University of Essex

Department of Mathematical Sciences

MA981 DISSERTATION

Employee Attrition and the Use of Protected Characteristics

Peter Hatton

2201687

Supervisor: **Dan Brawn**

September 22, 2023
Colchester

Contents

CHAPTER 1

Introduction

Employee turnover and retention has always been a major concern of corporations globally and across history. As far back as 1925, prior to the explosion in employee turnover research, Benge argued "What is needed is some weather vane which will show the way the labor winds are blowing before a gale sweeps valuable employees past your pay-off window and applicants away from your employment office" (p. 359) [2]; this clearly highlights how old and serious a problem this is.

Given the current state of low unemployment and the proceeding labour shortage that many developed economies are currently facing, the ability to identify important predictors and use them to predict employee turnover could be considered especially important now. As an employee leaving the company is less likely to be replaced, a business could be left with a major gap in their staffing requirements. Even if an employee is replaced, more resources will likely have to be dedicated to the process of finding, training, and paying the new replacement. Thus, the ability to predict which employees are likely to leave is an essential area of research for all companies; it could save the company time, money, and manpower on replacing lost employees. Furthermore, it is likely not unreasonable to assume that employees will also benefit from this research; policies designed to retain employees will likely counter some employee grievances. It is for the reasons stated above that this dissertation seeks to analyse employee data to first identify predicting factors indicating employee attrition, and then to use those factors to build a machine learning (ML) model to attempt to predict which employees will leave and which will stay.

In addition to the importance of predicting employee retention and attrition, there is the essential issue that many people, companies, and institutions are facing when it comes to the use of ML models for prediction - potential bias or discrimination through the use of protected characteristics (PCs) as predictors. Protected characteristics are defined in many different ways by many different countries, governments, or institutions, but for this dissertation the definition used will be that of the Government of the United Kingdom under the Equality Act 2010 [3], which states it is against the law to discriminate against someone because of:

- Age
- Disability
- Gender identity
- Marriage or Civil Partnership
- Pregnancy or maternity
- Race
- Sex
- Religion or Belief
- Sexual orientation

While it is assumed readers of this dissertation have a basic understanding of these protected characteristics, a short summary of these items and an explanation of them can be found here [4] to provide further clarity if needed.

It is because of fears surrounding the potential misuse of PCs by black-box ML models that this dissertation seeks to build two models, one with PCs included and one without. The intention is to establish the importance placed on PCs by certain types of predictive modelling and if their removal will noticeably impact what is considered the most important predictors and the model's ability to successfully predict whether an employee will leave.

1.1 Related Literature

As an area of key importance to businesses there has been much research into employee turnover, including attempts to identify important predictors and use them to predict it through various statistical and ML methods. Employee turnover has a long history of interest in the literature, as earlier mentioned Benge [2] identified the issue in the 1920s, however it wasn't until the 1950s and onwards that there was an explosion in studies on this topic. Brayfield and Crockett's (1955) [14] study is indicative of studies at the time, identifying an important predicting factor, but being limited in scope to only studying one or two potential predictors.

Later in the 1970s and 1980s it is possible to identify several large meta-analyses of turnover data and studies. These, such as Muchinsky and Tuttle [15] and Cotton and Tuttle [16], operate with a much wider range of data than many other studies, and identify the importance of variables which are much harder to measure and evaluate; noticeably job satisfaction, morale, and responsibility. Muchinsky and Tuttle also note the importance of the human relations skills of an employee's line manager or supervisor. In addition, Cotton and Tuttle identify the importance of external factors, especially membership in a workers union.

It is important to note, these studies were conducted with access to data that is much harder to acquire now; due to changes in data protection, especially post GDPR, it is much harder to access personal information, especially overlapping data such as employee data, union membership, etc. Time-series data is also very common amongst these studies, and has been used with great affect to study the changes in time and how adjustments in one factor impact on the same employee's likelihood of attrition at later dates.

It is likely not a coincidence that many large meta-analyses and high profile studies analysing turnover started appearing shortly before there was an explosion in the number of articles discussing the end of "jobs for life" such as this article by Dore [17]. Although articles such as this would not feel out of place in contemporary media, they truly started to draw attention in the 1990s.

Studies into employee turnover never ceased, being published continuously from the initial explosion mentioned earlier, well into the 2000s and into contemporary

media. Batt and Valcour's study [18] into work-life conflict was especially interesting. It highlighted the importance of aligning work and the rest of life, as variables that were associated with "life" over work related variables had a considerably greater correlation with turnover. Variables such as working hours, commuting distance, and overtime were identified as being positively correlated with increased work-life conflict, while age and tenure were negatively correlated. Thus, they argued that companies should focus on policies which aid in avoiding work-life conflict in order to increase retention.

Narrowing the focus, the chosen data set for this dissertation is the IBM Employee Attrition data set from Kaggle [1]. This data set is widely used across many different fields and universities as it is one of the few available data sets of its kind, and due to being easy to apply any number of statistical or ML models to it; Zhao *et al* [21] alone identifies over 11 different types of model which are commonly applied to it. Yang and Islam [19] offer a good example of a study regarding this data set, looking to apply models to predict employee turnover starting with all the predictors available in the data set. However, a key limitation of almost all studies attempting to predict employee turnover, but especially for this data set, is the near complete lack of research regarding the impact of PCs on the importance of predictors and the ability to predict turnover.

Several authors have already raised great concern in the wider literature over the use of ML predictive algorithms, especially in the context of PCs; this has even led to several legal cases. Notable is the case of $Bridges$ [5], the first and only successful legal challenge to the use of PCs in ML predictive models. Casale [22] discusses *Bridges* in her article on the widespread fear surrounding ML black boxes. A black box is an algorithm which operates without the human overseer knowing exactly how it operates; an input is taken and an output is given, but the knowledge on how it achieved this output is limited. This means it is difficult to understand how it interacts with PCs, which may lead to issues of the algorithm discriminating against protected groups. Sokol *et al* [23] does highlight that there are some ways to avoid the issues of black boxes, but admits that these are very "time and resource consuming" and so not compatible with wide-scale use currently.

The issues of potential discrimination via the use of ML algorithms is acknowledged by the UK government through the Centre for Data Ethics and Innovation [26]. However, the case of *Coll* [24] limited discrimination based on PCs to "exact correspondence",

meaning only a direct link between the PC and the discrimination suffered would be considered under the Equality Act. This grants black boxes more protection and greater scope for use. Counter to this ruling, Borgesius [25] argues that current legal protections do not offer enough protection against potential discrimination from ML algorithms, identifying that algorithms are very effective at using substitute attributes, even when PCs are removed. For example, he used the case of an algorithm discriminating via the use of nationality as a substitute for race.

While there is both research into employee turnover, and research into the issues surrounding predictive algorithms and PCs, there is almost no cross-over. Only three sets of authors could be found that have studied this intersection of ideas, all with their own limitations. Ghosh *et al* [28] noted that some "demographic-unaware" models can perform admirably, however this was focused around "sex" alone and is thus limited in scope. Castille and Castille [27] alongside Speer [29] were the only authors to advocate for the removal of PCs from predictive models. However, both sets of authors have a very limited definition of protected characteristics. Castille and Castille only included sex and race, and Speer only added age to that list. Speer was the also the only one to use a legal basis (US law) to define PCs; notably neither recognised a distinction between sex and gender, and none came close to the more expansive UK legal definitions.

While it is questionable that any manager would deliberately discriminate against a person based on these factors, whether they are correlated with turnover or not, it may lead to unconscious bias against that person. Therefore, especially within the public sector it may be considered important to avoid the use of protected characteristics as predictors. The lack of research and the potential real world impact to people via discrimination shows the need for more to be done in this field, something this dissertation seeks to study and hopefully draw more attention to.

CHAPTER 2

Methodology

All analysis and modelling will be conducted using the R coding language.

As the outcome attempting to be predicted is binary - Attrition, yes or no - the chosen model used will be logistic regression. Logistic regression is a simple prediction model, and one of the most widely used for "two-class classifications" [7]. Speelman [6] confirms that it is widely accepted as the preeminent method for predictions involving binary outcomes, as there is much that goes on "under the hood" of logistic regression, making it very effective with proper preparation. Logistic regression follows the statistical formula:

$$Pr(Y_i = 1|X_i) = \frac{exp(\beta_0 + \beta_1 X_i + \beta_2 X_2 + ... + \beta_i X_i)}{1 + exp(\beta_0 + \beta_1 X_i + \beta_2 X_2 + ... + \beta_i X_i)} \tag{2.1}$$

Logistic regression has been chosen over other modelling methods such as random forests due to its simplicity and ease of use. While both logistic regression and random forests are effective at predicting binary outcomes, random forests often operate with a slightly higher level of accuracy [8]. Nevertheless, logistic regression was chosen over random forests due to Tonkin *et al*'s [9] work that highlighted that tree-based models are much more prone to over-fitting, leading to much greater variation in the accuracy of the model when hold-out cross validation is applied. Because of this logistic regression was chosen as the method for this dissertation as - although its max accuracy is often slightly lower, it has far less variation when applied to different/non-training data. As it is usually considered to be important for a model to be applicable outside its

original training data set -especially for companies which wish to predict something important like turnover- it was believed that the benefits of reliability and creating an easily reusable model were more important than a slight increase in model accuracy.

Given the reasonable number of variables in this data set, and the knowledge that having too many predictors in a model can lead to sub-optimal results and computing times [34], two types of feature selection methods (FSM) have been used to identify the most important predictors and eliminate the least useful.

The two feature selection methods selected were Boruta and VSURF. These were identified by Speiser *et al* [10] as being effective methods. While Speiser did recommend Jiang over Boruta for data sets with less than 50 predictors, due to the lack of documentation on the Jiang method, this was not chosen. These choices are further confirmed throughout the literature with many authors advocating for their use; namely Kursa *et al* [12] who argues that due to the way Boruta incorporates random fluctuations into its system of classification, and the use of shadow attributes to account for accuracy loss between differing trees in the forest, it is very effective at identifying important predictors, especially from a large pool. Boruta identifies the importance of features by identifying the loss of accuracy based on the random permutations of the inclusion and exclusion of various features across the random forests [30].

Meanwhile, Speiser [35] endorses the VSURF method proposed by Genuer *et al* [13], highlighting that its "elegant and versatile framework" makes it expertly suited to feature selection, regardless of the number of starting predictors. VSURF operates by developing many random forests (default 50) averaging varying feature importance across all random forests to create a ranking of variables from most to least important. From there, VSURF removes the variables from its calculations which fall below a threshold of importance. Finally, it feeds variables into the forest one by one, retaining them only if the accuracy increase is larger than the previously established threshold, creating the final selection of important variables [35].

Cross-validation (CV) will be used to ensure the model is replicable and can be used on new data. This will also be useful for testing the true efficacy of the model and to compare differing models created from the various feature selection methods used. In line with Zakrani *et al*'s recommendation [36], a 30% holdout validation method will be used to evaluate how applicable the models are to generalisation. Here the data set will

be split into two non-overlapping sets via a random subject-wise split. Although Xu and Goodacre [37] argue in favour of bootstrapping over train/test splits, the latter was chosen due to compelling arguments from Zakrani *et al*, Saeb *et al* [38] and Wieczorek *et al* [39]. These authors all agree that subject-wise CV reduces the likelihood of over estimation of a model's effectiveness, and is generally considered to be essential when being used to predict a diagnosis; which is similar enough in application to apply to our prediction for the variable "Attrition" as both are binary.

The methodology explained above will be applied to two different logistic regression models.

The models will be assessed using a number of evaluation metrics; namely, Area Under Curve (AUC), Accuracy, Precision, Sensitivity, F-Measure, and Specificity. For this, several two letter acronyms must be understood: TP represents True Positives; TN represents true negatives; FP represents false positives; FN represents false negatives. True instances are observations correctly predicted, while false instances are observations incorrectly predicted.

AUC measures the model's effectiveness at discerning between positive and negative cases. It analyses the true positive rate against the false positive rate, and produces an output between 0 and 1, with 1 being perfect at predicting instances.

AUC = the calculated Area Under the Sensitivity(TP Rate)-(1-Specificity)(FP Rate) Curve

Accuracy measures how many instances are correctly predicted by the model. As it covers both true positives and true negatives it is often used as the principle measure of model effectiveness; it can easily be compared with the baseline accuracy of 84% for simply assigning all employees as retained.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

Precision is the measure of how many positively labelled instances are actually positive, it details a model's effectiveness at avoiding false positives.

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

Sensitivity or Recall shows of all positive cases how many were correctly predicted.

It details a model's effectiveness at identifying positive cases.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.4}$$

F-Measure is the harmonic mean of precision and recall, "combining recall and precision into a globally useful metric" [40]. It is used to evaluate a model's overall performance on positive instances.

$$F - Measure = 2 * \frac{Precision * Sensitivity}{Precision + Sensitivity} \tag{2.5}$$

Finally, specificity identifies of all the negative cases, how many were correctly predicted by the model; it details a model's effectiveness at identifying negative cases.

$$Specificity = \frac{TN}{TN + FP} \tag{2.6}$$

Logistic regression will be coded using the glm() function that comes with the base version of R. Boruta feature selection will be done using the boruta() command from the library of the same name [30]. Similarly, VSURF selection will be done using the VSURF() command from the library of the same name [13].

CHAPTER 3

The Data Set

The data set chosen for this dissertation is a collection of employee data from IBM [1] including information on internal variables such as income, overtime, and contracted hours, as well as survey data from each employee regarding their satisfaction with various subjective variables - for example, job satisfaction, work-life balance, and their relationships with their colleagues. This is one of the only high-profile data sets of its kind containing employee data. The original purpose for which IBM released the data set was to give members of the public a set of employee data with which they could predict attrition [33]. The data set contains 1470 observations with 34 variables.

It should be noted that this is a fictional data set [31]; as explored earlier, due to the tightening of data protection regulations it is far more difficult to acquire natural employee data, especially due to fears that even anonymised data can be used to identify personal information [32]. Because of these fears it was essential for this data set to be fictional. However, this data being fictional creates some differences between this data set and what could be expected of natural data. For example, this data set is exceptionally clean; additionally, it has some oddities, such as everyone recorded in the data set working the exact same number of hours despite some people doing overtime and others not. These will be explored further in the section on data cleaning.

3.1 Data Cleaning and Preparation

As mentioned earlier, due to this being an artificial data set it is incredibly clean. A quick check reveals there are no empty observations, retaining all 1470 observations. Checking the number of unique returns for each variable shows that the columns Over18, StandardHours, and EmployeeCount all have only one unique response and so are dropped. Furthermore, the variable EmployeeNumber was dropped due to it being a unique identifier for each employee and not necessary for our purpose.

Continuing, the variables Attrition, Gender, and OverTime are converted to binary numeric values, while BusinessTravel, and MaritalStatus are converted into a numeric scale. Just prior to converting Attrition, a copy of the column is made in its original form because it makes graph creation slightly easier and more readable.

