Analyzing IBM Employees and Attrition

By: Naveen Mirapuri

Github for Project: https://github.com/NaveenM12/DataChallenge_IBM_Employees/
Tableau Download: https://public.tableau.com/profile/naveen4955#!/

I. Introduction

This report will analyze survey data gathered from employees across IBM to gain a clearer image about the community within the company, while also striving to better understand and predict employee attrition. Using information from 1,470 employees, the following will first breakdown different groups of IBM workers. These groups will serve as the foundation for visualizing the company's workforce. Further, they will provide data for analyzing important company topics such as diversity and overall satisfaction. In the second half of the report, these groups will be used to develop a model that predicts attrition within IBM employees and can identify the most important contributing factors. Throughout the paper, this information will be used to draw conclusions on areas of future focus and improvement for HR.

II. Data Visualization

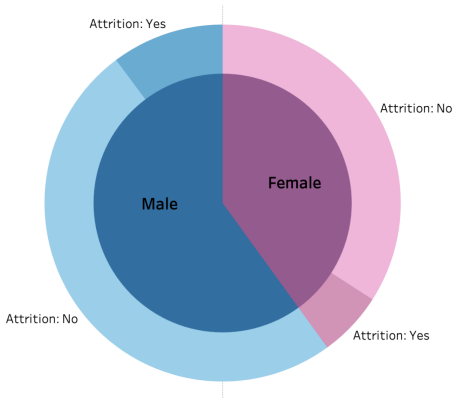
This section will analyze key demographics from the data to answer the question 'Who is the average IBM employee?' and perhaps just as importantly, 'Who isn't?'. I will focus on categories relating to workers' personal backgrounds in addition to their roles within the organization, looking to find any connections between the two as well. In order to visualize this data, I first created a short script in R to map the integer values to their corresponding String labels and then worked in Tableau to create Sunburst charts and segmented bar graphs for each attribute. For a majority of the graphs, I related the categories to attrition. This will allow us to visually identify any unusual trends that would impact the attrition-prediction model. This is by no means a proof of causality, but rather allows us to view the data through the lens of attrition and keep the goal of our model in mind when visualizing data.

Note: If you are inclined to go deeper, the workbooks are available for download on my Tableau Public account 'naveen4955' linked above; however, any relevant graphics have already been pasted throughout.

Gender

We will first analyze gender distribution. It is important that diversity thrive at IBM not only for the sake of equality but because a wider-range of people means more creative products, while avoiding the homogeneity that leads to outcomes like groupthink. Thus, in order for IBM to foster innovation, diversity metrics like gender should be extremely important. However, *Fig. 1* displays a different reality at the company, with just 40% of the sample as female, and although there doesn't appear to be a significant relation to attrition, there is no question that women are underrepresented at IBM. Looking a step deeper though, there is some promising gender-related data. While women as a whole are underrepresented, their proportional distribution across jobs and departments portrays a relatively fair community. As can be seen in *Fig. 2*, Women are spread amongst the three departments of Research and Development, HR, and Sales fairly proportional to men, with a small bias towards sales. When analyzing specific jobs in *Fig. 3*, the data become more even optimistic for women at IBM. Women are shown to be holding their own in leadership roles, comprising more than their overall 40:60 ratio in positions as Managers, Research Directors, and Sales Executives. One glaring point, however, is the absence of women as Lab Technicians, which should undoubtedly be a focus for HR moving forward. Thus, while women are in fewer overall numbers, they are proportionally equal.

