Approach:

I started out with logistic regression and few ensemble models to predict the result without any feature engineering. This was a very basic model I created for my reference. The performance metric calculated from the base model was too low. I tried hyperparameter tuning to find the best parameters but in vain. This is where I decided to try out the neural network approach to solve this problem. The initial ANN model gave me better score than other ML models. A lot of parameters in the ANN model was found out by trial-and-error approach as keras tuning method was not giving good result.

I had to encode the categorical features separately and consider them as integer datatype to find out the new features by interaction. I multiplied some of the significant features obtained using statistical tests which I have performed in the python file and multiplied them with non-significant features to obtain new ones. I used Min Max scalar for the numerical variables.

All the encoded features and new features were merged and used as a new dataset for training and testing. Once again, I started with the ML models and tried hyperparameter tuning for some of the models but still the results were not that satisfactory. The score was around 0.43. I switched to ANN and got a very good result of 0.59. I tried keras tuning to find the optimal number of hidden layers, number of neurons and learning rate which improved my score a little better and from there it was trial and error to find the optimal parameters for my ANN model.

Hence, my final model is an artificial neural network containing five hidden layers with ‘relu’ activation function and one output layer with ‘sigmoid’ activation function. I've also added two drop out layers to avoid overfitting after the 3rd and 4th hidden layers. After this I compiled my model with binary cross entropy loss function, Adam optimizer with a learning rate of 0.01 and a recall function to check my metrics. After predicting the final test dataset I used Youden index and a little bit of trial-and-error method to find the best cut off for the predicted values.