## Preface

Student IDs:

Chris Corby - 12958144

Adam Ozder - 12952471

# Introduction

Logistic regression is a statistical model that uses a logistic function to model binary dependent values given some parameter or parameters.

The input is any real variable, or a linear combination of independent variables, each of which can be either a binary of continuous variable. These values, also called predictors, are analysed to identify the link between them and the dependent variable.

The output of the model is the aforementioned dependent variable, a value between 0 and 1, where 0 represents absolute certainty of the first binary class, 1 represents absolute certainty of the second binary class, and values between represent the shifting probabilities of either class.

# Exploration

## Challenges

We encountered some issues during the process of developing the Logistic Regression model. Some were about the actual implementation of the model; we had significant discussion about the specific details of how we would implement the Logistic Regression model and what the best way to visualize and explore the data was.

Most however were more about the method of implementation through Python. Both of us were only slightly familiar with the language and as such we struggled with some elements of data storage and manipulation, and with creating a way to get Python to visualise the data in a useful form.

Though we did our best to simplify and streamline the process, and cut down the amount of data we had to parse into each function, it is likely that there are many places where the efficiency of the program, in terms of both runtime and memory usage could be greatly enhanced.

### Data Structure

Given the number of elements a machine learning system needs to reliably understand a problem, and the number of elements each machine learning system possesses, we elected to build the Logistic Regression model to take inputs in the form of an array, and output an array of floats between 0 and 1 for each element, which would then be rounded up or down according to the breakpoint function.

## Methodology

Details on the algorithm implementation are included in the code above.

For the purpose of demonstrating our model we used a training and testing dataset about the authenticity of bank notes.

The data set has four independent variables; in addition to the dependent variable. They are:

- Wavelet Variance
- Wavelet Skewness
- Wavelet Kurtosis
- Image Entropy

In addition to those four independent variables the data set also tracks a dependent variable, Authentic / Inauthentic, which is used to mark the true value for the purpose of training the model.

Although we could have used all four independent variables, through empirical testing it was found that two variables had significantly more impact on the end results than the other two, and so we trained the model on only two variables, Wavelet Variance and Image Entropy.

For the purpose of training, we split the dataset at random into two section*s, using 66% for the training set and 33% as the testing set.

## Evaluation

### Results

At the conclusion of training and testing over 100,000 iterations, we were able to achieve an average accuracy rating of 0.85 / 85%. The confusion matrix is displayed at the end of the code block.

The confusion matrix tells us that the logistic regression learning algorithm is good at identifying which banknotes are authentic and inauthentic. It correctly

identified 231 of 255 fake banknotes (91%), and 160 of 203 authentic banknotes (79%).

It is more likely to return a false positive and guess an authentic note is fake (9%) than it is to return a false negative and accept a fake note as authentic (5%).

When it predicted a note was fake, it was correct in 231 of 274 cases (84%).

When it predicted a note was real, it was correct in 160 of 184 cases (87%).

### Comparative Study

Logistic Regression is a form of linear model, analogous to Linear Regression, although it is built upon differing assumptions about the relationship between independent and dependent variables, and the distribution of errors.

Unlike a Linear Regression model, Logistic Regression predicts probabilities of outcomes, which are then rounded up or down to generate a final result, rather than predicting the outcome outright. This allows us, or the machine learning system, to optimize towards cutoff point for the probabilities that best suits the database. However, where a linear Regressor can predict any outcome, a Logistic Regressor can only predict the possibilities of two results, making it a dedicated binary classifier.

There are some cases where a Linear Regression model would be more suitable; any case where the outcome is to be measured as a continuous variable, for instance.

## Conclusion and Reflection

Although ultimately we were able to construct a functioning Logistic Regression model, which functioned with a relatively high level of accuracy, we are somewhat unhappy with the end result. Our implementation was hindered by our poor understanding of Python, as mentioned in the Challenges section*, and it is likely that there are many places where improvements could be made in the code.

Aside from coding optimization, there are likely systemic inefficiencies as well. We implemented only a fairly straightforward and simple logistic regression system and there are a number of alternative development avenues which could potentially have provided greater performance.

We utilized a very basic implementation of gradient descent to optimize the log likelihood by minimizing the loss function. Gradient descent is a very common method due to being relatively simple to implement, however it does in some cases have problems achieving the global minima due to becoming caught in local minima.

Other gradient-evaluation methods, or perhaps even Hessian-based optimization methods, may have been able to achieve a greater result, or achieved a similar

level of optimization in a small time frame, although we did not have sufficient time to implement those methods and test their comparative efficiencies.

Had we been faster and more efficient in establishing our initial model, we may have had time to further explore potential modifications and their impacts on prediction accuracy, and to pursue optimizations in the code to reduce the runtime and memory requirements of the machine learning system.

# Ethical Considerations

Logistic Regression models are commonly used to model the probability of a certain class or event given a binary choice; determining whether a mark is pass or fail, a bet is won or lost, or a patient is going to recover or not. There is unfortunately potential for the model to create ethical issues depending on implementation.

For instance, if the Logistic Regression algorithm is used in a medical capacity to determine the likelihood of a patient recovering from their illness, and for a given patient returns a negative result, then the objectively best choice, mathematically, is to stop providing care and focus medical resources and attention on others. However, not acting to preserve human lives is immensely unethical and requires absolute trust in the assessment of the logistic regression algorithm that the patient could not recover.

There are many other situations which face similar issues; placing trust blindly in the algorithm, even if it is faster or more efficient, removes a level of human oversight and places the weight of responsibility on an algorithm incapable of understanding empathy or morality.

In this sense, given similar inputs, many of these cases may be better suited by removing the function used to narrow down the probabilities into binary options. The percentage values would remove the certainty of the system, but also make it a much more useful tool for human staff in the work environment, who would be able to see the assessment made by the algorithm and how confident the algorithm is in its result, rather than only seeing the final prediction.

# Postface

*Video Pitch:*

https://github.com/C-Corby-016/UTS_ML2019_ID12958144/blob/master/ADA_A2_12958144_12952471_Video_Pitch.mp4

*Python File:*

https://github.com/C-Corby-016/UTS_ML2019_ID12958144/blob/master/ADA_A2_12958144_12952471_Python_Code.py

*Notebook File:*

https://github.com/C-Corby-016/UTS_ML2019_ID12958144/blob/master/ADA_A2_12958144_12952471_Logistic_Regressor.ipynb

*Report PDF:*

https://github.com/C-Corby-016/UTS_ML2019_ID12958144/blob/master/ADA_A2_12958144_12952471_Logistic_Regressor.pdf