

Predicting for Improvement: Using Machine Learning Models to Improve Financial Decision-Making and Employee Retention

Matthew Sandell

Data Analytics Senior Capstone Winter Quarter, Eastern Washington University

DSCI 483: ML-Applied Data Science

Dr. Alexandros Paparas

March 17, 2025

Introduction and Background

Efficiency in today's highly developed economy is more important than ever. Losing a step against competitors can make or break a company. While making sure that sales, marketing, and logistics are functioning smoothly is extremely important to the company's wellbeing, there are other aspects that are just as important. The key part of any business is its workforce. The importance of a competent and efficient employee base can't be understated. The current work environment is incredibly competitive, meaning finding and keeping skilled employees should be a top priority for any firm. Employee retention is an easily measured metric that can help guide internal decision making in terms of human resources and employee relations. It is also important to be aware of income distributions through various departments in the firm. Companies are always planning for the future, including growth. The budgeting process for both current and future expenses can be more accurately achieved through clear knowledge of factors predicting monthly income and employee retention, as well as the number itself.

Both metrics, employee retention and income, can be leveraged from raw data into actionable insights that allow companies to remain competitive in their industry and improve internal relations.

To illustrate these important concepts, we conducted research relating to these variables from a provided dummy dataset. The dataset contains 32 variables covering over 1400 employee entries. The variables relate to employee role, satisfaction, and numerous other topics. The variables are a mix of continuous (measured, such as Monthly Income and Age) and categorical (such as Job Satisfaction or Department). Some variables, such as Employee Number, are irrelevant and not used in this research.

The dataset came cleaned and with no missing values. A recurring step in the research is converting data types of certain variables to work in given models and splitting the data into testing and training subsets. Otherwise, the data was already prepared for analysis. The research questions approached in this project are as follows:

- What predicts Monthly Income?
- What predicts decision to leave the company?

To answer these questions, we utilize machine learning (ML) methods and models. In the context of this research, ML is the process of creating an algorithm to perform complicated statistics from a sample dataset. The hallmark of this process is the ability to test the trained model on observed data. The output goal is to maximize predictive accuracy of each dependent variable. This is achieved by using different methods of model creation and choosing the best one based on standardized metrics; because machine learning is sensitive to the type of data being analyzed, different models must be used for each research question.

Dependent Variable	Data Type	Method of Evaluation	Models Used
Monthly Income	Continuous	Root Mean Square Error (RMSE)	Linear Regression Regression Tree Bootstrap Aggregating Random Forests Gradient Boosting Machine
Leaving the Company	Binary	Percentage of outcomes correctly predicted (confusion matrix)	Logistic Regression Naïve Bayes Discriminant Analysis (Linear and Quadratic) Classification Tree Bootstrap Aggregating Random Forests Gradient Boosting Machine

Continuous Model Selection

Beginning with the continuous variable Monthly Income, we used five ML methods. Linear regression is a classic method for examining relationships between dependent variables and multiple independent variables. The assumptions are not too restrictive and can be easily checked. Regression Trees are used in case the modeled relationships are not strictly linear. This method utilizes decision trees split into subsets of data. Bootstrap aggregating (Bagging) is a versatile method because it trains numerous models from the data and averages the results, creating more stable predictions. Random forests are an ensemble method similar to bagging, with added diversification of the trees due to randomly selected subsets of the dataset used for training. Gradient boosting machine (GBM) is a unique ML algorithm that builds sequential models to reduce previous model error. Each one of these methods yields a similar but different result, so it is up to us to run all appropriate models and packages and pick the one that works best in the context of the data and research questions.

To create the multiple linear regression model, first we created a “full” model, containing all the variables in the dataset with unique values. We then dropped statistically non-significant variables to create a leaner model. The dropped variables were not significant at an alpha level of 0.05, meaning they don’t have a strong impact on the results of the model. Doing this allows us to simplify the model without sacrificing large amounts of predictive accuracy. Additionally, we created a third model that uses an interaction term. An interaction term is created by combining the effect of two or more thematically similar independent variables. In this case the interaction term combined Job Level and Job Role.

