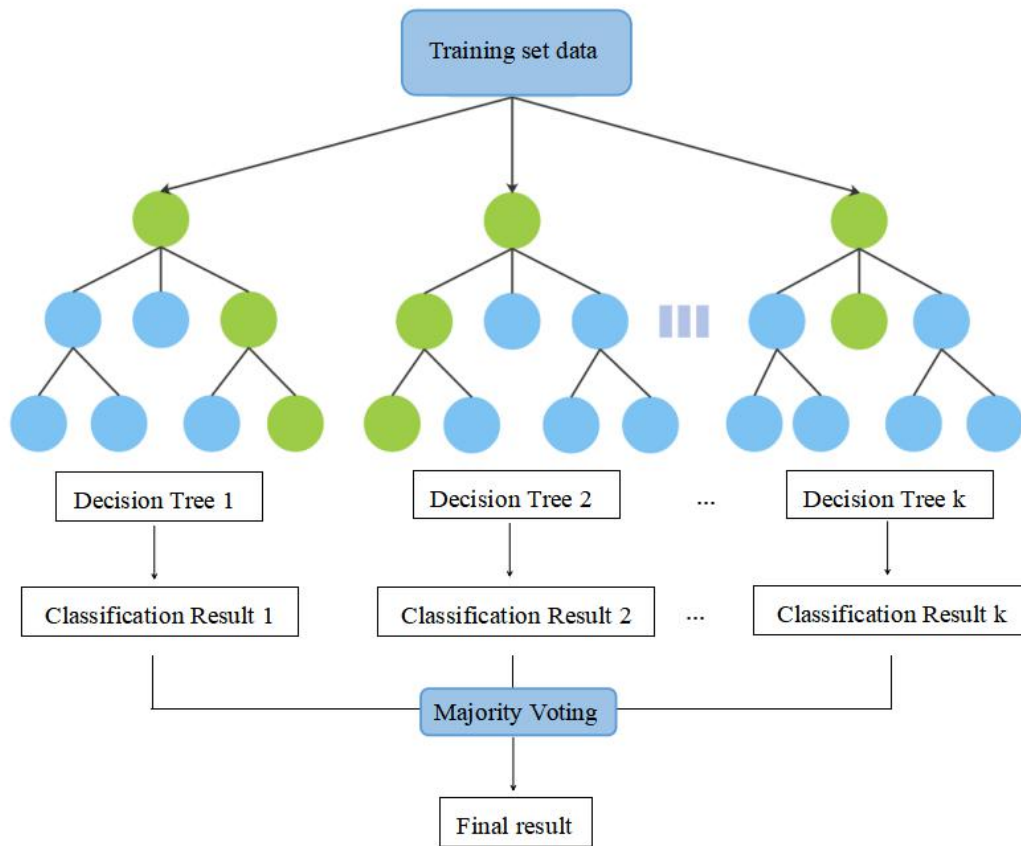


Figure 3. Random Forest Classification structure

3.3.2 Random Forest Regression

Random forest regression is composed of multiple regression trees. Random forest regression predicts continuous values. Random forest regression produces a single result by creating different decision trees, each making its own individual prediction, and then taking the average. The average of the prediction cannot exceed the highest and lowest value in the sample, which is a benefit to improve prediction accuracy.

