# **Employee Attrition Modelling**

## **Part 1: The Data Problem**

Employee attrition is the rate at which employees leave a company. The goal of this analysis is to model employee attrition and determine the most dominant contributing factors that govern this turnover. The benefits to the company are substantial: not only is the best talent retained, but recruitment and training costs may be reduced significantly as well.

Through this kind of analysis, we can understand how many employees are likely to leave, while also determining which employees are at the highest risk and for what reasons. Companies face significant costs for having to search for, interview, and hire new employees. In general therefore, a company is highly motivated to retain their employees for a significant period of time.

This analysis is particularly useful if a company wants to lower attrition levels but is unsure of the source of the problem. Conversely, if a company needs to decrease their labor costs and headcounts, then understanding employee attrition can ease the transition by preparing affected groups to lose personnel.

### **Exploratory Data Analysis**

The particular dataset used in this analysis can be found [here](https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset). The dataset contains several pieces of information about each employee such as their department, job satisfaction, years at company, work/life balance and so on. Of all these, there are five that can be used to subset the users in our Shiny app: age, gender, education level, monthly income and marital status. Note that we do not recommend subsetting the dataset by more than three (3) parameters as there may be no data satisfying.

In terms of a machine learning analysis, the data has to be initially cleaned before it can be used. For most projects, cleaning the data is often the most time consuming aspect of the entire process. Generally you would want to fill in missing values, understand (potentially discard) outliers, correct erroneous ones, fix formatting issues and standardize categories. The goal is to make the data as consistent and relevant across the board as possible. This will allow for the maximum accuracy of the final model.

We first conduct an exploratory analysis on the dataset using visual tools, which allows us to summarize the main characteristics of a dataset. From here, we perform machine learning modelling that will determine the probability that each individual will attrite, thus, uncovering the most important factors that lead to overall employee turnover. Based on the needs of the employer, this analysis can also be narrowed down to determine key factors governing attrition for particular demographics, job titles, working groups, and indeed specific individuals.

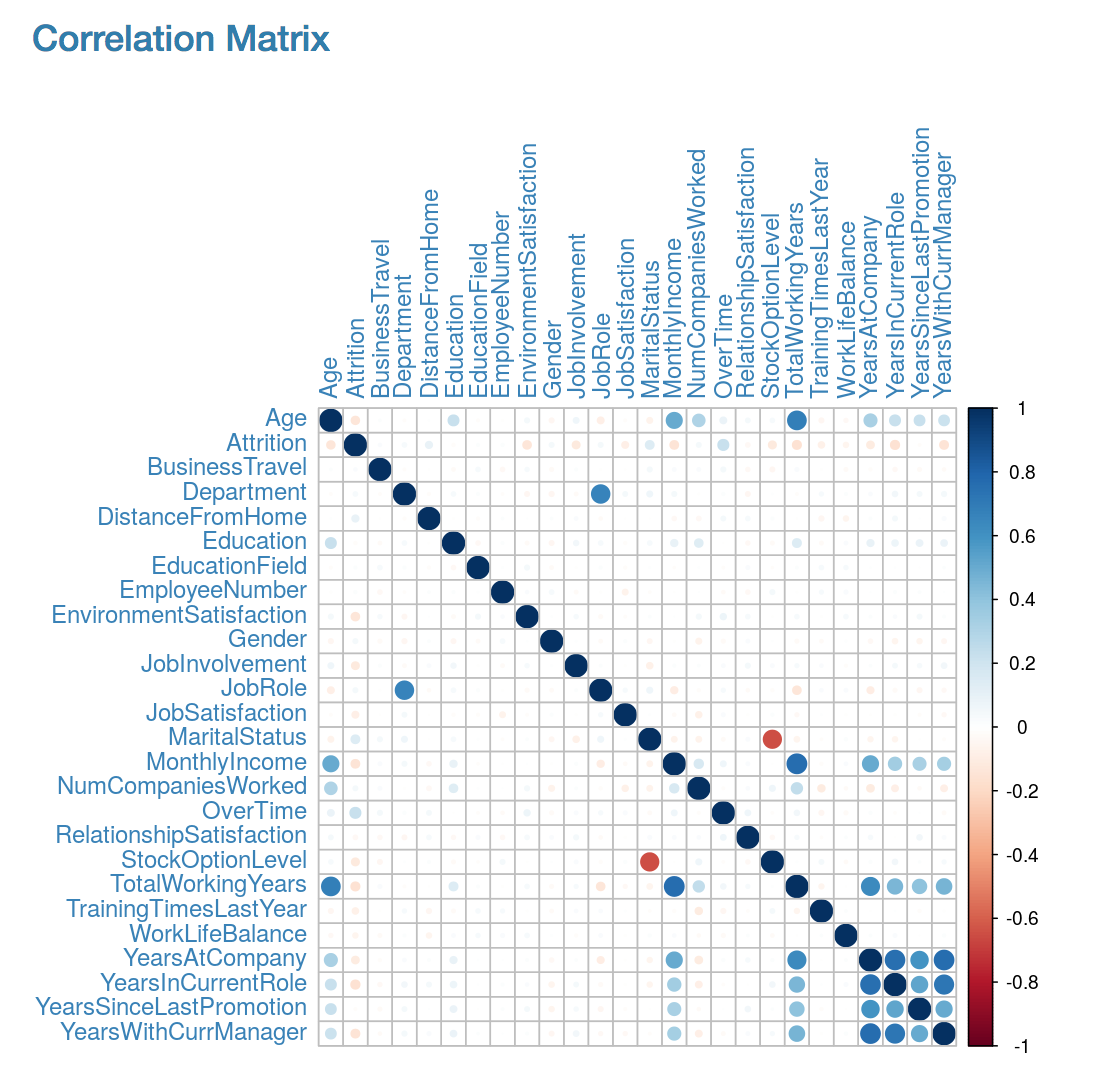


Figure 1: Correlation Matrix for all features

The above correlation matrix displays the linear correlation between every pair of features in the form of dots of varying colors and sizes. A larger dot indicates that the correlation between these selected features is stronger, whereas the color denotes the strength of the positive (blue) or negative (red) correlation coefficient. When two variables are correlated, we are essentially observing that change in one variable is accompanied by change in the other. In this large matrix, it is clear that the majority of features are uncorrelated. However, even for those variables that are correlated, care must be taken when interpreting the correlation as it does not necessarily imply a causal relationship.

