Employee Attrition Dataset

Rory McStay
14323268
mcstayr@tcd.ie

January 4, 2018

1 Introduction

The aim of this report is to record the process of building a classification model to predict when an employee will leave an organisation. We will begin with a brief description of the dataset and what preparation and cleaning was necessary in order to build our models. In the following sections, we will examine a handful of variables in greater depth based on their importance or predictive power. We will then document the procedure of how we came to our final models. We will discuss a single tree model and two ensemble techniques. Finally, we will evaluate our models both in there own right and in comparison to other models built.

1.1 A Brief Overview of the Dataset

The dataset consists of a target variable *Attrition*, which classifies each of our 1470 observations as yes or no, where yes implies that they left the company. 84% of cases are classified as no. We have 34 more variables outlined in table **??**. A number of the variables consisted of Survey like data assuming values 1 through 5. These variables will be considered as both as both ordered factors and as continuous. The data will be explored with these variables as ordered factors and experiment with both ways of displaying them in the model building stage.

1.2 Data Cleaning and Preparation

The first first item to consider is variables with zero or near zero variance. We will then consider missing variable cases and attempt to determine a reason as to why they are missing. Stock option level consisted of a high number of missing value cases. It is assumed that these missing cases are equivalent to zero or no stock option level. However, after doing so the variable had a variance close to zero so will provide little explanation of the attrition likelihood of an individual. The raw dataset also consisted of two more variables with a variance of zero, namely whether or not the individual was over 18 and the employee count. These variables were removed for analysis as they provide no explanatory power.

1.2.1 Derived Variables

A number of other derived variables were created from the dataset provided. These included a demographic variable, where employees were grouped into 5 equal groups of non overlapping age ranges. This was mainly done for easy graphical analysis of age as an explanatory variable. The number of hours worked in a month was created from dividing the monthly income by the hourly rate. As well as variables created directly from the dataset, a number of variables were created for model building and exploratory analysis. A grouping variable was created through hiearchial clustering methods to determine an underlying structure to the dataset. This was used in the exploratory phase as well as an input into the models built to predict the attrition rate. A support vector machine was run on the whole dataset for novelty classification. This was done to classify observations as outliers. This provided a way to determine if an observation was significantly different from others whilst accounting for all variables in the dataset. A principal component anlaysis was conducted in the explanatory phase for dimension reduction. These variables were also used in the model building phase.

2 Exploratory Analysis

This section outlines the steps taken in understanding the dataset. The purpose of this is to gain insight into what variables are important, determine the factors and trends as to why an individual may leave the company to aid model building. It will also offer insight as to how later models make their decision boundaries.

2.1 Correlated Variables

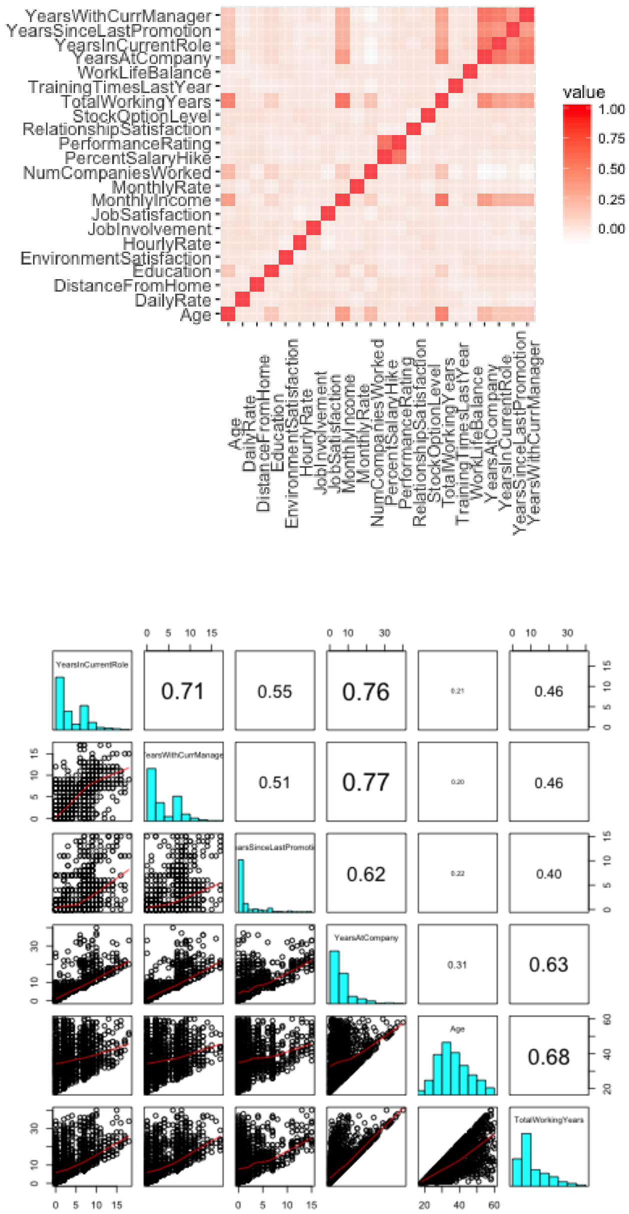


Figure 1: Correlation matrix and time variable set

A simple heat map of the correlation matrix (figure 1 Top)of continuous variables allows the visualisation of correlations amongst variables in the dataset. Multi collinearity does not appear to be a problem except for certain subsets of the data. The north east corner contains time related variables in relation to that company which are all highly correlated along with total working years and age. These variables may cause an upward bias of their importance in a fitted model. Examining the scatterplot matrix(figure 1 Bottom), illustrates the duplicate information within this set of variables.

Years in current role and years with current manager have a strong linear relationship. One can see there is both instances where managers and teams both move role, and where a new manager has taken over. However the strong linear relationship is likely to cause multi collinearity issues so will not be included in an optimised dataset . The pattern appears to be somewhat present amongst other pairs as expected as there is a strong linear component to the relationship amongst all variables and this is the threat. However the problems of including all the variables will not out way the loss of additional information outside of the linear component. We will seek a linear combination of these variables as to maximise the total variance amongst these variables to replace them in future models.

Examining the distribution of time variables, helps to explain the demographic of the company. The average age is 36, the average working years is 11 and the average time spent at the company is seven years how ever the mode lies between 2.5 and 0 years, The distribution of experience an individual has with this company is positively skewed, suggesting a large proportion of the cases have less than average experience. This could be a reason as to why individuals may not want to be with the firm. 40 % of observations recorded 0 years since last promotion which suggests that the rate of career progression is high. Points on either side of the fitted line for years with current manager & years in current role contain information of how managers change or not with role and how new managers come in. Likewise for years since last promotion & years in current role providing information of the type of promotion.

Categorical variables with three factors can be created for these variables for each logical condition $<, >, =$ and are illustrated by figure 2. Analysis of this mapping begins by highlight segments with high proportions of attrition. Comparisons can then be made across other segments of that factor. This will allow visualisation of potential decision boundaries of a classification model. The most clear instance of this is that all sales representatives which got promoted to a new role and new manager left the company. This is perhaps an indication of the quality of management in that hierarchy of job role. It also describes what type of movements of workforce occur within the company. It illustrates what type of roles have relatively more movement of workforce within roles by the area of the bars in the bottom segment and what roles have relatively more people entering by area of bars in the top segment. research scientists, laboratory technicians and sales executives see the highest rates of entry into that role with no promotion. An individual is promoted into the same role more often than they change role and more likely to change role through promotion than without.