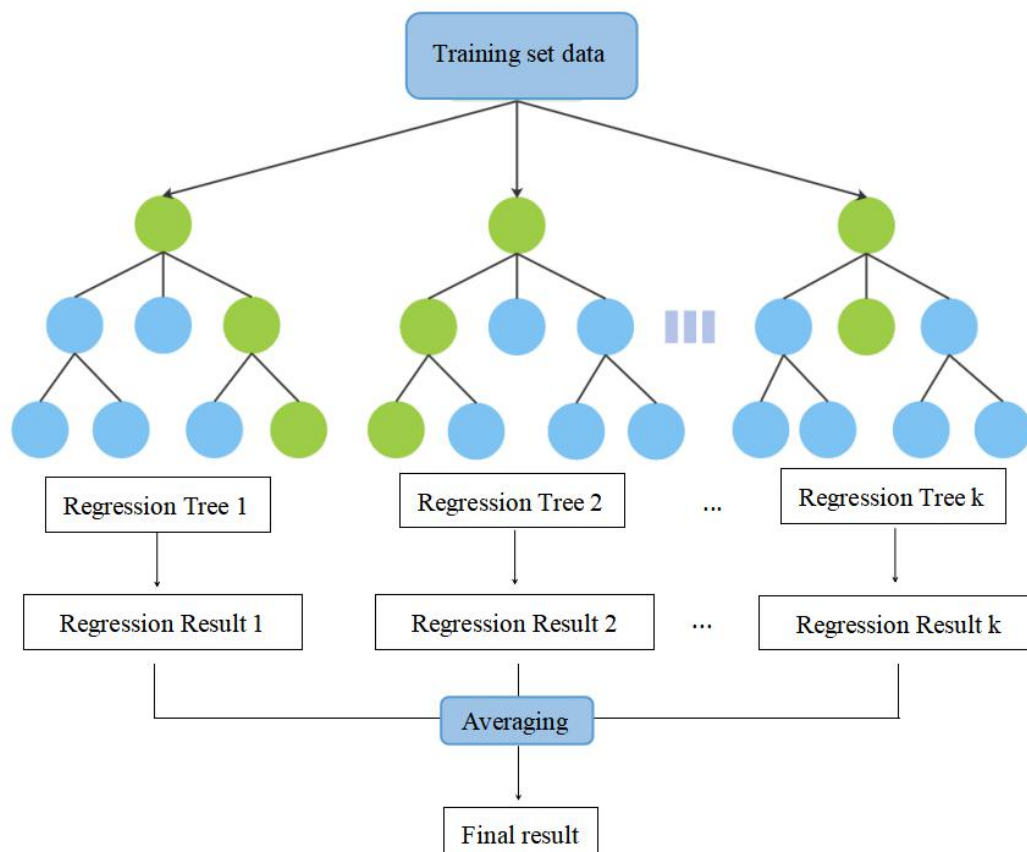


Figure 4. Random Forest Regression structure

3.3.3 Tuning Parameters in Random Forest

From package *randomForest* in R (Liaw & Wiener, 2002), we found there are many tuning parameters in random forest. When we created the model we found out which parameter can have a big impact on the model accuracy.

1. The number of trees, *ntree* . Usually an increase in the value of *ntree* reduces the Out-of-Bag error. The Out-of-Bag error always decreases sharply in the range of *ntree* between 0 to 100. When *ntree* is too large, it may increase the time or even lead to over-fitting. We need to make use of K-fold cross validation to find *ntree* .
2. Number of randomly selected attributes of nodes at each split, *mtry* . The correlation of the decision trees within the random forest are related to *mtry* . If

mtry is small enough, the correlation of the trees tends to weaken. On the other hand, the classification strength of the decision tree model increases as *mtry* increases.

3. The minimum size of terminal nodes, *nodesize* . This parameter implicitly sets the depth of the tree, which controls the complexity of trees in random forest.

4. The maximum number of terminal nodes, *maxnodes* . It is also used to control the depth of tree in the random forest.

Each bagged tree probably utilized only 2/3 of the observations, the others that did not get used are called Out-of-bag(OOB). During the Bagging, random forest method uses random sampling in the training process, there are some Out-of-Bag samples in the training set (Breiman, 2001). OOB error estimation is used to estimate the test error of a bagged model. In OOB prediction, there are OOB MSE for regression problem and OOB error for classification problem.

Grid search is a method to tune the parameters for the random forest algorithm. Grid Search is an exhaustive search means for tuning parameters. The best-performing parameter is the final result after iterating through all the candidates and trying every possibility. The metric for random forest classification is accuracy while using MSE for random forest regression.