Fig. 1: Gender Split w/ Attrition



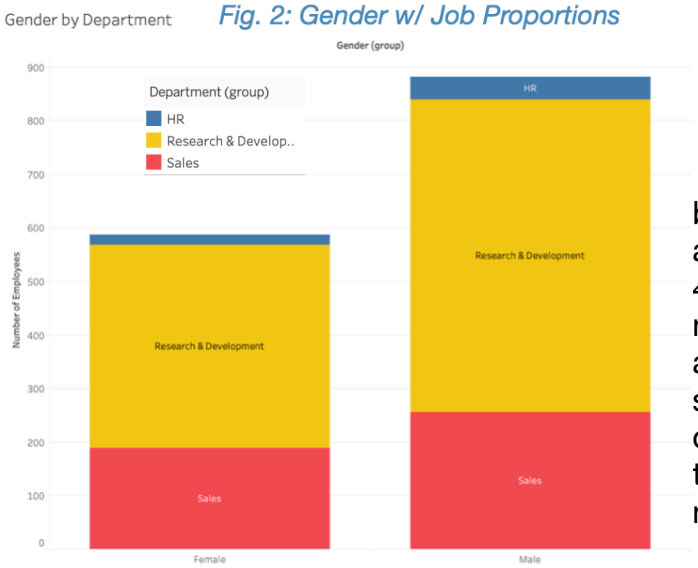


Fig. 2: Gender w/ Job Proportions

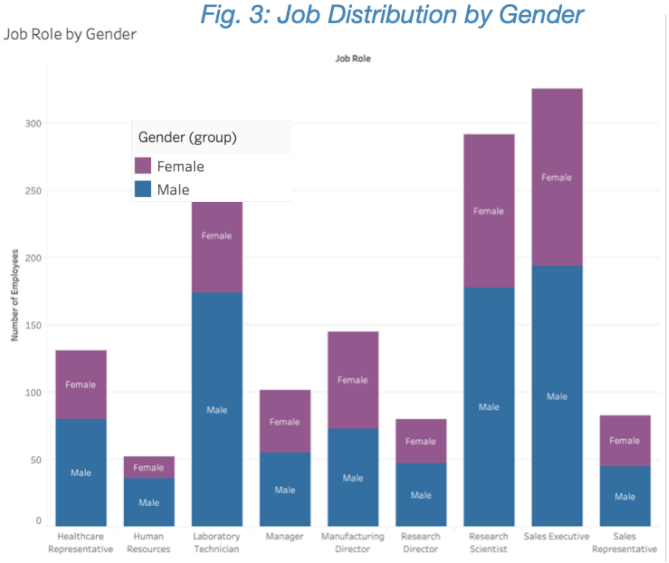


Fig. 3: Job Distribution by Gender

Job Satisfaction/Age

Job satisfaction is another key category, historically being tied to higher productivity, increased profits, better recruiting, and less attrition. Initially when looking at the data for Environment and Job Satisfaction (*Fig. 4)*, it is good to see that positive responses outweigh negative ones, but also interesting to note that, visually, attrition only slightly increases with decreased satisfaction (this will be contradicted later, however, by our Machine Learning model). Visually, this could point to attrition being driven by other factors, such as the notion that young people tend to change jobs quickly in technology-focused firms like IBM. The graph for total working years highlights this point, demonstrating that the highest levels of attrition are amongst people just entering the workforce (*Fig. 5).*

Honesty in Performance Ratings

A surprising result of my analysis found that across all 1,400+ employees, not a single person had a negative performance rating. Reasonably assuming that not all employees are 'excellent' or 'outstanding', the only explanation is a lack of open and honest evaluations. Feedback is the cornerstone of growth, and HR must do more to encourage this communication at IBM.