In preparation for later use in the models, a partition was created splitting the data frame into 70% train and 30% test sets. Then all continuous variables were selected and min/max scaling was used to normalise the data. This was applied to both train and test sets. With this complete the data was ready to be used for the first model.

3.2 Exploratory Data Analysis and Data Visualisation

Due to the large number of variables in the data set and the numerous different ways they can interact, it would take far too much time and space to properly explore all of them here; also doing so would go beyond the remit of this dissertation, and could potentially draw away from the stated purpose of this project. Therefore, beyond the protected characteristics, exploratory data analysis will focus on those variables which are likely to be impactful to the final model, as determined either through a correlation matrix, or through what could be expected to be impactful from the study of the literature found in 1.1.

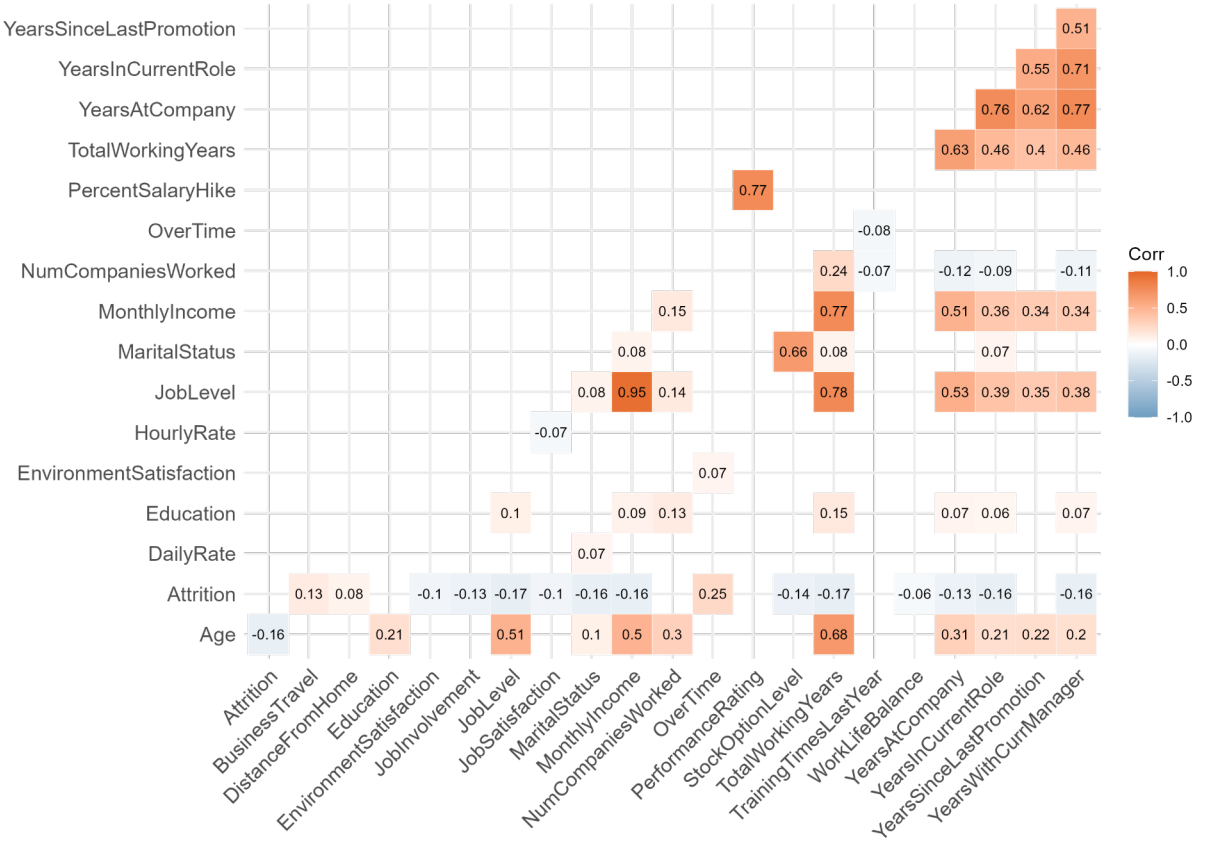

Figure 3.1: Correlation matrix of the full data set with low correlations filtered

3.1 is a correlation matrix formed from all variables in the data set, however every correlation below +/- 0.05 has been removed to allow for easier visual analysis of important correlations. From this correlation matrix the first notable piece of information is the lack of any strong correlation between predictors and Attrition - OverTime has the most significant correlation at just 0.25. Given the understanding taken from the literature that factors leading to work-life conflict are expected to have an out-sized impact on the likelihood of attrition, this is not unexpected. Aside from OverTime, there are several variables with correlations of (+/-) 0.16 or 0.17, but beyond that correlations with the dependent variable are very limited, with PerformanceRating

being the least correlated with a correlation of 0. It could be assumed that variables with more significant correlations with Attrition will likely be the variables which will be the most impactful for predicting it.

Also as noted earlier, TotalWorkingYears has a large correlation with a PC, namely Age, of 0.68. This is the highest correlation with any PC and has been identified as a potential substitute variable for a PC, thus it will be removed later. Additionally, and very interestingly, StockOptionLevel and MaritalStatus have a significant correlation of 0.66, and have no other strong correlations with other variables. A Chi-squared test was then conducted to confirm dependence between these two variables; with a p-value of 2.2e-16, it can be shown there is a clear level of dependence here. Therefore, StockOptionLevel will be considered as a potential substitute for a PC for the purposes of this dissertation, thus it will be removed during the model filtering stages later.

Rather unsurprisingly, the highest correlation is between JobLevel and income. Furthermore, there is a cluster of high correlations between highly linked factors such as YearsInCurrentRole, YearsWithCurrManager, and YearsAtCompany, etc. What is surprising is that monetary factors like pay and raises, and factors regarding progression such as training or promotions have a very low correlation with Attrition. RelationshipSatisfaction and WorkLifeBalance also had little correlation with Attrition; given the literature this is surprising as one of the key factors influencing attrition identified there was potential work-life conflicts, which it could be imagined WorkLifeBalance would be an effective rating of.

3.2.1 Protected Characteristics

Age

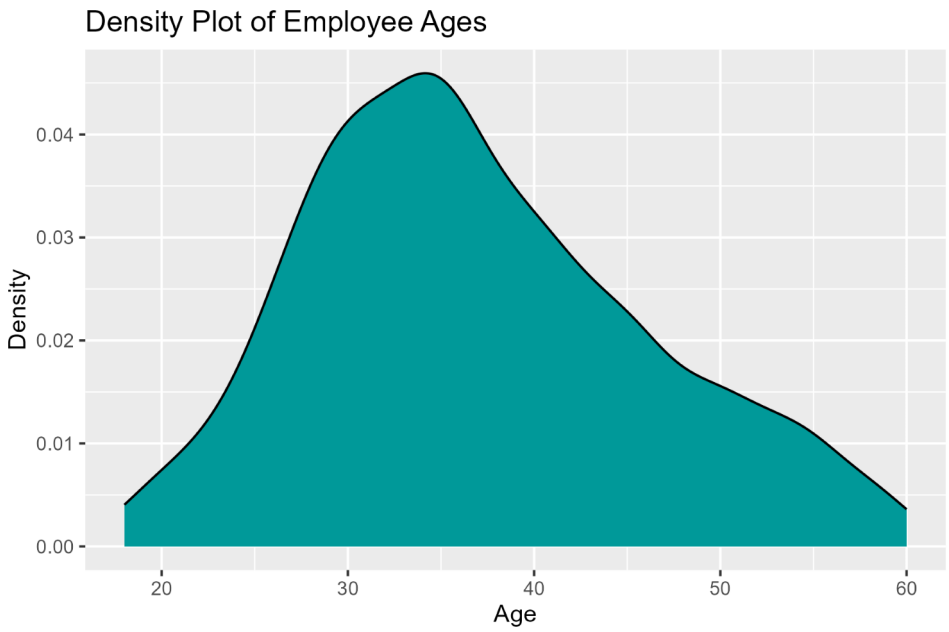

Figure 3.2: Density Plot of Employee Age

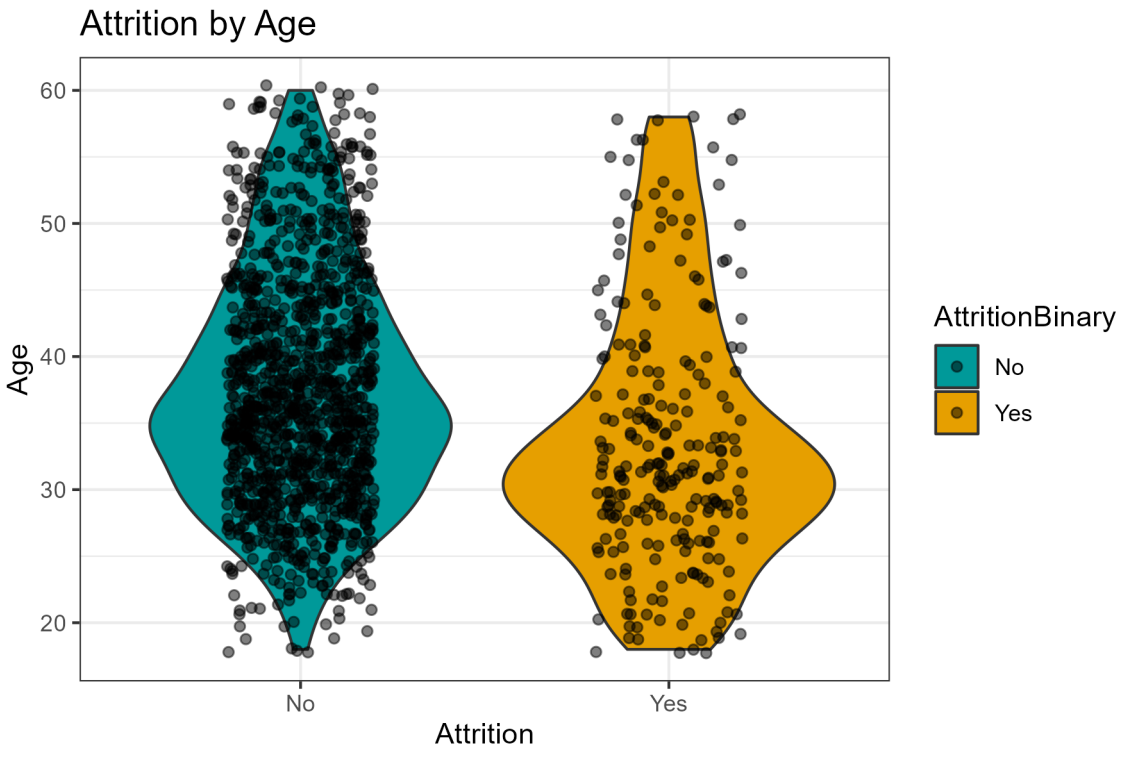

Figure 3.3: Violin plot of Employee Attrition by their Age

Firstly, by analysing what is shown in 3.3, it can be seen that the bulge in the violin plot is located in a younger age bracket for those who leave compared with those who are retained. The bulge is also notably more narrow, showing a greater concentration around one point rather than a more even spread. Those employees who are retained not only appear to have a higher average age but are more evenly spread throughout the entire age range. Additionally, conducting a Wilcox test confirms what could be identified from the plot, that the distribution of Ages between both Retained and Non-Retained employees are significantly different with a p-value of 5.304e-11. This suggests that Age could be an important and impactful predictor for Attrition, in-line with its relatively high negative correlation with Attrition of -0.17; while not a high correlation it is joint second for highest correlation with Attrition of all included variables. All this suggests that its removal from the predictive model may have a noticeable impact on its effectiveness to predict attrition.

Marital Status

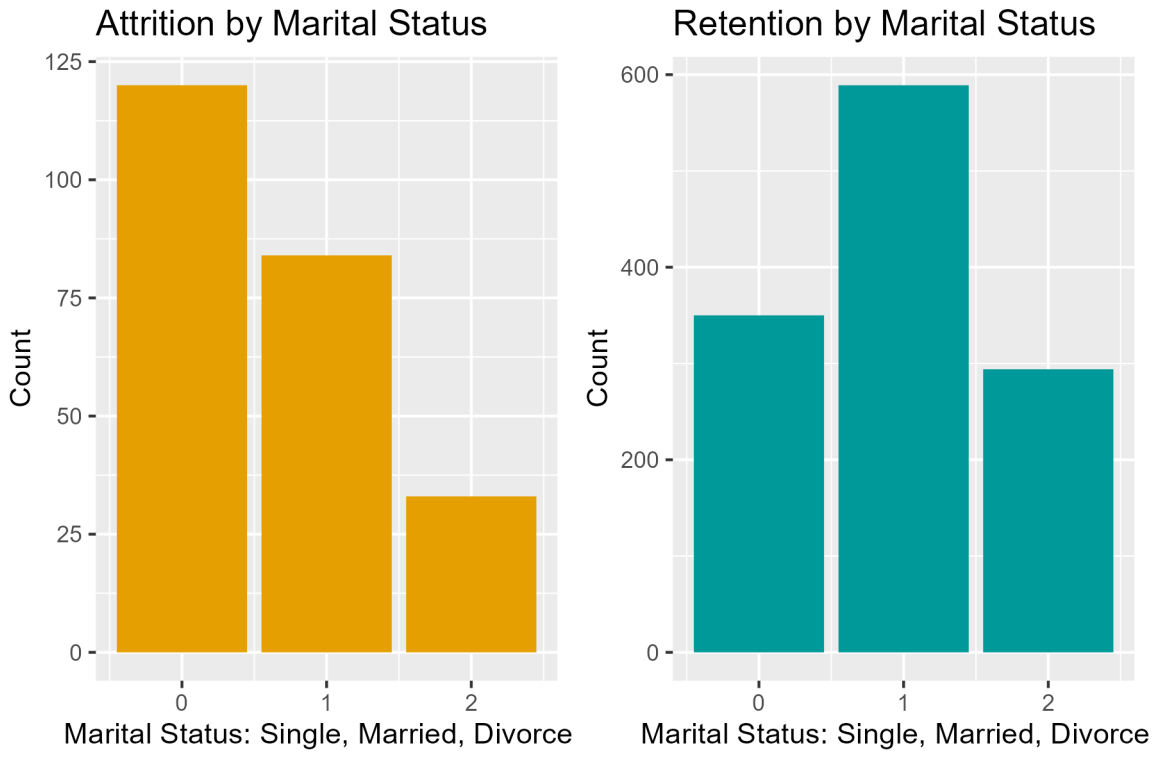

Figure 3.4: Employee Attrition by their Marital Status

Straight away it can be seen in 3.4 that there is a notable difference in MaritalStatus between Attrition and Retention. Of those leaving roughly half are single, compared with those who are married being a plurality of those who are retained. A Wilcox test confirms with a p-value close to zero that there is a significant difference in the distribution between the two displayed on 3.4. This signifies that this is likely an important variable for predicting Attrition, suggesting its removal may make a major impact on the predictive model's effectiveness.