3.5 Feature Selection Procedure

We cannot be sure that all the variables in the complete model are valid. These factors are put together as independent variables, and the complexity or covariance of the individual variables can affect the model construction. In this instance we need to do feature selection procedure. We used three methods for feature selection for logistic

regression and linear regression.

3.5.1 Stepwise Regression Algorithm

Stepwise regression algorithm is a method that combines backward selection and forward selection to find fitted regression model. Stepwise regression algorithm is used as a criterion to remove or add variables by selecting the smallest AIC value. If the value of AIC increases regardless of whether the variable is removed or added, the stepwise regression analysis is terminated and the current optimal regression model is obtained. In R, we need to use the “*stepAIC*” function in the *MASS* (Venables & Ripley, 2002) package and set the direction to "both" to achieve stepwise regression.

3.5.2 K-fold Cross-Validation

K-fold cross-validation is a resampling procedure used to measure the skills of the model. Firstly, we randomly divided training set into K subsets. Each subset data will be used as a validation set, and the remaining K-1 subsets will be used as the training set, which will result in K prediction models. A single estimate is eventually obtained by calculating the average of cross validation error from K models. The function is:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i$$

3.5.3 Lasso Regression

Lasso(Least Absolute Shrinkage and Selection Operator) regression is one of the shrinkage methods for regression. The main idea for LASSO is to construct a first-order penalty function to obtain a simple, sparse model, and to filter the features by eventually determining the coefficients of some variables to be 0. Lasso regression performs ℓ_1 regularization, which means adding absolute values to all regression

coefficients to minimize the model complexity. This avoids the possibility that when two variables have correlation, the coefficients for one variable may be significantly positive and the other significantly negative. The ℓ_1 -penalization function for Lasso regression is :

$$\sum_{i=1}^n |\beta_i| \leq \lambda$$

The value of λ is usually the minimum mean cross-validation error chosen by 10-fold cross validation. Then, to find the coefficients for Lasso estimates, use the function:

$$\hat{\beta}^{lasso} = \arg \min_{\beta} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}$$

In R, we use “*glmnet*” function for building lasso and “*cv.glmnet*” function to find cross validation value in *glmnet* package (Friedman et al., 2002) and set *alpha* = 1.

4 Result

4.1 Regression Method Result

For the linear regression, performing linear stepwise regression (*Table 3*) value of the adjusted R-squared was increased while decreasing the value of the AIC. It indicates that the model is gradually getting better. We can easily see there are 13 variables left in the model after doing the linear stepwise regression. In *Table 3*, the lowest AIC value is -2691.57, which occurs at the 13 variables model. Its AIC value dropped by 18.74 compared to the full model, while its adjusted R-squared went up by 0.0045.

Step	Variable Removed	Num of Variable in Model	AIC	Adjusted R-squared
0	-	25	-2672.83	0.2301
1	TSX1	24	-2674.79	0.2307
2	TSX5	23	-2676.73	0.2313
3	P90	22	-2678.63	0.2319
4	P96	21	-2680.46	0.2325
5	P86	20	-2682.23	0.2330
6	P80	19	-2683.99	0.2335
7	P20	18	-2685.34	0.2337
8	P84	17	-2686.60	0.2339
9	P60	16	-2687.83	0.2341
10	P70	15	-2689.69	0.2346
11	P94	14	-2690.73	0.2347
12	TSX8	13	-2691.57	0.2346

Table 3. Variable selection by stepwise linear regression

The left dashed lines in *Figure 5 and Figure 8* (in part 4.2) represent the minimized prediction error obtained when $\lambda = \text{lambda.min}$, and the right one represents one range of standard deviations for $\lambda = \text{lambda.1se}$ that maintains acceptable error and allows filtering to fewer values of the variables.

