Machine Learning Coursework

# Introduction

I plan to use classification methods to predict the binary outcome of a COVID case (‘Recovered’ or ‘Died’), based on the age, sex, country and time between the data of onset of the symptoms and the date of a confirmation (measured in days).

The question that I intend to build a classifier to answer is, “given their age, sex, country and the days they waited between the onset of symptoms and being confirmed to have COVID, will a patient diagnosed with COVID recover or die?”

My motivation for choosing this problem is to try and improve the understanding of how important it is to respond quickly to the onset COVID symptoms for different demographics. I would like to understand whether rapidly responding to a patient exhibiting COVID symptoms can help to save their life.

# EXPERIMENTAL PROCEDURE

## DATA PREPARATION

The features from the dataset that I’ve identified as useful for solving this problem are the age, sex, outcome, date\_onset\_symptoms, data\_confirmation and the country. As the other features in the dataset will not be used in training, I have dropped them from the dataset.

Many rows in the dataset are missing values for one of more of these features. In order to respect the highly individual nature of each case I have elected to remove these rows rather than attempting to impute values or fill values in with dummy variables.

#### Feature Encoding: The outcome field contains several different values with the same meaning (for instance ‘died’ and ‘dead’ refer to the same outcome). To encode this data I use the pandas replace() method to convert the different words to read 0 or 1, where 0 refers to instances with a bad outcome (death or ‘severe’ and 1 refers to outcomes with a positive outcome such as ‘recovered’ or ‘Alive’.

#### To encode the age data I have applied an ordinal encoding. The minimum precision of the data in this column is a 10 year range. To encode this data I have replaced all the values with the middle of the 10 year range. For example, the value ‘10-19’ is replaced with 15. Where the age data exists as a single integer (for example 17) I have left it as it is. To scale the data belonging to this feature I have used min-max scaling, this produces values between 0 and 1 which can easily be interpreted by the models.

#### To encode the sex data I have used the value 0 to represent the ‘male’ class and the value ‘1’ to represent the ‘female’ class.

#### To encode the ‘days waiting’ data I have taken the gap between the date\_onset\_symptoms and the date\_confirmation and stored it as an integer. These have then been scaled with min-max scaling.

#### To encode the ‘country’ data I have used a one-hot encoding method. This respects the fact that the country data cannot be considered ordinally and requires no further feature scaling.

## VISUALISATION

The data that I wanted to visualise for this task was the distribution of instances split across ages, genders and days waiting.

#### Outcome

This pie chart shows that 82% of my data points represent individuals who recovered. This means that my dataset is highly imbalanced and will require care to produce unbiased estimators.

#### Age

The leftmost histogram shows the number of people by the scaled ages, the middle histogram shows the distribution of ages amongst the individuals who died, and the rightmost histogram shows the distribution of ages amongst the individuals who recovered.

We can see that the total dataset has a roughly bell shaped distribution centered around the median age (with a spike around the 0.3 mark) whilst the set of individuals who died has a significant skew towards the older end and the set of individuals who recovered has a younger skew.

#### Gender

The following charts can be used to examine the distribution of sexes across the total dataset, those that died and those that recovered.

The total dataset contains 1785 entries labelled as Male and 1671 entries labelled as Female. Examining the distribution for the set of individuals who died suggests that if an individual died from COVID they were more likely to be male than female. In this subset 374 entries were labelled as Male whilst 250 entries were labelled as Female. In the set of individuals who recovered, there were 1421 Males and 1411 Females, indicating that if an individual recovered from COVID it is difficult to make predictions about their gender.

#### Days waiting

When considering the data for the days waiting between the onset of symptoms and being diagnosed with COVID, the distributions have a similar shape across each subset of the data

## PARTITIONING THE DATA

A larger training set is likely to result in a more accurate model, as my dataset is relatively small, I have opted for a 90% Training set and a 10% Testing set. I also use a cross validation technique during hyperparameter tuning in order to maximize the usefulness of the small dataset.