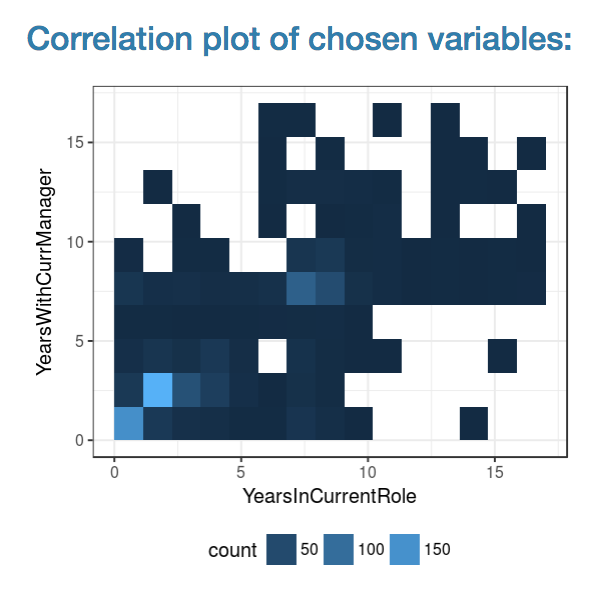


Figure 2: Correlation between Years With Current Manager and Years At Company

This application has an additional functionality: by clicking any element in the correlation matrix, a 2D histogram is displayed in order to better observe the correlation between those features. Correlation between variables allows us to determine the overlap between feature in the dataset. In general, the machine learning algorithm should be given as much uncorrelated information as possible to maximise the predictive accuracy.

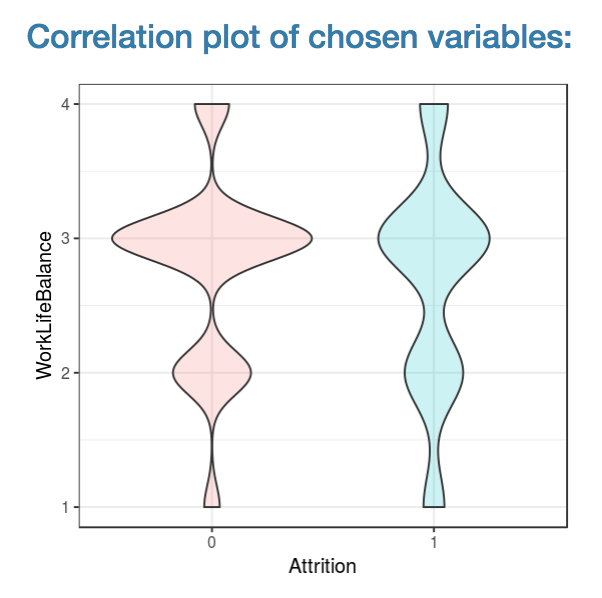


Figure 3: Violin plots of Work/Life Balance, separated by attrition

Alternatively, clicking the elements along the leading diagonal will output violin plots of the selected features, bucketed by the true underlying attrition value (1 indicating employees that attrite, and 0 indicated those that remain). Figure 3 shows the violin plots for the *WorkLifeBalance* variable with those that attrite tending to have fewer mid-range scores. Unlike box plots, violin plots show the full distribution of the data, which is particularly useful if the data is multimodal. If we wanted to, we could even redefine attrition to include mid level ranges and plot the violin plots of those levels.

For more details on this or any potential analyses, please visit us at <http://sflscientific.com> or contact [mluk@sflscientific.com](mailto:mluk@sflscientific.com).

## **Part 2: Data Modeling with Machine Learning**

In this study, we use several algorithms to model employee attrition: extreme gradient boosting (XGBoost), support vector machines (SVM), and logistic regression. XGBoost is a decision tree based algorithm. Multiple trees are ensembled to improve the predictive power of the model. An SVM is a discriminative classifier that takes labeled training data and constructs a hyperplane to categorize new examples. Finally, logistic regression is a simple classifier used to estimate the probability of a binary outcome based on several predictors and a logit function.

All three algorithms are supervised learning methods; these take in a set of labelled data, in this case a historic records of people who have either left the company or stayed on, and learns the underlying patterns in the available data - in this case, from features such as Age, Job Role, Overtime worked etc.

Feature generation is an important aspect of modeling in machine learning. When generating features, we take the data and either decompose or aggregate it in order to answer an underlying question. In this particular case, since we want to know why employees attrite, we build features that explain this phenomenon. The features should be meaningful; that is to say, they are a significant part of an observation that helps the model learn about the structure of a problem.

After the creation of features, we can understand the overall impact of features. The library for one of the algorithms we used, XGBoost, has a built-in ranking function that ranks the importance of features.

Not only does this feature ranking tell us about how important certain features cause for employee attrition but also, we can use it to manually eliminate redundant features. Performing these tasks provide huge benefits such as the speed and simplification of the model, whilst also ensuring greater generalization by reducing the potential for overfitting.

When it comes to building the model, there are several steps to take before any predictions can be made. The data is partitioned into three sets: training, validation and testing. The training set is responsible for initially teaching the model the causal relationship between all information and the attrition probability. The validation set is then used to estimate how well the model has been trained and fine tune the parameters to develop the best model. Once those two steps have been completed, the completed model is applied to the testing set in order to get accurate results on how the model would perform on real-world data. In this case, we can predict the likelihood for any employee to attrite in the future based solely on the quantifiable data that an HR department has access.

### **Fine-Tuning the Results**

Each algorithm gives a confidence score between 0 and 1 for each employee, indicating that the model thinks these individuals are somewhere between 0% and 100% likely to attrite, respectively. By setting the confidence score threshold, above which we predict an employee to leave, we end up with a control on the precision and recall statistics;the cutoff can be adjusted in real-time to optimize the model based on the needs of the business.

In a true analysis, these algorithms would be further tuned and potentially ensembled to provide the most accurate prediction of employee attrition possible. For now, they can be compared using the graphical plots in the app.