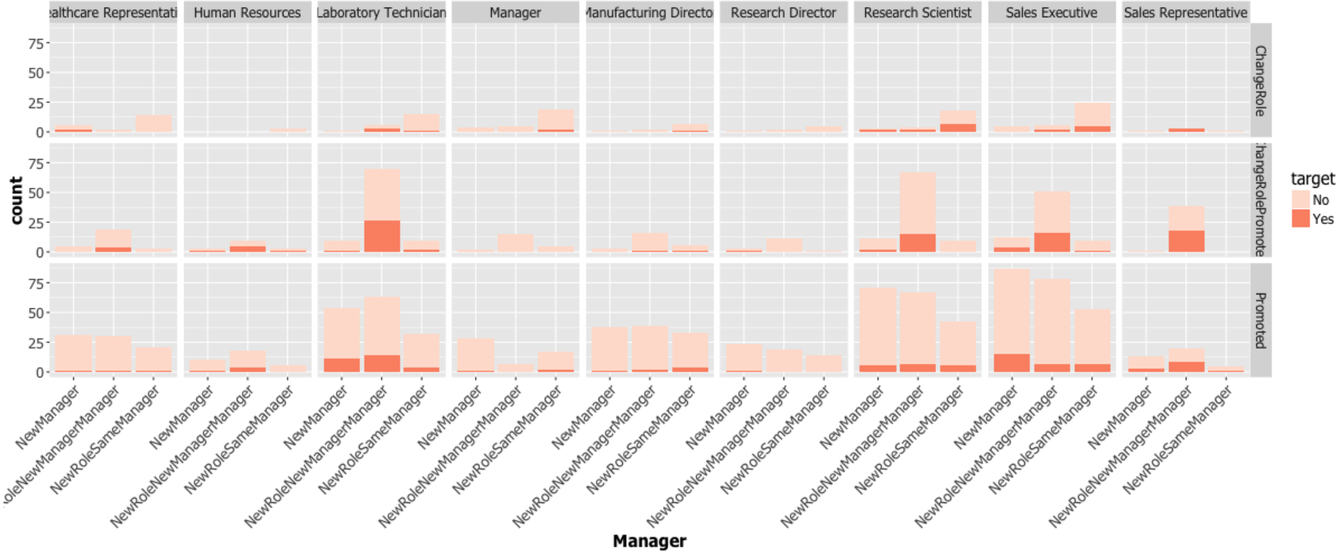


Figure 2: Managers and Promotions

2.2 Clustering

Exploratory analysis working towards a classification problem must consist of attempts to find underlying structure in the data. Investigating clustering techniques offers insight into how a classification model may define its decision boundary. This can be then applied to ensuring the relation is captured in later models. A Gower distance matrix which is a mixed dissimilarity measure was calculated and returned similiar observations within groups but poor dissimilarity between groups. The Gower dissimilarity measure found to agree on all factor variables and age for the most similar pair of observations (0.073) but failed to disagree with factors on the most dissimilar pair of observations (0.7). As such, no natural groupings were founded through this clustering method.

An alternative method of measuring the dissimilarity of observations is to use a proximity matrix returned from a random forest. An unsupervised `randomForest` model was run on 500 trees to return a proximity matrix. If two observations fall into the same node in one tree, their proximity is increased by one. The proximity score is then normalised by dividing by the number of trees generated. Through this method of clustering, most similar pairs of observations agreed with the majority of factor variables and had high agreement in time related variables. However, the pair of most dissimilar observations also agreed on factor variables but disagreed on time related and income variables. As there was agreement of factors in both the most similar and dissimilar pairs of observations, no discernible grouping was determined.

Considering the fact the proximity matrix from the random forest was generated with attrition as the target variable and that observations were deemed similar or dissimilar based on time related variables. It was determined that time related factors and experience with that company were the most important variables in determining whether or not an individual will leave. After examining the variable importance of the random forest used to generate the proximity matrix, this observation made sense as time related variables occurred most frequently in the top 10 most important variables as defined by the mean decrease in the gini measure.

This suggests that observations are not all that dissimilar. If this method of analysis returned groupings with uneven proportions of attrition across its groupings, it would provide later models with a headstart on classification. Despite this not being the case,if the clusterings do bear significance, including them in models will be beneficial. By picking a relatively small number of segments, the bias associated with variables with many factor levels causing a bias, its potentially unnecessary inclusion will not be a concern. It will also be possible to choose based a value for the number of groups as to best pick the proportions of the target based on the training data. This way the relevant information gain is maximised.

2.3 An Analysis of Principal Components

In search of a pair of dimensions to visualise decision boundaries to gain insight into the classification problem, a principal component analysis is conducted. By creating two new variables which encapsulate each observation's original values, such that the new variables accounts for as much of the total variation as possible, the visual interpretation of the data is greatly increased.

We begin by examining the scores obtained of each individual for the first two components. Two relations become apparent through doing so. Moving in the positive x direction, point sizes become smaller. This suggests that Job Level is an important variable in the first two components. By examining loadings in the PC1 direction, we can see that age, total working years and number of years at company have large negative loadings for PC1. This would suggest that the larger these variables are, the individual will have a relatively greater job level. Thus implying, PC1 captures seniority of an individual in the workplace. It also appears that the frequency of attrition is greater, the higher the PC1 score. This is indicated by the number of dark red points to the right of the origin compared to the left. By considering the associated loadings mentioned previously, it is reasonable to assume that an individual is less likely to leave a company when they have relatively higher experience with that company, with the exception being when an individual reaches retirement observed by labelling of those over 57.

To examine this relation further, turn attention to the box plot of PC1 scores by Job Level. The average PC1 score is strictly decreasing through higher job levels. Considering the previous interpretation of PC1 as seniority, observe that the average PC1 score for job level 1 is lower with more outliers in the negative PC1 direction for those who did not leave. This suggests that individuals with a higher level of seniority at lower job levels are less likely to leave. This relation holds up until after level 3. Where there is a significant difference in average PC1 for those who left and those who did not. A higher PC1 score is associated with a higher age, so may suggests individuals with a lower PC1 score in higher job levels are approaching retirement.

Examining the line plot of job level and attrition rate, the rate of attrition by job level, is highest amongst those with the lowest job level. It has an overall decreasing trend but increases at three and five. An individual is more likely to leave at job level three than job level two. Also notice that the difference between average PC1 score for those who left and those who did not, job level held constant, is correlated with the change in attrition rate through varying job levels. Suggesting that, those who are not on the appropriate job level for their level of seniority, will leave the company to go elsewhere. This can be seen by examining the outliers at job level 3. The PC1 scores are larger for those who did not leave.

