Machine Learning Scientific Report

1. Problem framing

The Covid-19 pandemic is still developing today. Between December 2019 and May 2021, there have been over 160 million cases and over 3 million deaths globally. To better identify trends and assist in medical care, we can turn to machine learning (ML) for support.

In this paper, we build and compare predictive models using ML algorithms and epidemiological data from the Covid-19 outbreak. The dataset we use is by Xu et al. (2020) and is publicly available on GitHub.[1] The problem we want to solve is of predicting confirmed cases' outcomes as either *died* or *discharged* (from hospital). The motivation for solving this problem is that the prediction model built may be helpful for quickly allocating medical care to patients at a time when there are limited resources.

2. Experimental procedure

2.1 Data preparation

First, to better understand the data, we produced a number of visualisations: we created a bar plot of up to the 10 most frequent values of each categorical attribute (e.g., *symptoms*); we created a scatter plot of the *latitude* and *longitude* of each case; and, we plotted the daily and monthly number of cases globally and by country. From these visualisations, we got a strong overview of the spread of Covid-19 by location and by time.

Second, we began preparing the dataset for ML algorithms by removing irrelevant instances. These were instances that were not useful for the task. Since we were training models to predict outcome, irrelevant instances were those whose value of the target, *outcome*, was not one of either *died* or *discharged*. To remove such instances, we had to correct structural errors, such as inconsistent capitalisation and wording among target values: we replaced *Died*, *dead*, or similar values with *died*; and, we replaced *Discharged*, *recovered*, or similar values with *discharged*. Instances with target values whose meaning was dissimilar to either of these two values, including *null* values, were dropped.

Third, we kept attributes which we thought would be useful. The attributes kept, which became features, were *age*, *sex*, *latitude*, *longitude*, *date_onset_symptoms*, *symptoms*, *chronic_disease_binary*, and *travel_history_binary*. After this, we replaced the *symptoms* and *date_onset_symptoms* features with a new *symptoms_binary* feature, and cleaned up the dataset by dropping any remaining instances with missing values.

Fourth, we transformed numerical data. We performed data binning on the *age* feature, and created categorical values for the age ranges *0-14*, *15-34*, *35-59*, *60-79*, and *80+*. These were chosen because the *age* feature was in fact already categorical, and the latter four categories were the four most frequent. We also performed feature scaling on the *latitude* and *longitude* features by standardisation, also known as Z-score normalisation, which rescales the feature values to have zero-mean and unit-variance.

1. github.com/beoutbreakprepared/nCoV2019/

Fifth, we transformed categorical data. We performed one-hot encoding on the *age* and *sex* features since they had nominal data. Also, we encode the target values so that *discharged* corresponds to 1 and *died* corresponds to 0. The reason for performing these transformations is that ML algorithms require features to be numbers.

Finally, we randomise the resulting dataset, which has 5,206 instances, as we do not want the order of the instances, which is irrelevant, to affect the model training process; and, we split the data into a training set, validation set, and test set to prevent overfitting. We create a test set by selecting a 20% subset of the dataset so there are at least 1,000 test instances, which is large enough to yield statistically meaningful results. Since we suspect the *chronic_disease_binary* attribute to be an important feature to predict *outcome*, we chose to ensure that the the test set was representative of the overall *chronic_disease_binary* distribution by doing stratified sampling. The overall distribution was that 98.2% of values were *False* while the remaining 1.8% were *True*. We also observe that we maintain an exact balance of train and test target values (75.3% being *discharged*), which makes it easier to train models.

2.2 Model selection, training, and testing

The three ML algorithms we chose to build predictive models were the logistic regression classifier (LRC), gradient boosting classifier (GBC), and support vector classifier (SVC). For each of the three models, we iteratively train the model on the training set, evaluate the model on the validation set, and tweak the model according to the results on the validation set. In particular, we employ 4-fold cross-validation so that the size of the validation set is equal to the size of the test set. This further reduces the chance of overfitting. Then, we pick the model with the set of hyperparameters that does best on the validation set and confirm the results on the test set. This approach allows us to compare different trained models in an unbiased way, by comparing model performance using the test set, which is kept apart from the training process and is therefore unseen data.

2.3 Hyperparameter tuning

We tried different hyperparameter combinations for each model. For the LRC, we tried values 0.001, 0.01, 0.1, 1, 10, 100, and 1000 for C, the regularisation hyperparameter; and, we tried L_1 and L_2 regularisation for the norm used in the penalisation. For the GBC, we tried values 2, 4, 8, and 16 for the minimum number of samples required to be at a leaf node (*min_samples_leaf*); and, values 0.001, 0.01, and 0.1 for the learning rate. For the SVC, we tried values 0.75, 0.85, 0.95, and 1 for C; and, linear, polynomial, radial-basis, and sigmoid kernel types.