As shown in *Figure 5*, the MSE for the linear lasso regression model from 25 variables to 18 variables remains stable. MSE starts decreasing when the model has 18 variables and continues until it reaches its lowest point at 4 variables.

$\text{lambda.min} = 0.01552$ model obtained the best performance linear model with MSE around 0.255. The selected four variables are P98, P40, P10, and TSX3. Only one variable TSX3 appears for $\text{lambda.1se} = 0.14477$ model, with the MSE value around 0.277.

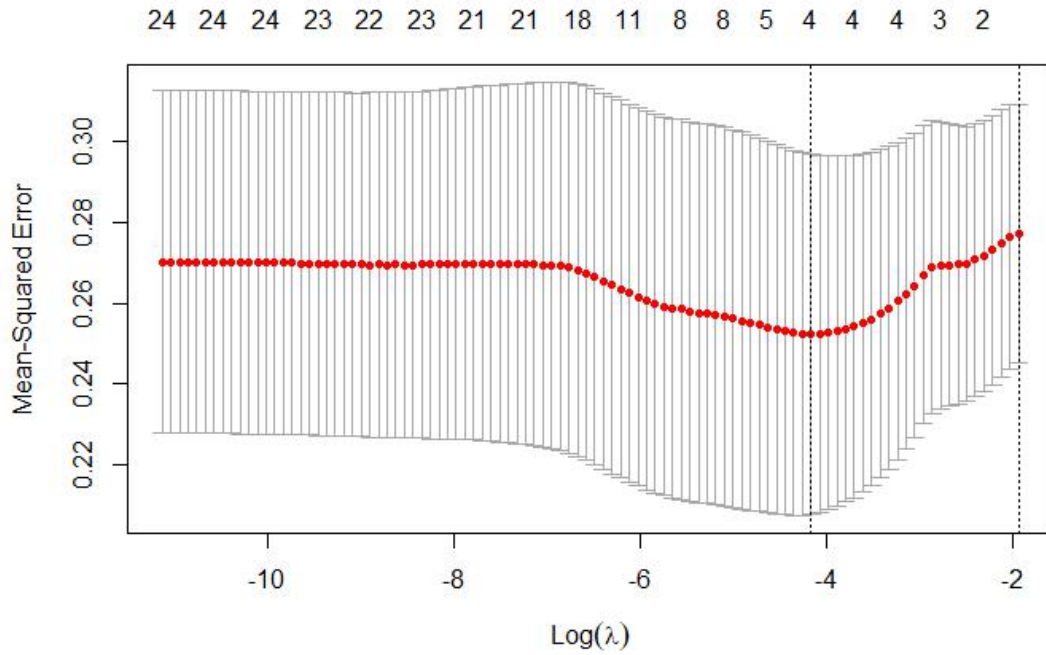


Figure 5. The CV error variation number of variables for linear lasso regression

The RMSE results when we perform the 10-fold cross-validation are shown in Figure

6. It illustrates that the 16 variables model has the lowest cross-validation RMSE value of 0.3217. The 16 variables are P98, P96, P90, P88, P80, P70, P60, P50, P20, P10, TSX2, TSX3, TSX4, TSX5, TSX7, and TSX8.

