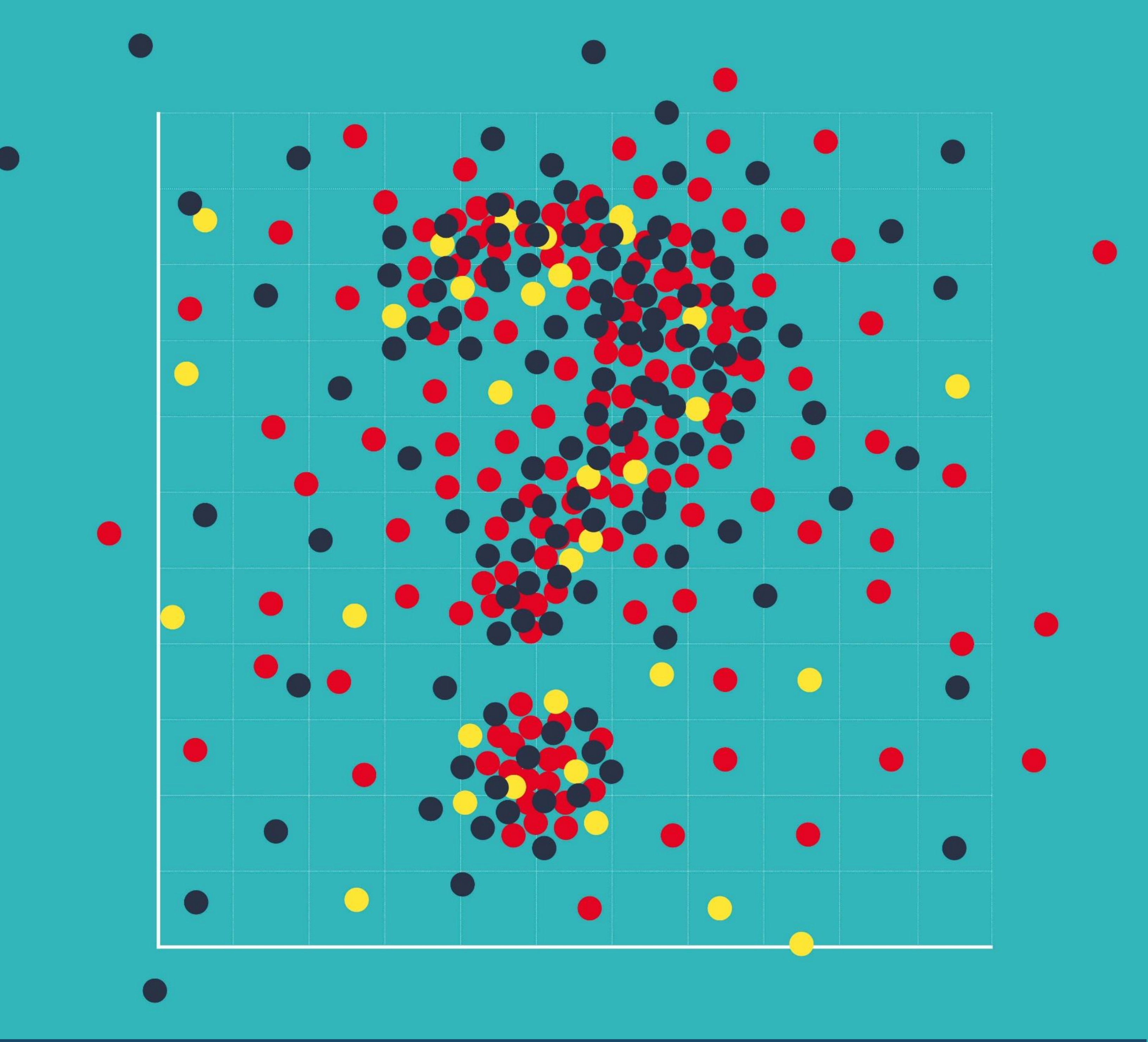


MATHEMATICS SPECIALISTS

BY ARTHUR LUND

INVESTIGATION: LOGISTIC GROWTH

Outline

Because our world is ever-changing, people are constantly looking for better and better ways of modelling and predicting that change in order to increase our understanding of our current situation and improve our predictions of what might happen in the future. This report will consider a mathematical function- the logistic/sigmoid function -that that has been traditionally used for modelling population growth. This report will explore two additional real-world applications of this function, including the modelling of the spread of the novel coronavirus and machine learning.