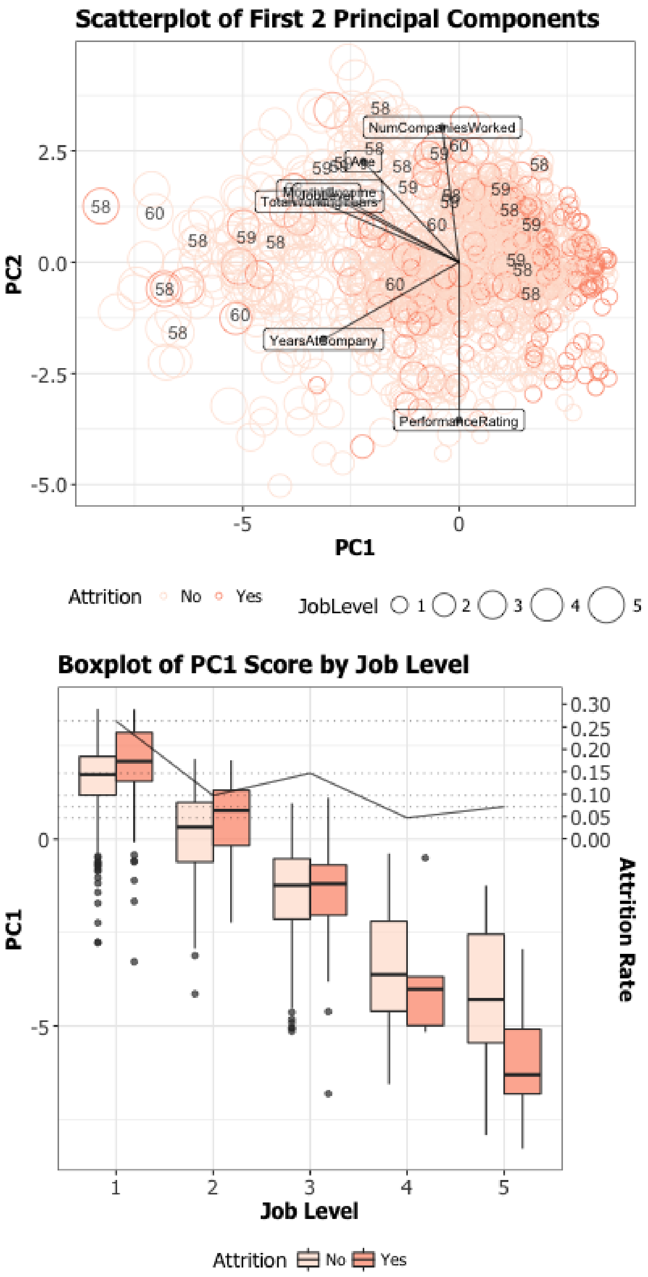


Figure 3: Principal Component Analysis

Examining the PCs envokes good seperation in the data. Therefore the inclusion of PCs will be considered. Where there is variable selection bias present in models, variables in question will be replaced with linear combinations derived from an eigendecomposition. In this way, un correlated variables are put in therir place with as much information retained as possible.

2.4 Why Individuals May Leave

Previous sections began to consider why an individual would leave based on their level of seniority in the firm and associated job level was given. This will now be explored in finer detail, why an individual will leave a firm. Attrition can occur in a work place for a number of reasons. Involuntarily, where an individual is no forced to resign or is fired. Voluntarily where an individual chooses to leave as the opportunity cost of working at that firm is too high (Obtained a better work offer, not enough compensation for working or other commitments) or reaches retirement. In this way, a motive may be defined as to why it was set out to make these predictions. How later models satisfy these motivations will be considered.

Table 1 shows the average job level of each job role, ordered by the corresponding rate of attrition. There exists a negative relation between job level and the rate of attrition. The highest rate of attrition out of all job roles are sales representatives. Why is it that this rate is almost double that of any other position as well as the average job level being considerably lower than any other for this position.

Figure 4 shows there is significant difference in the average performance rating for sales representatives who left the company and those who did not. It is also clear that the percent salary hike is in line with this observation. The average hours worked and job involvement have no significant difference for sales representatives. This perhaps means that sales representatives with lower performance ratings do not receive a salary increase and as such a proportion may leave the firm due to lower job satisfaction or involuntarily leave due to a lower performance rating.

Research directors with higher job involvement and hourly rate are more likely to leave the company. This would suggest that the better Research Directors leave the company perhaps due to being *poached* by other firms. However, there is insignificant difference in the average performance rating of those who left and those who did not. From the aggregated dot plot, in figure 4, we can see that Job Roles with lower average hours worked were more likely to leave the company. This is illustrated by the inverse relationship of the first and second plot in figure 4.

In order to arrive at a final model, it is important to consider why you are trying to predict. In this instance, is it to identify an employee who will leave to try reverse that outcome? Then a model may be seaked out as to minimise the false negative rate. This way of analysis allows for the consideration of the different environments a positive outcome may have occured from when evaluating later models.