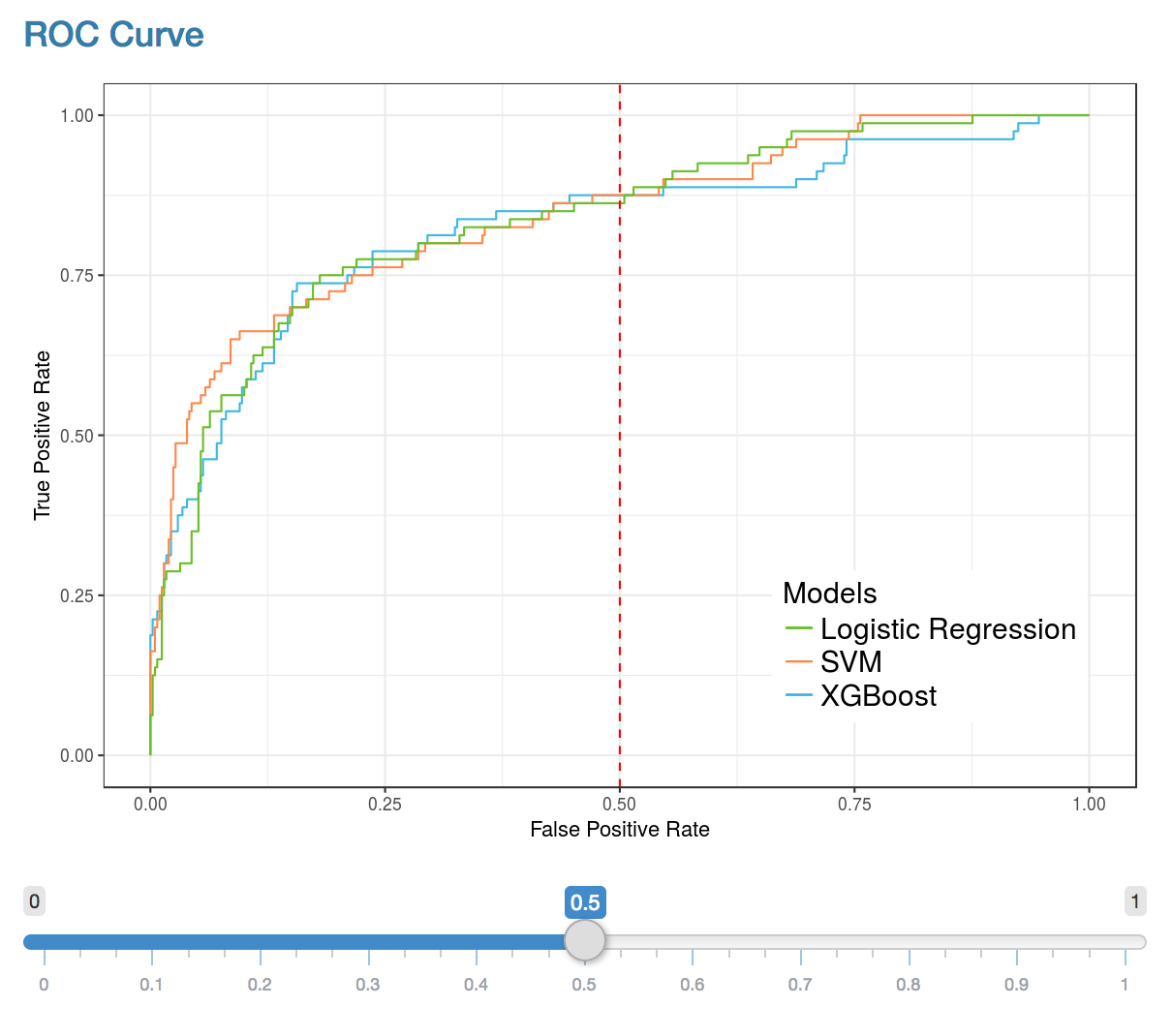


Figure 4: Receiver operating characteristic (ROC) curve

A receiver operating characteristic (ROC) curve is the result of plotting the true positive rate against the false positive rate. The closer the ROC curve is to the top left corner, the greater the accuracy of the test.

The slider allows the user to change the operation point of the algorithm by setting the false positive rate. The changes made to this cut-off are reflected in the confusion matrices shown, where each confusion matrix shows the performance of the predictions of the various algorithms with respect to the true label.