Part One – Covid-19 Spread

Unlike exponential growth ($f(x) = a^{x-b} + c, a > 1$) which approaches infinity as x approaches infinity, logistic growth only increases up to a **limiting value** K. Because of this, the logistic growth is ideal for modelling situations in the real world when a population's growth may initially seem exponential, but eventually levels off due to finite resources. Take, for example, a bacteria colony with population P starting with a single bacterium that divides once every hour ($P_t = 2^t$). Hypothetically, after 12 days in a 'perfect' environment, there would be more bacteria in the colony ($P_{12\times 24} = 2^{288} \approx 4.97 \times 10^{86}$) than atoms in the known universe ($\sim 10^{80}$). But obviously, in the real world, factors like limited resources, competition and crowding limit the growth of a population, causing it to follow something closer to a logistic curve.

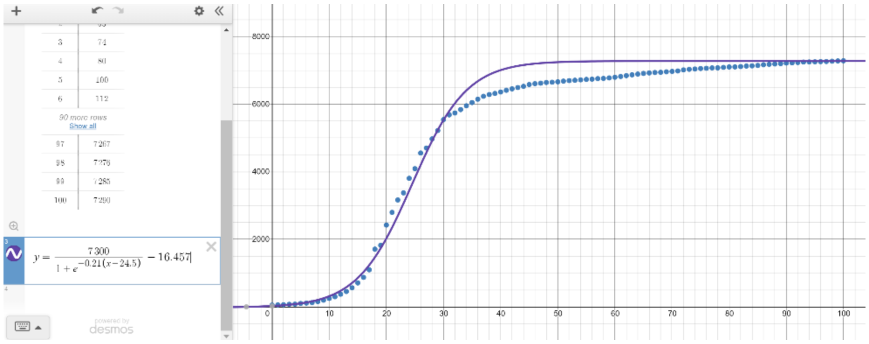
$$\boldsymbol{P(x) = \frac{K}{1 + e^{-a(x-b)}}}$$

Here, a is the steepness of the curve and b is the x-coordinate of the point of inflection.

How does this relate to the coronavirus pandemic? Because the spread of diseases can be modelled with logistic curves- initially they spread from person to person and the cases seem to grow exponentially, but as the cases approach the limiting value, the population of people susceptible to the disease decreases and thus, the infection rate also decreases. But does the spread of coronavirus spread according to a logistic curve, when taking into account real-world factors such as border closures, mask usage, business closures and individual self-imposed social isolation? Consider the following data showing the total number of covid cases from 3 countries over a 100-day period from the day they had their 50th case.

<u>**Australia (5th March – 25th May)**</u>

Here, the logistic curve closely follows the data up until day 30, but then the infection rate slows suddenly. From day 37 until day 100 it almost follows a linear progression, which could be due to the Emergency Response Plan that put harder restrictions into place since February 27th.



3	74
4	80
5	100
6	112

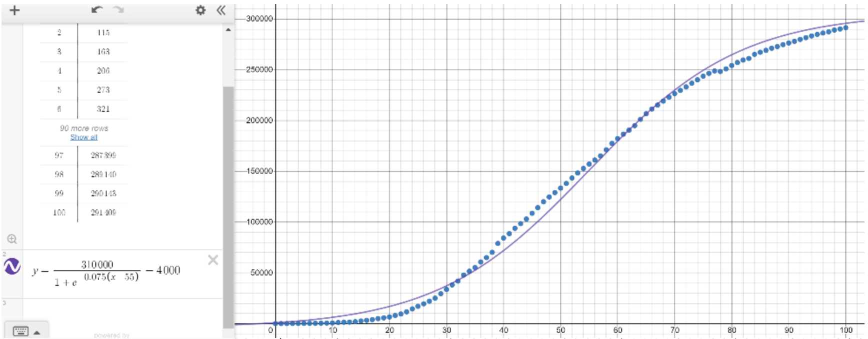
90 more rows
Show all

97	7267
98	7276
99	7285
100	7290

$$y = \frac{7300}{1 + e^{-0.21(x-24.5)}} - 16.457$$

United Kingdom (3rd March – 11th June)

Overall, the graph seems to generally follow the real-world data, although the infection rate seems to be more constant than predicted from day 35 to 65. It is notable that the UK took much longer than Australia to reach their peak cases, but this is reasonable considering their greater population.