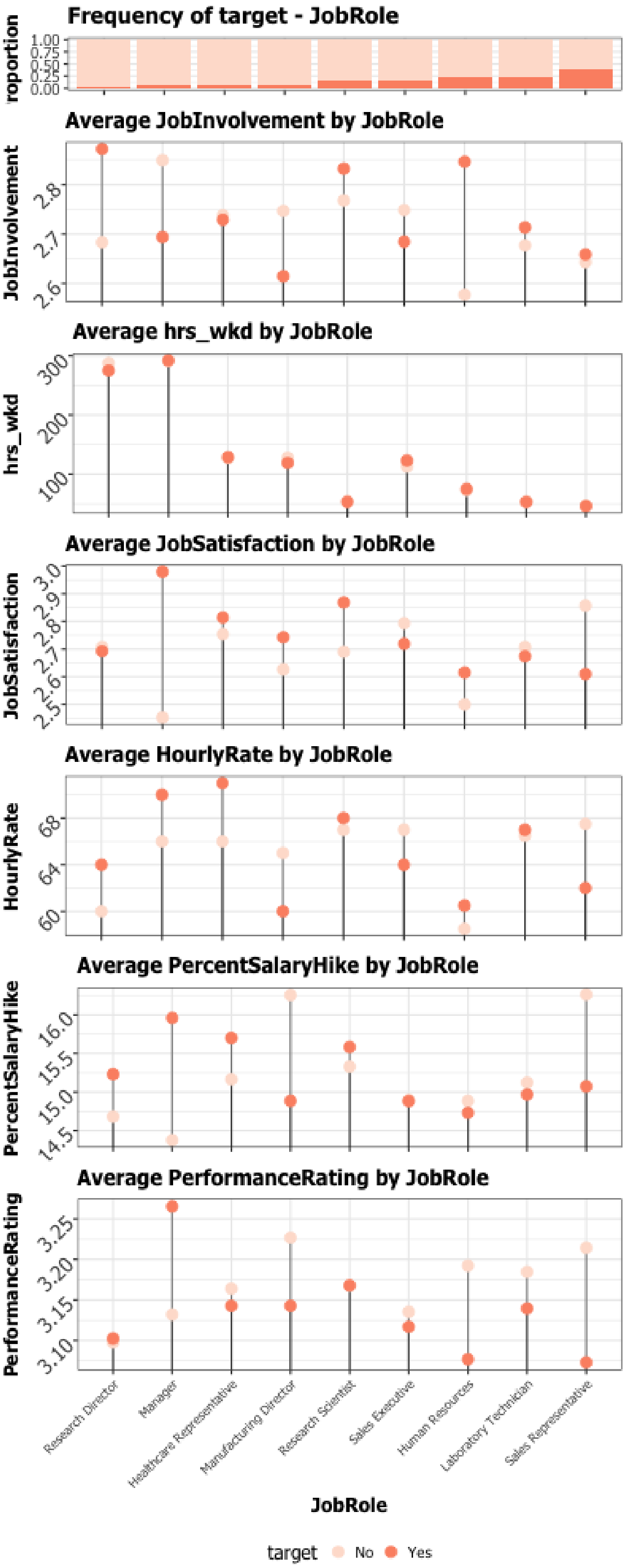


Figure 4: Attribute Analysis

3 Model Building

In this section, we describe the steps taken in building models to predict whether or not an individual will leave the company. We begin by discussing the considerations made when generating our test and training data. We then fit a simple tree model to our dataset and discuss its performance and what we can learn from it in building later models. We will then describe two ensemble techniques used in predicting the attrition rate. Throughout this section, we will discuss the evaluation of our models where necessary to describe why certain model characteristics were chosen. Later on, we will compare and evaluate our final models through misclassification rates and ROC curves as well as subjective considerations as to why either model may be more applicable to the task of classifying employee attrition and implementation.

3.1 A Single Tree Model

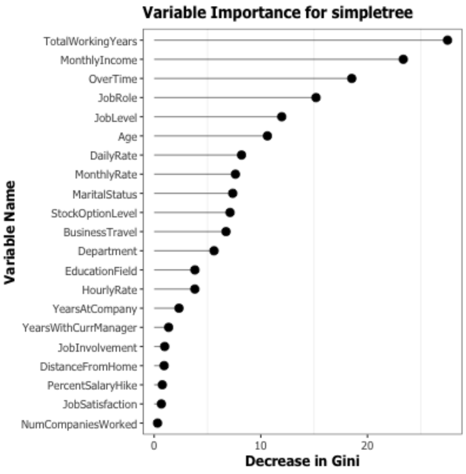


Figure 5: A Simple tree model

In this section, we will discuss the process of how we arrived at our single tree model for classifying whether or not an employee will leave the company. We will outline the considerations made of what variables to use and how the observations made in our exploratory analysis may impact our final model. We will first discuss the variable importance as defined by a single tree on the complete dataset.

3.1.1 Methods for growing trees

The standard decision tree for classification purposes described by the CART algorithm with a Gini purity measure was fitted to the raw dataset. We will use this as a foundation for building future single tree models. An 1000 bootstrap replications of the AUC value were generated and returned a value of .699 with a standard error of .0305. To visualise what it is that this model is doing, we can illustrate the fist split on a selection of the most important variables total working years, monthly income and overtime through a scatter plot below. The green region corresponds to the 11th node consisting of a 52.5% portion of individuals classified as yes. These are individuals who do not complete overtime, have a monthly income of less than 3745 and have worked less than 1 year. We compare this method of This is an example of how our simple model correctly picked up on our observations of experience in the workplace has a negative relation with the likelihood of leaving the company. Therefore, we will consider adding our first principal component to our tree model as this did a good job of describing an individuals work experience or seniority. We will do this by both adding it to the original dataset and replacing it with variables with a significantly large weight in the PC1 loading. As over time was identified as an important variable, We will include our variable for working hours in a week derived from monthly income and daily rate with the hope that a continuous type measure of the number of hours an individual works will have greater explanatory power. This will be both replaced with over time and added to the original dataset. It was also considered to use a conditional decision tree where by a test of proportions is done on every possible subset of the data. Suppose we have a dataset Y, can we partition the dataset on x_j such that we reject the null hypothesis $D(Y|X_j) = D(Y)$ for a given test of proportions. We stop growing the tree until we cannot reject the null hypothesis for a given level of significance α. This method provides a more statistical approach to growing trees however it failed to outperform a tree with a gini index defined split so it was decided to continue the development of a single tree model with the CART algorithm.

3.1.2 Model Training Data

The frequency of our target variable in our complete dataset is 84% of no cases and the rest yes cases. The first consideration in sub setting our dataframe is whether or not we use stratified sampling where by we set a desired proportion of yes and no cases to appear in each subset for model training and testing. We began investigating this by bootstrapping the average attrition rate for a random subset of 66% of the data, our desired training set size.

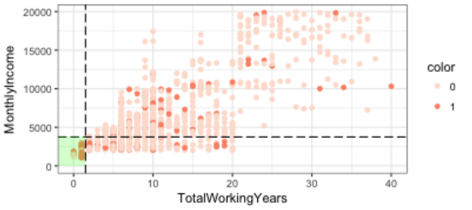


Figure 6: Data subsetting

3.1.3 Derived Variables

The derived variables we discussed in our initial analysis were tested within a single tree model. Due to the explanatory power of the principal components witnessed in the previous section, we investigate the explanatory power of the observed PC scores for each case. This was done initially by fitting a decision tree to our target variable, including all variables in our original dataset, as well as the PC scores. This failed to outperform a simple rpart tree with default parameters. Upon consideration, decomposing principal components of a dataset seeks to maximize the variance captured with as few linear combinations of the whole dataset. The linear relationship amongst variables may cause problems in a single tree model due to it the model complexity necessary to pick up on linear relationships. A single tree works by picking the best split of the data. A decision tree will only accurately describe a linear decision boundary once the number of subsets is relatively large. This can lead to over fitting. To combine principal components to our model, we will generate a new variable by running a general linear model on all the PCs and the predicted variable will be used as an input into our tree model.

In our initial single tree model, two time related variables mentioned in our analysis of correlations appeared prominently in our model, figure 5. Our concern is that models which contain similar information may promote a bias. To mitigate this, whilst including the maximum amount of information in our inputs, we seek out a linear combination of the time related variable set which maximises the variance of that set. A principal component analysis of the time variable set was conducted and the components which accounted for 85% of the variance in the data were selected. This was the fist two components. Adding the scores of PC1 and PC2 for each individual into our model, whilst we removed the time related variable set yielded an AUC of 0.775 with a standard error of 0.0298 standard error. This proves to be significantly different AUC value from our initial single tree model. The first principal component of the time related variable set achieved the greatest decrease in the gini coefficient in this model (figure 5). Thus we will include it in our single tree model.

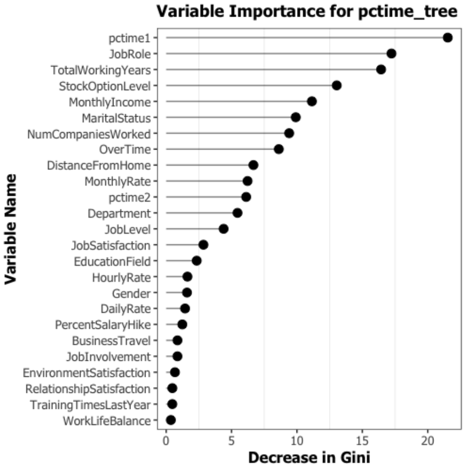


Figure 7: Combining the time set

The next derived variable we will investigate is a general linear model prediction based upon all the principal components of the model. The GLM model alone managed to achieve a bootstrapped AUC value of .76 and .01 standard error. This offered significant improvement over a GLM on the original dataset. Decomposing the data into its principal components increased the AUC from 0.63. However, adding the predictions into our model failed to make a significant improvement to the performance of our model.