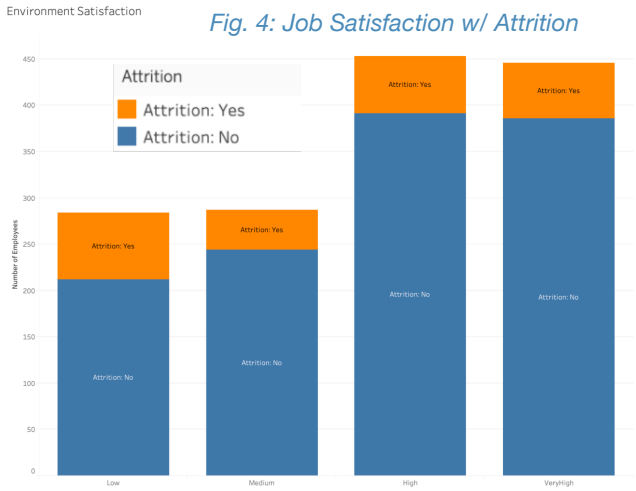


Fig. 4: Job Satisfaction w/ Attrition

Job Distribution

We can also look at the job distribution of workers at IBM and relate specific roles to satisfaction in *Fig. 6* on the next page. We can see there is a relatively high rate of dissatisfaction throughout all jobs, especially within Laboratory Technicians, Managers, HR, and Sales Reps. When comparing this to the proportion of attrition in each job (*Fig. 7*), there is a clear carry-over of attrition in Technician and Sales Rep positions.

Work-Life Balance and Relationship Satisfaction

When looking at satisfaction, work-life balance and relationships are crucially important fields. According to the data (*Fig. 8)*, work-life balance has lots of room for improvement at IBM. While very few marked 'Bad', a vast majority chose the second-worst option of 'Better', and it is no surprise that this has the highest amount of attrition-positive employees. On the other hand, relationship satisfaction tells a different story (*Fig. 9)*. Relationship satisfaction is stacked towards the positive responses, with not much variance in attrition.

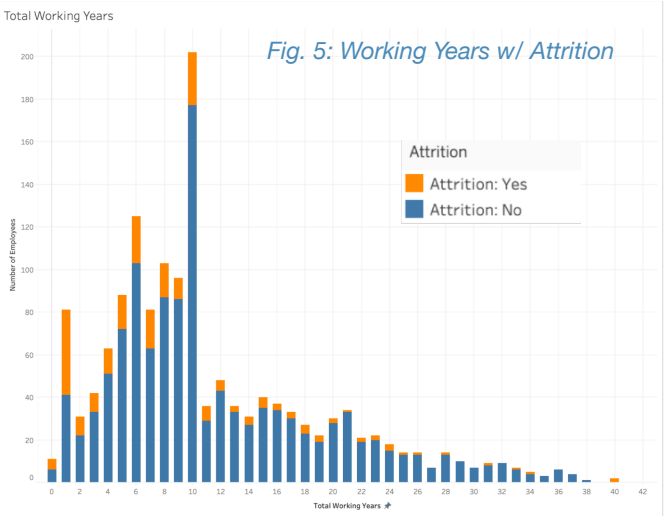


Fig. 5: Working Years w/ Attrition

Fig. 6: Job Roles w/ Satisfaction

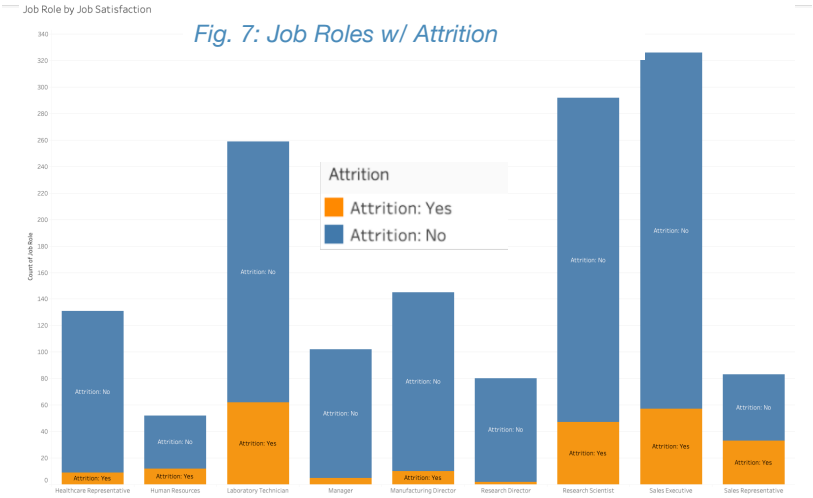


Fig. 7: Job Roles w/ Attrition

Note: These are just the most significant categories, you can see Tableau Public page for all categories