2	115
3	163
4	206
5	273
6	321

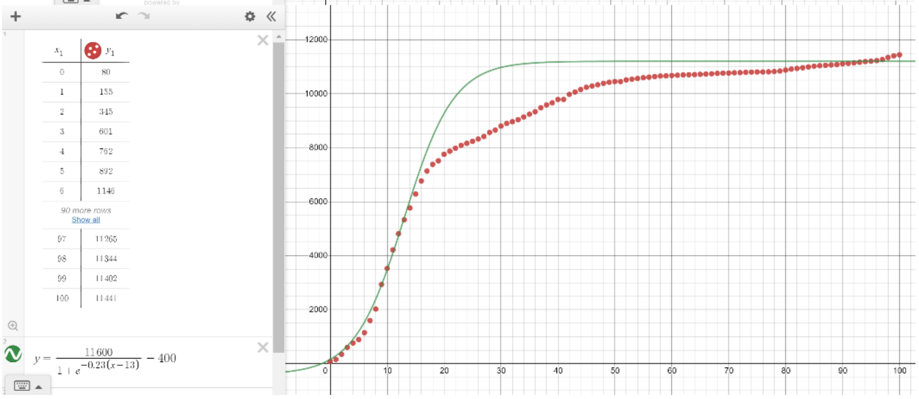
90 more rows
Show all

97	287 399
98	289 140
99	290 143
100	291 409

$$y = \frac{310000}{1+e^{-0.075(x-55)}} - 4000$$

South Korea (19th February – 29th May)

Here the real-world growth diverges from the graph in a similar (but far more obvious) way to that of Australia- the graph maps very closely until day 14, but then around day 16 the infection rate suddenly slows but continues growing with no limiting value in sight. This reflects the aggressively-implemented restrictions since January 27th in an effort to 'flatten the curve', especially since the Patient 31 incident around February 18th, which caused an explosion of covid cases in South Korea.



x_1	y_1
0	80
1	155
2	345
3	601
4	762
5	892
6	1146

90 more rows
Show all

x_1	y_1
97	11 265
98	11 344
99	11 402
100	11 441

$$y = \frac{11600}{1+e^{-0.23(x-13)}} - 400$$

Conclusions

While the general shape of the logistic curve was roughly followed by the data, especially during the first few weeks when the spread resembles exponential growth, lockdown restrictions and social distancing distort the graph significantly by slowing the infection rate. Unlike in the graph, where the initial conditions (i.e. limiting value, inflection point, gradient) are set and followed perfectly, in the real world, these conditions are constantly changing due to numerous factors such as social gatherings (e.g. protests, parties), travel restrictions, school/business closures, cleaning practices at institutions and how individuals social distance. These restrictions eventually slow the growth rate to a point where the graph appears to approach a limiting value, but the true value of K is the population of the country, so individuals who fail to get tested or socially isolate are able to cause cases to continue increasing slowly. The data shows that the more aggressive and sudden government restrictions were implemented, the greater the deviation from the graph. Therefore, a plain logistic curve is not suitable for modelling the spread of diseases. A more comprehensive model is needed that allows for the altering of the variables K, a and b in response to real-world conditions.

Part Two – Machine Learning

Machine learning is a branch of artificial intelligence that involves creating a computer program to perform a task by feeding it a large amount of training/testing data, based on which it can evaluate its performance and adjust its own algorithm to increase its accuracy. They are particularly useful for recognising patterns in extremely large or complex datasets or for automating extremely repetitive human tasks.