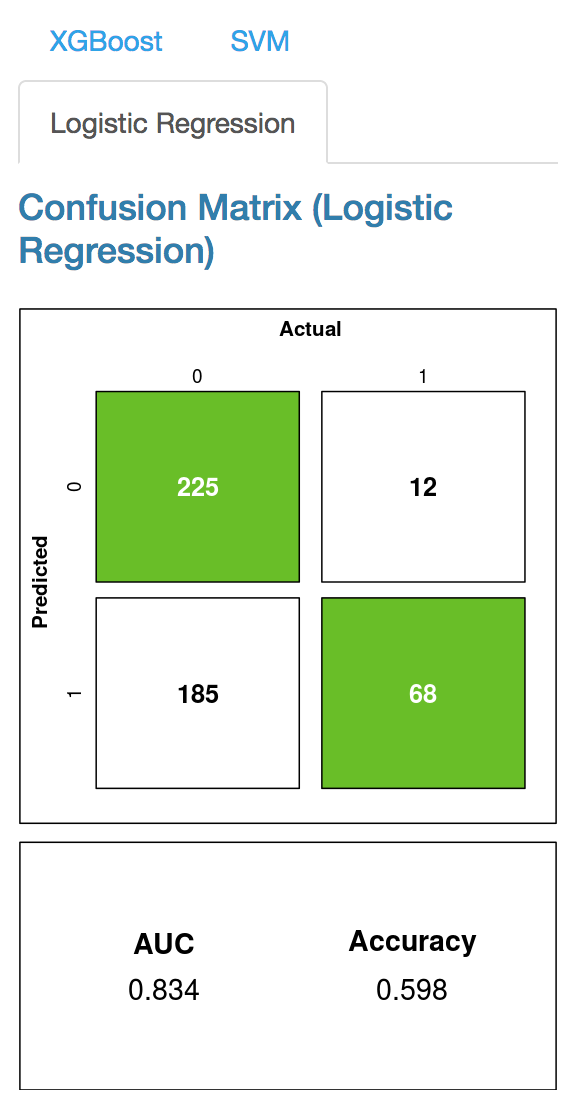
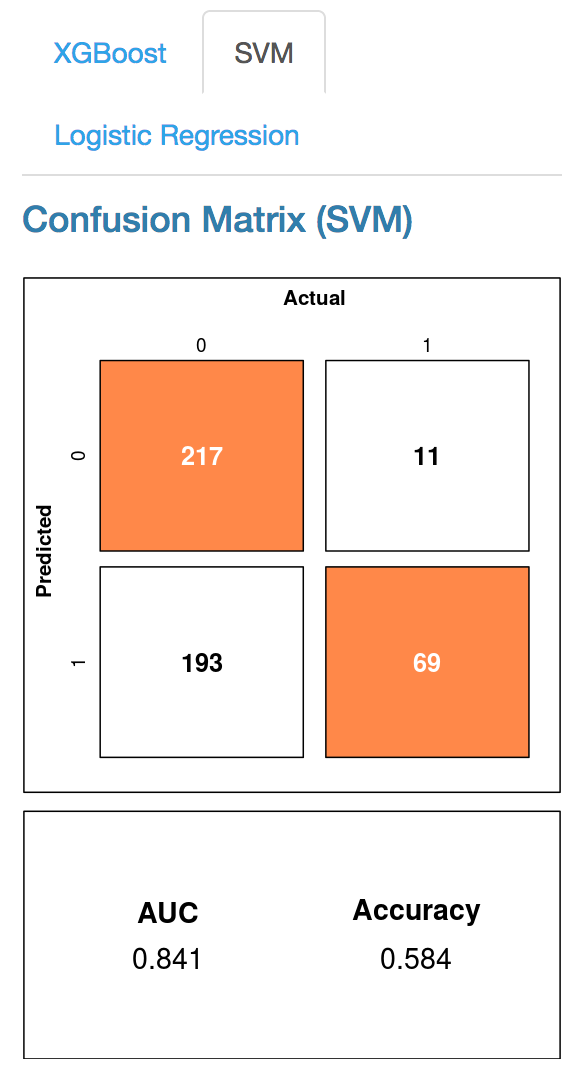
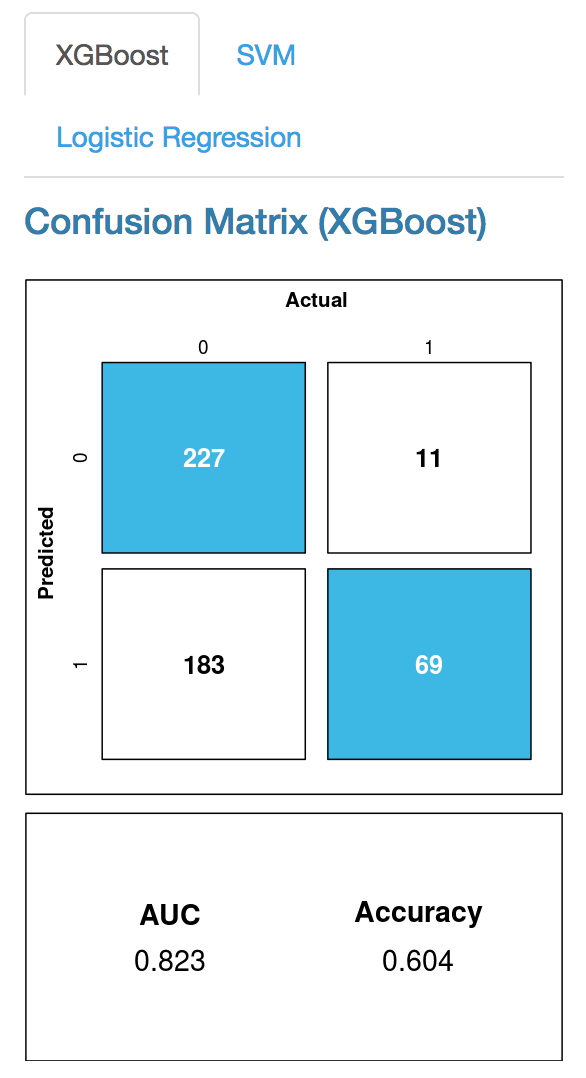


Figure 5: Confusion Matrices for all three models

Another way to visualise this result is to look at precision and recall. These two statistics are important aspects of any classification model and are components of the overall F1 score, given by:

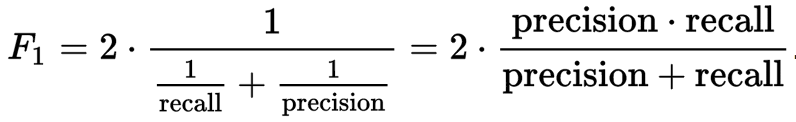


Figure 6: Equation for F1 Score

An ideal model has both high precision and recall; in general, this is difficult to achieve and instead we can choose to change the cutoff point directly to trade some precision for recall and vice-versa. By controlling the tradeoff between False Positives and False Negatives, businesses can determine where they would like to skew their analysis towards.