III. Attrition Prediction Model

Defining the Model

The survey data will now be used to create a model that can accurately predict employee attrition at IBM. To create such a program, we will work in R, employing various machine learning methods until arriving at a sufficiently accurate model. Before jumping into development, it is important to understand a few key traits about the data we are trying to predict. First, the majority of employees are attrition-negative and thus the results are largely skewed. In fact, if we were to simply assume that everyone is attrition-negative, we would already be correct 83 percent of the time! This is an important number to know, as when we begin to evaluate our models, anything with an accuracy less than this threshold must be ignored. Another key point is how we will define our model in Machine Learning terms and the corresponding evaluation metrics to choose.

Our program to differentiate between two attrition

Correlating the two categories' opposing results, HR's focus should be less on workers' communication (this is still very important) and more on how they can work with themselves to create manageable situations.

Data Visualization Conclusion

Based on each section of our visualizations, we can find insights and action points for HR.

Gender: Hire women in greater than 50:50 ratio, specifically in technician roles.

Job Satisfaction: Cater towards younger workers, those with lesser satisfaction

Performance Ratings: Encourage managers to be more honest with their workers, help make constructive criticism mainstream

Job Distribution: Honest discussions about dissatisfaction in Technician, Manager, HR, and Sales roles.

Work-Life/Relationships: Create a focus on balancing personal-life & demands of work