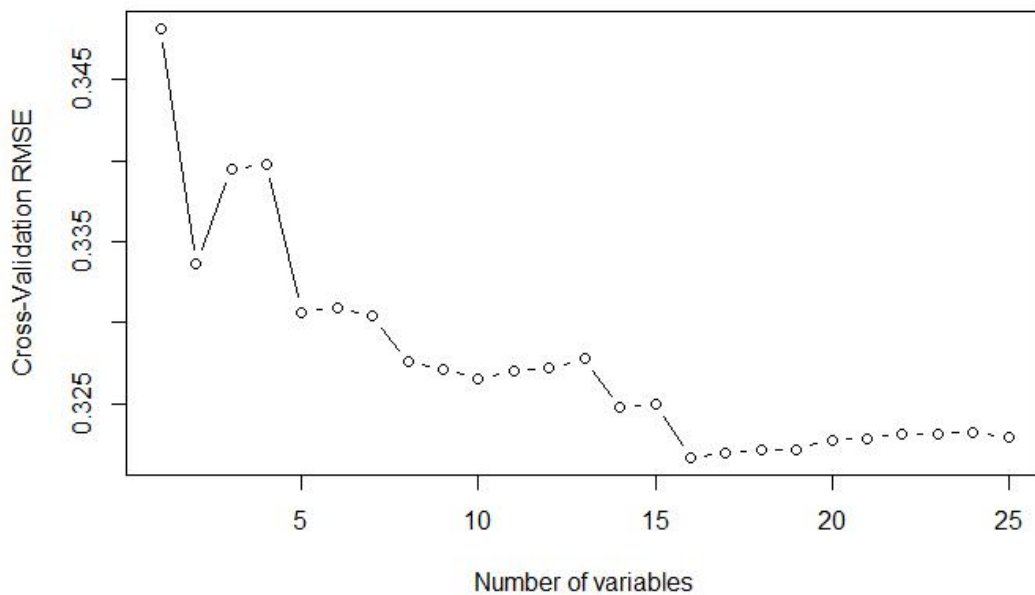


Figure 6. The CV RMSE for different number of variables in linear regression

Table 4 shows a comparison table of the best models obtained by our method for each variable selection. We can see that the least variable in the model is the linear lasso regression. The smallest AIC and the highest adjusted R-squared appear in the linear stepwise regression model. The lowest value of RMSE of the training set occurs in the linear regression full model, while linear stepwise regression has the lowest value of RMSE for the test set.

Feature selection method	Num of Variable in Model	AIC	Adjusted R-squared	RMSE for Training Data	RMSE for Testing Data
Full model	25	-2672.83	0.2301	0.3191782	0.31335381
Stepwise Regression	13	-2691.57	0.2346	0.3198820	0.31273683
K-fold Cross Validation	16	-2684.16	0.2317	0.3200712	0.31480644
Lasso regression	4	-2667.49	0.2317	0.3255730	0.31507498

Table 4. Compared linear regression models by different feature selection method

The best range and optimal number for each parameter find by grid search in *Table 5*.

Name of Parameter	Range for RF Regression	Optimal Num for for RF Regression
ntrree	[30,100]	100
mtry	[8,15]	8
nodesize	[4,12]	8
maxnodes	[20, 40]	36

Table 5. Tune the parameters in Random Forest Regression

The RMSE for the optimal random forest regression training data is 0.2709697, and the test data error is 0.3244217. The predict plot is as below (*Figure 7*):