From our exploratory analysis, we attempted to find a grouping structure within our dataset using clustering techniques. Following on from this and now using it as an input into our model, we used the dendrogram to split the data into k groups for a particular k which gave us the greatest frequency of attrition for anyone level k. The level of k which achieved this was k=3. This yielded unequally sized groups with the frequency of attrition of the largest group was equal to 25%. Using this new factor variable with three levels, along with what we learned in previous models, we managed to achieve an AUC of 0.78 from 1000 bootstrap replications, with a standard error of 0.021.

3.1.4 Tuning our model

Now that we have established the inputs that we will use in our single tree model, we will now consider the effects of the tuneable parameters in our tree growing method. In all of our previous fitments of a decision tree on our model, we never considered specifying prior probabilities of classification. The reasonable assumption to make in this scenario is to specify the average rate off attrition as the prior probability. We then checked this through cross validation. A model was built for every possible combination of prior probabilities in .01 intervals. The maximum value for AUC achieved was with the sample frequency rate of attrition. So this is what we used in our final model.

The next parameter we will investigate is the complexity parameter. Through examining the cross validation of this parameter, the value of 0.012 achieved the lowest relative error. This parameter presents a tradeoff on model complexity and accuracy. The lower the CP value the greater the number of splits in the tree. By specifying a complexity parameter, the tree will only split at a node if the improvement in relative error is greater than that CP. The default in this package is 0.01 which is close enough to our tuned parameter so adjusting this downwards will make little difference to the performance of our model.

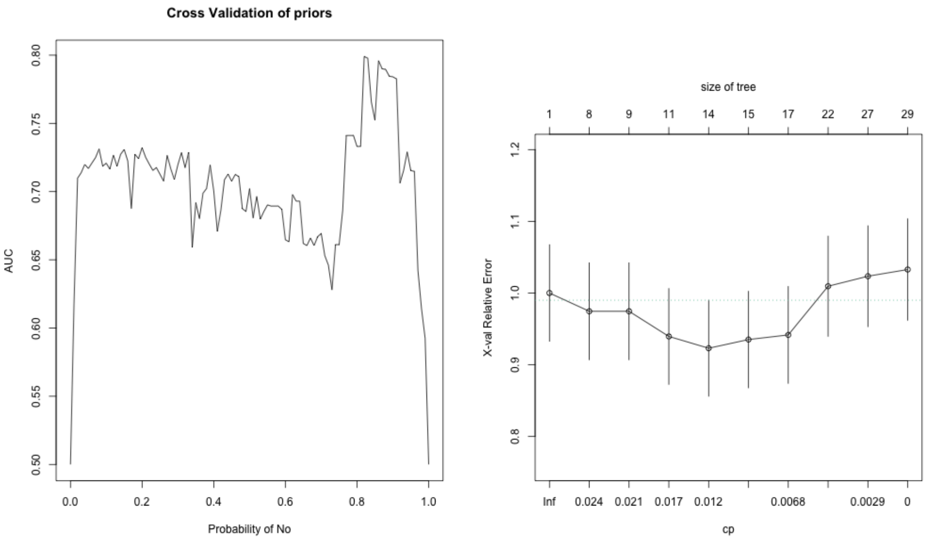


Figure 8: Cross Validation

3.1.5 Examining the Trees Output

Using `rpart.plot()` and increasing the complexity parameter of the tree, it is possible to show a simplified version of our tree. At the first split, if PCTime 1, expressed in section 3.2.3 is greater than or equal -12 then the case will go to the left. This splits the cases up in an 80:20 split. 80% going to the left and the remainder to the right. Individuals with less experience go to the right. Experience is penalisd most by lower scores in age. This split improved the proportion of yes cases by 7% in the right node. Then, overtime is the next best split in both instances. If a case is recorded as yes for overtime, they were more likely to leave the company according to the model. This graph reinforces the conclusions made in the explanatory section. There were significant differences found to exist when subsetting the dataset in experience related dimensions such as Job Role and Job Level. As well as time related variables being a critical factor, based on a principal component analysis.

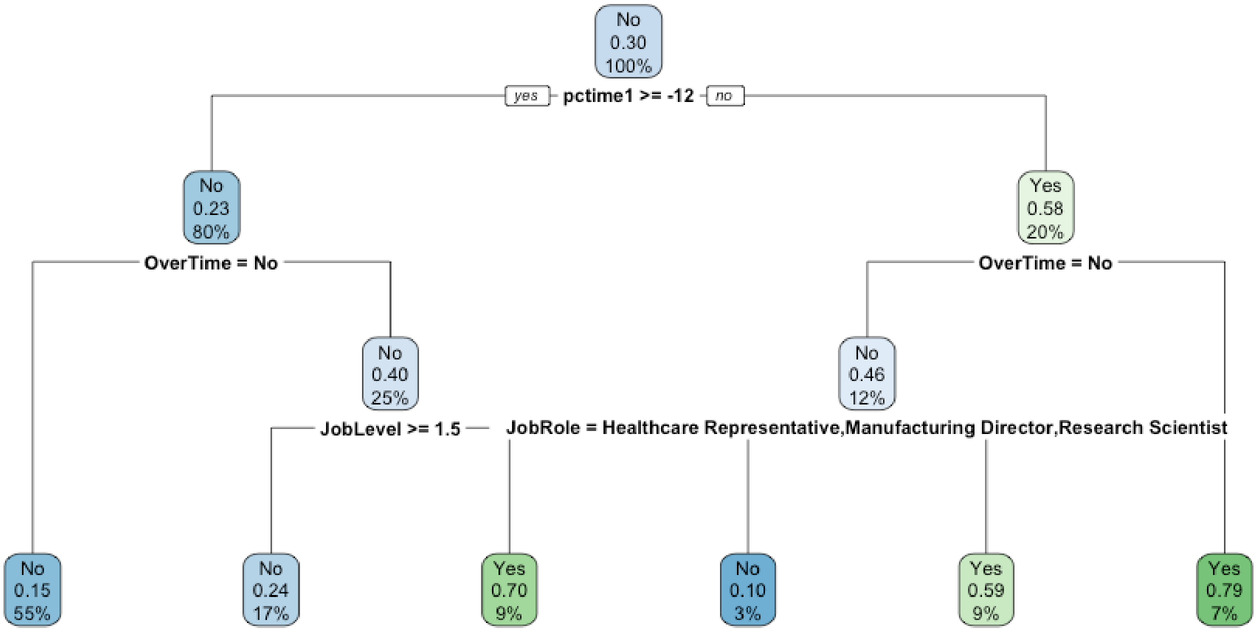


Figure 9: Tree Output

3.2 Ensemble Techniques

In this section, we will discuss how we arrived at our final two ensemble methods for prediction. We will discuss why it is advantageous to use ensemble techniques, what we considered when attempting to combined classifiers and how we selected what type of individual models to include.

