Building a Better Attrition Model

Vemo

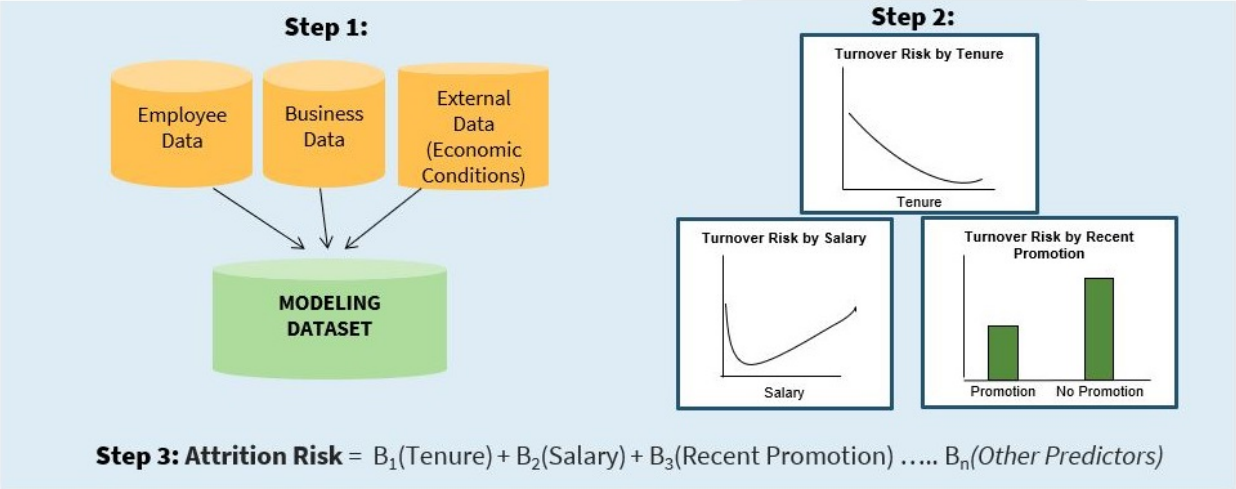
April 11, 2023

Introduction

Employee Attrition is among the most popular applications of predictive modeling, and is often employed as a standard case example in introductory courses. This can be effective as a teaching tool to help acquaint first-time learners with the basics, but the cookie-cutter approaches taught in introductory courses should not be mistaken for the real thing. When building an Employee Attrition model in the field, careful attention must be paid to specifics of the problem at hand, otherwise the model can fail to properly address the business need.

Here is an overview of the process of building an attrition model:

Step 1: Compile modeling dataset using internal and external data sources
Step 2: Use data mining techniques to find patterns in data
Step 3: Build a mathematical function to predict turnover



Step 3 produces an equation that can be used to predict an individual employee's turnover risk over some upcoming time period – often one year – based on their relevant characteristics. Here's a simplified example of a turnover equation, using just two predictors, in it's general form:

$$TurnoverRisk = .2 - .01(Tenure) - .002(Age)$$

The equation states that an employee's turnover risk is a function of their age and tenure. An employee has a baseline risk of .2 or 20%, which is reduced by .01 or 1 percentage point for each year they have been with

Table 1: Turnover Risk By Employee

EmployeeID	Tenure	Age	Job.Type	Location	TurnoverRisk
1	4	30	A	North	5.0%
2	8	45	A	North	2.5%
3	15	56	B	Central	1.0%
4	9	35	A	South	3.0%
5	3	40	C	South	5.0%
6	3	41	C	North	3.5%
7	10	50	B	Central	3.0%
8	4	36	D	South	5.0%
9	1	25	A	North	7.0%
10	1	24	A	West	10.0%

the company, and then reduced by another .005 or 0.5 percentage point for each year of age. This means that in general, older employees with more tenure are less likely to turnover than young employees with low tenure.

If **Employee X** is 50 years old with 7 years of tenure, their risk of leaving within the next year is 3%:

$$TurnoverRisk = .2 - .01(7) - .002(50) = .03 = 3\%$$

This equation is useful for **Risk Segmentation**: it can identify which employees are at the greatest risk of turning over and which employees are at lowest risk. In the case of our two-factor model, it may seem quite intuitive that older, higher-tenure employees have lower turnover risk, but modern modeling techniques allow us to segment risk based on a much wider range of predictors, some of which are less intuitive and require data-mining to uncover.

Supposed we have a four factor model. Table 1 (above) shows how such a model might assign risk to a sample of 10 employees.

Note the extra segmentation afforded by including the Job Type and Location predictors. Employees 5 and 6 have nearly identical age and tenure profiles and work the same job, but employee 5 is higher risk. This is because employee 5 works in the South location, which the model has determined is higher risk. Beyond **Risk Segmentation**, organizations also use turnover models for **Forecasting** turnover rates across larger groups of employees. This can be done by averaging the model-assigned turnover risk across the group of interest. For the sample of ten employees above, the forecast turnover rate would be 4.5%. For the subset of employees working Job A, the forecast rate is 5.5%, while for Job B, it's just 2%. These forecast rates can be very helpful in workforce planning, as they allow organizations to reliably anticipate the number of vacancies that will be caused by attrition. A final use for turnover modeling that will be discussed in this presentation is **Policy Evaluation**, whereby the insights gleaned with respect to each predictor are used to craft interventions to reduce attrition. The goal of this presentation is to show how a predictive model optimized for one application, such as **Risk Segmentation**, will not necessarily serve other applications reliably. Instead, a model needs to be explicitly optimized and tested for each of its intended uses. I will start by reviewing the basics of Employee Attrition modeling that are typically taught at the introductory level, with a focus on **Risk Segmentation**. I will then show how this foundational approach must be adapted for other applications, such as **Forecasting** and **Policy Evaluation**. I will show some code along the way, using the R programming language.