A few steps are undergone to find the best model. The linear model that performs best in predicting the dependent variable Monthly Income has the following characteristics:

- Higher R-Squared
- Lower sigma
- Has a significant p-value
- Higher logLik
- Lower AIC
- Lower BIC
- Lowest Root Mean Square Error (RMSE)

The interaction term model performs best in all these metrics, indicating it is the best linear model. The four key assumptions of linear regression – linearity, independence, normality, and equal variances are all satisfied.

For the regression tree, the algorithm splits the data into tree nodes based on minimizing the Mean Squared Error (MSE). This process repeats, with the tree branching out recursively until a stopping criterion is met. This method tends to overfit - that is, to create too many nodes at the end that contain too few observations in each. To fix this issue, we used a grid search to find the optimal hyperparameter settings. This process finds the optimal minsplit, maxsplit, and cp values. Minsplit is the minimum number of observations in a node before it can be split further. Maxsplit limits the number of splits the tree can have. The cp value trims non-significant nodes in the model. These settings all work to correct for possible overfit in the model. We created numerous regression tree models with various tuning values, which we compared to the optimal hyperparameter model. As expected, this model performed the best in terms of having the lowest RMSE. This indicates it is the best regression tree model.

Various bagging models were also created. Bootstrap aggregating is the process of creating a given number of random single complete regression trees. This method helps control for variance by training models from different portions of the dataset and averaging them together. The results are therefore generally more stable and reliable. Two separate bagging packages in R were used, ipred and caret. Ipred uses an optimal number of complete regression trees, determined by minimization of prediction error and highest level of model stability. The caret package functions similarly but uses cross-validation to divide the dataset into ten equal parts. Nine of them are used for training the data while the tenth is the testing dataset. This process is repeated until each subset has been used for testing. Like ipred, this optimizes variance and model stability. The RMSE of the ipred and caret models are compared, with the lower value from ipred indicating better model fit.

To create the random forest models, three separate packages were used – RandomForest, Ranger, and H2O. The process for each is the same. A default model is created to get baseline values for model accuracy and hyperparameters. Then a tuning grid search is conducted; different aspects of the model are tweaked, such as the number of trees and sample size of the dataset. By running hundreds of combinations of hyperparameters we find the best model for the chosen metric, in this case RMSE. Once the grid search is done, we sorted the models to find the

lowest RMSE and ran a new model with these hyperparameters. After that we run predictions using this model. The Ranger package worked the best in terms of RMSE.

The final ML method used for the continuous variable is gradient boosting machines (GBM). Numerous packages are available in R, but we chose XGBoost due to its mix of computational speed and accuracy. Categorical variables are converted into binary columns – called one-hot encoding – and a baseline model is made. Initial models are adequate, but we rely on tuning to improve predictive results. GBM builds sequential poor models that progressively fix issues with the previous iterations. By varying the learning rate, tree depth, and other metrics of the model we can improve accuracy. After the grid search gives us the best hyperparameters, we run this model and run predictions. Due to its iterative nature, it generally performs quite well, as we see below.

Now, each method – linear regression, regression tree, bagging, random forests, and gradient boosting machines – has been used to create models to predict Monthly Income. We took the model from each method that had the best (lowest) RMSE and compared them to see which continuous model overall performed the best in predicting the dependent variable. The table below shows the results.