3.2.1 Why Combined Classifiers?

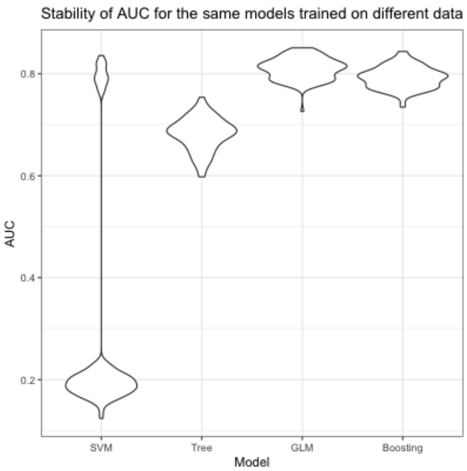


Figure 10: Training 1000 different models

To develop a single classification model from a dataset, it is necessary to subset the original dataset into a training and testing set. The model is built on the training set and evaluated on the test set. This is a practical consideration for implementation of the model. We want to know how our model will perform on unseen cases in practice. This is where a single classification model can become unstable. The performance of a model will change based on different train and test subsets of the data. We investigated this by subsetting the original dataset in 1000 different ways into testing and training data. A model was then fitted to each training subset and evaluated on its associated test set. The AUC value was returned for each 1000 models. We did this for a single decision tree, a support vector machine and a general linear model. The results of this analysis are illustrated in figure 13. As you can see, some methods are very unstable in their performance when varying the training subset. How do we know which model to chose? Picking the model with the highest AUC doesn't guarantee that it will perform any better on new data in the future. Each model was fit to the data in the exact same way. The only difference was what data it was trained on and what data it was tested on. Instead of picking one classifier, why not combined them all?

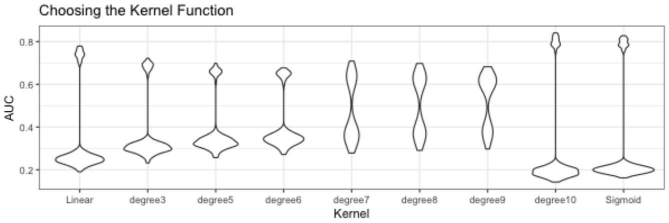
By sub setting our training dataset in many different ways in order to independently train many different models, we can take advantage of the variability of individual models to more accurately classify observations. If we consider the prediction for each model as a new variable, we can construct a final model to encapsulate all the predictions made for each observation. In this way, we hope to be able to explore each relation apparent in the dataset. In saying this, there is one important consideration to make at this stage. A model on a dataset, with the majority of variables containing the same information, will be no better than a model on a handful of relevant variables on that dataset. As we considered our individual model predictions as a new dataset, it is clear that these models must be as different as possible. More formally, we want our models to be as independent as possible.

The most variant of the three models is the support vector machine. In some instances it performs very well, otherwise it will perform very poorly with nothing in between. This is an interesting observation, and further investigation as to reasons why, is beyond the scope of this project time line. However, it will be considered in later models. A simple Tree model was the second most variant before general linear models but offered on average, a lower prediction ability. This graph gives us a good visual way to determine what is a good model to include in a model ensemble. A general linear model is the most stable, so we cannot ensure that our models will be all that different so combining this type of classifier in our ensemble techniques will be avoided. Recall that it was fully grown trees fitted to our trainins subsets. By shortening the depth achieved by each tree, we can make our models perform worse, giving more variance in predictions across models. This technique is encapsulated in a random forest and will be investigated as a model to include.

3.2.2 A Bootstrap Method for Support Vector Machines

Support vector machines (SVMs) have generated a lot of excitement in the statistical machine learning community and have proven themselves in the literature for classification tasks. Little attention has been given to the development of combining many independently trained support vector machines. In this section we will discuss the development of our support vector machine ensemble technique.

- **Choosing the optimal kernel -** We have already seen the variability of independently trained SVMs. This provided the motivation to experiment with them in an ensemble model. We will first discuss our considerations of the individual model components. The SVMs in figure 13 were fitted with a linear kernel. We will search for a Kernel which minimises the gap between the good and bad predictors witnessed in the figure 13.



The above figure shows the spread of AUC values for 1000 bootstraps of an SVM model for a linear kernel, a sigmoidal kernel and polynomial kernels of varying degrees. We can see that the gap between the good and bad predictors converges through increasing degrees of polynomial then jumps back to similar distribution of degree 1. This task was very computationally intensive. To investigate if this relationship would repeat itself through higher orders would require more time. We can see that the best kernel to fit our needs is a polynomial of order 9. The proportion of good predictors is greater than the proportion of bad predictors and the median predictor is approximately 0.6.

- **Splitting up the data and fitting the models -** First the training dataset is sub setted into n subsets consisting of $p\%$ of the training set. Rows are selected without replacement. A model is then fitted to each subset of the training subset. We will first examine the necessary number n of SVMs to include then the proportion of the test set to include in each subset $p\%$ through cross validation.
- **Generating fitted values -** Through using the technique outlined in splitting up the data and fitting the models, not every observations will have n fitted values. Therefore, after the models have been fitted to each subset, the whole training set is predicted by each model made in this stage. Now we have n fitted values for each case in our training set. These values are combinded into a dataframe with n variables for each case along with the corresponding true target value.
- **Combining predictions with a decision rule -** Next, a model is fitted to the dataset of predictor values to predict the true target value. This could also be a simple average of predictions across all machines, then a cutoff may be chosen.

Smaller values of p will build more independent support vector machines. However, to ensure that each point in the training set is sampled enough times, The number of SVMs to fit must be increased. Increasing the number of machines yields diminishing returns. Due to the imbalanced proportion of the target variable, too low a p value and yes classified observations will not be selected enough as training points. this could lead to a model which is very good at classifying true negatives but poor at predicting true positives which is likely not what is necessary to predict. Through all levels of p, When using a model based decision rule on the dataset of predictions, large values for iterations N will result in an over fitted decision rule. It is necessary to redefine our decision rule as the average prediction. This yielded much greater performance when using many machines. The optimised model was fit with $p = 0.5$ with 200 SVMs with a sigmoidal kernel function and a mean based decision rule.

3.2.3 Boosting methods

In the models investigated in the last section, the individual components were built independently. In the methods to be discussed in this section, the individual models are not independently generated. The main concept behind a general boosting method is to alter the weights of each observation based on the prediction in the previous model. A summary of the algorithm first implemented by Freud and Schapire (1996) named `AdaBoost.M1` is as follows.

- **Weighting observations -** The algorithm begins by initializing the weight of each observation at $\omega_i = 1/N$, where N is the number of individual cases in the training set.
- **Developing classifiers -** It then fits $y = h_m(X)$, a classifier to the data and compute

  $$W_m = \sum_{i=1}^{N} \omega_i I\{y_i h_m(x_i) = -1\} \tag{1}$$

  where I is an indicator function equal to one whenever the condition holds for a false prediction (data is coded with ±1). Note that $I_m = 1i$ if and only if$h_m(x_i) \neq y_i$ or when the model incorrectly predicts. Note that this will be equivelant to the error rate. We then compute

  $$\alpha_m = Log(\frac{1 - W_m}{W_m}) \tag{2}$$

- **Finding the new weights -** The new weights are then calculated with

  $$\omega_i = \omega_i e^{\alpha_m I\{y_i \neq h_m\}} \ldots \text{Scaled to sum to one} \tag{3}$$

  If the model incorrectly classifies observation i, The new weight will be greater. Otherwise, it will return to the weight it had before.