## Reweighting

In the visualization stage I discovered that my dataset is highly imbalanced.82% of my data points belong to the Recovered class. This means a classifier can predict the majority class each time and achieve an accuracy rating of 82%, to combat this effect I have used an upweighting method to increase the relative importance of the ‘Died’ instances during training.

## MODEL SELECTION

I began the training process by training 6 models with the default hyperparameters. The models used were Naive Bayes, Support Vector Machine, Decision Tree, Logistic Regression, Random Forest and K-Nearest Neighbours. These models were chosen as they represent a good spread of different, popular models for binary classification tasks.

To choose between them it was imperative that I defined an evaluation metric to use.

#### Evaluation: To evaluate a model I wanted to consider the model’s accuracy, balanced accuracy and f1 score. As my training set is quite imbalanced, the f1 score may be misleading so is weighted lower than the balanced accuracy. Here is how I’ve defined my score function for a model’s predictions on a set of data:

*Score = (3\*accuracy)+(2.5\*balanced accuracy)+(1.5\*f1 score)/7*

When sorting the models based on their score I found that the best performing models were the Random Forest Classifier, Logistic Regression and the k-Nearest Neighbours model. These are therefore the models that I continued to use, moving forwards into the model improvement section.

## TRAINING, EVALUATION AND TUNING

To tune my models’ hyperparameters I used Sci-Kit Learn’s GridSearchCV. This tests all the combinations of hyperparameter values from a set provided. It also uses 5-fold cross validation in order to determine the best performing hyperparameters on the training data (using one fold as a validation set each time). The score function provided to GridSearchCV was the custom evaluation metric, defined above.

# PRESENTATION, VISUALISATION AND INTERPREATION OF THE RESULTS.

The first metric that I would like to present is the accuracy score for my final models.

|  |  |
| --- | --- |
| Model | Accuracy Score |
| Logistic Regression | 72.25% |
| k-Nearest Neighbours | 81.5% |
| Random Forest | 76.3% |

These accuracy scores suggest that each of these models is useful for accurately answering the question, “given their age, sex, country and the days they waited between the onset of symptoms and being confirmed to have COVID, will a patient diagnosed with COVID recover or die?”.

The second metric that I would like to present is my own custom evaluation score for each model.

|  |  |
| --- | --- |
| Model | Custom Score |
| Logistic Regression | 0.6854 |
| k-Nearest Neighbours | 0.7927 |
| Random Forest |  |

# DISCUSSION OF MODELS AND EXPERIMENTAL PROCEDURE

For each of my models the experimental procedure was very similar. To produce fair and comparable results I used the same training and testing data for each model. This data was prepared using the method detailed above.

I then fitted an initial model to the training data, using the default hyperparameters. These models served as a benchmark to improve upon using the hyperparameter tuning.

## LOGISTIC REGRESSION

The logistic regression hyperparameters that I chose to tune were the solvers, the C values and the maximum number of iterations.

## k-NEAREST NEIGHBOURS

The hyperparameters that I chose to tune for the k-Nearest Neighbours model were the number, k, of neighbours that would be queried when making a prediction; the weights for the ‘voting power’ of the neighbours and the algorithm used for computing the nearest neighbours.

## RANDOM FOREST

The hyperparameters that I chose to tune for the random forest model were the number of decision trees in the forest; the function for measuring the quality of a split; the maximum depth of a tree in the forest and the maximum number of features considered when choosing a split.

# CONCLUSIONS AND LESSONS LEARNT

The best lesson I learned during this coursework was the importance of taking measurements to evaluate my models as often as possible. During the process of completing the coursework I slowly added more metrics to my evaluation function.

I also learned that investing more time into cleaning the dataset makes later steps easier and saves time overall.

Initially I was worried that my dataset with 3493 rows may be too small to train useful and accurate models, but I was surprised to see that the accuracy scores were mostly around 80%.

I also learned that hyperparameter tuning can take a long time when using the grid search method, but similar results can be obtained much faster using the randomised approach.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.