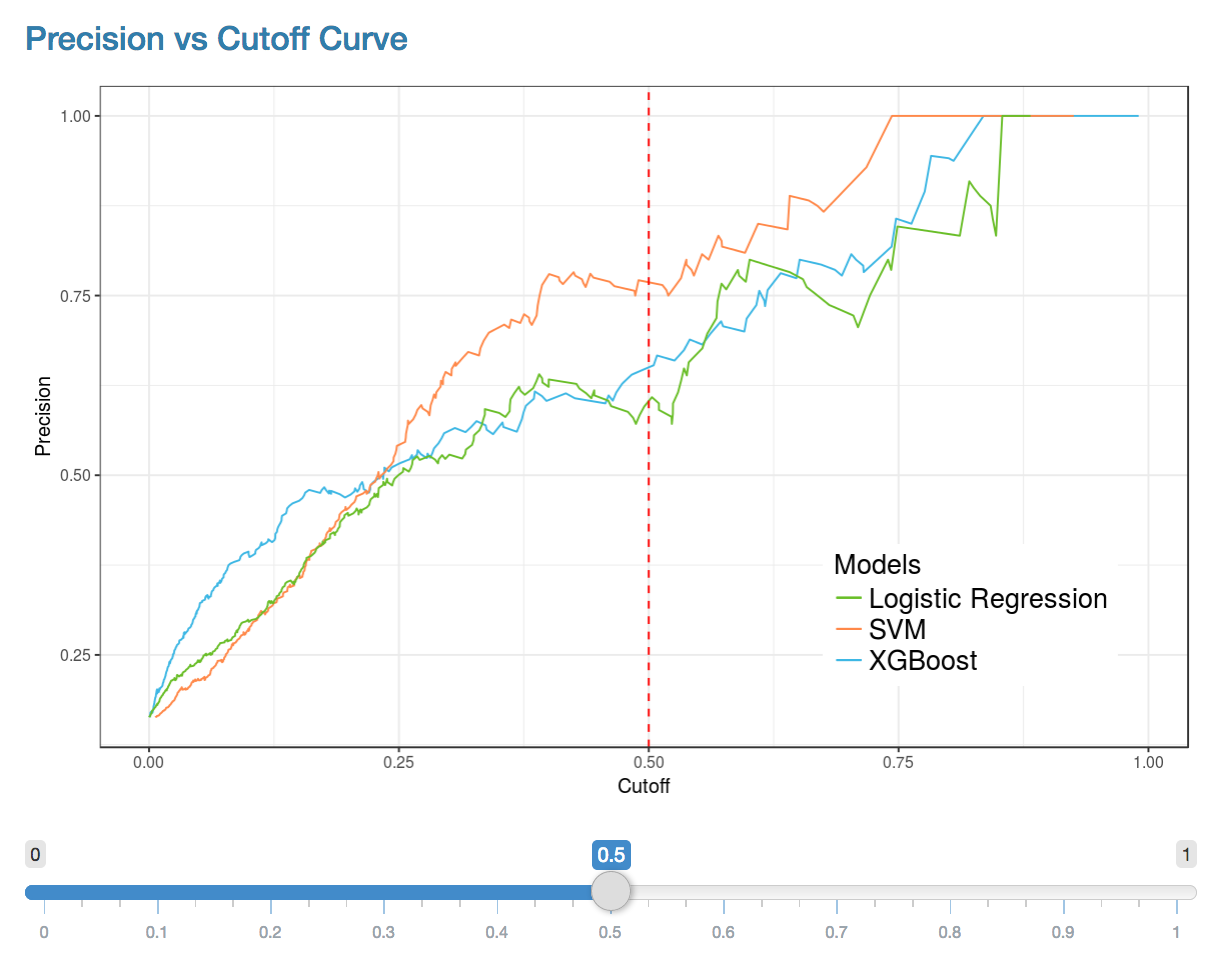


Figure 7: Precision vs Cut-off Curve

As with the ROC curve, we have provided confusion matrices for each algorithm so that the user can see the effect that changes in cutoff have on the final results.

For example, in cases such as medicine where undiagnosed issues have much greater importance, we can skew the operation point to the left - where we find far fewer false negatives (but more false positive) instances. Conversely for businesses with very limited HR resources, it may be best to go for fewer false positives (to the right of the curve), rank the confidence scores given by the raw algorithm, and target only the highest risk employees with incentives.

The confidence score can be combined with any HR metrics (which themselves can be modelled algorithmically if need-be) to give an expected value lost per individual. We can then use this to rank employees in terms value to the company that is likely to be lost. Ultimately this provides the threshold of spending power that HR departments should be given access to retain each specific employee.

To actively address overall employee retention issues, we need to look more closely at the most important features that determine the attrition probability and see if we can improve company retention.

### **Feature Importance**

One of the benefits of using XGBoost is the in-built estimation of feature importance. The importance metric provides a score indicating how valuable each factor was in the construction of the boosted decision trees. Higher relative importance indicates a larger impact on the algorithm and final prediction. Since the importance is calculated explicitly for each attribute, these attributes can be ranked and compared to each other.

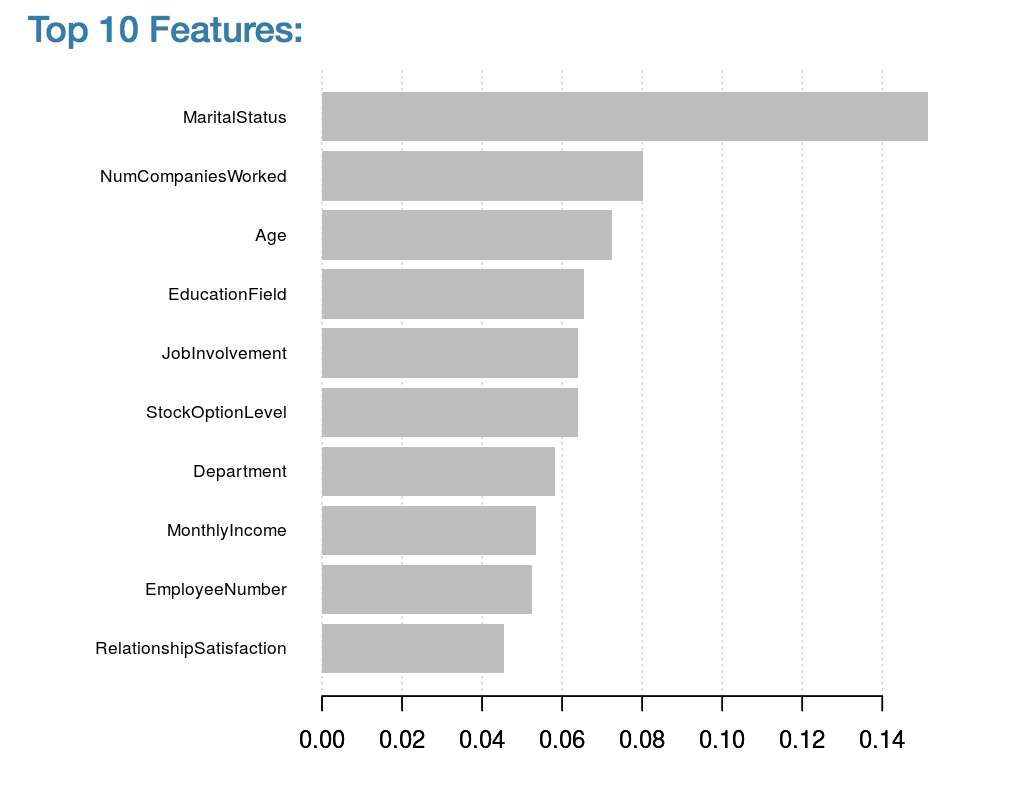


Figure 7: XGBoost Top 10 Features

When the model is run on the entire dataset, the results show that Marital Status, Number of Companies Worked For, and Age are the dominant drivers of employee attrition for this dataset. In terms of HR adjustable parameters, we note that adjustments to Job Involvement, Stock Option and Monthly Income might be used as incentives for high value employees. In addition to this, we can use similar feature importance methods to create an employee specific ranking of controllable factors that can then be used to target employees on an individual-by-individual basis.