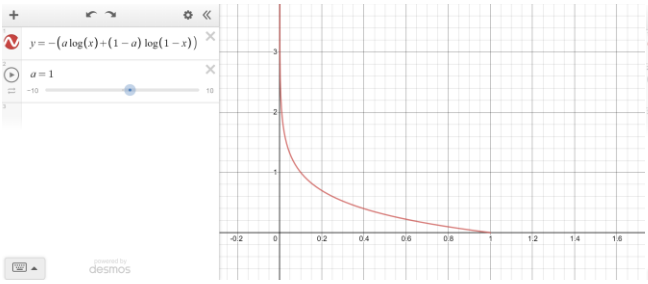
In short, an ML program contains 3 layers- the input layer, the hidden layers and the output layer. The data is processed in the hidden layers. Unlike linear and polynomial functions have ranges from negative to positive infinity, what is useful about logistic functions in ML is that their ranges can be bounded between 0 and 1 ($\frac{1}{1+e^{-a(x-b)}}$) and therefore used to express probabilities. This comes in handy in **binary classification problems**, that is, sorting a group of objects into two types (called **classes**). When deciding which class the object belongs in, that is, whether it will round up to 1 or down to 0, the programmer will give the model some **threshold** value (e.g. 90% confidence = 0.9). Some examples of binary classification ML programs that you maybe have encountered in day-to-day life include your email's spam filter (spam vs not spam), your bank observing your transactions for potential fraud and captcha tests on websites (is the user a robot or not).

In ML, the **hypothesis function** $\boldsymbol{h_\theta(x^i)}$ refers to what your program thinks the output will be for a certain input $\boldsymbol{x^i}$, and the **predicted output** $\boldsymbol{y^i}$ refers to the corresponding correct output. Since we are dealing with a binary situation, $\boldsymbol{y^i}$ can only take the value of 0 or 1. **Logistic regression** refers to the process of minimising the error of a model that uses a logistic function as $\boldsymbol{h_\theta(x)} = \left(\frac{1}{1+e^{-(\theta_0+\theta_1x_1+\cdots+\theta_nx_n)}}\right)$, where $\boldsymbol{\theta_0}$ to $\boldsymbol{\theta_n}$ are variables (called **parameters**) that the algorithm will adjust to improve the model's accuracy.

That error is calculated using the **cost function** $\boldsymbol{J(\theta)}$ which is averaged across the data points. The error of a single data point can be determined the piecewise function below (we'll get to why in a moment).

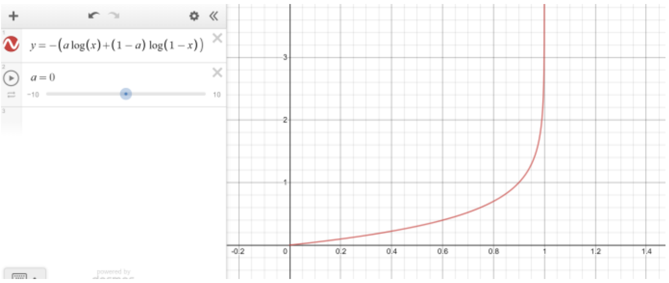
$$\boldsymbol{f(\theta)} = \begin{cases} \boldsymbol{-log\left(h_\theta(x)\right),} & \boldsymbol{for\ y = 1} \\ \boldsymbol{-log\left(1 - h_\theta(x)\right),} & \boldsymbol{for\ y = 0} \end{cases}$$

Which are graphed as follows:



$y = -\left(a\log(x) + (1-a)\log(1-x)\right)$

$a = 1$



$y = -\left(a\log(x) + (1-a)\log(1-x)\right)$

$a = 0$

So, in order to calculate the average error of $\boldsymbol{m}$ datapoints given parameters $\boldsymbol{\theta}$, we can express this as a single function:

$$\boldsymbol{J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^i log\left(h_\theta(x^i)\right) + (1 - y^i) log\left(1 - h_\theta(x^i)\right)\right)}$$

But how does this relate to adjusting the algorithm to make it more accurate? Enter **gradient descent**, the process adjusting the parameters $\boldsymbol{\theta_j}$ so as to minimise the cost function above. This is done by subtracting from each parameter the **gradient of the cost function** $\boldsymbol{\frac{d}{d\theta_j}J(\theta)}$ at the point $\boldsymbol{x^i}$, then multiplying it by the **learning rate** $\boldsymbol{\alpha}$ which is a small constant (often set to 0.01 by default) that allows the variable to be adjusted in small increments.