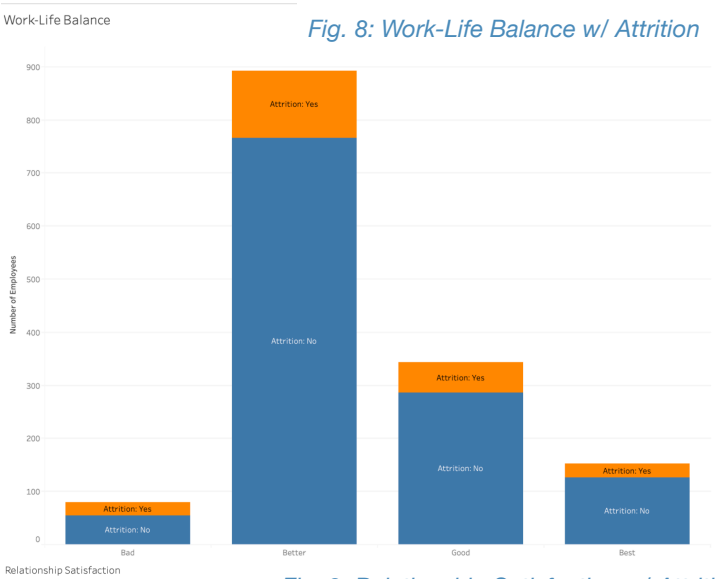


Fig. 8: Work-Life Balance w/ Attrition

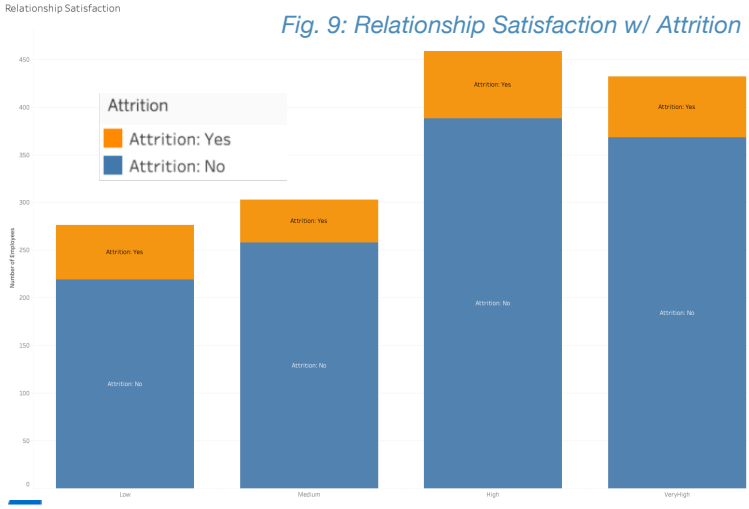


Fig. 9: Relationship Satisfaction w/ Attrition

outcomes must be a binary classifier system, which means that we will use a Receiver Operating Characteristic (ROC) curve and AUC score to track performance. The ROC plot is a probability graph showing the performance of a classification model at all classification thresholds, with AUC representing the area under this curve. Essentially, the "[ROC] tells how much a model is capable of distinguishing between binary classes…, higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s." Thus, AUC will be used to distinguish our different implementations, which we can then visualize through an ROC curve.

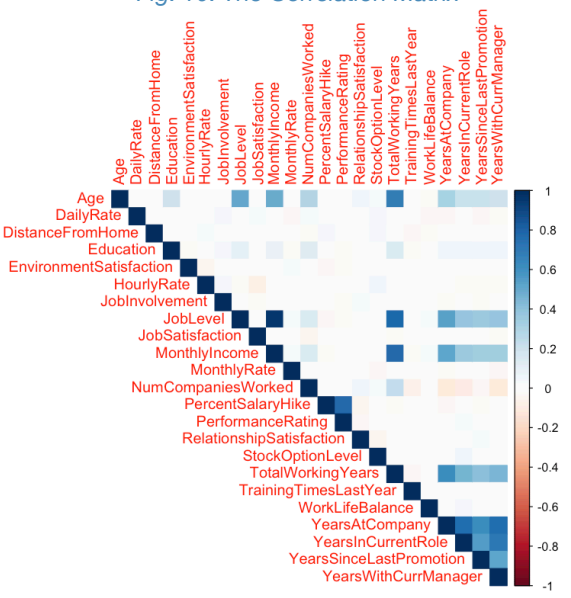
Cleaning the Data

The code for this sub-section can be found in the R_ModelData file in the Github

Prior to programming the data science models, we must first clean the data, as the saying goes, "garbage in, garbage out". Luckily the data we are using does not have any missing cells, so no values will have to be imputed or ignored. Nevertheless, there are a few columns containing arbitrary data that must be removed. First through inspection, the detail of 'Employee Number' can be removed from the dataset. There are no repeated employee entries to account for and the number assignment appears random, so 'Employee Number' is clearly an unnecessary variable. Next, we will remove any column with the same value in each row, as they are constant for everyone and will provide no new information. This leads to the removal of the 'Employee Count' attribute, as each entry is always for just one employee.

Now it's time to eliminate correlated data from the dataframe. Correlated data are variables that have a strong association with each other and will thus tend to follow related patterns (note: not causation). This is an important step in cleaning the data, as we do not want multiple independent variables conveying the same information to the model. Correlated data can make models unstable and introduce overfitting to the dataset by double-counting. However, we can only calculate correlation between numerical variables, so after removing categorical values, the plot in *Fig 10* can be constructed. An absolute value close to one indicates that the variables on each side are highly correlated. As shown, there is a diagonal of ones where each variable intersects with itself and a few correlations worth evaluating. Any correlation above 0.75 is considered high and makes it important to remove one of the two variables. After filtering, we will remove the 'Total Working Years', 'Years at Company', 'Job Level', and 'Percent Salary Hike' categories. Our last data-cleaning step will be to fix the aforementioned skew in our response variable 'Attrition'. The usual fix for skewed data is to take the logarithm of all values, but seeing as Attrition is a categorical variable, this won't work. Instead, we will have to use a program to create synthetic, realistic minority cases. Using the SMOTE library, we can balance the dataset by generating attrition-positive cases, allowing for better training and testing. Our data is now finally ready for the model!

