APPENDIX

Glossary

Target column/response variable	The dependent variable which will be the outcome of the model prediction.
Supervised machine learning algorithms	Algorithms in which labelled datasets are used to train the model, for it to classify the data or predict the output. The model would know the target it has to achieve.
Linear combination of features	An expression created from a set of features by multiplying the features with constants and then adding the results to form a linear equation like aX + b Y.
Sigmoid function	It is a mathematical function with curve in an S shape and has the ability to take any real value and map it between 0 and 1. It is also known as logistic function.
Decision boundary	A boundary line created by the classifier to segregate the data into different classes.
Threshold	It is a barrier point based on which the probability outcome is converted into classification by assigning the probability into one of the labelled classes.
Gradient descent	It is an optimization method used to minimizer function by following the gradients of a cost function plot. Cost function is a measure of how bad the model that uses the estimation of losses of the model during the training phase.
Overfitting	This is a condition where a model tries to fit all the data during the training, but is unable to predict the relationship among the data precisely.
Hyperparameters	It is a constituent of model configuration which helps the model in learning process and can be used to tune or optimize the model.
Complete separation	A condition when all the predicted values are perfectly classified into the outcome classes.
Underfitting	This is a condition where a model is neither able to fit to the data, nor is able to predict the relationship among the data.
Decision Tree	A Supervised machine learning algorithm where the decision making is done in the form of a tree starting from the node of the tree and concluding at the leaves.
Bagging	A Process in which subsets of data are made randomly such that a subset may contain repetition of the same data point and these subsets are assigned to individual decision trees. The final output is given by averaging the predictions of each individual decision tree or by considering the majority vote made by the decision trees.
Ensemble Techniques	Techniques used for grouping the classifiers in a particular fashion.
Holdout Method	A technique in which a data set is split into training and test sets based on the proportion of partitioning the data given by the user.
Cross Validation	A method to estimate how well a machine learning model can perform in general for any data.
Link function	A way of assigning a nonlinear function to a linear function.
Diagnostic plots	Plots of observed diagnostics for trained model.
ROC curve	Receiver Operating Characteristic curve is a plot of false positive rate against true positive rate of prediction of classes.
AUC	Area Under Curve represents the ability of the model to precisely predict the labelled classes.

True positives	It is the value of the positive class data of a classification problem that have been predicted as positive and are actually positive.
True negatives	It is the value of the negative class data of a classification problem that have been predicted as negative and are actually negative.
False positive	It is the value of the negative class data of a classification problem that have been predicted as positive but are actually negative.
False negative	It is the value of positive class data of a classification problem that have been predicted as negative but are actually positive.
Sensitivity	It is the fraction or proportion of positive class of a classification problem that have been correctly identified as positive.
Specificity	It is the fraction or proportion of negative class of a classification problem that have been correctly identified as negative.
Precision	It is the fraction of positive class that is identified correctly out of all the positives including false positives.

Intermediate results including any negative results

Logistic Regression

The threshold for logistic regression was determined after running multiple iterations for different values. For each value between 0.1 and 0.9 the error of the trained model and the AUC for train model were evaluated and the best AUC was achieved for 0.5 as threshold value. Then further trials for values between 0.5 and 0.6 showed that the best threshold was achieved for 0.52.

The diagnostic plots of the trained model indicated that the shape of the curve for each variable together formed a curve which is almost like a sigmoid function. But to check how bad the model will perform if some features are removed additional trial was made to train a model with some reduced features and as expected the error of the model increased drastically.

Although cross validation was not available or applicable for *fitglm* in MATLAB, there were defined functions created by other MATLAB users which would cross validate a logistic regression model. But using that function only cross validation error could be calculated. There was no means to further evaluate that cross validated model using any performance matrix and hence that section was omitted from the code.

Random Forest

After training the second model which is the first iteration to improve the baseline model, a plot of classification error versus number of trees was plotted. Although 100 was the optimal value of number of learning cycles as given by the machine for the baseline model, from the plot mentioned above it appeared that the least error was when the number of learning cycles equals to 27. So an additional trial run was made with number of learning cycles as 27 and the error of the model for that run was same as that for original number of learning cycles of 100. So this section was omitted from the code. This was perhaps due to the fact that even though the error was lowest for 27 number of trees but the plot of error classification was stabilised as it approached 100.

Initially while writing the code for model 5, it was tried to assign the method directly as a value of best hyperparameters variable assigned, similar to the usage done for number of learning cycles and the minimum leaf size. But unlike them, the assignment of a method for the ensemble code could only be given as a string value and that too, it is to be selected from the options given by the function itself. Show the value of method of ensemble was entered manually.

APPENDIX

Implementation details including main implementation choices

Heat map in MATLAB was most useful to compare the relation between two columns but it was not as flexible as Python for comparing correlations among all the variables of the data and hence the heat map of correlation among the variables was done in Python.

Similarly, the descriptive statistics such as mean, standard deviation, quintiles, etc could be evaluated in MATLAB, but only individually and not as a whole set. Whereas in Python it can be done as a whole set using the described method on the data frame, so it was done there.

Out of holdout method and Kfold method for cross validation partition of the data into training and testing sets, the Kfold is definitely better one, but only if the models are to be cross validated later. In our case random forest can be cross validated but logistic regression cannot be. So instead of Kfold method holdout method was chosen for cross validated partition of the data.

Logistic Regression

Since the relation of the variables in the data was a linear relationship the distribution for *fitglm* was considered as binomial.

Since the task was a binary classification the *fitglm* function was preferred instead of *mnrfit* which is more useful for multinomial logistic regression.

Since there were not many options to optimize or improve the logistic regression model available on MATLAB, there were no trials for improving the model. Additionally, since it's a generalised model it can be expected that the performance of the model would be consistent and reliable.

Random Forest

Although, *Treebagger* was an alternative MATLAB function for random forest, the *fitcensemble* function was chosen keeping in mind the advantage of "*optimizehyperparameters"* argument available only to *fitcencemble*.

For the baseline model MATLAB chose one of the boosting methods most of the time. But for the sake of accuracy boosting methods may lead to high variance and hence bagging method was chosen to reduce the variance.

In some of the trial runs of model 4, in which the hyper parameter optimization was set to auto, it was observed that the error was the least in certain runs because the method chosen by the system as a part of optimization was one of the boosting methods. But this trend was not consistent so this model who is considered not to be consistent.

Similar was the case for model 5 and again it was not performing the best consistently and hence was not considered as the best model.

Initially execution of codes for all the models for both logistic regression and random forest were done without seeding. Then it was observed date the results of error and accuracy were different for every run or every instance of execution of the code. Therefore, seeding was done for every section of training a model for both the algorithms.