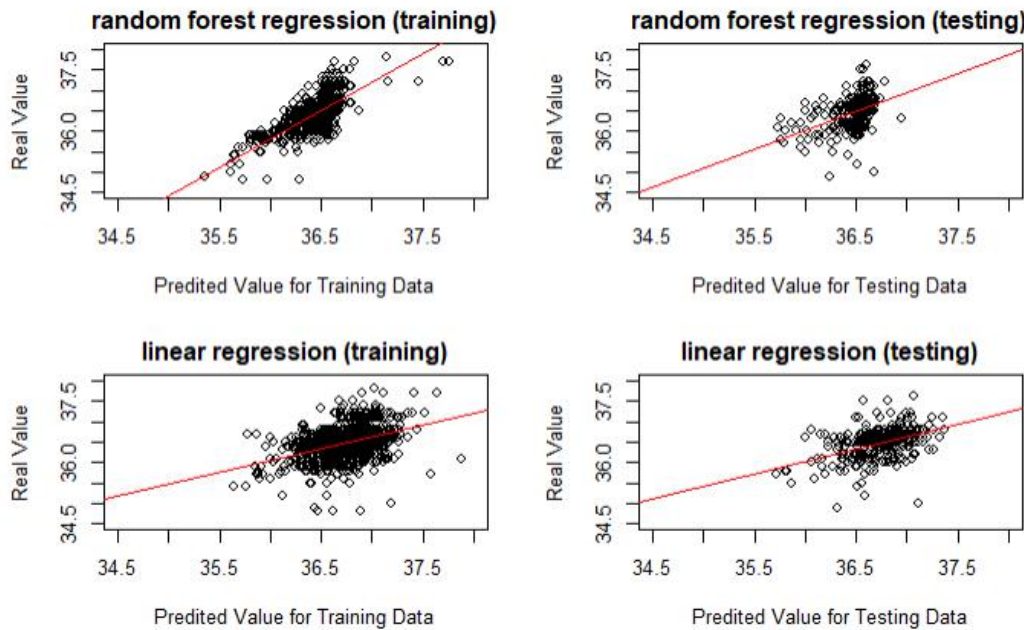


Figure 7. Random forest and linear regression prediction plots

4.2 Classification Method Result

The logistic regression model (*Table 6*) improved because their AIC value decreased after variable selection. The lowest AIC, 2587.93 for the logistic regression model appears with 20 variables in *Table 6*. It differs from the AIC value of the full model by -6.73. The value of AUC remained greater than 0.8 for all step in logistic stepwise regression.

Step	Variable Removed	Num of Variable in Model	AIC	AUC
0	-	25	2594.66	0.802
1	P40	24	2592.66	0.802
2	P92	23	2590.69	0.802
3	TSX1	22	2589.08	0.802
4	P86	21	2588.19	0.802
5	P20	20	2587.93	0.801

Table 6. Variable selection by stepwise logistic regression

For the logistic lasso regression (Figure 8), when $\lambda_{min} = 0.00700$, the model will minimize the cross-validation error around 1.245, and the number of variables is eight. The eight selected variables are P94, P92, P80, P40, P30, P10, TSX3, and TSX6. When the $\lambda_{1se} = 0.04501$, the model allows to filter the number of variables to three with the Cross-validation error around 1.31. The three selected variables are: P40, P10, and TSX3.

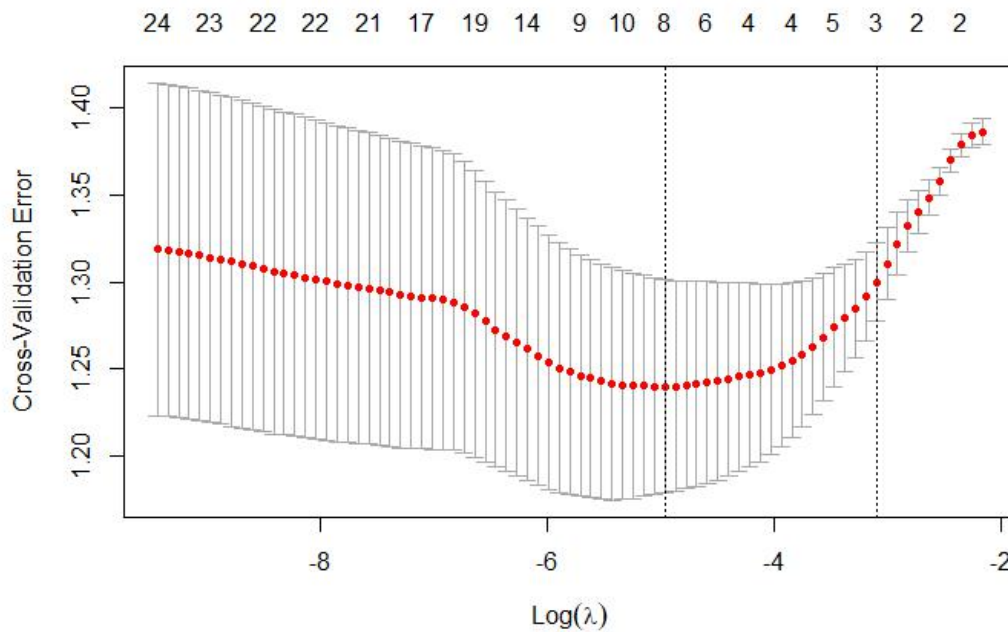


Figure 8. The CV error variation number of variables for logistic lasso regression. After using 10-fold cross-validation on logistic regression, the 8 variables model has the highest cross-validation accuracy, 0.9254 (Figure 9). The 8 variables are P80, P20, P10, TSX1, TSX3, TSX4, TSX7, TSX8.