The final output of the classifier can be both a ranked list of individuals most likely to leave and also a ranked list of contributing factors that govern each employee’s attrition probability. Both are highly useful for any HR department aiming to minimize talent loss and ultimately save company dollars.

For more details on this or any potential analyses, please visit us at <http://sflscientific.com> or contact mluk@sflscientific.com.

## **Part 3: Construction of the R Shiny App**

R Shiny is a powerful yet intuitive tool for creating interactive web applications in R. Part of its appeal comes from the fact that it handles the code conversion for you; instead of learning HTML, CSS and Javascript, all you need to know is R. These apps can perform any R calculation in the backend, meaning they are just as powerful as a program run on your desktop. Further, Shiny has many options for constructing a user interface that fit the requirements of any project.

The application itself has two main components: the user interface and the server. The user interface controls the appearance and layout of the app, including all the interactive sliders, text boxes and buttons. The server is responsible for performing the calculations and contains the instructions for building the app.

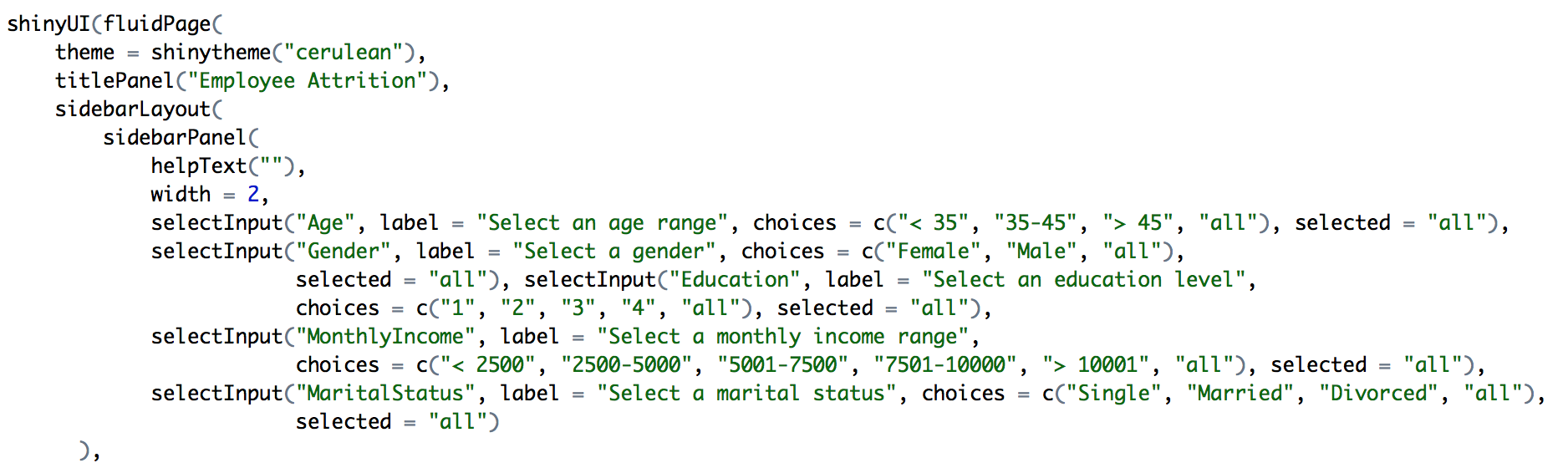


Figure 8: User interface code snippet

Figure 8 shows the code snippet used to set up a section of the user interface. In this case, we created a simple sidebar to allow the user to select from certain sub-groups of employees.



Figure 9: Server code snippet

On the server side, the options from the side bar are stored in variables which can then be applied to the model. The two work as a cohesive unit, with the user interface receiving the changes from the user and passing them along to the server to execute.

The level of interactivity of a Shiny app can vary based on the scope of the project and the developer's needs. When dealing with employee attrition in particular, it is useful to look at specific employees and thus we added several options to allow the user to subset out such groupings.

Note: We limited the subsetting options to five factors: age, gender, education level, monthly income and marital status. The size of the dataset should be taken into account when deciding how many options to provide the user. Certain algorithms need a certain amount of data in order to properly run, and unless the user knows exactly how much data is in each subset, there are likely to be errors. Even within our application, we do not recommend subsetting the data by more than 3 options as plots will become unpopulated. Developers can find workarounds for this issue, for example, by dynamically limiting the options for the user or displaying different plots if necessary.

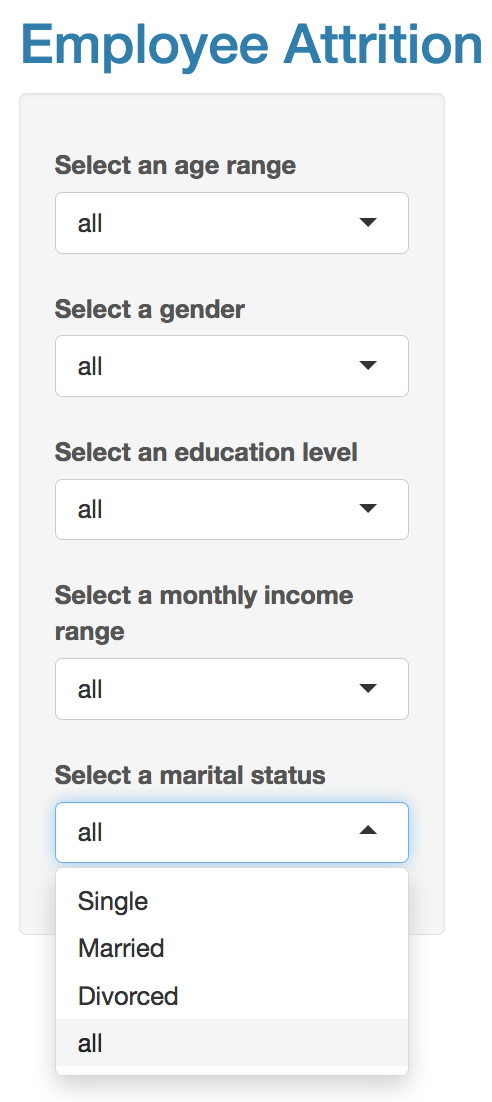


Figure 10: Options to select groups of employees

As previously demonstrated in the algorithms section, the user is able adjust the false positive rate along with the cutoffs for precision and recall. All of these options depend on the current needs of the business and it is for that reason that they appear as a choice. Most of the work goes on behind the scenes with the modeling and calculations done in R. However, having a variety of options in these applications is incredibly important for representing every possible need that might arise.

