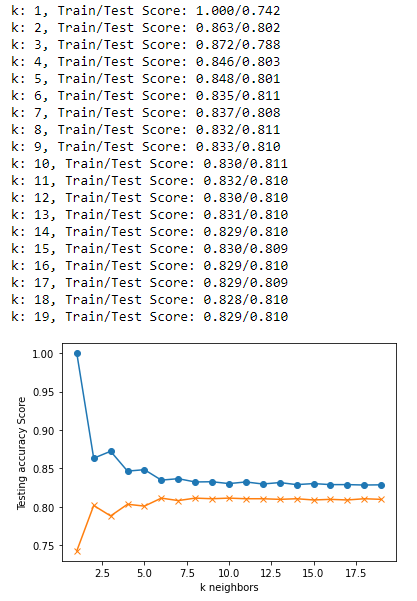
**KNN Model Optimisation and Scoring**

Compiling, Training, and Evaluating the Model

Data was randomly split for training and testing. 75% was split for training and 25% for testing. Data was scaled and initial run of the model was initialised to access the performance of the model with various n neighbours hyperparameter, resulting in an Elbow graph (Figure1). Model has scored quite well with its performance plateau nearing n neighbours of 6. Models’ accuracy resulted nearing 83.5% for training and 81.1% for testing. To optimise model hyperparameters such as: leaf size, n neighbours, and p value we used GridSearchCV library, after it optimised the necessary values, model was run with leaf size =1, n neighbours = 11, and p value = 1. This model’s accuracy was the same as the one that was run the first time. This tells us that model is already performing as best as it can. Finally, for the final model Confusion Matrix was created.

Figure 1. *Elbow Graph*

Model Optimisation

Chart, treemap chart

Description automatically generatedAs it can be seen from the confusion matrix heatmap (Figure 2), there is a significant proportion of clients that were predicted to stay but instead they would churn, also only a tiny percentage of people were predicted to churn correctly. To address the issue, sensitivity threshold was implemented. Through trial-and-error threshold value of 0.23 was used. Which drastically improved the rate of True Positives and reduced the rate of False Negatives, though the rate of False Positives were increased. We believe that the improvement to the model outweighs the negatives. (Figure 3)

Figure 2. *Confusion Matrix without threshold*

*Chart, treemap chart

Description automatically generated*

Figure 3. *Confusion Matrix with the threshold*