Gender

Looking at 3.5 there appears to be little difference in the distributions between Retained and Non-retained. A Wilcox test confirms this, with a p-value of 0.259 failing to confirm a statistically significant difference in the distribution between the two outcomes. Additionally, a Pearson's Chi-Squared test shows there isn't significant association between gender and attrition in this data set, with a p-value of 0.29. Ultimately this likely means Gender is not an effective predictor of Attrition, and its removal from the predictive model will likely make little impact in its effectiveness.

Figure 3.5: Employee Attrition by Gender

TotalWorkingYears

As well as the true protected characteristics there were two other variables with high correlations to PCs which could potentially be used as substitute variables for PC. Thus, these warrant some exploration. Firstly, TotalWorkingYears and Age would normally have distributions separated by many years as is logical considering someone just out of university may have 1 year in work but be 23. Thus, using data from [41], TotalWorkingYears was adjusted to assume that these employees found work within 1 year of graduating from a 4 year bachelors degree which they started at 18, as is the majority in the US (where this data set is from). With this assumption, it can be seen in 3.6, that Age and TotalWorkingYears have very similar peaks, and their drop off follows a similar pattern. Therefore, their high correlation can be clearly seen and understood.

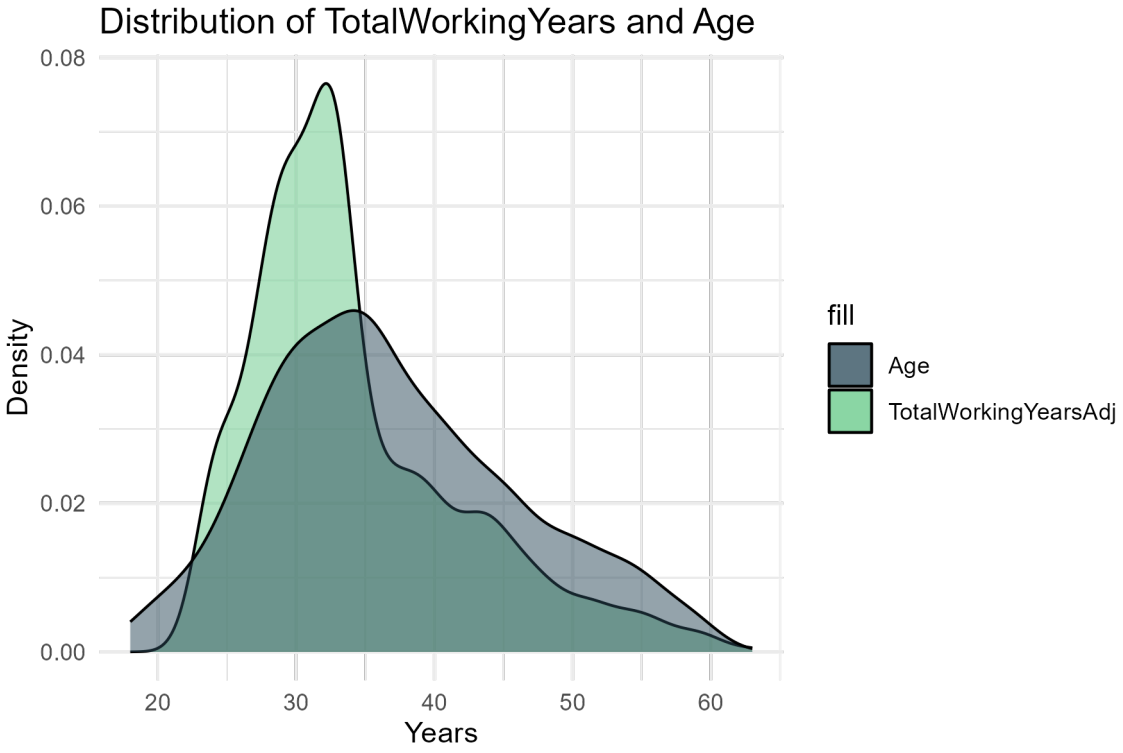

Figure 3.6: Density plot of adjusted TotalWorkingYears overlaid with Age

StockOptionLevel

From looking at 3.7 it is obvious there is a difference in the levels of Stock options given to employees based on their marital status. Those who are single have been given no stock options at all, meanwhile only a very limited number of employees who are

divorced are not offered some level of stock option. Given this, it is understandable why there is such a high correlation between these two variables, therefore this could be a variable which is used by the model to simulate MaritalStatus.

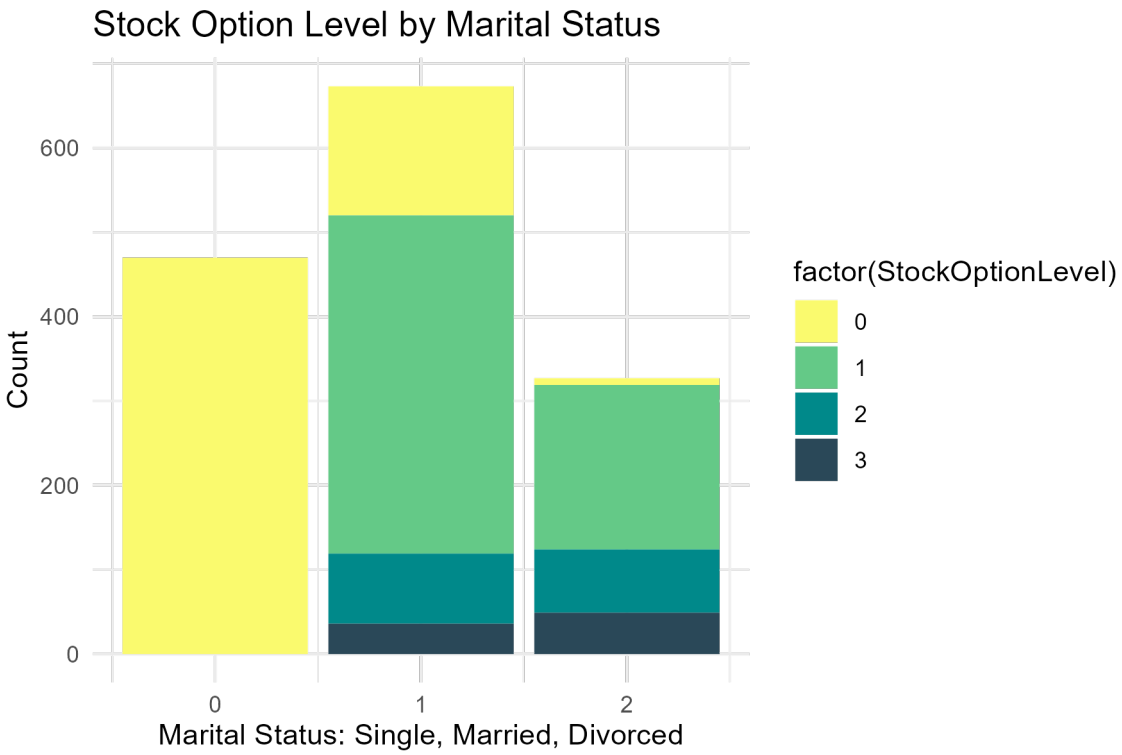

Figure 3.7: Stacked bar graph of StockOptionLevel by employee Marital Status

3.2.2 Other Variables

Despite the high correlation, a K-S test highlights that there is significant difference between the distributions of MonthlyIncome and JobLevel. Furthermore, Wilcox tests confirms that visualisation of there being a noticeable difference between the two levels of Attrition. This can be further seen by the difference in medians and position of outliers between the two variables in 3.8.

Figure 3.8: Attrition by Job Level and Attrition by Monthly Income

3.9 was built to evaluate the various variables which can be identified as showing Years. It shows these factors are all highly linked, as can be seen by their high correlation with each other. Additionally, many of these have noticeably different correlations dependent on Attrition - especially YearsWithCurrManager, which has a 0.2 correlation difference between Attrition and Retention, and has the most difference between the two levels of any variable here. This likely identifies it as an influential variable for predicting Attrition. It is also worth noting that all the factors have reasonably similar median points when split by Attrition, although there is a notable difference in inter-quartile range for most of them.

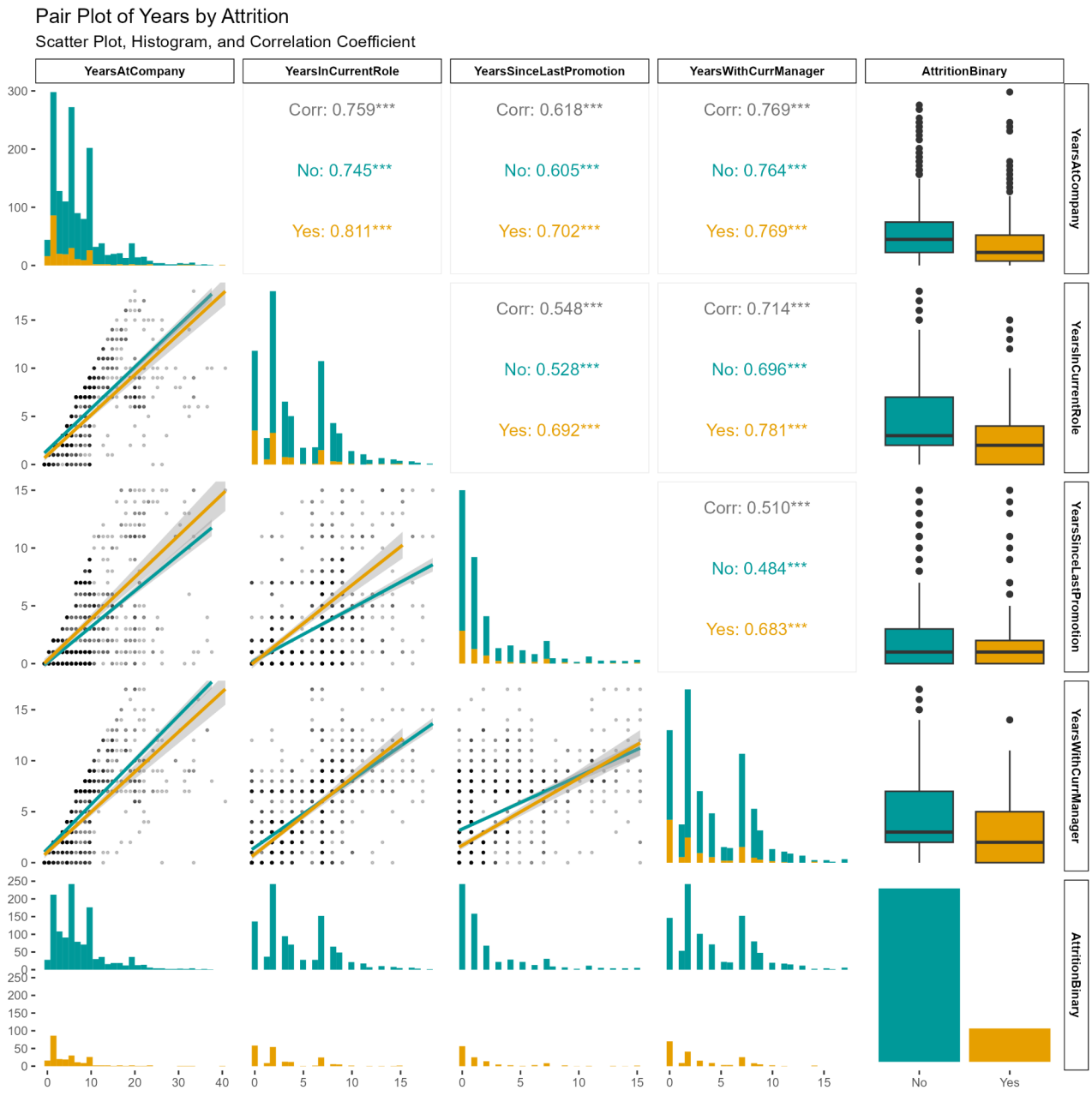

Figure 3.9: Plots showing the interactions of various Years Spent variables against Attrition

Looking at 3.1 it can be seen that those who have a negative opinion towards both their work environment and their work colleagues are more likely to leave than those who have a positive opinion towards both. Going from double positive to double negative opinion over doubles the percentage of employees leaving in both instances of the variable OverTime. The most important and noticeable thing to be taken from the table is how much working overtime impacts the percentage Attrition rate; there is no overlap at all in the percentages of people leaving between doing or not doing overtime,

OverTime	EnvironmentOpinion	JobOpinion	TotalEmployees	EmployeesLeaving	Percentage
0	Negative	Negative	158	28	17.72%
0	Negative	Positive	270	28	10.37%
0	Positive	Negative	258	28	10.85%
0	Positive	Positive	368	26	7.07%
1	Negative	Negative	56	30	53.57%
1	Negative	Positive	87	29	33.33%
1	Positive	Negative	97	26	26.8%
1	Positive	Positive	176	42	23.86%

Table 3.1: Table showing Attrition grouped by Overtime and various subjective Satisfaction ratings

regardless of employee opinions towards their environment or colleagues.

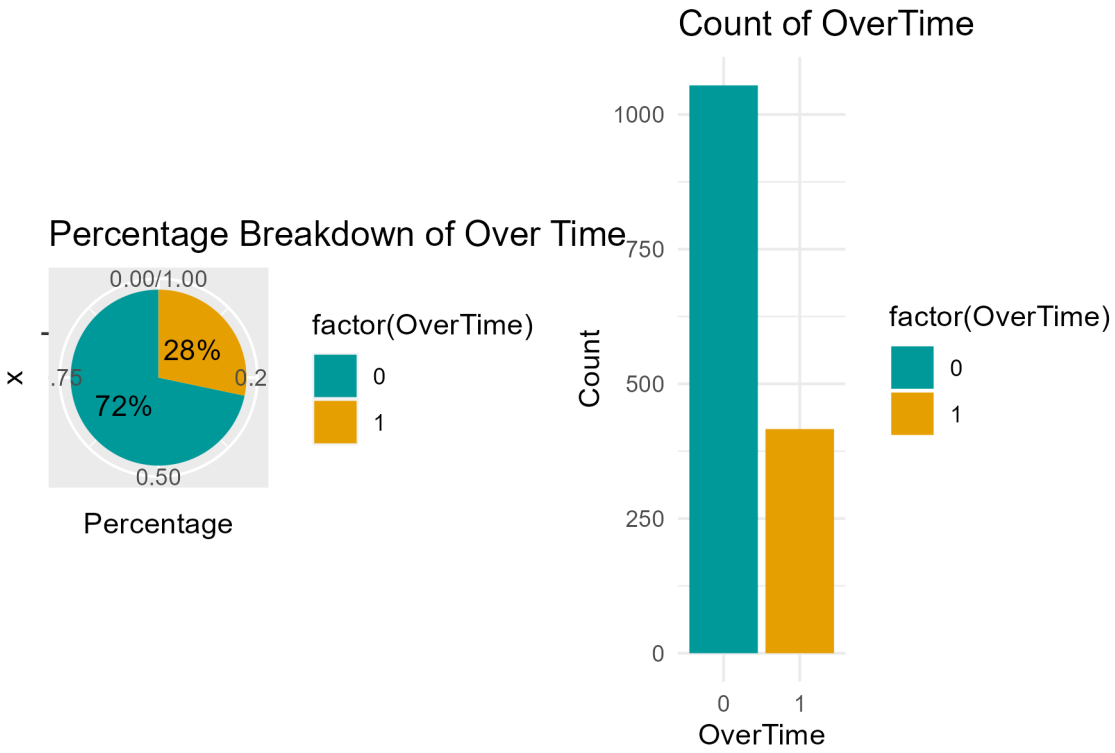

Figure 3.10: Plot breaking down OverTime rates and count

Given the importance of OverTime identified in this section a quick assessment of this variable is important and is show in 3.10.