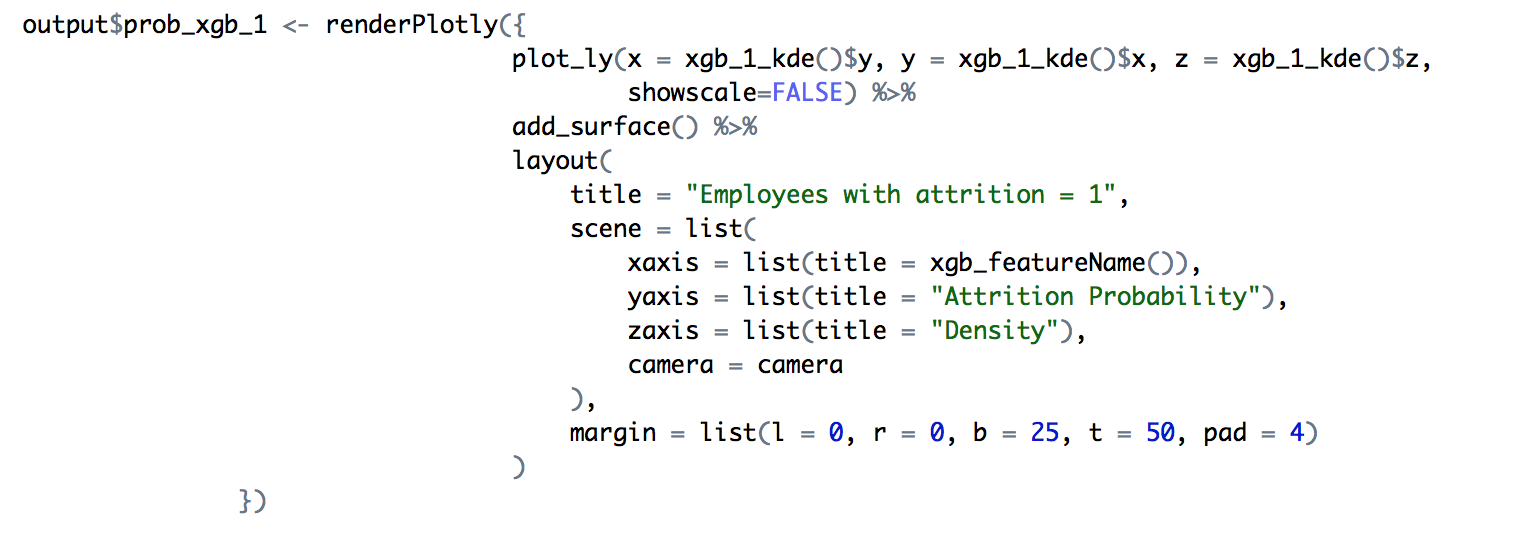


Figure 11: Density plot code snippet

Data visualization is key to any presentation and that is no different for a Shiny application. Creating plots is fairly straightforward; the server seamlessly handles any changes to refresh the information and provide insights on the shape of the data.

The plots themselves are interactive and are displayed side by side to facilitate comparisons.

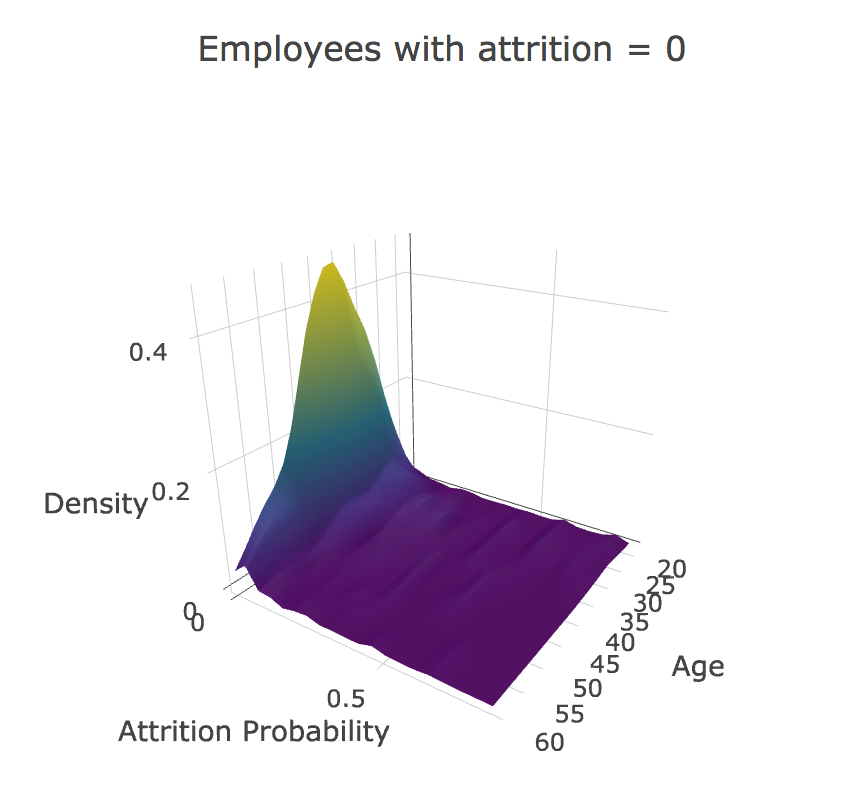
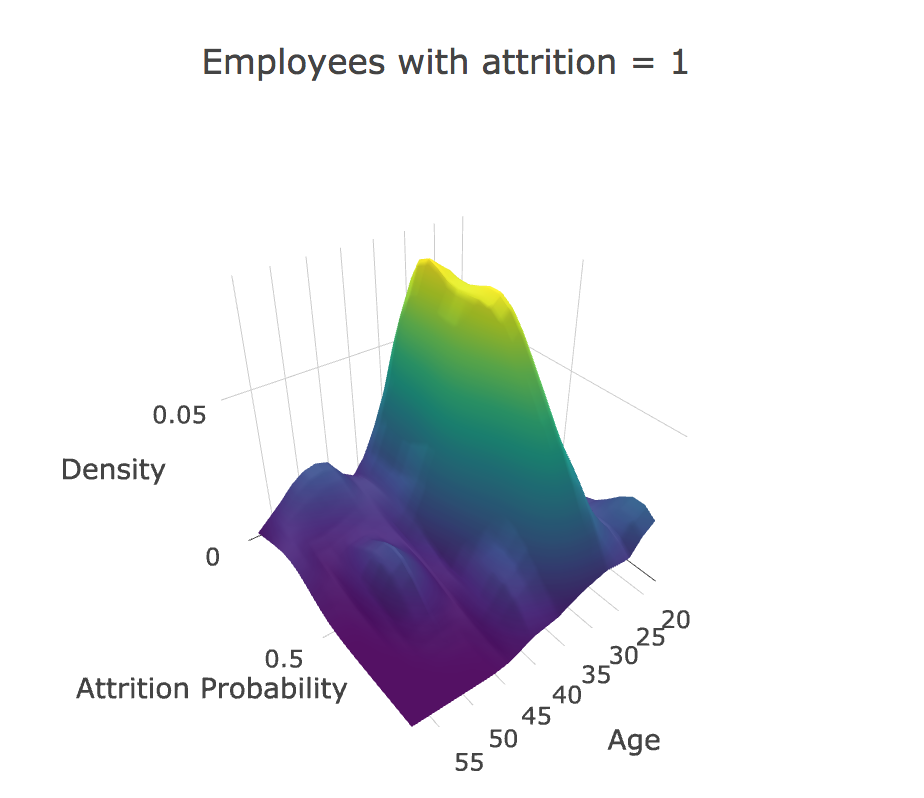