Fig. 10: The Correlation Matrix



The other technical preparation steps for coding (such as creating dummies, merging, etc.) can be found in the Github file

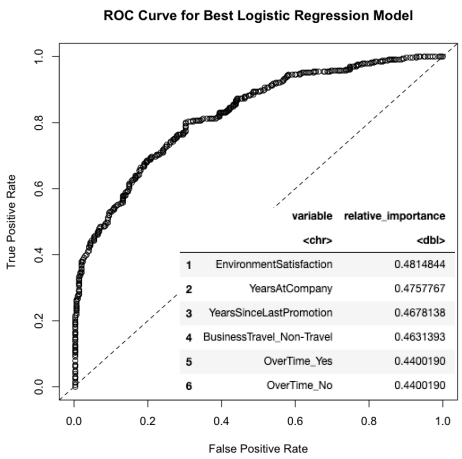
The Model

The code for this sub-section can be found in the R_Attrition_Model and Python_Attrition_Model file in the Github

After reading in the data frame and splitting into train, validation, and test sets, we will begin building our four model types in R: Logistic Regression, Random Forest, Gradient Boosting, and a Neural Network. I used the H2O library in R to construct each variation. For each type, we will conduct a hyperparameter optimization to improve network performance. Hyperparameters are all the small variables (iterations, batches, max time, etc.) that can

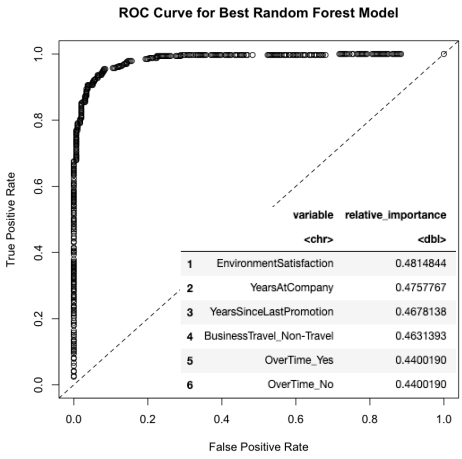
significantly change the performance of a model. To find the best combination, different models will be tried against the validation set to identify the top performer. The best of each method will have its ROC plotted alongside its calculations for AUC and the most important variables. We will choose which model to use against the test set from the validation set.

Logistic Regression (AUC: 0.8864)



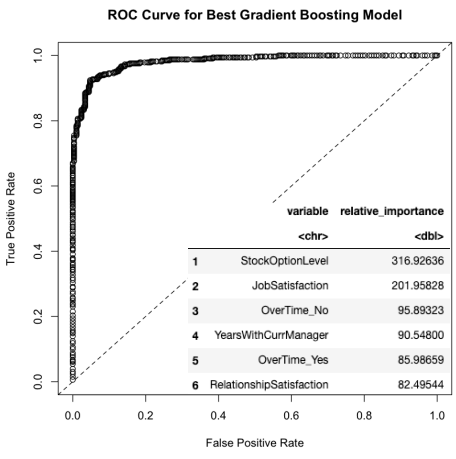
	variable	relative_importance
	<chr>	<dbl>
1	EnvironmentSatisfaction	0.4814844
2	YearsAtCompany	0.4757767
3	YearsSinceLastPromotion	0.4678138
4	BusinessTravel_Non-Travel	0.4631393
5	OverTime_Yes	0.4400190
6	OverTime_No	0.4400190

Random Forest (AUC: 0.9839)



	variable	relative_importance
	<chr>	<dbl>
1	EnvironmentSatisfaction	0.4814844
2	YearsAtCompany	0.4757767
3	YearsSinceLastPromotion	0.4678138
4	BusinessTravel_Non-Travel	0.4631393
5	OverTime_Yes	0.4400190
6	OverTime_No	0.4400190

