In this group project, we were tasked with exploring different techniques for binary classification and compare them using an agreed upon performance metric. To do this we required a dataset with a binary target feature, we chose the 'Breast Cancer Wisconsin' dataset, which contained 30 features and a diagnosis of either benign or malignant. The models we decided to investigate were logistic regression, support vector machines, neural networks and decision trees / random forests, I decided to work on the logistic regression. To decide our performance metric, we considered the standard metrics used in health, these are sensitivity and specificity e.g. true positive rate and true negative rate. The choice of metric then depends on the hypothetical use of our models, if we were screening patients, false positives have little negative effect whereas false negatives mean could have serious, avoidable health complications for the patient, so we would choose sensitivity. On the other hand, if we were doing a final diagnosis, false positives could lead to healthy patients receiving harmful treatment that they do not require, and false negatives are unlikely due to previous screening., so we would choose specificity. It is more realistic that a statistical model would be used for screening, therefore we decided to use sensitivity as our performance metric.

To begin, we set up a Git repository and decided to work in python as we are all familiar with it. We met after lectures to discuss progress and ideas as well as communicating on whatsapp. We decided to work on a single shared Google colab, this is so that we could have a standard train/test split across all models as well as share useful functions. Colab was a useful tool in creating a coherent report, however we ran into issues when working on the report at the same time in different browsers as when one is saved, the others work was lost. This caused some headaches and lost time so next time I would return to working on separate folders.

We agreed to each work on a different model and in my section, I investigated logistic regression. The benefits of the logistic model is that it is very simple and therefore benefits from being easy to implement, interpret and having low computational complexity. Logistic regression has also been shown to be more accurate for some simple datasets than other approaches, especially when the dataset has linearly separable features. It is usually less inclined to overfitting , however in high dimensional datasets, such as our one, this can happen. The simplicity of the model does have drawbacks in that it will struggle to capture complex relationships in the data. Logistic regression is a supervised machine learning algorithm, used to estimate the parameters of the logistic model. The logistic model assumes that the log-odds of an outcome (of the target feature) is a linear combination of some independent variables (features), and the parameters we wish to estimate are the coefficients of this linear combination. To do this we use the log-likelihood function, with an optional regularisation term (such as L1 or L2) to prevent overfitting, and then find values of the parameters that maximise this function using numerical optimisation techniques. L1 (or lasso) regularisation adds the square of the coefficients multiplied by some factor, L2 adds the absolute value multiplied by some factor.

I started by implementing a basic logistic regression and obtained average results, this could have been due to the fact that there were no measures to prevent overfitting and the high level of multicolinearity between features. To reduce the multicolinearity, I tried removing features that reduce VIF. This approach failed to improve on sensitivity, which could be because some of the features I have removed had large coefficients in the first model (like fractal dimension) so our target feature had high dependence on them and removing them lead to too much information being lost. Next, I experimented with different regularisation techniques L1 and L2, these penalties help with overfitting as they penalise

large coefficients to promote a simpler model, this can also reduce the coefficients of highly correlated variables, thus reducing the effect of multicolinearity. I found that L1 improved on the basic model, and that L2 performed better than L1. The intuition for this is that L1 is a more powerful regulariser, driving many coefficients to 0 and oversimplifying the model. And we saw that increasing or decreasing the regularisation factors did not improve results and so our best model was a default L2 (lasso) logistic regression.

Finally, I would say that logistic regression was a good model for me to work on as I have little experience in data science, and I was able to easily understand the methods and implement them quickly. The only negative is that it left me with little room to tweak, and this made it hard to produce ideas to include in my section, and led to fairly simple approaches.