CHAPTER 4

Predictive Modelling

As identified earlier, 16% of employees leave, this gives a baseline accuracy of 84% for the models to beat, as that would be the model accuracy if it assigned all employees as not leaving. Thus, achieving greater than this in the evaluative metrics shows the effectiveness of the model, the higher above 84%, the more effective the model. This is a reasonably high baseline which, although it may be more difficult to achieve, does mean should that the model perform better, a highly accurate and effective model will have been made. It should be noted that due to the highly imbalanced nature of this data set, Accuracy will not be the only, or potentially even the primary, evaluative metric used. As stated in 2, several evaluative metrics will be used, the model's effectiveness will be compared using all of these - accuracy is the only metric which the data gives us without needing to build a model.

4.1 Full Model

Using the training data set from the earlier data preparation stage, an initial logistic regression model is built using all variables in the set as predictors for Attrition. This resulted in a model AIC of 684.58 and use of McFadden's pseudo R-Squared (PR2) score, revealed a score of 0.3320. Due to the nature of logistic regression, R-Squared scores are not produced, thus McFadden's PR2 score is used instead, which operates in a very similar way, summarising the proportion of variation in the dependent that is explained by the predictors. The above AIC and PR2 scores would be the baseline to compare

models against going forwards. Using the vif() command to check for co-linearity shows Department as having a score of over 95, clearly highlighting the need for its removal. After removing this variable from both the train and test sets another vif() check confirms no variable has a value greater than 4, indicating its required removal. After this logistic regression is run again, reducing model AIC to 682.16 and PR2 to 0.3302.

Analysing the results of this model, of the PCs MaritalStatus and Age were considered statistically significant; Gender, TotalWorkingYears, and StockOptionLevel all had p-values over 0.5 so are likely unimportant to the model. OverTime, both Job Satisfaction and Involvement, alongside BusinessTravel, and EnvrionmentSatisfation, were the only non-PC predictors to be statistically significant at 0.001>. While other variables were considered statistically significant, only the ones with the lowest p-values were mentioned here.

Boruta feature selection is then applied to the latest model using the training data. This gave each of the predictors an importance ranking and dropped those considered least important. The importance rankings for all predictors can be seen in 4.1, this includes those dropped from the model.

It is clear to see that OverTime is considered the most impactful of all the predictors, and by some margin. This is expected given OverTime had the highest correlation with Attrition of any variable. MonthlyIncome, TotalWorkingYears, Age, and JobLevel make up the rest of the top 5 impactful predictors. OverTime and MonthlyIncome are in line with what could be expected from the literature, given it was identified that variables that increase work-life conflict could be expected to have a disproportionate impact. However, under this logic it is strange that WorkLifeBalance had a middling impact in comparison. Important to note, that of the top 5 impactful predictors, Age was earlier identified as Protected Characteristic, and TotalWorkingYears was identified as a potential PC substitute; thus the later removal of these predictors in the filtered model may have a major impact on its effectiveness. Furthermore, it can be seen that MaritalStatus - another PC - is also considered highly important to the model alongside StockOptionLevel. This would not have been an obvious conclusion, given the findings from the earlier discussed literature.

Comparatively, MonthlyRate, PerformanceRating, and Gender were highlighted as

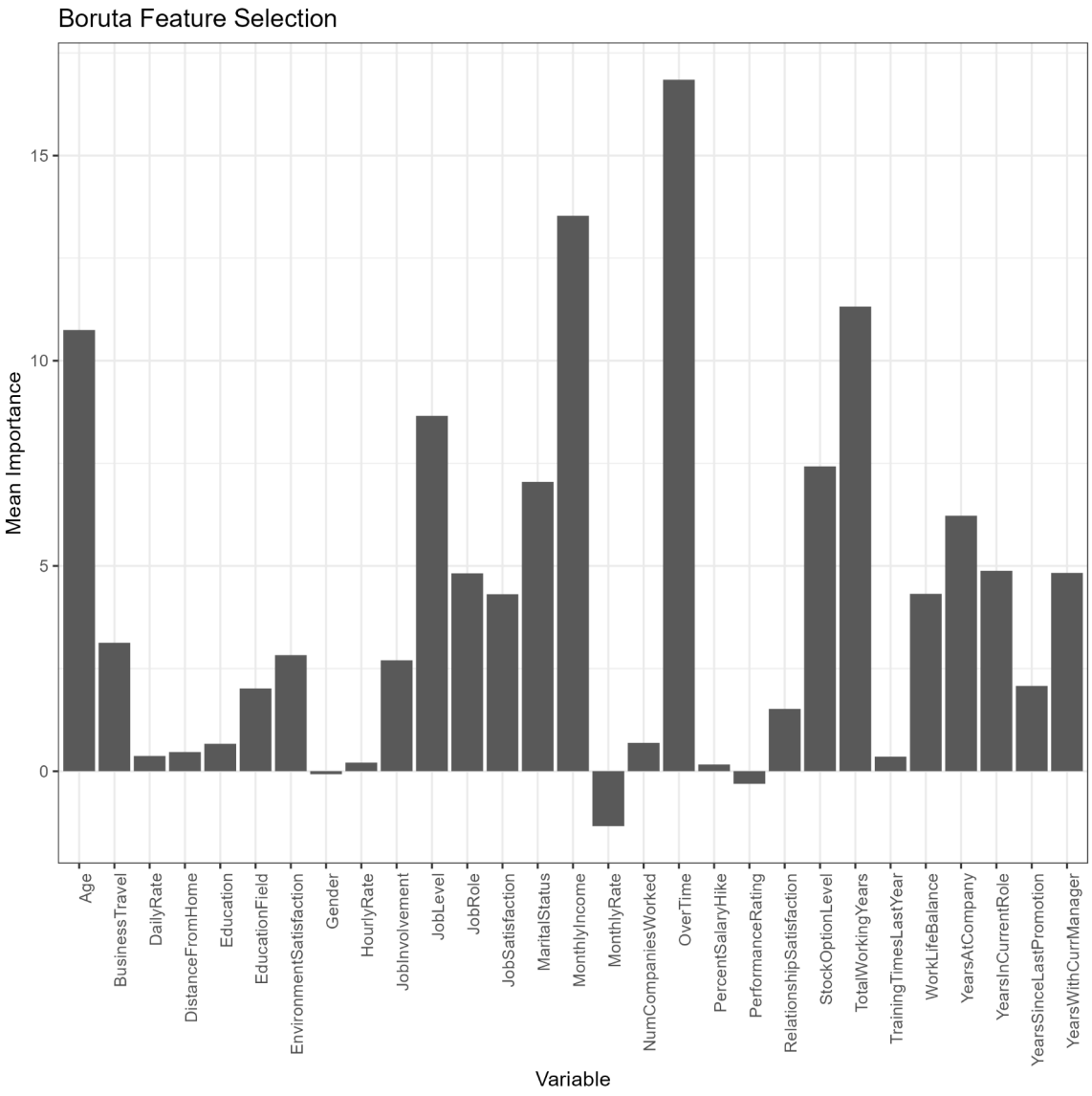

Figure 4.1: Boruta Feature Selection Predictor Importance Rankings

being of negative value to the prediction model and were thus removed from the model by Boruta. Additionally, HourlyRate, PercentSalaryHike, and TrainingTimesLastYear were considered by Boruta to be of minimal value to the model. Additionally, while the top 5 most or least important variables is not related to the FSM and is generally an arbitrary number, it is an easy way to assess the outputs across different FSM and data sets. Ultimately Boruta reduced the number of predictors to 14. Of these relatively unimportant or detrimental variables, only one (Gender) is considered a PC.

With the application of Boruta feature selection a new logistic regression model was made with the recommended variables. Again Age and MaritalStatus were the only PCs considered statistically significant, with TotalWorkingYears and StockOptionLevel

have p-values of >0.5 - Gender was removed from the model by the FSM. OverTime, BusinessTravel, and JobSatisfaction were the only non-PC variables to have a p-value below 0.001. 4.1 shows a limited selection of predictors from the Boruta model, with only the most impactful variables or PCs shown in it. As can be seen in 4.1 YearsAtCompany has the highest odds ratio (OR) of any variable, although it must be noted the 95% confidence interval (CI) has an incredible level of variance - the highest of any variable. OverTime and BusinessTravel are two predictors consistently showing very low p-values, and also have very low variance in their OR. Looking at the PCs, it can be seen that Age and MaritalStatus have ORs below 1 even at the 97.5% CI level; meanwhile TotalWorkingYears has the greatest variance in its CI of the PCs.

Coefficient	OR	2.5%	97.5%
Overtime	4.6980	3.1430	7.0868
TotalWorkingYears	1.5708	0.1183	19.1189
StockOptionLevel	0.9045	0.6324	1.2734
MaritalStatus	0.4977	0.3316	0.7358
BusinessTravel	1.9404	1.3438	2.8184
JobSatisfaction	0.6938	0.5817	0.8250
Age	0.1295	0.0311	0.5045
YearsAtCompany	22.1957	1.0427	412.0708

Table 4.1: Odds Ratio and Confidence Interval for Boruta model

This boruta model has an AIC of 724.17 and a PR2 of 0.2409. Which is oddly worse than the full model; Boruta does not use Accuracy as a metric for feature selection, so this outcome is not the result of the highly imbalanced data set. It could be that the number of variables is impacting Boruta's ability to successfully select the best predictors. While Speiser [10] did recommend Boruta as a FSM, he did also argue that it worked most effectively when there were over 50 predictors, while this data set had only 34 variables at the beginning, before cleaning and preparation.

In an attempt to achieve a better result, the application of VSURF feature selection to the training data set resulted in what can be seen in 4.2. Due to the nature of VSURF's output, a figure showing the rating it gave to all variables cannot be obtained; thus, 4.2 only shows the rating given to the retained predictors. Nevertheless, OverTime

has clearly been identified as the most important predictor here as well as with Boruta. Importantly, both methods identified MonthlyIncome, Age, TotalWorkingYears, and JobLevel as the 5 most influential variables - although VSURF and Boruta place Age and TotalWorkingYears in a slightly different order. This is important as both FSM have selected 2 PCs in the top 5, which does not bode well for the later filtered model.

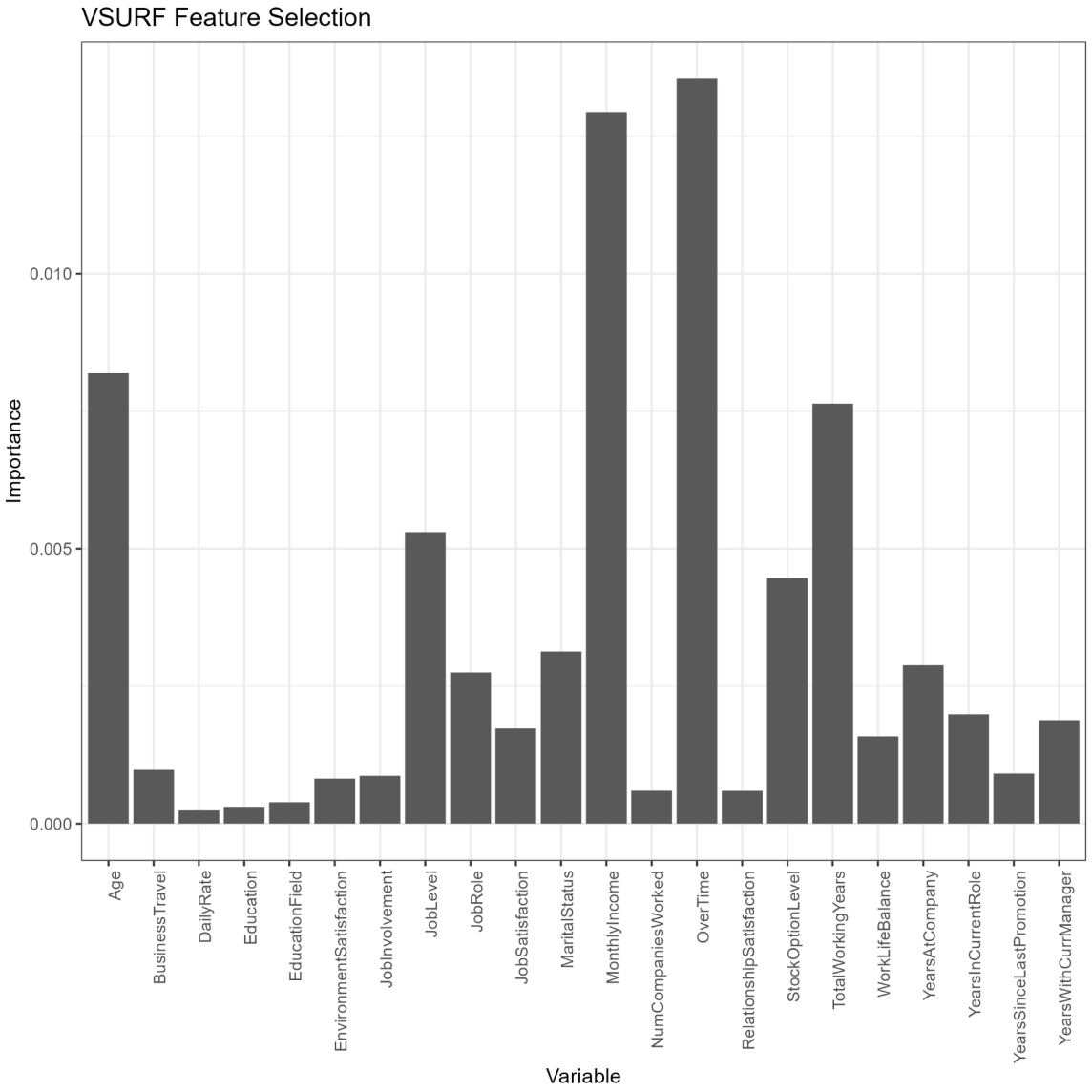

Figure 4.2: VSURF Feature Selection Predictor Importance Rankings

It can also be seen that the two FSM do not agree on the least important features, with VSURF highlighting DailyRate, Education, EducationField, NumCompaniesWorked, and RelationshipSatisfaction as the least impactful of the retained features. Due to their exclusion from 4.2, it can be established that VSURF considers MonthlyRate, PerformanceRating, and Gender, as well as HourlyRate, PercentSalaryHike, and Train-

ingTimesLastYear to be of little use to the model. This is inline with Boruta; however unlike Boruta, VSURF retained many more variables, only lowering the number to 22.

