Logistic Regression Model Optimisation and Performance

## Compiling, Training, Evaluating the Model

Text

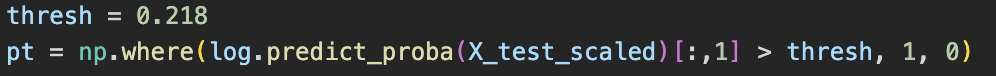
Description automatically generatedWe randomly split and scaled the data for training and testing, keeping 25% of the data for testing the model. Both training and testing scores were relatively high and close together after applying the Logistic Regression model, at 83.3% and 81.4% respectively, and a total accuracy of 81%. This was achieved through tuning the parameters of the model: ‘max\_iter’, ‘solver’, ‘penalty’, and ‘l1\_ratio’.

Chart, treemap chart

Description automatically generatedHowever, the proportion of False Negative results was 16.7%. These customers would churn (leave the company) while our model predicted them to stay, and so would be lost customers. In addition, only 2% of the results were True Positives (where our model correctly predicted the customers likely to churn). In comparison, the proportion of False Positive results was low at 2%; these customers are predicted to churn however would remain with the company. As it is more important for our model to predict those who leave, so the company can carry out any necessary retention strategies, we decided to optimise our model to increase its sensitivity.

## Model Optimisation

Through amending the probability threshold from the default of 0.5 to 0.218, we were able to more than halve the proportion of False Negative results to 8%, while increasing the proportion of True Positives to 11%. Although total accuracy reduced to 77% and the proportion of False Positive results increased, the benefit received from the higher sensitivity would outweigh any cost incurred from the additional False Positive customers.



Chart, treemap chart

Description automatically generated