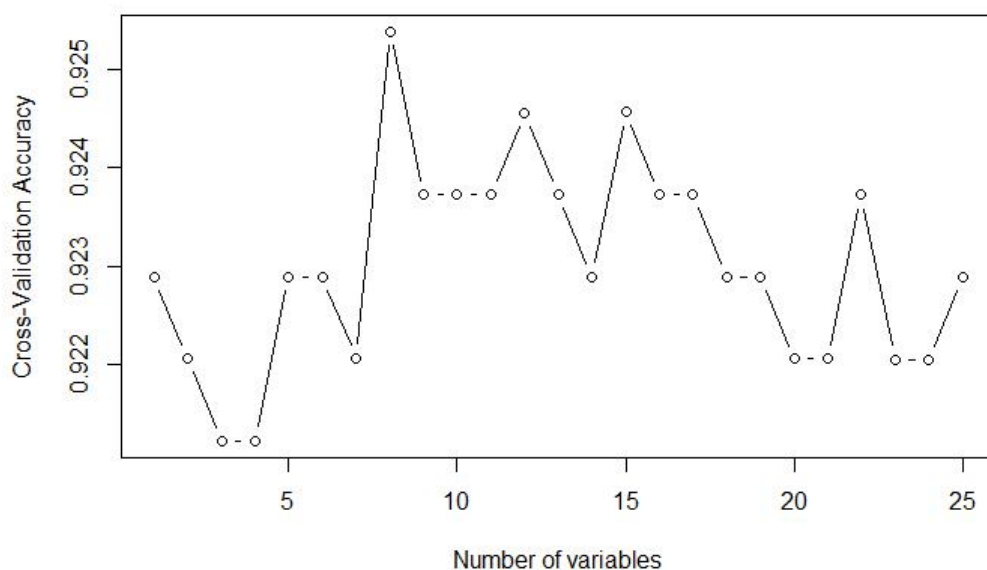


Figure 9. The CV accuracy for different number of variables in logistic regression

The number of variables and the variables selected in each category of feature selection are different. The models obtained by each method of feature selection can be found in Table 7.

Feature selection method	Num of Variable in Model	AIC	AUC	Accuracy for Training Data	Accuracy for Testing Data
Full model	25	2594.66	0.802	0.7234	0.7066
Stepwise Regression	20	2587.93	0.801	0.7200	0.7066
K-fold Cross Validation	8	2698	0.771	0.7091	0.7035
Lasso regression	8	2727	0.765	0.6882	0.6309

Table 7. Compared logistic regression models by different feature selection method

The lowest value of AIC appears in logistic stepwise regression, and the highest value of AUC appears in the logistic full model. The highest accuracy of the training set appears at the logistic full model, and the logistic full model and logistic stepwise regression test sets.

By using grid search method for random forest classification, the best range and optimal number for each parameter in *Table 8*.

Name of Parameter	Range for RF Classification	Optimal Num for RF Classification
ntree	[30,80]	40
mtry	[1,5]	2
nodesize	[6,10]	8
maxnodes	[15,30]	20

Table 8. Tune parameters in Random Forest Classification

The accuracy for optimal RF classification is 0.8499581 for training data (*Table 9*), while 0.807572 for testing data (*Table 10*), with an OOB error rate of 0.2252.