For the LRC, the best model achieved a 0.80 accuracy score with hyperparameter combination $C = 1000$ and L_2 regularisation. For the GBC, the best model achieved a 0.85 accuracy score with hyperparameter combination *min_samples_leaf* $= 2$ and learning rate 0.01. For the SVC, the best model achieved a 0.84 accuracy score with hyperparameter combination $C = 1$ and radial-basis kernel type.

3. Results (25%)

Next, we evaluate the trained models on the test set. To enable evaluation, we produce the following: the confusion matrix, positive predictive value (PPV), negative predictive value (NPV), sensitivity, specificity, F_1 score, accuracy, and receiver operating characteristic (ROC) curve, for each model.

Table 1: Confusion matrices produced by each predictive model on the test set.

(a) LRC.

Pred. Real.	0	1	all
0	185	55	240
1	148	654	802
all	333	709	1042

(b) GBC.

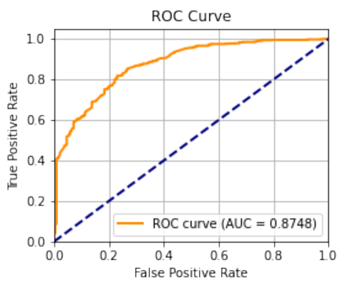
Pred. Real.	0	1	all
0	112	128	240
1	19	783	802
all	131	911	1042

(c) SVC.

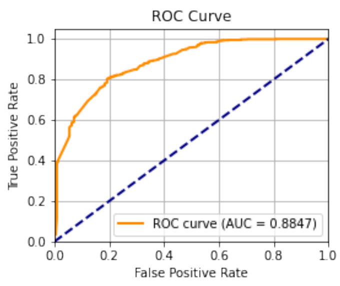
Pred. Real.	0	1	all
0	109	131	240
1	24	778	802
all	133	909	1042

Table 2: Statistics produced by each predictive model on the test set.

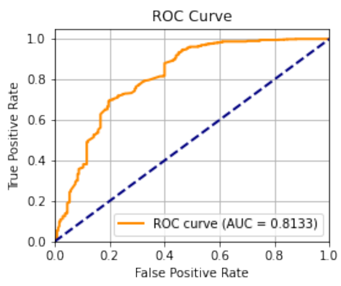
Model	**NPV**	**PPV**	**Specificity**	**Sensitivity**	**F1 Score**	**Accuracy**
LRC	0.56	**0.92**	**0.77**	0.82	0.87	0.81
GBC	**0.85**	0.86	0.47	**0.98**	**0.91**	**0.86**
SVC	0.82	0.86	0.45	0.97	**0.91**	0.85



(a) LRC.



(b) GBC.



(c) SVC.

Figure 1: ROC curves produced by each predictive model on the test set.

- Make comparisons between the 3 predictive models

- Provide necessary tables and charts to summarise and support the comparisons.

4. Discussions (20%)

4.1 Chosen models

We chose the first two algorithms because they were demonstrated in *Exercise for Logistic Regression*[2] with breast cancer prediction, and the third because it is recommended by the ML Cheat Sheet (for scikit-learn)[3].

4.2 Experimental procedure

- standardisation is commonly used by KNN, SVM, PCA, etc. but not necessary for logistic reg, tree-based, and random forest
- standardisation helps better deal with outliers, but min-max can generate smaller std; could have tried fitting model to raw, normalised, and standardised data da the ncompare their performances for the best results
- could have imputed

4.3 Limitations

- Was going to include 'country', then saw that data over represents Philippines
- Further work can include date_confirmation
- We notice we have an imbalanced data set, so we employ down sampling and up-weighting during model training.
- Larger k means less (pessimistic) bias. Leave-One-Out Cross-Validation is too computationally expesive.
- deviance and exponential loss functions to be optimised for gradient boosting

5. Conclusions and lessons learnt (10%)

- Discuss the results and draw conclusions from your experimentation

References

Bo Xu, Bernardo Gutierrez, Sumiko Mekaru, Kara Sewalk, Lauren Goodwin, Alyssa Loskill, Emily Cohn, Yulin Hswen, Sarah C. Hill, Maria M Cobo, Alexander Zarebski, Sabrina Li, Chieh-Hsi Wu, Erin Hulland, Julia Morgan, Lin Wang, Katelynn O'Brien, Samuel V. Scarpino, John S. Brownstein, Oliver G. Pybus, David M. Pigott, and Moritz U. G. Kraemer. Epidemiological data from the COVID-19 outbreak, real-time case information. *Scientific Data*, 7(106), 2020. doi: doi.org/10.1038/s41597-020-0448-0.

2. tinyurl.com/COMP2261prac6
3. scikit-learn.org/stable/tutorial/machine_learning_map/index.html