Method/model type	RMSE
Linear Regression – with interaction term	1,014
Regression tree – with optimized hyperparameters	1,176.07
Bagging – using ipred package	1,111.59
Random Forest – using Ranger package	1,077.49
Gradient Boosting Machine – using XGBoost	989.36

Root Mean Square Error (RMSE) is used here for comparison because it is a standardized measure of model fit for continuous variables. The differences between each predicted and actual value, called residuals, are squared to remove negative numbers and give more weight to large deviations; then these deviations are averaged, forming Mean Square Error (MSE). The square root, which converts the value from MSE to RMSE, is used simply to reduce the size of the numbers, as each MSE here is somewhere near a million. Because this measures the deviation of predicted values from actual values, it is intuitive why a lower score is better – this indicates less variance from observed reality and therefore a better model fit. The table above highlights that the gradient boosting machine model using the XGBoost package performed the best.

Gradient Boosting Machine Model Interpretation

A variable importance plot of the final model indicates that Job Level is the most important variable in predicting Monthly Income. Having the Job Role of Manager or Research Director, and Total Working Years are the next most influential. These results are largely intuitive: as an employee moves upward in the company to higher positions, income would also increase. Additionally, Managers and Research Directors are generally higher-level positions and would see better base pay compared to other roles. Total Working Years is another indicator of an employee's age/experience, which in general should correspond to higher positions in the company.

Partial dependence plots (PDPs) were also used to model the relationships between dependent and independent variables. The Job Level PDP indicates an approximate \$4,000 increase in Monthly Income per unitary Job Level increase. The PDPs for Managers and Research Directors each show roughly a \$1,000 increase in Monthly Income for employees with those roles compared to others. The plot of Total Working Years indicated that as employees move through their first twenty years, they can expect to increase their Monthly Income by about \$800 compared to less experienced employees. After the first decade and a half, this increase largely plateaus. See Appendix for variable important plot and PDPs of these variables.

Binary Model Selection

Moving on to the binary research question, we used a multitude of machine learning models to answer the question of what determines the decision to leave the company. The methods were all chosen due to their robust and varied approaches to predicting binary outcomes. Logistic regression is a simple and easily interpretable model that follows a similar process to the linear model used earlier. Naïve Bayes is a robust probabilistic classifier. Based on the classic Bayes' Theorem, it is a fast model that handles categorical and text variables well. Discriminant Analysis is used here in two ways – Linear and Quadratic (LDA and QDA, respectively). Working for both equal and unequal covariances among classes, these models are adaptable at addressing relationships that are either linear or non-linear. Classification Trees function in the same fashion as the regression tree process used earlier, altered only to address a binary outcome instead of continuous. The bagging process is also almost identical to the one used earlier. We also used random forests and GBM for the binary outcome. These processes are very similar to the models made for the continuous variable, with minor changes. Accurately predicting Yes values, employees who leave the company, is more important than predicting those that don't. This is because employee turnover is costly and disrupts stability, where employees that remain at the company do not drain resources in the same manner.

We created three logistic regression models to predict the dependent variable. The full model, with all appropriate variables, had extreme multicollinearity, meaning high correlations

between variables are present in the model. To fix this, we removed this model from consideration and created an improved version that dropped the variables causing the multicollinearity. We also ran a leaner model, with statistically non-significant variables dropped. Tests of the residual distributions and influential observations indicated both models were valid. To find the model that performed better, we compared predictive results via a confusion matrix, a table containing the total predictions against observed outcomes. Tuning was done by adjusting the threshold value of Yes and No. Lowering the threshold value means more predictions are considered Yes. This introduces an inherent trade off - by increasing the predictive accuracy of true Yes values, called sensitivity, we reduce the accuracy of true No values, called specificity. The goal is to measure the gain in sensitivity with the loss in specificity to create the most balanced model possible, given sensitivity is more important to us for this question. Each model performs similarly, but the model without non-significant variables performed slightly better in terms of confusion matrix results and McFadden pseudo r squared.

Naïve Bayes is another effective method. This model is a simple probabilistic classifier, with the ability to function well even when core assumptions, such as independence and normality as in this dataset, are violated. The caret package, used previously for bagging the continuous variable, is used here as well. This includes cross validation, creating subsets of the data and training models from them. The initial Naïve Bayes performs well overall, with over 80 percent accuracy, but correctly predicts those that leave the company only about half the time.