- **For** $i = 1, \ldots, M$ **-** M models are created one after eachother. At each iteration, the sample used to build the model is a bootstrapped sample of N observations. If a sample has a higher weight ω_i. It has a higher chance of being selected in the next model. **Combining the classifiers -** There is now M models of the form $h_m(X)$ and M weights consisting of some function of an error rate for that model. We can now combined the models to have a prediction

  $$f_m(x_m) = \sum_m = 1^M \alpha_m h_m(x_i) \tag{4}$$

  The better the individual $h_m(X)$ is, the higher the greater the weight $alpha_m$ will be.

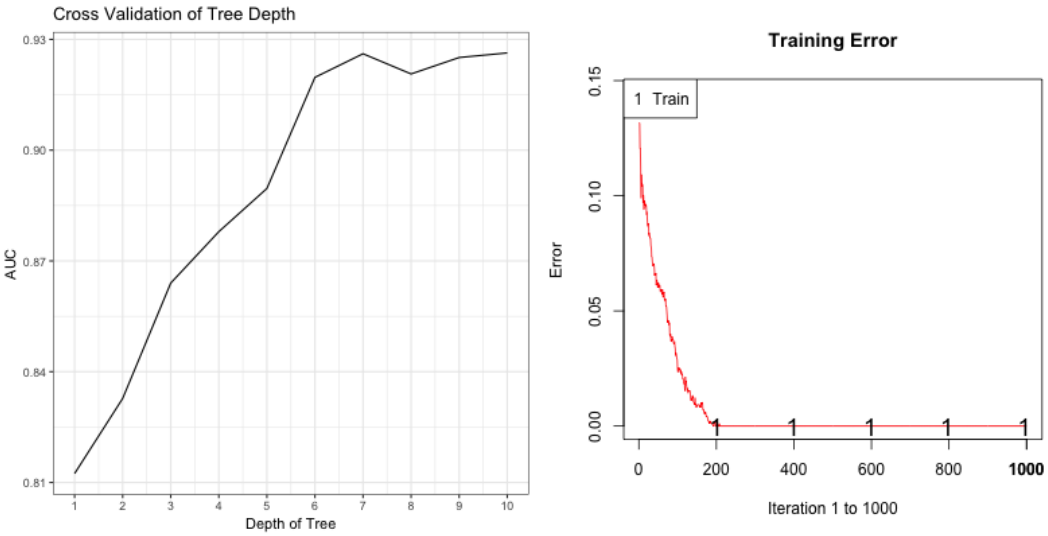


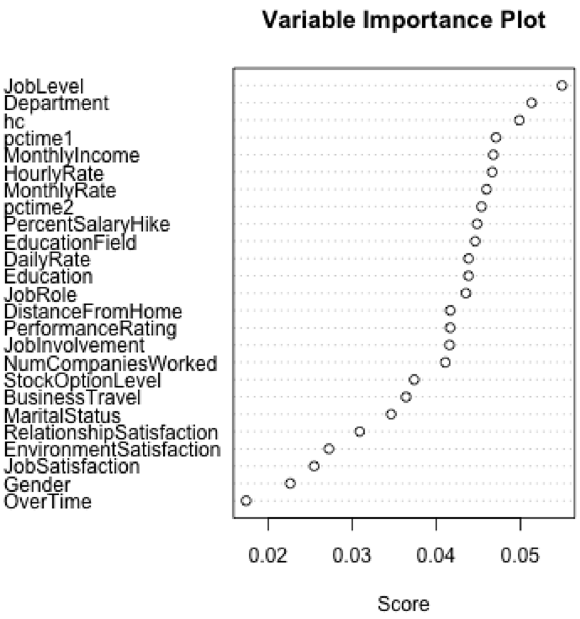
Figure 11: Cross Validation of the boosting method

In this particular algorithm known as discrete boosting, the weights are set arbitrarily and the algorithm then seeks to find the optimal weights from there. A modification of this algorithm, is known as gradient boosting. The main difference is the search for the optimal weights is optimised by a loss function who's first derivative is evaluated at each point to determine the starting weights. That loss function is then minimised. The output is some linear combination of classifiers which encapsulates a learning rate to determine weight on models built earlier on. This technique will be investigated in the later stage of the project.

The package `ada` is used which contains methods for the extensions to the standard boosting. To use the discrete boosting method, the loss function is set to `"exponential"` and the learning rate η equal to `sign(x)`. In this way, the model will have a closed form solution to $\alpha_m = 0.5log\frac{1-err_m}{err_m}$. The individual classifiers will be a classification tree from the `rpart()` package. The model is optimized through varying the maximum depth of tree and the number of iterations to process the algorithm with. The advantage of using an `rpart` tree with `ada` is that the high flexibility and customisation of `rpart` trees is gained as opposed to a built in splitting function such as seen in the `mboost` package.

To optimise the model, first the optimal depth of tree was found by cross validating tree depth from one through ten with 50 iterations. Figure 11, shows an increasing AUC value as the individual trees are grown deeper up until it reaches its maximum at a depth of 7 and tails of there after.. This maximises the AUC value and is not too deep as to make the model too complex. From examining the training error in the above plot, it is clear that over 200 iterations is unnecessary. Adding more then this adds unnecessary model complexity and computational need. By having fewer iterations, insight can be more easily gained as to the relationship between predictor variables and target.

A bootstrapped calculation of the AUC value for this model trained on the optimized dataset yielded a 12% gain in the AUC value when compared to the original dataset. Intuitively this makes sense as including variables containing duplicate information will cause a bias in the model. By replacing five correlated variables with two orthogonal variables accounting for 85% of the variation of the dataset, the bias reduction in predictions outweighs the information loss. The variable importance plot is based on the improvement criteria of a standard tree. An individuals job level, time proxy score (PCTime1) and income related variables featured most prominently in the model. The reduction in AUC when using the unrefined dataset is likely to the model being bias towards choosing the time related variables. This bias will be more apparent in ensemble methods. The relationships between the most important variables in the model will be examined with partial dependency plots and compared to the relations found in the exploratory analysis.