$$\boldsymbol{\theta_j \leftarrow \theta_j - \alpha\frac{\delta}{\delta\theta_j}J(\theta)}$$

$$\boldsymbol{\theta_j \leftarrow \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}\left(\left(h_\theta(x^i) - y^i\right)x^i\right)}$$

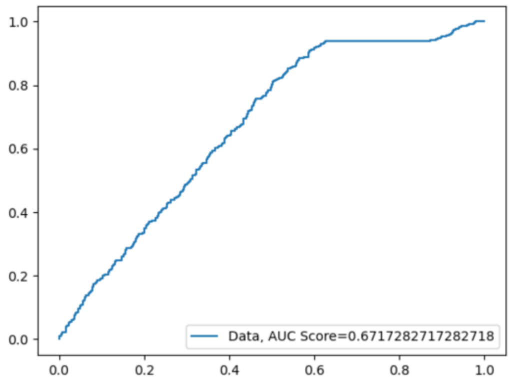
This process is repeated until the function can sufficiently distinguish the 2 classes, that is, until the gradient of the cost function $\boldsymbol{\frac{\delta}{\delta\theta_j}J(\theta)}$ approaches 0.

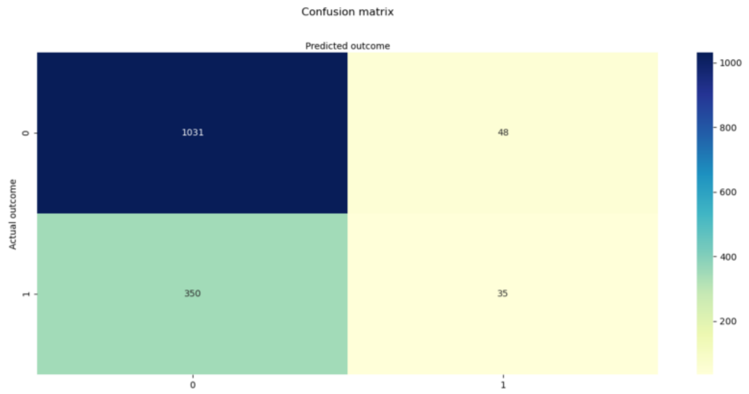
To put the use of a logistic regression model in binary classification to the test, I have created a machine learning algorithm with 2 settings in the Python programming language (much of the code has been taken from this website) that take in a dataset of 14,635 surgical records from TSHS (Teaching of Statistics in the Health Sciences) in order to predict whether there was a complication during the patient's surgery or if they died within 30 days (mort30). Of the available records, 90% were used to train the model and 10% were used to test its accuracy. The training data was further split into 50 batches (equally sized chunks) to help the model learn gradually, and the data is passed through the logistic model 20 times (called epochs).

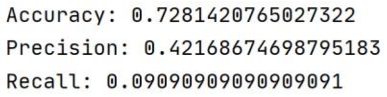
In the results, precision refers to the number of complications the algorithm correctly predicted divided by the total number of predicted complications ($\frac{true\ positives}{true\ positives\ + false\ positives}$), recall refers to the correctly predicted complications divided by the number that should have been predicted ($\frac{true\ positives}{true\ positives\ + false\ negatives}$) and accuracy is self-explanatory ($\frac{correct}{total}$). The ROC (Receiver Operating Characteristic) curve plots the true positive rate against the false positive rate, and the AUC refers to the Area Under this Curve, with 1.0 being a perfect score and 0.5 being equivalent to a random guess. The other graphic is a confusion matrix, which plots the predicted outcome against the actual outcome.

Setting 1

This program took in only raw features (inputs)- no advanced or pre-processed statistics allowed, with the exceptions of CCI -for the purpose of testing whether a logistic model can draw complex conclusions by itself. The 16 features include age, gender, race, the hour of day, day of the week, month, moon phase, and whether they have diabetes, cancer, dementia, osteoarthritis, cardiovascular disease, pulmonary disease, or a psychiatric disorder. The results were as follows:



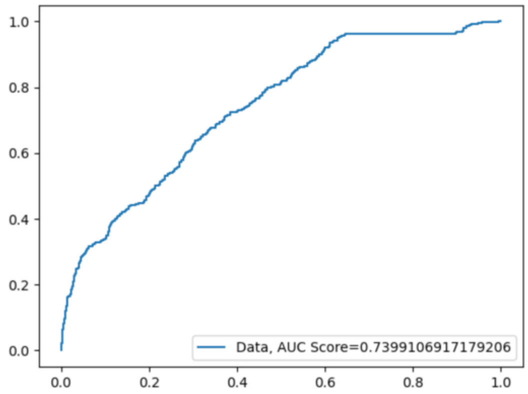


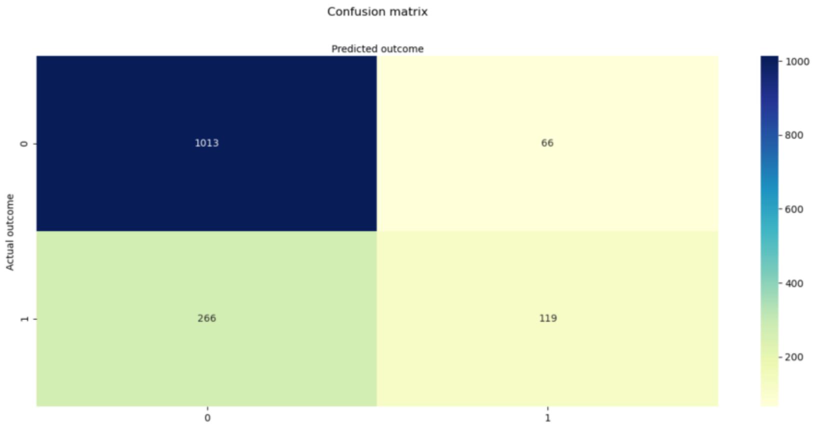


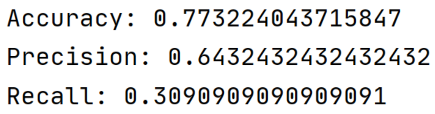
```
Accuracy: 0.7281420765027322
Precision: 0.42168674698795183
Recall: 0.09090909090909091
```

Setting 2

This program took in all inputs, including all the data from program 1 as well as CCI (Charlson Comorbidity Index), ASA status (American Society of Anesthesiologist physical status), RSI (Risk Stratification Index) for mort30, RSI for complications, CCS (Clinical Classifications Software) complication rate and CCS mort30 rate. This was to test a logistic algorithm's ability to filter out the useless inputs and prioritise that which has a meaningful impact on predictions. The results were as follows:







```
Accuracy: 0.773224043715847
Precision: 0.6432432432432432
Recall: 0.3090909090909091
```

Conclusions

The first setting produced quite poor results. Considering 50% accuracy is equivalent to a random guess, the 73% accuracy rate isn't too bad for only using raw inputs as features. However, the 42% precision and 9% precision calls into question the usefulness of accuracy regarding what the logistic model was actually doing. Because most patients (75%) in the records had neither a complication nor died within 30 days of their operation, the algorithm seems to have just guessed negative in most circumstances, perhaps because this would take the error function to a local minimum rather than the global minimum. This suggest that a logistic model is not suitable for processing complex data with multiple features that may influence each other. It should be noted, though, that the data was of limited quality and relevance, as basic patient characteristics such as BMI and age cannot even come close to capturing the full picture of a patient or the minute details of the operation.

The second setting produced much better results. The accuracy increased by 5.4%, precision increased by 22% and the recall more than tripled. This suggests that a logistic model is successfully able to prioritise features by increasing the weighting of advanced statistics such as such as complication RSI and 30-day mortality CSS and decreasing the weighting of irrelevant features such as moon phase. Therefore, while this model is not nearly accurate enough to be used in real-world medical scenarios where people's lives are on the line, its moderate level of success and ability to prioritise many inputs shows that a logistic regression model could be used for predicting binary outputs in a simpler, less serious environment, such as a DIY email spam filter.

Conclusion

This report has investigated the usefulness of the logistic curve with regard to modelling the spread of the covid-19 virus. While this model may look similar to the real-world data during the initial growth that seems exponential, over time it deviates from the graph due to real-world factors such as government restrictions. Therefore, a more complex model is necessary to take into account that the variables change over time (e.g. limiting value).