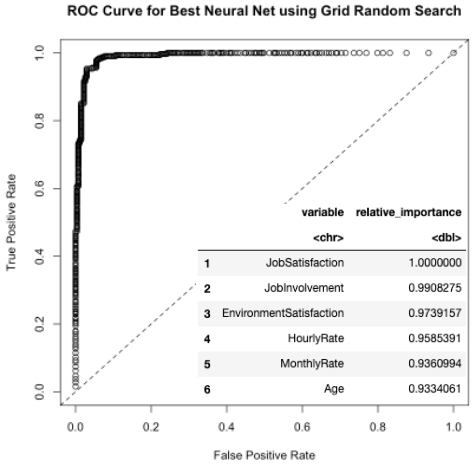
Gradient Boosting (AUC: 0.9812)



	variable	relative_importance
	<chr>	<dbl>
1	StockOptionLevel	316.92636
2	JobSatisfaction	201.95828
3	OverTime_No	95.89323
4	YearsWithCurrManager	90.54800
5	OverTime_Yes	85.98659
6	RelationshipSatisfaction	82.49544

We can now compare each of the models. The logistic regression, which fits a polynomial function to the data, works well, but not as efficiently or accurately as the Random Forest model which employs randomized initial points to build an 'ensemble' of decision trees. Comparing the RF to the Gradient Boosting Model, both have similar variable importances, but the GBM's use of gradient descent for optimization …

Neural Network (AUC: 0.9950)



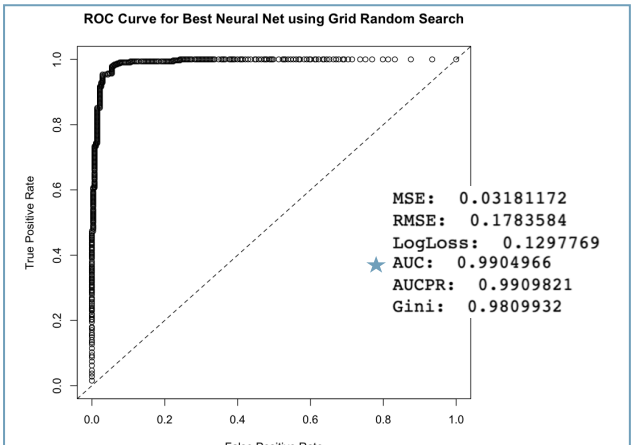
	variable	relative_importance
	<chr>	<dbl>
1	JobSatisfaction	1.0000000
2	JobInvolvement	0.9908275
3	EnvironmentSatisfaction	0.9739157
4	HourlyRate	0.9585391
5	MonthlyRate	0.9360994
6	Age	0.9334061

… appears to be slightly less powerful that the Random Forest method. Lastly we implemented what is usually one of the best classifiers: a neural network! This type of model uses activation layers, connectivity weights, and hidden layers to feed the parameters through a network. Once tuned, the network is extremely accurate in identifying attrition, so we choose our neural net to analyze the test

Applying the neural net to the test set did not disappoint. As can be seen below, our model finished with an AUC of 0.9905! Its other performance metrics are equally as promising, achieving a mean squared error of just 0.0318, max F1 score of 0.967, max F2 score of 0.980, and a maximum accuracy score of 0.963!

Attrition Model Conclusion

Having developed a powerful classification model, there are two key points of focus for HR moving forward. First, HR should employ this neural network (or something similar) in order to anticipate and prevent employee attrition with high accuracy. Next, we can see in the above graphic that the most important factors in attrition are Job Satisfaction, Job Involvement, Environment Satisfaction, Pay Rate, and Age. Some traits are hard to solve for, such as the fact that young employees naturally tend towards attrition. However others, such as satisfaction, involvement, and pay scale, are areas that can be addressed through clear and open discussion. Going forward, HR should monitor these 4 variables with weekly surveys and conduct check-ins focused on these topics with employees.



Maximum Metrics: Maximum metrics at their respective thresholds

	metric	threshold	value	idx
1	max f1	0.521855	0.966565	257
2	max f2	0.344085	0.979792	263
3	max f0point5	0.854202	0.970384	240
4	max accuracy	0.613159	0.963087	253
5	max precision	0.999998	1.000000	0