Task 2: Binary classification

In this task, it was required to create a ML model for binary classification. I was provided with a training set that contained 3700 records and a validation set that contained 200 records. Each set consisted of 18 variables and a class label. No other information about the nature of the problem or the meaning behind those variables was provided which made the task more challenging as I had no expectations of the outcome.

First, I started off by checking the bias in the training set. The data was strongly biased towards class "Yes." with more than 90% of the records belonging to that class. This made me expect that the model will not have high accuracy but it should have a very high recall.

Then, I counted the number of NAs in each column. Variable 18 had more than 2000 missing variables which made it logical to drop it out and not use in building our model. Other than variable 18, the rest of the columns had a small number of missing values so I removed all rows that have missing values as dropping them out did not largely affect the size of the training set

Next, I started observing the correlations between the different variables to remove highly correlated variables and found out that variable 4 and variable 17 had a correlation coefficient equal to 1. Variable 17 was a multiple of variable 4 so I dropped it out.

The variables in the training set included both continuous and categorical variables. The categorical variables were then converted into dummy variables.

After trying different models (NN, logistic regression, SVM and random forest) , I opted for using random forest as it showed the best performance on both the training set and the validation set. After training the model using all variables but variables 17 and 18, I checked the importance of each variable and found out that variable 1 had 0 importance. I dropped variable 1 and retrained the model and tested it on the validation data and its precision increased from 54.5% to 59.4%

Performance metric

As the training data showed a large bias towards class "Yes." and the validation set had a near perfect distribution between the 2 classes, it was not logical to use the accuracy alone as a performance metric as the model is more likely to predict a yes than a no. In such case, the recall (the true positive rate) captured the performance of the model better as it is expected from the model to predict most of the class yes instances correctly while it is more likely to fail to predict a class No. The table below shows the confusion matrix of the validation set

Actual / Predicted	Yes	No
Yes	91	62
No	2	38

As predicted, the model was able to predict most of the yes labels correctly and predicted around a third of the objects belonging to class no as a yes. The model overall performance can be summarized in the following table:

Recall	97.84%
Precision	59.47%
Accuracy	66.83%