We have also discussed the application of the logistic curve as a regression model for binary classification in machine learning. My experiment with the surgical data shows that a logistic function is effective at prioritising which features are important, but is not suitable for processing complex data where multiple features may affect one another.

Sources

http://www.sthda.com/english/wiki/regression-analysis-essentials-for-machine-learning#:~:text=Regression%20analysis%20consists%20of%20a,function%20of%20the%20x%20variables.

https://bionumbers.hms.harvard.edu/bionumber.aspx?s=n&v=2&id=106614

https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

https://en.wikipedia.org/wiki/COVID-19_pandemic_in_Australia#February_2020

https://en.wikipedia.org/wiki/COVID-19_pandemic_in_South_Korea#Timeline

https://en.wikipedia.org/wiki/COVID-19_pandemic_in_the_United_Kingdom#Timeline

https://en.wikipedia.org/wiki/Logistic_function

https://en.wikipedia.org/wiki/Precision_and_recall#Recall

https://getpocket.com/explore/item/how-statistics-lost-their-power-and-why-we-should-fear-what-comes-next

https://machinelearningmastery.com/logistic-regression-for-machine-learning/

https://machinelearningmastery.com/machine-learning-in-python-step-by-step/

https://machinelearningmastery.com/types-of-classification-in-machine-learning/

https://meltingasphalt.com/interactive/outbreak/

https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html#:~:text=Gradient%20descent%20is%20an%20optimization,the%20parameters%20of%20our%20model.

https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html#gradient-descent

https://note.nkmk.me/en/python-function-return-multiple-values/

https://pandas.pydata.org/pandas-docs/stable/user_guide/reshaping.html

https://pandas.pydata.org/pandas-docs/version/0.18.1/generated/pandas.DataFrame.html

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

https://stackoverflow.com/questions/17315737/split-a-large-pandas-dataframe/28882020#28882020?newreg=6872d036ff7e4ab28863e413508347d9

https://stackoverflow.com/questions/31951980/what-exactly-does-numpy-exp-do#:~:text=The%20exponential%20function%20is%20e,x%20in%20your%20input%20array.

https://stackoverflow.com/questions/46572475/module-sklearn-has-no-attribute-cross-validation

https://stackoverflow.com/questions/49841324/what-does-calling-fit-multiple-times-on-the-same-model-do

https://stackoverflow.com/questions/51595162/how-to-update-logistic-regression-model

https://stackoverflow.com/questions/62658215/convergencewarning-lbfgs-failed-to-converge-status-1-stop-total-no-of-iter

https://stats.stackexchange.com/questions/324561/difference-between-convex-and-concave-functions#:~:text=A%20non%2Dconvex%20function%20is,to%20tell%20when%20this%20happens.

https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb

https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

https://www.csis.org/analysis/timeline-south-koreas-response-covid-19

https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python

https://www.desmos.com/calculator

https://www.health.gov.au/news/health-alerts/novel-coronavirus-2019-ncov-health-alert/coronavirus-covid-19-current-situation-and-case-numbers

https://www.kaggle.com/omnamahshivai/surgical-dataset-binary-classification

https://www.khanacademy.org/science/biology/ecology/population-growth-and-regulation/a/exponential-logistic-growth

https://www.researchgate.net/post/Logistic_Growth_Model_Is_it_suitable_for_COVID-19

https://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer

https://www.youtube.com/watch?v=54XLXg4fYsc

https://www.youtube.com/watch?v=gxAaO2rsdIs

https://www.youtube.com/watch?v=Kas0tIxDvrg&ab_channel=3Blue1Brown

https://www.youtube.com/watch?v=vN5cNN2-HWE&frags=wn&ab_channel=StatQuestwithJoshStarmer

https://www.youtube.com/watch?v=VyWAvY2CF9c&ab_channel=freeCodeCamp.org

https://www.youtube.com/watch?v=yIYKR4sgzI8&ab_channel=StatQuestwithJoshStarmer

universetoday.com/36302/atoms-in-the-universe/#:~:text=At%20this%20level%2C%20it%20is,hundred%20thousand%20quadrillion%20vigintillion%20atoms.

Sadler, A. 2017, *Mathematics Specialist Student Book*