To improve sensitivity, we tune the hyperparameters by altering the initial probabilities. The optimal model weighs correct values at 0.74 and no values at 0.26. As a result of this, total accuracy of the model has gone down about 10 percent. However, sensitivity has improved from about 54 percent to about 74 percent. This optimized model better suits our goal of correctly predicting those that leave.

Two discriminant analysis models are used. Linear discriminant analysis (LDA) is used when assuming constant covariance among variables, i.e. a linear dataset. Quadratic discriminant analysis (QDA) works better when we are uncertain about constant covariance and can be used to model non-linear relationships. After the models are fit to the data, we adjust the posterior threshold value to increase sensitivity, based on a confusion matrix. The results indicate LDA is better suited to this data than QDA. This conclusion is confirmed by two graphs - Receiver Operating Characteristic (ROC) and Area Under Curve (AUC), which illustrate performance of the models as the threshold value changes and measures the ability to distinguish between classes respectively.

Like the regression tree process used earlier, we also created classification trees and bootstrap aggregating. Several models were made to provide baseline confusion matrix results before an optimized model was used. As in the regression tree, minsplit, maxdepth, and cp are tuned to give the most accurate and stable model. Caret, oversampling, and a loss function are bagging methods employed. Tuning of the threshold value was used for each to find the most

accurate model. The best bagging model and the optimized classification tree are shown in the results table later.

We also leveraged random forests for the binary outcome. While this ML method is very similar to bagging, random forests are unique in that they leverage subsets of the dataset features -that is, instead of all variables in the model, random forest models pick only some of them. This reduces correlation between trees and generally leads to a more stable model. The same three packages were used here as with the continuous variable. Each package follows a similar process but predicts slightly differently. We created a baseline model and then ran a grid search to optimize the hyperparameters for each. The ranger package performed the best in terms of balancing higher sensitivity with specificity.

As a robust model creation method, we were also able to use gradient boosting machines to predict employees leaving the company. An extensive grid search found the optimal model, which we used to generate a confusion matrix of prediction results.

Model Method	Sensitivity	Specificity	Overall Accuracy
Logistic Regression	76.56 %	87.65 %	85.89 %
Naïve Bayes	74.29	69.83	72.06
Linear Discriminant Analysis	78.22	77.38	77.53
Classification Tree	60.81	78.24	75.12
Bagging	72.97	75.29	74.88
Random Forest – using Ranger package	52.70	90.59	83.82
Gradient Boosting Machine using XGBoost	88.89	63.56	67.29

As seen above, with our goal of maximizing sensitivity, correctly predicting employees who leave the company, there are multiple models we can choose from that perform well. Logistic regression, GBM, and LDA all perform highly in sensitivity. If the goal is simply to maximize sensitivity, GBM is the best model. However, GBM performs poorly in correctly predicting No values. Therefore, we chose to use the logistic regression model. This is because the model performs very well not only in sensitivity but also specificity. We decided it is not worth losing such a large degree of specificity for the increase in sensitivity. Similar logic applies to the LDA model. While it technically outperforms the logistic model in sensitivity, it does so at a high cost of specificity. Because the logistic model is so balanced that it performs well at predicting both outcomes, we choose that as the overall model. It is important to note that given a situation in which more emphasis is placed on sensitivity, GBM would be a great option.

Assumptions and Interpretation of Logistic Regression Model

As with conducting linear regression, some assumptions must be met to confirm the validity of the model. First, we can assume independence of the observations because we know from the original dataset that each observation corresponds to a unique employee; further, the data is not time series, so we can satisfy the assumption of independence. Second, the variable Years At Company possesses the highest VIF value in the model at approximately 4.74. This is well within the acceptable range of values, meaning we can assume the lack of multicollinearity. Once the assumptions have been satisfied, we examine the variables in the model. The coefficients are not interpreted as in linear regression, but we can see which variables are the most influential in the model.

