A machine learningtechnique called a **random forest** was used to predict **x**. The data journey, particularly the inner workings of the technique, are described step-by-step below:

1. **Data collection:** Data tables were extracted from data storage within **{insert organisation}**. The tables were merged into one large table for data cleaning.
2. **Data cleaning:** The data table was cleaned to ensure accurate predictions of **x** could be made from the data. First, data variables irrelevant to the project were completely removed from the table. Next, missing values were identified and substituted with the average (mean) of that variable or middle number (median) of the range of values of that variable, to ensure the data captured an approximate and statistically complete picture of **x**. Finally, a process called **feature engineering** was performed, where additional variables were calculated from existing variables to enhance the accuracy of predicting **x**. A full list of all variables used to predict **x** can be found on page **x**.
3. **Predictor and Target variables:** The variables used to predict **x** are called predictor variables. A separate variable that identifies whether **x** occurred or notwas also computed from predictor variables within the dataset. This variable codifies the correct **x** result to be predicted and is called a target variable. Both predictor and target variables are listed on page **x** and were used in subsequent steps.
4. **Train-Validation-Test split:** To place data (predictor and target variables) into the random forest technique, a process called **train-validation-test splitting** was performed. Here, the predictor and target variables were split into three groups. The first group was called a training dataset and contained 80% of the overall data. The second group was called a validation dataset and contained 10% of the training data. The third group was called the testing dataset and contained the remaining 20% of the overall data. For this project, many models were initially trialled to accurately predict **x**, before concluding that the random forest technique was the best model for the problem. To make this assessment, the validation dataset was used to identify the lowest error in predicting **x** among all trialled models. The validation dataset was mentioned here for completeness of data processing however, where the random forest technique is concerned, the main focus for Steps 5-7 below are the training and testing datasets.
5. **Bagging:** To allow the random forest to map underlying data patterns and learn to accurately predict **x**, the training dataset was introduced to the technique. Within the technique, repeated samples were taken from the single training dataset in a process called **bagging**. In the first instance, a random forest is a collection of smaller data learners called **decision trees** (discussed in Step 6). Decision trees have two key limitations that bagging serves to address. The first limitation is that decision trees have low predictive accuracy of data compared to other techniques that classify data. The second limitation is that a small change in supplied data can cause a large change in the final data prediction from a decision tree. Bagging allows data variation to be modelled through repeated sampling of the training set and improves predictive performance of decision trees by aggregating trees within a random forest. For each repeated sample, two-thirds of the original training dataset was used, with the remaining third being used to output predictions (discussed in Step 6f).
6. **Building a model:** Each repeated training sample was fed into the basic building block of a random forest technique, the **decision tree**. Decision trees split data observations into simple groups, the details of which are captured in Figure 1 and are described below.

**A picture containing diagram

Description automatically generated**

**Figure 1:** A simple decision tree.

In a random forest, a decision tree is grown on a repeated sample in the following way:

1. **Selecting predictors:** A random portion of predictor variables is selected from the total number of predictor variables in the repeated sample. The number of selected predictors is approximately equal to the square root of the total number of predictors.
2. **Creating the tree root:** Only one predictor is used from the random portion of predictors to split data into groups. The grey dot at the top of Figure 1 represents the first predictor variable and data split that initiates splitting of the repeated sample into more refined data groups. This first decision point is called an **internal node**.
3. **Growing the tree:** From the first internal node, subsequent internal nodes are created as described in 6A and 6B, with a fresh portion of predictors taken for new splits. In each new split, data is placed into ever more distinct groups in a two-group or binary fashion.
4. **Stopping tree growth:** Data splitting stops when the decision tree determines that data observations are in distinct groups. Here, data observations cannot be split further and are placed into **terminal nodes** or **leaves**, presented as black dots in Figure 1.
5. **Tree pruning:** Throughout sub-steps 6A-6D, as many predictor variables as practicable are used to split repeated samples into distinct groups. This results in more complex decision trees than illustrated in Figure 1. In addition to the diagrammatic complexity, complex decision trees can use predictors that do not contribute very much to splitting data into distinct groups. Therefore, large decision trees are **pruned** to obtain simpler trees with all the possible relevant data splits and predictors that contribute to these splits in it. During pruning, the random forest technique uses a quantitative variable called the **Gini index** to measure how effectively a predictor variable categorised data into groups. The Gini index measures the number of odd data observations out in a decision tree node, that do not belong to the most common group of observations in that node. A small Gini index identifies a terminal node with observations predominately from a single group, and therefore indicates a better data split. The total amount that the Gini index decreased by data splits over a given predictor is summed and then averaged over all decision trees built within the random forest, to rank predictors by how effectively they split data. For this project, the rank order of predictor variables by Gini index can be found in table **x** on page **x**.
6. **Predicting from data:** Repeated samples comprising two-thirds of the training dataset are used to fit each decision tree of a random forest (sub-steps 6A-6E). The remaining one-third of training observations (called **out-of-bag observations**) are used to predict the group or **class** of training data observations. For a given out-of-bag observation, the majority class in the terminal nodes of each decision tree is recorded. To obtain the overall prediction of the out-of-bag observation, the most commonly occurring class among all the out-of-bag decision tree predictions is selected.
7. **Testing the model:** As the final step of the data journey, the testing dataset is used, as in sub-step 6F, to predict the class of data observations. These testing predictions are then compared to the target variables (previously mentioned in Steps 3 & 4), which represent the actual **x** result of the data observation. As the testing dataset was not used to train the random forest technique, a comparison of the testing predictions to the actual outcome of the observation are a method to identify how well the random forest performs on data it has not been exposed to before. This is an important step to determine how effectively the model will provide predictions for new data.

# **References**

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning (Data Mining, Inference and Prediction).* New York: Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning (with Applications in R).* New York: Springer.

Nord, C., & Keeley, J. (2016). An Introduction to the HPFOREST Procedure and its Options. *MWSUG 2016*, 12.