Training data set		
	Predicted: Normal	Predicted: Fever
Actual: Normal	946	155
Actual: Fever	24	68

Table 9. The confusion matrix of optimal random forest classification (training set)

Testing data set		
	Predicted: Normal	Predicted: Fever
Actual: Normal	248	51
Actual: Fever	10	8

Table 10. The confusion matrix of optimal random forest classification (testing set)

5 Discussion

Our research findings suggest that for the classification analysis, the prediction accuracy of random forest classification is better than other methods (*Table 7, Table 9 & Table 10*). The other classification models obtained using feature selection and the full model have training set accuracy around 0.7 while the testing set accuracy is between 0.63 to 0.7. The accuracy of both the training and test sets of the classification random forest improved to above 0.8. It is significantly higher than the accuracy of the logistic regression model.

There is not a clear difference between the different prediction methods for regression (*Table 4*). The RMSE of the predicted training set for all regression models was in the middle of 0.319 to 0.325, and the RMSE of the testing set was around 0.314. Random forest regression drops the RMSE of the training set to 0.271 but the test error increased to 0.324. Since the adjusted R-squared for each model is around 0.233, which is a number very far from 1, we suggest that our data is more suitable for classification analysis than regression analysis.

There are some limitations to our research. Since we had insufficient fever data, we set the fever threshold at no less than 37 degrees Celsius. In future study, we should collect more data for the fever set to test if our model is accurate. Furthermore, the thermal camera is not being set up specifically to do person temperature reading. The greyscale representation of each image makes it more difficult to determine any apparent temperature differences. Finally, our regression model did not work well,

and in the future, we can adjust some new variables or use other regression models to create a more valid model.

6 Reference

- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. "O'Reilly Media, Inc."
- Breiman, L. (2001). Random forest. *Machine Learning*, 34, 5-32.
<https://doi.org/10.1023/A:1010933404324>
- Bernard, V., Staffa, E., Mornstein, V., & Bourek, A. (2012). Infrared camera assessment of skin surface temperature – Effect of emissivity. *Physica Medica*, 29(6), 583-591. <https://doi.org/10.1016/j.ejomp.2012.09.003>
- Brownlee, J. (2016). *Tune Machine Learning Algorithms in R (Random forest case study)*. R Machine Learning.
<https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/>
- Fawcett, T. (2006). *An introduction to ROC analysis*, Pattern Recognition Letters. 27, 861–874.
- Friedman, J., Hastie, T., & Tibshirani, R. (2010). “Regularization Paths for Generalized Linear Models via Coordinate Descent.” *Journal of Statistical Software*, 33(1), 1–22. <https://www.jstatsoft.org/v33/i01/>
- Kuhn, M. (2008). Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5), 1 - 26. <http://dx.doi.org/10.18637/jss.v028.i05>
- Ladislav, Mariš., & Veronika, Adamová. (2021). Global Economic Impacts of COVID 19 to Increase the Protection of Society By Using Thermal Imaging Cameras to Measure the Temperature of Human Body. *SHS Web of Conferences*, 92, 01031.

- Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. *R News* 2(3), 18--22.
- Peters, J., Bernard, D. B., Verhoest, N. E. C., Samson, R., Degroeve, S., Piet, D. B., & Huybrechts, W. (2007). Random Forests as a Tool for Ecohydrological Distribution Modelling. *Ecological Modelling*. 207. 304-318.
<https://doi.org/10.1016/j.ecolmodel.2007.05.011>
- Ranstam, J., & Cook, J. A. (2018). LASSO regression. *British Journal of Surgery*, 105(10), 1348. <https://doi.org/10.1002/bjs.10895>
- Venables, W. N., Ripley, B.D. (2002). *Modern Applied Statistics with S*, Fourth edition. Springer, New York. ISBN0-387-95457-0.
<https://www.stats.ox.ac.uk/pub/MASS4/>
- Wagemakers, E. J., & Farrell, S. (2004). AIC model selection using Akaike weights. *Psychonomic Bulletin & Review*, 11(1), 192-196.