Variable	Variable Importance Score
OverTime: Yes	9.36
Marital Status: Single	5.02
Environment Satisfaction	4.37
Years Since Last Promotion	4.20

* Variance Importance Scores rounded for simplicity. For a full list of the model scores, see Appendix.

The table above leads us to some interesting insights. First, the status of being an overtime employee greatly impacts employee retention. This makes sense, as employees that feel overworked and drained are less likely to stay with the company. This issue can be rectified by potentially reducing overtime hours for these employees and redistributing the lost efficiency across other employees and/or departments. Employees that aren't in relationships leaving makes some sense as well – employees that have families are usually primary financial providers and therefore can't afford to quit a job as often as someone with relatively few ties, such as a single employee. This can't really be addressed by the company. Environment satisfaction and years since last promotion are both interesting variables. Employees who see less upward mobility – evidenced by lack of promotion – would naturally be less-inclined to stay with the company. Further, if an employee has low environment satisfaction, this clearly represents a potential liability of the employee leaving. Potential solutions could include a focus on quality-of-life adjustments for employees, as well as reevaluating the promotion process to improve the path of upward mobility.

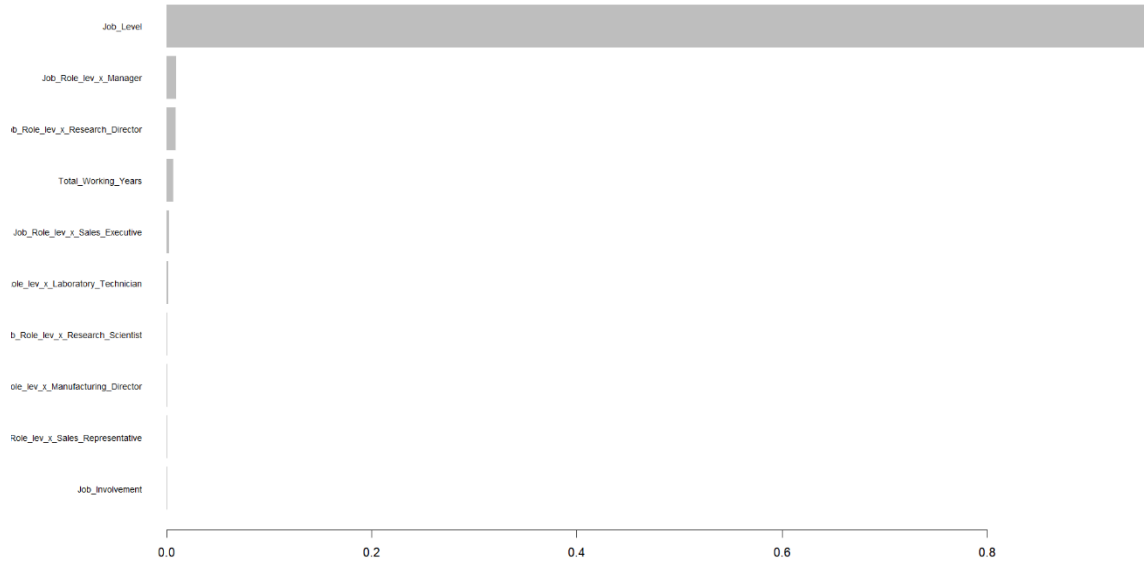
Conclusion

Machine learning processes applied to both research questions yielded insightful results that teach us more about the nature of employee behavior. These insights can aid the company in improving its efficiency. As seen in the tables, these models were chosen for their balanced performance in standardized metrics. Assumptions were addressed and the validity of the models confirmed. The GBM model showed us that certain job roles such as manager or research director are highly significant factors predicting income, with indicators of upward mobility,