Coefficient	OR	2.5%	97.5%
Overtime	6.1045	3.9442	9.5853
TotalWorkingYears	0.4225	0.0221	6.9487
StockOptionLevel	0.9159	0.6236	1.3237
MaritalStatus	0.4914	0.3168	0.7498
BusinessTravel	2.1073	1.4247	3.1420
JobSatisfaction	0.6798	0.5628	0.8180
Age	0.0944	0.0211	0.3967
YearsAtCompany	37.2527	1.0383	1266.9702
JobInvolvement	0.5896	0.4414	0.7850
EnvironmentSatisfaction	0.6776	0.5593	0.8175

Table 4.2: Odds Ratio and Confidence Interval for VSURF model

With the application of VSURF feature selection, a new logistic regression model was created with the recommended predictors. This lead to a model with an AIC of 683.34 and a PR2 of 0.3133; this is notably better than Boruta based solely on these metrics, yet only somewhat better than the full model. As could be expected, Age and MaritalStatus were the only PCs considered statistically significant, with TotalWorkingYears and StockOptionLevel have p-values of >0.5 - Gender was again removed from the model by the FSM. The variables with p-values of 0.001> were the same as those from the full model. Similarly to 4.1 from the Boruta paragraph, this table 4.2 shows a limited selection of predictors from the VSURF model, with only the most impactful variables or PCs shown in it. Firstly, this model gives OverTime, BusinessTravel, and YearsAtCompany a higher OR than the Boruta model, however YearsAtCompany has much higher CI variance in this model than the Boruta model, with the upper bound being around 3 times higher. Age, MaritalStatus, and StockOptionLevel all have ORs below 1 and low variance in their CI range.

Following from application of the two methods of feature selection, two new data sets were created by copying the training data and stripping out all variables not selected by each of the feature selection methods - leading to one with only variables selected by

Boruta, and one with only variables selected by VSURF. The same was then done for the test data sets. This now ensured for each feature selection method there was a training and test set that contained only the variables considered important by each FSM.

These data sets were then used to train two models; the first based on Boruta's selection - henceforth called Boruta model - and the second based on VSURF's selection - henceforth called VSURF model. These were then used to train the models, before being used on the test data with instructions to predict whether instances should be labelled Attrition or Retained. Prediction was based on a 0.5 probability, meaning if the model considered the likelihood of an instance being classed as Attrition as being above 0.5 then it would assign it to the "Attritioned" category. The results of the models testing can be seen in 4.3.

Model	AUC	Accuracy	Precision	Sensitivity	F-Measure	Specificity
Boruta_Model	0.7881	0.8617	0.7059	0.3200	0.4404	0.9727
VSURF_Model	0.8386	0.8685	0.6977	0.4000	0.5085	0.9645

Table 4.3: Evaluative metrics for Boruta and VSURF logistic regression models

On all metrics apart from Specificity and Precision the VSURF model is markedly better than the Boruta model, and even on those two metrics, Boruta was less than a percentage point better. It is good to note that both models performed better than the baseline accuracy of 84%. However, it is disappointing to note the particularly low sensitivity on both models.

4.2 Filtered Model

As the VSURF model is the best performing model built so far, it will be this model which will be used as the base for this section. After creating a new training and testing data split from the sets used for the VSURF model, each PC was removed and then re-added to assess its impact to the model. It is important to note that Gender will not be tested here, this is because every FSM removed it as a predictor and it had a very low correlation with Attrition - it is assumed that it will be of limited importance and can be safely removed from the models.

Protected Characteristic	AIC Change	McFadden PR2 Change
Age	+8.62	-0.02
MaritalStatus	+9.01	-0.02
TotalWorkingYears	-1.65	-0.0004
StockOptionLevel	-1.79	-0.0002

Table 4.4: : Change in AIC and PR2 Value by Protected Characteristics

As can be seen by 4.4, MaritalStatus is the most influential PC on the model, and its removal will have the most impact out the PCs. Age is not far behind with both leading to an increase in the AIC value of roughly 9 points. However, the removal of both TotalWorkingYears and StockOptionLevel actually improve the model, despite their inclusion by both FSMs. However, this was done on an individual basis, removing and then re-adding each predictor, to see how they each impacted the model. When all four remaining PCs are removed from the model the AIC increases to 723.66 and the PR2 changes from 0.3133 for the VSURF model to 0.2549 for the filtered model.

To ensure that this was the best model that can be created within this dissertation's framework, another model was created using all variables - minus the PCs - as predictors. This model had an AIC of 726.03, which is higher when compared with the filtered VSURF model's AIC of 723.66. Thus, the filtered VSURF model will remain as this dissertation's point of comparison.

Coefficient	OR	2.5%	97.5%
Overtime	5.2900	3.4970	8.0918
YearsSinceLastPromotion	6.7555	1.7721	26.6123
YearsInCurrentRole	0.0680	0.0111	0.4047
BusinessTravel	2.0099	1.3835	2.9400
JobSatisfaction	0.7143	0.5983	0.8506
RelationshipSatisfaction	0.8161	0.6772	0.9823
JobInvolvement	0.5635	0.4283	0.7382
EnvironmentSatisfaction	0.6881	0.5738	0.8224
WorkLifeBalance	0.7254	0.5542	0.9489

Table 4.5: Odds Ratio and Confidence Interval for Filtered VSURF model

With the removal of all PCs the number of statistically significant variables has dropped noticeably. 4.5 shows the OR and 95% CI for the filtered VSURF model; note this only contains coefficients which were considered statistically significant at 0.05>, other coefficients are not shown here. OverTime has consistently had one of the highest OR and least variance in its CI of any coefficient, and this remains the same in this model. Unlike in previous models, YearsAtCompany is not considered statistically significant, and thus its position as coefficient with the highest variance in its CI has been taken by YearsSinceLastPromotion, although this variable does not have a CI anywhere near as wide as YearsAtCompany previously did. Surprisingly, only OverTime, YearsSinceLastPromotion, and BusinessTravel have an OR above one, even when looking solely at the upper end of the 95% CI.

	0 (Actual)	1 (Actual)
0 (Predicted)	352	50
1 (Predicted)	14	25

Table 4.6: Confusion Matrix for the Filtered VSURF Model

Looking at 4.6 it can be seen that - rather unsurprisingly - the biggest area for incorrect predictions is for false negatives. Given the imbalanced nature of the data set, it is not unexpected that the model would have a preference for predicting no turnover. As can be seen throughout this dissertation, all models have a disappointing sensitivity level, likely stemming from this imbalance. However, this imbalance does lend itself to getting very good specificity scores, which can be seen from the very high number of true negative cases in the confusion matrix and the low number of false positives.

Model	AUC	Accuracy	Precision	Sensitivity	F-Measure	Specificity
Filtered_VSURF_Model	0.8353	0.8549	0.6410	0.3333	0.4386	0.9617
VSURF_Model	0.8386	0.8685	0.6977	0.4000	0.5085	0.9645

Table 4.7: Evaluative metrics for the best unfiltered and filtered models

4.7 Shows the best models created with and without the PCs. As can be seen, the filtered model is worse than the model with the PCs, something confirmed by a chi-squared test with a p-value of 2.29e-10; furthermore, the deviance between the two

models is 50.95, which is actually lower than the deviance between the VSURF and the Boruta model. As seen by the low deviance between the two, and from the AIC scores, it can be seen that the filtered vsurf model is actually better than the Boruta model with all the PCs included. Additionally, looking at the differences in the evaluation metrics between the vsurf and filtered vsurf models, it can be seen that the filtered model, while worse, is not that much less effective than the unfiltered model.

CHAPTER 5

Conclusion

The purpose of this dissertation has been a comparative study to identify the most important predictors for employee turnover, and use these to build predictive models, both with and without Protected Characteristics included. This was done to identify if it makes a noticeable difference in the model results, and if the removal of PCs - as defined by UK law - in-line with the recommendations of several prominent thinkers would negatively impact the ability to predict employee turnover. To test this several logistic regression models have been built using two FSMs, with the manual removal of protected characteristics in later models.

Through use of FSMs it was established that a more effective model could be built than simply using all variables as predictors. Additionally, it highlighted the relatively minimal usefulness of Gender as a predictor. This meant one of the five identified protected characteristics, or likely substitutes, could be removed from the model quickly and without majorly impacting the ability to effectively predict employee turnover. The FSMs also highlighted the importance to the model of several predictors, of which OverTime, MonthlyIncome, and JobLevel were considered highly important by both FSMs. Of the remaining four PCs, all were considered to be important, with Age and TotalWorkingYears being considered among the top five most influential by both FSMs. This meant their later removal was expected to noticeably reduce the effectiveness of the predictive model.

With VSURF providing the most effective model before PC filtering, it was used as a base for the filtered model. Testing showed that the removal of some of the remaining

PCs could actually improve the model, with the removal of TotalWorkingYears and StockOptionLevel leading to a reduction of AIC. However, it was also identified that the removal of Age and MaritalStatus were likely to have a major impact on the effectiveness of the model. The removal of all PCs led to a AIC increase of 31.2 points over the unfiltered VSURF model, creating a model worse than the VSURF model and the unfiltered full model. However, it is good to note that their removal led to a better model than using Boruta FSM and simply removing the PCs from the full model.

Unfortunately due to the time and scope limitations inherent in a dissertation, more analysis could not be done on other predictive modelling methods, such as random forests or XGBoost, therefore additional study in using these methods could be useful to gain greater insight into the impact of protected characteristics on prediction. Furthermore, due to the data set used being artificial, it could be useful for further study to be done using natural data, as there may be some subtle yet important changes which cannot be effectively represented in artificially created data.

To conclude, the removal of PCs will negatively affect the effectiveness of a predictive model created to predict employee turnover. However, the decrease seen in various evaluative metrics is often only a few percent when compared with an equivalent unfiltered model 4.7. This signifies that an effective predictive model can still be created even with all protected characteristics removed. It should also be noted that this is using a more stringent list of protected characteristics than has been seen in all literature studied in preparation for this dissertation, and is in-line with the legal definition as described the UK Government. This dissertation has shown that the removal of all protected characteristics still allows the creation of effective predictive models, while also protecting subjects from potential discrimination from their inclusion.