Figure 12: 3D Density Plots for Age, separated by attrition

To validate the prediction on a hold out set, we can also look at the final probability output from each of three classifiers. The histogram shows the confidence scores separated by the true attrition class. This plot shows that those that stay at the company (attrition group = 0) are well classified by our simple algorithm, typically having scores below 0.20. In contrast, many of those who have actually left the company (attrition group = 1) have scores above 0.20.



Figure 13: Histogram plots code snippet

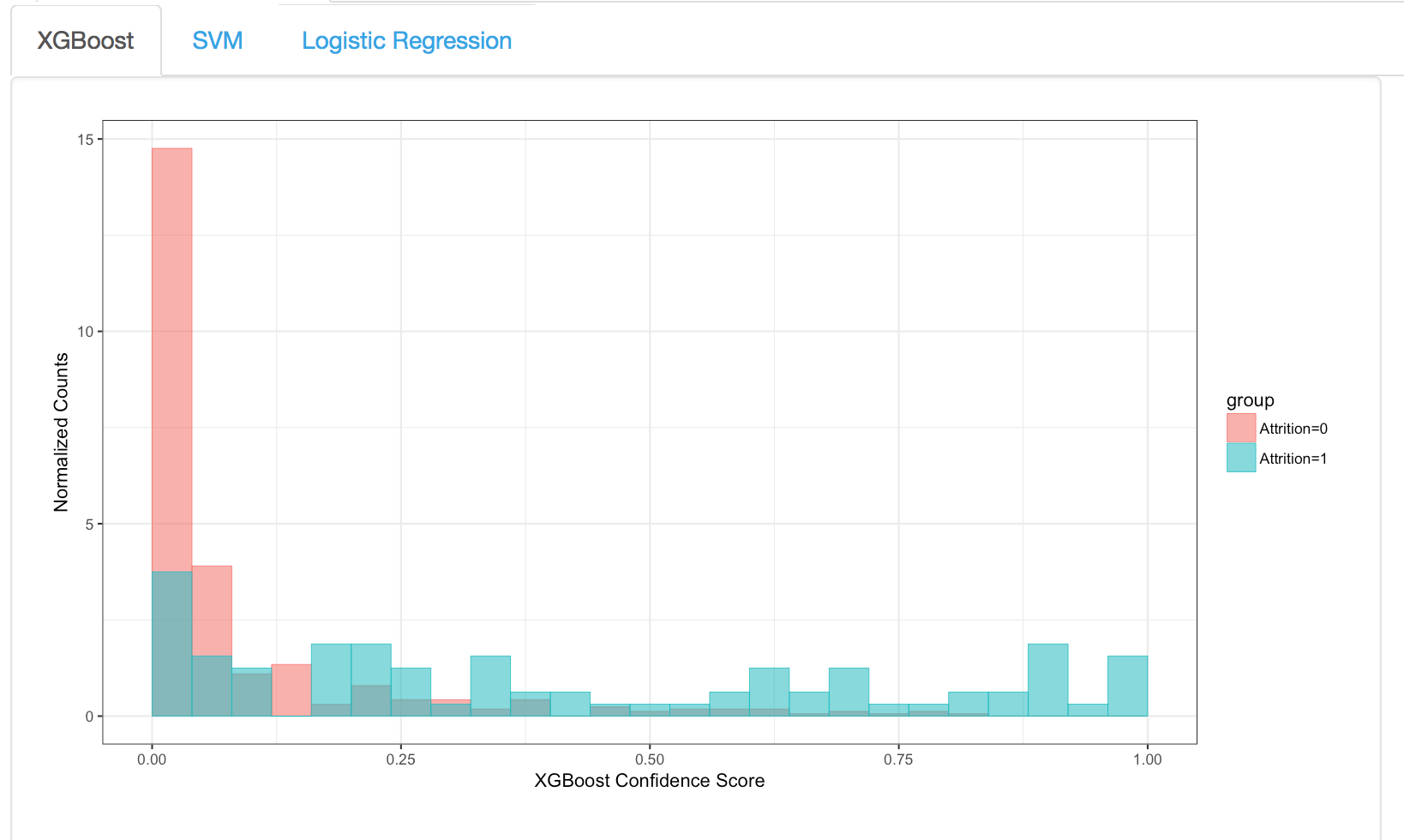


Figure 14: Histogram plots separated by attrition group

Note that due to the overlap of these two histograms, there will always be some number of False Positives and/or False Negatives. This, there is still potential for algorithm improvement that can be made by developing our methods beyond this toy analysis.

For more details on this or any potential analyses, please visit us at <http://sflscientific.com> or contact [mluk@sflscientific.com](mailto:mluk@sflscientific.com).