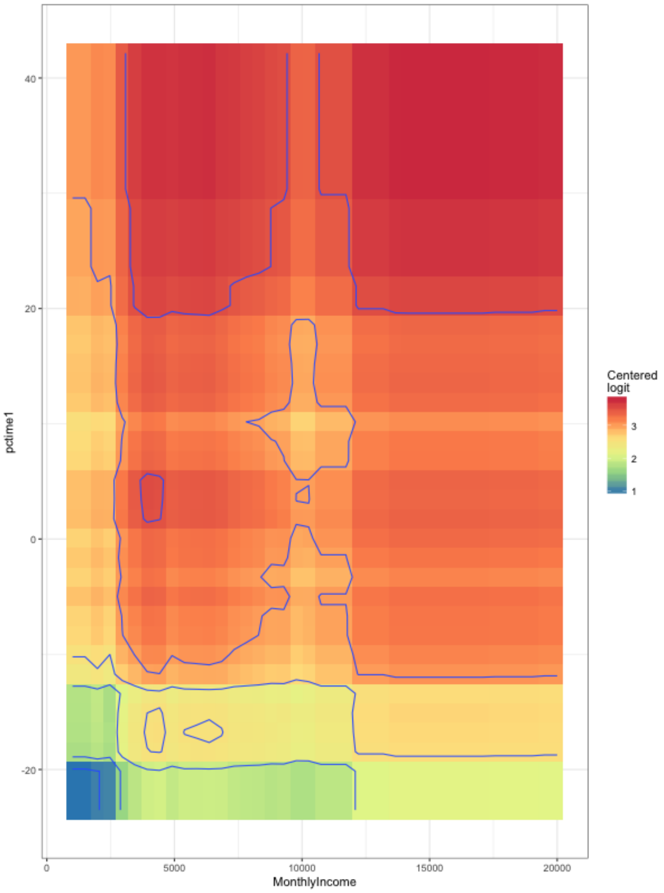


Figure 12: PcTime1 and Monthly Income Model Interpretation

The partial dependency plot contour scale is intepreted as a centred logit probability odds ratio of a positive prediction. Higher rates of monthly income increased the likelihood of predicting an observation as a yes for PcTime1 score of less than -10.

$$PcTime1 = \begin{bmatrix} YrsInCurrRole \\ YrsWithCurrMan \\ YrsSinceLastPromo \\ YrsAtCompany \\ Age \\ TotalWorkYrs \end{bmatrix}^T \begin{bmatrix} -0.363 \\ 0.158 \\ -0.128 \\ 0.354 \\ 0.567 \\ 0.602 \end{bmatrix}$$

An individual with a high value for years since last promotion and current role will have a lesser chance of leaving. It was previously observed that the loading on age in the first two PC dimensions in figure 3 invoked a seperation in the target variable by reducing the frequency of yes labelled points (dark red) in the negative x direction of the graph. From the partial dependancy plot, a larger value for PCTime1, which is most asscociated with age, decreases the likelihood of leaving the company. Individuals with more experience in general are more likely to be predicted to leave.

4 Model Evaluation

In this section, comparisons are made between models investigated through out this report. Observations will be made based on how models are in line with varying motivations for prediction and by comparing overall performance. Based on this, the model which best suits the nature of this classification problem will be picked.

4.1 Overall model performance

To get a grasp of overall model performance, attention is given to the ROC curve. A model which has a false positive rate of 0 and a true positive rate of 1 is a perfect predictor. Therefore, it would show on the ROC curve as higging the y axis and where y=1. Hence, the area underneath it curve is 1. In this way, it is possible to assess the performance of the models discussed both individually and in comparison to one another. Here the difference in classifier performance is clear for ensemble methods as opposed to a single tree method. The first model discussed had an AUC

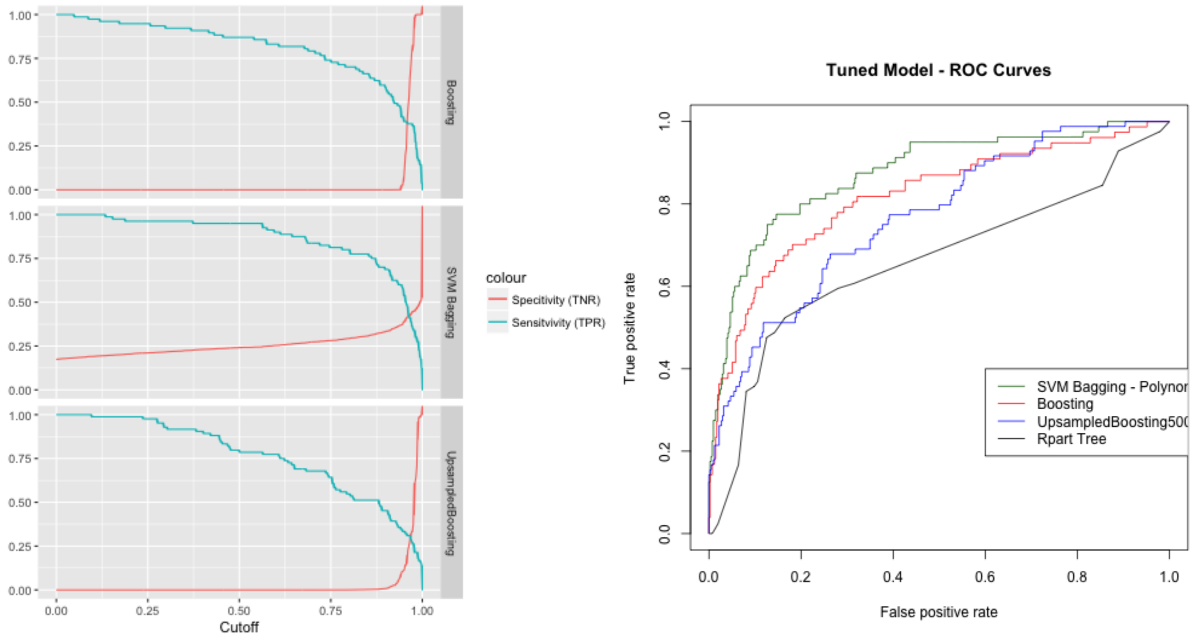


Figure 13: PcTime1 and Monthly Income
Model Interpretation

value of 0.78. The ensemble methods just discussed achieved values of 0.88 for the support vector ensemble and 0.85 for the boosting method. The values for the variance of AUC value was also much lower for ensemble techniques, sugesting that they may predict, more predictably into the future.

From analysing the ROC curve, it is clear that the model which performed the best was the bootsrap method for ensemble machines.

4.2 Predicting What is Relevant

In a perfect scenario, a model to predict attrition from a sample of the workforce would have a TPR and a TNR of 1. That is, it would predict precisely whether or not a person will stay or a person will leave. The models discussed throughout this report have been far from perfect. As a result a trade off is met. It must be decided what is more important, correctly predicting if a person will leave or correctly predicting when a person will not. By examining the sensitivity and specificity of a model for different cutoffs, this trade off can be graphically illustrated. For A cutoff of 0, the model will predict every event as a positive classification. This is expressed by the blue lines of figure at a value of 1 when $x = 0$. As we move to the right, the sensitivity degrades. Where these curves intersect, is where the true positive rate and the false negative rate equal eachother. This point is highest in the Bagged SVM model. This would be a natural point to chose a cut off. Doing so, the SVM bagging model would perform the best in practice.

If it is the case that the motives for prediction is to try and stop someone from leaving, then a model which only at predicts when a person will not leave wouldn't be any use. Therefore, the optimal model in this scenatio is the bootstrap method for support vector machines. The blue line hugs $y = 1$ the most out of the other two models. Also the support vector machine model happens to be excellent at predicting true negatives. This is indicated as the red line does not touch the x-axis. This gives some insight as to how the support vector defines its decision boundaries much more definitively than a tree based model. Furthermore, it may give insight into the large variation in the bootstrapped performance of an SVM seen earlier.

For any given level of cutoff, the support vector machine performs better at predicting true positives and true negatives. This makes the trade off earlier discussed easier to make. Therefore is the optimal model in this scenario,