Appendix

A.1 Appendix: Source Codes

```
#install required packages for this report
library(dplyr)
library(ggplot2)
library(gridExtra)
library(ggcorrplot)
library(corrplot)
library(plotly)
library(tidyverse)
library(car)
library(pscl)
library(Boruta)
library(factoextra)
library(pdfCluster)
library(flextable)
library(pROC)
library(DT)
library(caret)
library(VSURF)
library(nortest)
library(GGally)

setwd("C:\\Users\\Peter\\Desktop\\Dissertation") #set working directory

df <- read.csv("C:\\Users\\Peter\\Desktop\\Dissertation\\WA_Fn-UseC_-HR-Employee-Attrition.csv
    ")
head(df)
str(df)
```

```
unique_count <- function(column) { #take the number of unique entries for each column and add
    them
  length(unique(column))
}
col_unique <- sapply(df, unique_count)
print(col_unique) #this is used to get a better understanding of each column

sum(is.na(df))

df <- df %>% select(-Over18) #Drop a column with only 1 possible characteristic
df <- df %>% select(-StandardHours) #Drop a column with only 1 possible characteristic
df <- df %>% select(-EmployeeCount) #Drop a column with only 1 possible characteristic
df <- df %>% select(-EmployeeNumber)
df$Gender <- ifelse(df$Gender == "Male", 1, 2) #change Male to 1 and Female to 2
df$AttritionBinary <- df$Attrition #create a copy for ease of use in graphs
df$Attrition <- ifelse(df$Attrition == "Yes", 1, 0)#change Yes to 1 and No to 0
df$Attrition <- as.integer(df$Attrition)
df$OverTime <- ifelse(df$OverTime == "Yes", 1, 0) #change Yes to 1 and No to 0

df$BusinessTravel <- ifelse(df$BusinessTravel == "Non-Travel", 0, #set a numeric scale of
    travel in role from 0 to 2
                            ifelse(df$BusinessTravel == "Travel_Rarely", 1,
                                   ifelse(df$BusinessTravel == "Travel_Frequently", 2, df$
                                       BusinessTravel)))
df$BusinessTravel <- as.numeric(df$BusinessTravel) #change the factor to numeric

df$MaritalStatus <- ifelse(df$MaritalStatus == "Single", 0, #set a numeric scale of marital
    status in role from 0 to 2
                           ifelse(df$MaritalStatus == "Married", 1,
                                  ifelse(df$MaritalStatus == "Divorced", 2, df$MaritalStatus))
                                  )
df$MaritalStatus <- as.numeric(df$MaritalStatus) #change the factor to numeric




##### DATA VISUALISATION
attach(df)
#Pie chart showing % of people leaving (Figure 1)
f1 <- ggplot(df, aes(x = "", fill = AttritionBinary)) + #create a plot using ggplot2, leave x-
    axis empty
  geom_bar(width = 1, position = "fill") +
  scale_fill_manual(values = c("#009999", "#E69F00")) + #set colours manually
  geom_text(stat = "count", aes(label = paste0(scales::percent(stat(count) / sum(stat(count)))
      , "" , "")),
            position = position_fill(vjust = 0.5),color = "white") + #Add text labels to the
                plot displaying the percentage count
  coord_polar("y", start = 0) +
```

```
  ggtitle("Percentage␣of␣People␣Leaving") +
  guides(fill = guide_legend(title = "Attrition")) #set custom title and legend title
ggsave("f1.png", plot = f1, width = 6, height = 4)
print(f1)
#16% Leaving their job

#Correlation Matrix of all variables (Figure 2)
nonNumeric <- c(5,8,14,32) #select columns which are non-numeric
Cols <- df[, -nonNumeric] #select all columns minus non-Numeric columns
corrM <- cor(Cols) #correlation matrix of all useable variables
corrM[abs(corrM) <= 0.06] <- NA
f2 <- ggcorrplot(corrM, type = "lower", outline.col = "white", lab = TRUE, lab_size = 3,
    colors = c("#6D9EC1", "white", "#E46726")) #plot corr matrix
ggsave("f2.png", plot = f2, width = 10, height = 10)
print(f2)
#a correlation analysis is used to help identify potentially most useful predictor values
#and to see if there are any predictors which are highly correlated and could probably be
    removed from the model

chisq.test(df$MaritalStatus, df$StockOptionLevel) #check for dependency

#Density plot of age (Figure 3)
f3 <- ggplot(data = df, aes(x = Age)) + #using ggplot2 and df create density plots
  geom_density(fill = "#009999") + #create density plot
  labs(x = "Age", y = "Density") +
  ggtitle("Density␣Plot␣of␣Employee␣Ages") #custom title and axis labels
#Looks close to a normal distribution of ages between 18 and 60
shapiro.test(Age)
#A quick test shows that Age is not normally distributed at p<0.05
ggsave("f3.png", plot = f3, width = 6, height = 4)
print(f3)

ks.test(MonthlyIncome, JobLevel)

ks.test(YearsInCurrentRole, YearsWithCurrManager)

#Boxplot of Attrition by Job Level
f10p <- ggplot(df, aes(x = AttritionBinary, y = JobLevel, fill = AttritionBinary)) +
  geom_boxplot(outlier.shape = NA) + #Create boxplot using ggplot2
  geom_jitter(position = position_jitter(width = 0.2), alpha = 0.5) + #Jitter datapoints
  labs(x = "Attrition", y = "Job␣Level",
       title = "Attrition␣by␣Job␣Level") + #Custom title and axis labels
  scale_fill_manual(values = c("#009999", "#E69F00")) + #Set colors
  theme_bw()
#Boxplot of Attrition by Monthly Income
f11p <- ggplot(df, aes(x = AttritionBinary, y = MonthlyIncome, fill = AttritionBinary)) +
  geom_boxplot(outlier.shape = NA) + #Create boxplot using ggplot2
  geom_jitter(position = position_jitter(width = 0.2), alpha = 0.5) + #Jitter datapoints
  labs(x = "Attrition", y = "Income␣($)",
```

```
        title = "Attrition by Monthly Income") + #Custom title and axis labels
  scale_fill_manual(values = c("#009999", "#E69F00")) + #Set colors
  theme_bw()
f11 <- grid.arrange(f10p, f11p, nrow = 1) #combine the two
ggsave("f11.png", plot = f11, width = 6, height = 4)
print(f11)

#Violin of Attrition by Age (Figure Age)
fage <- ggplot(df, aes(x = AttritionBinary, y = Age, fill = AttritionBinary)) +
  geom_violin(outlier.shape = NA) + #create boxplot using ggplot2
  geom_jitter(position = position_jitter(width = 0.2), alpha = 0.5) + #jitter datapoints to
      make it easier to see where they are located on the plot
  labs(x = "Attrition", y = "Age",
       title = "Attrition by Age") + #custom title and axis labels
  scale_fill_manual(values = c("#009999", "#E69F00")) +
  theme_bw()
ggsave("fage.png", plot = fage, width = 6, height = 4)
print(fage)

#Attrition by gender (Figure 12)
f12 <- gridExtra::grid.arrange(grobs = list( #set to plot multiple graphs
  ggplot(subset(df, AttritionBinary == "Yes"), aes(x = Gender)) + #plot 1
    geom_bar(fill = "#E69F00") + #set colour for plot 1
    labs(title = "Attrition by Gender",
         x = "Male (1) or Female (2)",
         y = "Count"), #custom labels
  ggplot(subset(df, AttritionBinary == "No"), aes(x = Gender)) + #plot 2
    geom_bar(fill = "#009999") + #set colour for plot 2
    labs(title = "Retention by Gender",
         x = "Male (1) or Female (2)",
         y = "Count")), #custom labels
  nrow = 1) #plot should be arranged on a single row
ggsave("f12.png", plot = f12, width = 6, height = 4)
print(f12)

#Attrition by Marital Status (Figure MS)
fms <- gridExtra::grid.arrange(grobs = list( #set to plot multiple graphs
  ggplot(subset(df, AttritionBinary == "Yes"), aes(x = MaritalStatus)) + #plot 1
    geom_bar(fill = "#E69F00") + #set colour for plot 1
    labs(title = "Attrition by Marital Status",
         x = "Marital Status: Single, Married, Divorced",
         y = "Count"), #custom labels
  ggplot(subset(df, AttritionBinary == "No"), aes(x = MaritalStatus)) + #plot 2
    geom_bar(fill = "#009999") + #set colour for plot 2
    labs(title = "Retention by Marital Status",
         x = "Marital Status: Single, Married, Divorced",
         y = "Count")), #custom labels
  nrow = 1) #plot should be arranged on a single row
ggsave("fms.png", plot = fms, width = 6, height = 4)
```

```
print(fms)

###########

#Table showing attrition grouped by OverTime, EnvironmentSatisfaction, and JobSatisfaction (
    Table 1)
df <- df %>% #Add new column grouping ratings into positive and negative to make table more
    readable
  mutate(EnvironmentOpinion = case_when(
    EnvironmentSatisfaction %in% c("1", "2") ~ "Negative",
    EnvironmentSatisfaction %in% c("3", "4") ~ "Positive",
  ))
df <- df %>% #Add new column grouping ratings into positive and negative to make table more
    readable
  mutate(JobOpinion = case_when(
    JobSatisfaction %in% c("1", "2") ~ "Negative",
    JobSatisfaction %in% c("3", "4") ~ "Positive",
  ))
df_attritioned <- df %>% #create new dataframe,
  group_by(OverTime, EnvironmentOpinion, JobOpinion) %>%
  summarize(EmployeesMeetingCriteria = n(), #summarise the data, finding total employees
      meeting criteria
            EmployeesLeaving = sum(AttritionBinary == "Yes")) %>% #calculate number of
                employees within the total that left
  ungroup() %>% #remove grouping
  mutate(Percentage = round(EmployeesLeaving / EmployeesMeetingCriteria * 100,2 )) #calculate
      percentage
t1 <- datatable(df_attritioned, #create datatable visualising the above
                options = list(pageLength = 20, lengthChange = FALSE))
print(t1)
df <- df %>%
  select(-EnvironmentOpinion)
df <- df %>%
  select(-JobOpinion) #Remove added columns that were used for this table
#Overtime seems to be the obvious influencing, how many people do over time then?


OverTime_grouped <- df %>% #Group and establish percentage breakdown of OverTime column
  group_by(OverTime) %>% #This is done to avoid issues with how R was reading the OverTime
      column
  summarise(Percentage = n() / nrow(df))
plot1 <- ggplot(OverTime_grouped, aes(x = "", y = Percentage, fill = factor(OverTime))) +
  geom_bar(stat = "identity", width = 1, position = "fill") +
  geom_text(aes(label = scales::percent(Percentage)), position = position_fill(vjust = 0.5)) +
  coord_polar("y", start = 0) +
  scale_fill_manual(values = c("#009999", "#E69F00")) +
  ggtitle("Percentage␣Breakdown␣of␣Over␣Time")
plot2 <- ggplot(df, aes(x = factor(OverTime), fill = factor(OverTime))) +
  geom_bar() +
```

```
  labs(title = "Count of OverTime", x = "OverTime", y = "Count") +
  scale_fill_manual(values = c("#009999", "#E69F00")) +
  theme_minimal()
# Use grid.arrange to arrange and display the plots
f13 <- grid.arrange(plot1, plot2, nrow = 1)
print(f13)
ggsave("f13.png", plot = f13, width = 6, height = 4)

#Select relevant columns for the pair plot and create a pair plot with customized aesthetics
selected_cols <- df %>%
  select(starts_with("Years"), AttritionBinary)
fpair <- ggpairs(selected_cols,
                 aes(color = AttritionBinary),
                 lower = list(continuous = wrap("smooth", alpha = 0.2, size = 0.5, color = "
                     black")),
                 diag = list(continuous = "barDiag"),
                 upper = list(continuous = wrap("cor", size = 4))) +
  scale_color_manual(values = c("#009999", "#E69F00")) +
  scale_fill_manual(values = c("#009999", "#E69F00")) +
  theme(axis.text = element_text(size = 8),
        panel.background = element_rect(fill = "white"),
        strip.background = element_rect(fill = "white"),
        strip.background.x = element_rect(colour = "black"),
        strip.background.y = element_rect(colour = "black"),
        strip.text = element_text(color = "black", face = "bold", size = 8)) +
  labs(title = "Pair Plot of Years by Attrition",
       subtitle = "Scatter Plot, Histogram, and Correlation Coefficient",
       x = NULL, y = NULL)
print(fpair)
ggsave("fpair.png", plot = fpair, width = 10, height = 10)

#stacked bar plot showing SOL by MS
fsol <- ggplot(df, aes(x = MaritalStatus, fill = factor(StockOptionLevel))) +
  geom_bar(position = "stack") +
  labs(title = "Stock Option Level by Marital Status",
       x = "Marital Status: Single, Married, Divorced", y = "Count") +
  scale_fill_manual(values = c("#fafa6e", "#64c987", "#00898a", "#2a4858")) +  #Define custom
      colour scale
  theme_minimal()
print(fsol)
ggsave("fsol.png", plot = fsol, width = 6, height = 4)


#Create a density plot of totalworkyears
ftwy <- ggplot(df, aes(x = TotalWorkingYears, fill = "TotalWorkingYears")) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of TotalWorkingYears",
       x = "TotalWorkingYears", y = "Density") +
  theme_minimal()
```

```
#Create a density plot for Age and overlay it
fd2 <- ftwy +
  geom_density(data = df, aes(x = Age, fill = "Age"), alpha = 0.5) +
  labs(title = "Distribution of TotalWorkingYears and Age",
       x = "Years", y = "Density") +
  scale_fill_manual(values = c("TotalWorkingYears" = "#64c987", "Age" = "#2a4858"))
print(fd2)
ggsave("fd2.png", plot = fd2, width = 6, height = 4)

#Adjust TWYs to account for age of people
df$TotalWorkingYearsAdj <- df$TotalWorkingYears + 23
#recreate above plot with new column
fadj <- ggplot(df, aes(x = TotalWorkingYearsAdj, fill = "TotalWorkingYearsAdj")) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of TotalWorkingYearsAdj",
       x = "TotalWorkingYearsAdj", y = "Density") +
  theme_minimal()
#Create a density plot for Age and overlay it
fta <- fadj +
  geom_density(data = df, aes(x = Age, fill = "Age"), alpha = 0.5) +
  labs(title = "Distribution of TotalWorkingYears and Age",
       x = "Years", y = "Density") +
  scale_fill_manual(values = c("TotalWorkingYearsAdj" = "#64c987", "Age" = "#2a4858"))
print(fta)
ggsave("fta.png", plot = fta, width = 6, height = 4)
#drop column created for this plot
df <- df %>%
  select(-TotalWorkingYearsAdj)

#p.44
Attritioned <- df[df$Attrition=="1",]
Retained <- df[df$Attrition=="0",]

wilcox.test(DistanceFromHome~Attrition, data=df) #Run statistical comparison for x between the
    two levels.
wilcox.test(WorkLifeBalance~Attrition, data=df) #repeat for WLB
#shows the distribution of WLB and DFH is noticeably different between the two outcomes
wilcox.test(Age~Attrition, data=df) #Statistically significant
wilcox.test(MaritalStatus~Attrition, data=df) #Statistically significant
wilcox.test(Gender~Attrition, data=df) #Not Statistically significant

wilcox.test(JobLevel~Attrition, data=df)
wilcox.test(MonthlyIncome~Attrition, data=df)

ks.test(Age, TotalWorkingYears) #The two have a significant difference in distribution

str(df)
#Perform statistical tests on variables with 2 levels.
chisq.test(df$Gender, df$Attrition) #nothing statistically significant
```

```
chisq.test(df$OverTime, df$Attrition) #very significant, high xsq and vlow p-val



##### Predictive Modeling 1 - Logistic Regression

#####Normalisation and Train/Test Splitting
#Normalise continuous variables to ensure best results from random forest feature selection
    methods
set.seed(123)
str(df)
to_drop <- c(32) #Select columns to drop, really just duplicates
df <- df[,-to_drop] #drop columns
index <- createDataPartition(df$Attrition, p = 0.7, list = FALSE) #split data into train and
    test sets
train_data <- df[index, ]
test_data <- df[-index, ]
continuous_vars <- c(1,4,6,11,17,18,21,25,28,29,30,31) #select continuous variables to
    normalise

train_data_norm <- data.frame(train_data) #copy original data into data frame to normalise
for (var in continuous_vars) { #use min-max scaling to normalise data in training set
  min_val <- min(train_data[[var]])
  max_val <- max(train_data[[var]])
  train_data_norm[[var]] <- (train_data[[var]] - min_val) / (max_val - min_val)
  attr(train_data_norm[[var]], "min_val") <- min_val
  attr(train_data_norm[[var]], "max_val") <- max_val
}

test_data_norm <- data.frame(test_data) #copy original data into data frame to normalise
for (var in continuous_vars) { #use min-max scaling to normalise data in training set
  min_val <- attr(train_data_norm[[var]], "min_val")
  max_val <- attr(train_data_norm[[var]], "max_val")
  test_data_norm[[var]] <- (test_data[[var]] - min_val) / (max_val - min_val)
}

str(train_data_norm)
str(test_data_norm)

#Preliminary Feature Selection and testing (VIF, AIC, McFadden...)
model1 <- glm(Attrition ~., family = "binomial", data = train_data_norm) #create the first
    model using all other variables
summary(model1) #show output of above regression model
vif(model1) #check for colinearity
pR2(model1)["McFadden"] #psuedo R-Squared (pr2) value
#AIC: 684.58, pr2: 0.3320

train_data_norm <- train_data_norm %>% select(-Department) #drop Department column as it has
    very high colinearity
```

```
test_data_norm <- test_data_norm %>% select(-Department)
model2 <- glm(Attrition ~., family = "binomial", data = train_data_norm) #run the new model
summary(model2) #lowered AIC slgihtly
pR2(model2)["McFadden"] #slightly lower pr2 score
#AIC: 682.16, pr2: 0.3302
vif(model2) #no colinearity left

"""
OverTime, JobSat, MaritalStatus, JobInvol, EnvrionSat, BusinessTrav: the only ones with p-val
below 0.001. Age statsig at 0.0017. Not sig: Gender 0.55; SOL 0.53; TWY 0.54.
"""

##Boruta Feature Selection
#Boruta applies a random forest to rate features by their importance in the model
boruta1 <- Boruta(Attrition ~ ., family="binomial", data = train_data_norm, doTrace = 1) #run
    boruta feature selection
str(boruta1)
decision <- boruta1$finalDecision
signif <- decision[boruta1$finalDecision %in% c("Confirmed")] #select significant predictors
    according to boruta algorithm
print(signif)#print significant predictors


attStats(boruta1) #(Table 2) create table showing feature importance

boruta_scores <- attStats(boruta1)
boruta_scores$variable <- rownames(boruta_scores)
f14 <- ggplot(boruta_scores, aes( x = variable, y = meanImp)) +
  geom_bar(stat = "identity" ) +
  labs(x = "Variable" , y = "Mean Importance", title = "Boruta Feature Selection" ) +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  #Rotate labs by 90 degrees
#ggsave("f14.png", f14, height = 8, width = 8)
print(f14)

#Create model from Boruta selection
model4 <- glm(Attrition ~ Age + BusinessTravel + JobLevel + JobRole + JobSatisfaction +
                MaritalStatus + MonthlyIncome + OverTime + StockOptionLevel +
                    TotalWorkingYears +
                WorkLifeBalance + YearsAtCompany + YearsInCurrentRole + YearsWithCurrManager,
              family = "binomial", data = train_data_norm)
summary(model4)
pR2(model4)["McFadden"]
#AIC: 724.17, pr2: 0.2409
exp(cbind(OR = coef(model4), confint(model4)))

coef_names_model4 <- names(coef(model4)) #put all coefs in a list
print(coef_names_model4)
coeff_names <- c("OverTime", "TotalWorkingYears", "StockOptionLevel", #Specify coefs to