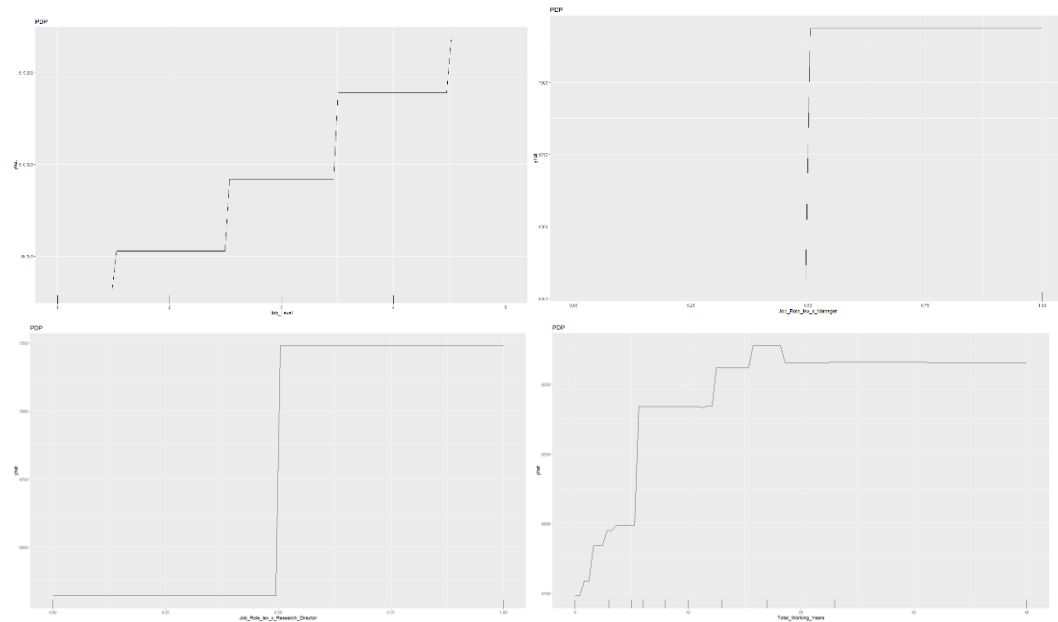
working years and job level, also showing a clear positive relationship with income. The logistic model helped us learn more about employee retention factors. The presence of overtime is by far the most influential variable, while environment satisfaction levels and time since the last promotion are also strong predictors. These can all be reviewed and adjusted by the company to decrease the chances of given employees leaving. For the gradient boosting model, we faced a mild constraint in that finding the optimal model via hyperparameter grid search is computationally intensive. Therefore, optimizing this model would take a large amount of computing power and time, which is beyond the scope of this project.

Appendix

Variable importance graph for GBM final model



Partial Dependency Plots (PDPs) of the most influential variables in the GBM model:



Full list of logistic regression variables with importance scores:

	overall
Age	3.893670132
BusinessTravelTravel_Frequently	3.500731926
BusinessTravelTravel_Rarely	2.474973646
Distance_From_Home	2.817452924
Environment_Satisfaction	4.368054093
Job_Involvement	3.281105548
Job_RoleHuman Resources	3.350319522
Job_RoleLaboratory Technician	3.348713994
Job_RoleManager	0.011118491
Job_RoleManufacturing Director	0.004346683
Job_RoleResearch Director	1.084615969
Job_RoleResearch Scientist	1.445126305
Job_RoleSales Executive	1.970404946
Job_RoleSales Representative	3.932460150
Job_Satisfaction	3.550966833
Marital_StatusMarried	2.296406362
Marital_StatusSingle	5.023054787
Number_of_Companies_worked	3.715982970
OverTimeYes	9.358011514
Relationship_Satisfaction	1.831173092
Training_Times_Last_Year	2.225666039
Years_At_Company	2.347627821
Years_In_Current_Role	2.517289161
Years_Since_Last_Promotion	4.195915605
Years_With_Current_Manager	3.452349089