```

```
    display
                "MaritalStatus", "BusinessTravel", "JobSatisfaction", "Age", "YearsAtCompany"
                    )
invalid_names <- setdiff(coeff_names, coef_names_model4) #find difference
if (length(invalid_names) > 0) {
  stop(paste("Invalid coefficient names:", paste(invalid_names, collapse = ", ")))
} #check to make sure the following code will work
selected_coeffs <- coef(model4)[coeff_names] #Extract coefs and ci
selected_ci <- confint(model4)[coeff_names, ]
result <- exp(cbind(OR = selected_coeffs, selected_ci)) #combine for use in table
print(result)

"""
p-val >0.001: BusTrav, JobSat, MS, OT. StatSig: Age 0.004. NotSig: TWY 0.727; SOL 0.57.
Gender dropped.
YearsAtCompany OR 22.2, 1.04-412.07 CI, highest impact and variance; Age low OR at 0.13;
BusTrav 1.94 OR and little variance in CI; MS 0.5 OR with all variance under 1; SOL 0.9 OR
    with +/-0.3;
TWY 1.57 OR, but variance between 0.1 and 19.1; OT at 4.7 OR and limited var between 3-7;
"""

##VSURF Feature Selection
str(train_data_norm)
#Move column to the end of the dataframe to make life easier
Attrition <- train_data_norm[, "Attrition"] #Extract Attrition
train_data_norm <- train_data_norm[, -2] #Remove the column from its current position
train_data_norm <- cbind(train_data_norm, Attrition) #Re-add the column to the last position

vsurf1 <- VSURF(train_data_norm[,1:29], train_data_norm[,30]) #run the feature selection
    function
vsurf1
summary(vsurf1)
head(vsurf1)
plot(vsurf1, var.names=TRUE) #plot the findings from the vsurf function (figure 14)

predictors <- train_data_norm[,-30]
var_order <- vsurf1$varselect.thres
var_imp <- vsurf1$imp.varselect.thres
var_name <- names(predictors)[var_order]
vsurf_df <- data.frame(Variable=var_name, Importance=var_imp)

f15 <- ggplot(vsurf_df, aes(x=Variable, y=Importance)) +
  geom_bar(stat = "identity") +
  labs(x = "Variable", y = "Importance", title = "VSURF Feature Selection") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  #Rotate x-axis labs by 90 degrees
ggsave ("f15.png", f15, height =8 , width =8)
print(f15)
```

```
vsurf_cols <- vsurf1$varselect.thres
train_vsurf <- train_data_norm %>% select(all_of(vsurf_cols))
train_vsurf <- cbind(train_vsurf, Attrition)
model5 <- glm(Attrition ~ ., family = "binomial", data = train_vsurf) #build a model off
    recommended predictors
summary(model5)
pR2(model5)["McFadden"]
#AIC: 683.34, pr2: 0.3133

exp(cbind(OR = coef(model5), confint(model5)))

coef_names_model5 <- names(coef(model5)) #put all coefs in a list
print(coef_names_model5)
coeff_names5 <- c("OverTime", "TotalWorkingYears", "StockOptionLevel", #Specify coefs to
    display
                  "MaritalStatus", "BusinessTravel", "JobSatisfaction", "Age", "YearsAtCompany"
                      ,
                  "JobInvolvement", "EnvironmentSatisfaction")
invalid_names5 <- setdiff(coeff_names5, coef_names_model5) #find difference
if (length(invalid_names5) > 0) {
  stop(paste("Invalid␣coefficient␣names:", paste(invalid_names5, collapse = ",␣")))
} #check to make sure the following code will work
selected_coeffs5 <- coef(model5)[coeff_names5] #Extract coefs and ci
selected_ci5 <- confint(model5)[coeff_names5, ]
result5 <- exp(cbind(OR = selected_coeffs5, selected_ci5)) #combine for use in table
print(result5)

"""
OT,␣JobSat,␣BusTrav,␣JobInv,␣EnvironSat␣=␣0.001>.␣Age,␣MS␣=␣0.01>.␣TWY,␣SOL␣=␣0.5<.
OT␣at␣6.1;␣YearsAtComp␣37.25,␣but␣with␣1-1267␣95%␣range;␣BusTrav␣2.11␣but␣with␣only␣+/-1␣95%;
SOL␣0.92␣+/-0.3␣95%;␣TWY␣0.42␣but␣has␣potential␣to␣be␣0.02-6.95;␣Age␣0.094,␣0.02-0.4;
JobRole␣consistent␣above␣1␣OR,␣but␣high␣variance␣95%␣CI.
"""

#Create new train/test splits with the columns selected from earlier feature selection
train_boruta <- train_data_norm[,(colnames(train_data_norm)
                                  %in% c("Attrition","Age","BusinessTravel", "JobLevel",
                                         "JobRole","JobSatisfaction","MaritalStatus","
                                             MonthlyIncome",
                                         "OverTime","StockOptionLevel","TotalWorkingYears","
                                             YearsAtCompany",
                                         "WorkLifeBalance", "YearsInCurrentRole","
                                             YearsWithCurrManager"))]
test_boruta <- test_data_norm[,(colnames(test_data_norm)
                                %in% c("Attrition","Age","BusinessTravel", "JobLevel",
                                       "JobRole","JobSatisfaction","MaritalStatus","
                                           MonthlyIncome",
                                       "OverTime","StockOptionLevel","TotalWorkingYears","
                                           YearsAtCompany",
```

```
                                        "WorkLifeBalance", "YearsInCurrentRole","
                                            YearsWithCurrManager"))]

str(vsurf_cols)
str(test_data_norm)
Attrition <- test_data_norm[, "Attrition"] #Extract Attrition
test_data_norm <- test_data_norm[, -2] #Remove the column from its current position
test_data_norm <- cbind(test_data_norm, Attrition) #Re-add the column to the last position
#this now matches train_data_norm.

test_vsurf <- test_data_norm %>% select(all_of(vsurf_cols)) #populate that dataframe with the
    predictors from the model
test_vsurf <- cbind(test_vsurf, Attrition) #add attrition to new dataframe

str(test_vsurf)
##Comparison of Feature Selection Methods
#Build a table comparing several different metrics judging model success between the three
    main feature selection methods

attach(train_boruta)
predict_boruta <- predict(model4, newdata = test_boruta, type = "response") #Make predictions
    for boruta logistic regression model using test data
odds_boruta <- ifelse(predict_boruta > 0.5, "1", "0") #Store prediction

matrix_boruta <- table(odds_boruta, test_boruta$Attrition) #Make confusion matrix (Rows =
    Prediction, Columns = Actual)
matrix_boruta

ROC_boruta <- roc(test_boruta$Attrition, predict_boruta) #Establish ROC curve
TP_boruta <- matrix_boruta["1", 2] #Select true positives
TN_boruta <- matrix_boruta["0", 1] #Select true negatives
FP_boruta <- matrix_boruta["1", 1] #Select false positives
FN_boruta <- matrix_boruta["0", 2] #Select false negatives

accuracy_boruta <- round((TP_boruta + TN_boruta)/(TP_boruta + FP_boruta + TN_boruta + FN_
    boruta), digits = 4) #Establish model accuracy
sensitivity_boruta <- round(TP_boruta / (TP_boruta + FN_boruta), digits = 4) #Establish model
    sensitivity/recall
specificity_boruta <- round(TN_boruta / (TN_boruta + FP_boruta), digits = 4) #Establish model
    specificity
precision_boruta <- round(TP_boruta / (TP_boruta + FP_boruta), digits = 4) #Establish model
    precision
fm_boruta <- round(2 * (precision_boruta * sensitivity_boruta) / (precision_boruta +
    sensitivity_boruta), digits = 4) #Establish model F1 measure

cat("Area␣Under␣ROC␣Curve␣(AUC):", round(auc(ROC_boruta), digits = 4), "\n")
cat("Accuracy:", accuracy_boruta, "\n")
cat("Precision:", precision_boruta, "\n")
cat("Sensitivity/Recall:", sensitivity_boruta, "\n")
```

```
cat("F␣Measure:", fm_boruta, "\n")
cat("Specificity:", specificity_boruta, "\n") #Print the evaluation metrics
detach(train_boruta)


#For the final time, repeat the above for model5, or the vsurf feature selection model
attach(train_vsurf)
predict_vsurf <- predict(model5, newdata = test_vsurf, type = "response") #Make predictions
    for vsurf logistic regression model using test data
odds_vsurf <- ifelse(predict_vsurf > 0.5, "1", "0") #Store prediction

matrix_vsurf <- table(odds_vsurf, test_vsurf$Attrition) #Make confusion matrix (Rows =
    Prediction, Columns = Actual)
matrix_vsurf

ROC_vsurf <- roc(test_vsurf$Attrition, predict_vsurf) #Establish ROC curve
TP_vsurf <- matrix_vsurf["1", 2] #Select true positives
TN_vsurf <- matrix_vsurf["0", 1] #Select true negatives
FP_vsurf <- matrix_vsurf["1", 1] #Select false positives
FN_vsurf <- matrix_vsurf["0", 2] #Select false negatives

accuracy_vsurf <- round((TP_vsurf + TN_vsurf)/(TP_vsurf + FP_vsurf + TN_vsurf + FN_vsurf),
    digits = 4) #Establish model accuracy
sensitivity_vsurf <- round(TP_vsurf / (TP_vsurf + FN_vsurf), digits = 4) #Establish model
    sensitivity/recall
specificity_vsurf <- round(TN_vsurf / (TN_vsurf + FP_vsurf), digits = 4) #Establish model
    specificity
precision_vsurf <- round(TP_vsurf / (TP_vsurf + FP_vsurf), digits = 4) #Establish model
    precision
fm_vsurf <- round(2 * (precision_vsurf * sensitivity_vsurf) / (precision_vsurf + sensitivity_
    vsurf), digits = 4) #Establish model F1 measure

cat("Area␣Under␣ROC␣Curve␣(AUC):", round(auc(ROC_vsurf), digits = 4), "\n")
cat("Accuracy:", accuracy_vsurf, "\n")
cat("Precision:", precision_vsurf, "\n")
cat("Sensitivity/Recall:", sensitivity_vsurf, "\n")
cat("F␣Measure:", fm_vsurf, "\n")
cat("Specificity:", specificity_vsurf, "\n") #Print the evaluation metrics
detach(train_vsurf)


table3 <- data.frame( #Create a dataframe containing all the model effectiveness metrics (
    Table 3)
  Model = c("Boruta_Model", "VSURF_Model"),
  AUC = c(round(auc(ROC_boruta), digits = 4), round(auc(ROC_vsurf), digits = 4)),
  Accuracy = c(accuracy_boruta, accuracy_vsurf),
  Precision = c(precision_boruta, precision_vsurf),
  Sensitivity = c(sensitivity_boruta, sensitivity_vsurf),
  FMeasure = c(fm_boruta, fm_vsurf),
```

```
  Specificity = c(specificity_boruta, specificity_vsurf)
)
print(table3) #print the above dataframe to check for errors

save_table3 <- flextable(table3) #convert dataframe into a flextable
save_as_docx(save_table3, path = "table3n.docx") #save flextable

anova(model4, model5, test = "Chisq")
#p-val of 2.932e-09 shows signifcant difference and that model5 is noticeably better
#deviance shows model 5 is better than model 4
#Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
#1      1007     680.17
#2       995     615.34 12   64.829 2.932e-09 ***

#######

"""
drop␣from␣train_data_norm␣the␣PCs␣and␣build␣a␣model␣without␣the␣PCs␣from␣that.

So␣now␣we␣take␣model5,␣the␣better␣model,␣and␣selectively␣take␣away␣the␣PCs␣to␣find␣out␣how
these␣affect␣the␣overall␣AIC.

VSURF␣Model␣(model5)␣is␣now␣the␣competitor,␣the␣new␣model␣build␣without␣the␣PCs␣will␣be
compared␣to␣this␣model.

PCs␣=␣Age,␣MaritalStatus,␣TotalWorkingYears,␣StockOptionLevel
Gender␣has␣been␣removed␣by␣two␣FSMs,␣thus␣it␣will␣not␣be␣included␣(however␣could␣be␣readded␣to
    ␣tes)
"""

train_vsurf_filtered <- train_vsurf #create new data frames for us to change
test_vsurf_filtered <- test_vsurf

str(train_vsurf_filtered)
str(test_vsurf_filtered)

#model2: 682.16
#model4: 724.17
#Model5: AIC 680.71 PR2 0.3117
#So Boruta makes the model worse, however Vsurf prove effective as a FSM.
#VSURF model is the basis moving forward.

#Age test
train_vsurf_filtered <- train_vsurf_filtered %>% select(-Age)
str(train_vsurf_filtered)
modela <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(modela)
pR2(modela)["McFadden"]
#AIC 689.33 pr2 0.2999
```

```
#So Age is equal to ~9 AIC increase and ~0.02 drop in pr2

#MS test
train_vsurf_filtered <- train_vsurf #reset
train_vsurf_filtered <- train_vsurf_filtered %>% select(-MaritalStatus) #rerun with new PC
str(train_vsurf_filtered)
modelms <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(modelms)
pR2(modelms)["McFadden"]
#AIC 689.79 pr2 0.2994
#So MS is equal to ~9 AIC increase and ~0.02 drop in pr2

#TWY test
train_vsurf_filtered <- train_vsurf #reset
train_vsurf_filtered <- train_vsurf_filtered %>% select(-TotalWorkingYears) #rerun with new PC
str(train_vsurf_filtered)
modeltwy <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(modeltwy)
pR2(modeltwy)["McFadden"]
#AIC 679.06 pr2 0.3113
#TWY is equal to ~0.8 AIC decrease and a pr2 change of -0.0004
#Removing this PC actually improves the model

#SOL test
train_vsurf_filtered <- train_vsurf #reset
train_vsurf_filtered <- train_vsurf_filtered %>% select(-StockOptionLevel) #rerun with new PC
str(train_vsurf_filtered)
modelsol <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(modelsol)
pR2(modelsol)["McFadden"]
#AIC 678.92 pr2 0.3115
#SOL is equal to ~2 AIC decrease and a pr2 change of -0.0002
#again removing SOL benefits the model


#Continuing the removal of PCs one at a time from the model to see the final result.
#Remove TWY + SOL
train_vsurf_filtered <- train_vsurf_filtered %>% select(-TotalWorkingYears) #rerun with new PC
str(train_vsurf_filtered)
model7 <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(model7)
#AIC 677.26
#Further reduction in AIC of ~1.7

#Remove TWY + SOL + Age
train_vsurf_filtered <- train_vsurf_filtered %>% select(-Age) #rerun with new PC
str(train_vsurf_filtered)
model8 <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(model8)
```

```
#AIC 692.48
#AIC increase of ~15.2

#Remove TWY + SOL + Age + MS
train_vsurf_filtered <- train_vsurf_filtered %>% select(-MaritalStatus) #rerun with new PC
str(train_vsurf_filtered)
model8 <- glm(Attrition ~ ., family = "binomial", data = train_vsurf_filtered)
summary(model8)
pR2(model8)["McFadden"]
#model8 - AIC 723.66 0.2549
#AIC increase of ~31.2 from last model
# model5 - AIC: 683.34, pr2: 0.3133

#Run full model minus PCs to see if VSURF is better
str(train_data_norm)
full_data_filtered <- train_data_norm %>% select(-Age, -MaritalStatus, -Gender,
                                                 -TotalWorkingYears, -StockOptionLevel)
str(full_data_filtered)
model9 <- glm(Attrition ~ ., family = "binomial", data = full_data_filtered)
summary(model9)
pR2(model9)["McFadden"]
#model2 - 682.16 0.3302 (full model)
#model9 - 726.03 0.2701

coef_names_model8 <- names(coef(model8)) #put all coefs in a list
print(coef_names_model8)
coeff_names8 <- c("OverTime", "YearsSinceLastPromotion", "YearsInCurrentRole",
                  "BusinessTravel", "JobSatisfaction", "RelationshipSatisfaction",
                  "JobInvolvement", "EnvironmentSatisfaction", "WorkLifeBalance") #Specify
                      coefs to display
invalid_names8 <- setdiff(coeff_names8, coef_names_model8) #find difference
if (length(invalid_names8) > 0) {
  stop(paste("Invalid coefficient names:", paste(invalid_names8, collapse = ", ")))
} #check to make sure the following code will work
selected_coeffs8 <- coef(model8)[coeff_names8] #Extract coefs and ci
selected_ci8 <- confint(model8)[coeff_names8, ]
result8 <- exp(cbind(OR = selected_coeffs8, selected_ci8)) #combine for use in table
print(result8)



"""
Testing has shown that model5 (VSURF model) is the best model compared with Boruta and full.
When filtering of PCs is applied to both vsurf and full model, vsurf is still superior.
As can be seen in the table, the vsurf model is better than the filtered vsurf model, but the
difference is not so drastic.

We are not here to assess the effectiveness of vsurf compared with no FSM, therefore the final
table will only contain information from vsurf and vsurf_filtered so we can more easily
```

```
    comapre
the␣differences␣found␣from␣removing␣Pcs

anova␣chi␣squared␣test␣-␣vsurf␣model␣is␣better␣than␣vsurf_filtered␣with␣deviance␣of␣50.95,␣
    however␣this␣is␣actually
#a␣lower␣deviance␣than␣between␣vsurf␣and␣boruta.␣So␣model8␣is␣better␣than␣model4.␣p-value␣
    2.291e-10
"""

str(train_vsurf_filtered)
str(test_vsurf_filtered)
test_vsurf_filtered <- test_vsurf_filtered %>% select(-Age, -MaritalStatus,
                                                      -TotalWorkingYears, -StockOptionLevel)

#Build a table comparing the two models through eval metrics
attach(train_vsurf_filtered)
predict_filtered <- predict(model8, newdata = test_vsurf_filtered, type = "response") #Make
    predictions for boruta logistic regression model using test data
odds_filtered <- ifelse(predict_filtered > 0.5, "1", "0") #Store prediction

matrix_filtered <- table(odds_filtered, test_vsurf_filtered$Attrition) #Make confusion matrix
    (Rows = Prediction, Columns = Actual)
matrix_filtered

ROC_filtered <- roc(test_vsurf_filtered$Attrition, predict_filtered) #Establish ROC curve
TP_filtered <- matrix_filtered["1", 2] #Select true positives
TN_filtered <- matrix_filtered["0", 1] #Select true negatives
FP_filtered <- matrix_filtered["1", 1] #Select false positives
FN_filtered <- matrix_filtered["0", 2] #Select false negatives

accuracy_filtered <- round((TP_filtered + TN_filtered)/(TP_filtered + FP_filtered + TN_
    filtered + FN_filtered), digits = 4) #Establish model accuracy
sensitivity_filtered <- round(TP_filtered / (TP_filtered + FN_filtered), digits = 4) #
    Establish model sensitivity/recall
specificity_filtered <- round(TN_filtered / (TN_filtered + FP_filtered), digits = 4) #
    Establish model specificity
precision_filtered <- round(TP_filtered / (TP_filtered + FP_filtered), digits = 4) #Establish
    model precision
fm_filtered <- round(2 * (precision_filtered * sensitivity_filtered) / (precision_filtered +
    sensitivity_filtered), digits = 4) #Establish model F1 measure

cat("Area␣Under␣ROC␣Curve␣(AUC):", round(auc(ROC_filtered), digits = 4), "\n")
cat("Accuracy:", accuracy_filtered, "\n")
cat("Precision:", precision_filtered, "\n")
cat("Sensitivity/Recall:", sensitivity_filtered, "\n")
cat("F␣Measure:", fm_filtered, "\n")
cat("Specificity:", specificity_filtered, "\n") #Print the evaluation metrics
detach(train_vsurf_filtered)
```

```
table5 <- data.frame( #Create a dataframe containing all the model effectiveness metrics (
    Table 3)
  Model = c("Filtered_VSURF_Model", "VSURF_Model"),
  AUC = c(round(auc(ROC_filtered), digits = 4), round(auc(ROC_vsurf), digits = 4)),
  Accuracy = c(accuracy_filtered, accuracy_vsurf),
  Precision = c(precision_filtered, precision_vsurf),
  Sensitivity = c(sensitivity_filtered, sensitivity_vsurf),
  FMeasure = c(fm_filtered, fm_vsurf),
  Specificity = c(specificity_filtered, specificity_vsurf)
)
print(table5) #print the above dataframe to check for errors

save_table5 <- flextable(table5) #convert dataframe into a flextable
save_as_docx(save_table5, path = "table5.docx") #save flextable

anova(model8, model5, test = "Chisq")
#vsurf model is better than vsurf_filtered with deviance of 50.95, however this is actually
#a lower deviance than between vsurf and boruta. So model8 is better than model4.
```

Bibliography

[1] IBM HR Analytics Employee Attrition and Performance. Kaggle

[2] Benge, E.J., (1925). An index for predicting labor turnover. Journal of Personnel Research, 3, pp.359-365.

[3] Equalities Act 2010, Government of the United Kingdom of Great Britain and Northern Ireland. Gov.UK

[4] Equality and Human Rights Commission. (2021). Protected characteristics. Accessed 17/08/23: Equality Human Rights

[5] *Bridges*. Bridges vs South Wales Police. (2020). Court of Appeal, C1/2019/2670. Judiciary.UK

[6] Speelman, D. (2012). Logistic regression: A confirmatory technique for comparisons *Corpus Linguistics*. PDF

[7] Ponnuru, S., Merugumala, G., Padigala, S., Vanga, R., and Kantapalli, B. (2020). Employee attrition prediction using logistic regression. *IJRASET, 8*(5), pp.2871-2875. Google Scholar

[8] Daghistani, T. and Alshammari, R. (2020). Comparison of statistical logistic regression and random forest machine learning techniques in predicting diabetes. *Journal of Advances in Information Technology Vol, 11*(2), pp.78-83. Semantic Scholar

[9] Tonkin, M., Woodhams, J., Bull, R., Bond, J.W. and Santtila, P. (2012). A comparison of logistic regression and classification tree analysis for behavioural case linkage. *Journal of Investigative Psychology and Offender Profiling, 9*(3), pp.235-258. Wiley Online Library

[10] Speiser, J., Miller, M., Tooze, J., and Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications, 134*, pp.93-101. Science Direct

[11] Mehmood, T., Saebo, S., and Liland, K. H. (2020). Comparison of variable selection methods in partial least squares regression. *Journal of Chemometrics, 34*(6). Wiley Online Library

[12] Kursa, M., Jankowski, A., and Rudnicki, W. (2010). Boruta - A System for Feature Selection. *Fundamenta Informaticae 101*, pp.271-285. DOI: 10.3233/FI-2010-288

[13] Genuer, R., Poggi, J., and Tuleau-Malot, C. (2015). VSURF: An R Package for Variable Selection Using Random Forests. *The R Journal, R Foundation for Statistical Computing, 7*(2), pp.19-33. Semantic Scholar

[14] Brayfield, A., and Crockett, W. (1955). EMPLOYEE ATTITUDES AND EMPLOYEE PERFORMANCE. *PSYCHOLOGICAL BULLETIN, 52*(5), pp.369-424. PDF

[15] Muchinsky, P.M. and Tuttle, M.L., (1979). Employee turnover: An empirical and methodological assessment. *Journal of vocational Behavior, 14*(1), pp.43-77. Science Direct

[16] Cotton, J., and Tuttle, J. (1986). Employee Turnover: A Meta-Analysis and Review with Implications for Research. *The Academy of Management Review, 11*(1), pp.55-70. JSTOR

[17] Dore, R. (1995). The end of jobs for life?: corporate employment systems: Japan and elsewhere. *Centre for Economic Performance, 11*. PDF

[18] Batt, R., and Valcour, M. (2003). Human Resources Practices as Predictors of Work-Family Outcomes and Employee Turnover. *Industrial Relations, 42*(2). Wiley Online Library

[19] Yang, S., and Islam, M. T. (2020). IBM employee attrition analysis. *arXiv preprint arXiv:2012.01286*. PDF

[20] Fallucchi F, Coladangelo M, Giuliano R, and William De Luca E. (2020). Predicting Employee Attrition Using Machine Learning Techniques. *Computers, 9*(4). DOI: 10.3390/computers9040086

[21] Zhao, Y., Hryniewicki, M.K., Cheng, F., Fu, B., and Zhu, X. (2019). Employee Turnover Prediction with Machine Learning: A Reliable Approach. In: Arai, K., Kapoor, S., Bhatia, R. (eds) Intelligent Systems and Applications. *IntelliSys 2018. Advances in Intelligent Systems and Computing, 869*. DOI: 10.1007/978-3-030-01057-7_56

[22] Casale, E. (2022). Around the black box: applying the carltona principle to challenge machine learning algorithms in public sector decision-making. *LSE Law Review, 7*(3), pp.369-389. HeinOnline

[23] Sokol, K., Flach, P. (2020). One Explanation Does Not Fit All. *Kunstl Intell 34*, pp.235-250. DOI: 10.1007/s13218-020-00637-y

[24] *Coll*. Coll vs Secretary of State for Justice. (2017). Supreme Court, UKSC 2015/0148. Supreme Court

[25] Borgesius, F.(2018). DISCRIMINATION, ARTIFICIAL INTELLIGENCE, AND ALGORITHMIC DECISION MAKING. *Directorate General of Democracy, Council of Europe*. Council of Europe

[26] Centre for Data Ethics and Innovation. (2020). Review into bias in algorithmic decision-making. UK Government. Gov.UK

[27] Castille, C.M. and Castille, A.M.R. (2019). Disparate treatment and adverse impact in applied attrition modeling. *Industrial and Organizational Psychology, 12*(3), pp.310-313. DOI: 10.1017/iop.2019.53

[28] Ghosh, A., Kvitca, P., and Wilson, C. (2023). When Fair Classification Meets Noisy Protected Attributes. *AAAI/ACM Conference on AI, Ethics, and Society (AIES 23)*, Montreal, QC, Canada. ACM, New York, NY, USA. PDF

[29] Speer, A.B. (2021). Empirical attrition modelling and discrimination: Balancing validity and group differences. *Human Resource Management Journal*, pp.1-19. Wiley Online Library

[30] Kursa, M., and Rudnicki, W. (2010). Feature Selection with the Boruta Package. *Journal of Statistical Software, 36*(11). DOI: 10.18637/jss.v036.i11

[31] Choudhary, I. (2020). IBM HR Attrition Case Study. *Towards Data Science*. Towards Data Science

[32] Henriksen-Bulmer, J. and Jeary, S. (2016). Re-identification attacks-A systematic literature review. *International Journal of Information Management, 36*(6), pp.1184-1192. DOI: 10.1016/j.ijinfomgt.2016.08.002.

[33] Swaminathan, S., and Hagarty, R. (2019). Data science process pipeline to solve employee attrition. *IBM*. IBM

[34] Kohavi, R., and John, GH. (1997). Wrappers for Feature Subset Selection. *Artificial Intelligence, 97*, pp.273-324. Science Direct

[35] Speiser, J. (2021). A random forest method with feature selection for developing medical prediction models with clustered and longitudinal data. *Journal of Biomedical Informatics, 117*. https://doi.org/10.1016/j.jbi.2021.103763 DOI: 10.1016/j.jbi.2021.103763

[36] Zakrani, A., Hain, M. and Idri, A. (2019). Improving software development effort estimating using support vector regression and feature selection. *IAES International Journal of Artificial Intelligence, 8*(4), p.399-410. Research Gate

[37] Xu, Y., and Goodacre, R. (2018). On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. *J. Anal. Test. 2*, pp.249-262. DOI: 10.1007/s41664-018-0068-2

[38] Saeb, S. *et al* (2017). The need to approximate the use-case in clinical machine learning. *GigaScience, 6*(5). DOI: 10.1093/gigascience/gix019

[39] Wieczorek, J., Guerin, C., and McMahon, T. (2022). K-fold cross-validation for complex sample surveys. *Stat, 11*(1). DOI: 10.1002/sta4.454

[40] Jansche, M. (2005). Maximum expected F-measure training of logistic regression models. *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pp. 692-699. PDF

[41] Gitnux. (2023). Getting A Job After College Statistics 2023: Key Insights And Trends. Accessed: 06/09/2023 GitNux