Data and Drivers Selection

This analysis uses Employee Snapshots data from 2020-2021 for a company of ~4,000 employees. For the sake of this notebook presentation, we are going to skip over the data wrangling part of the project, and

assume that all of our data sources have been processed and combined into a single table. Here is a look at the data:

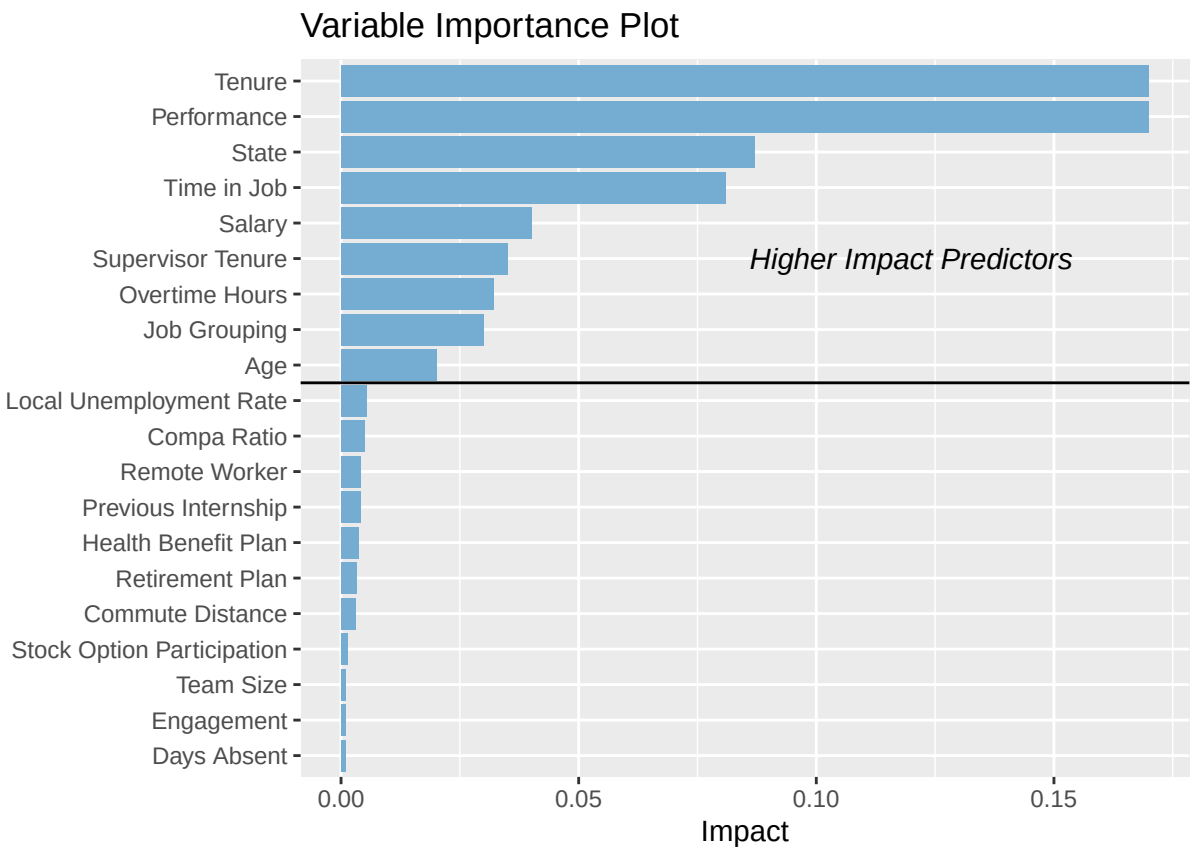
```
##    EmployeeID        SnapshotDate Tenure JobTenure Age State .. Turnover
## 1           1 Jul 2020 - Dec 2020   10.0       5.5  58    NY  ...        0
## 2           1 Jan 2021 - Jun 2021   10.5       6.0  58    NY  ...        0
## 3           1 Jul 2021 - Dec 2021   11.0       6.5  59    NY  ...        0
## 4           2 Jul 2020 - Dec 2020    5.5       1.0  39    TX  ...        0
## 5           2 Jan 2021 - Jun 2021    6.0       1.5  40    TX  ...        0
## 6           2 Jul 2021 - Dec 2021    6.5       2.0  40    TX  ...        0
## 7           3 Jul 2020 - Dec 2020   24.5      20.5  61    FL  ...        0
## 8           3 Jan 2021 - Jun 2021   25.0      21.0  62    FL  ...        1
## 9           4 Jul 2020 - Dec 2020   10.0       2.5  57    TX  ...        0
## 10          4 Jan 2021 - Jun 2021   10.5       3.0  57    TX  ...        0
```

Each row represents a six-month period of history for a given employee. A sample of available columns are shown in the above output; each column value indicates the value for that employee at the beginning of the six-month period. The Turnover indicator is 1 if the individual left during the period and 0 otherwise.

Each column may be considered as a potential driver of turnover behavior. Building an attrition model requires the data scientist to narrow down the field of potential drivers into a short list of impactful drivers that can be used to reliably predict turnover behavior.

One strategy for narrowing down the list of potential predictors is known as the *Variable Importance Plot*. The methodology used to predict this plot is beyond the scope of this presentation, but you can read more about it by exploring the randomForests package in R.

For now, assume that we've generated the variable importance plot, and that it looks like this:



The measure on the X-axis is a gauge of the potential usefulness of the predictor. There is a clear drop-off in impact between Age and Local Unemployment Rate, so for the purpose of this example, we will consider the predictors above the line as the short list of useful predictors.

Build Attrition Model

In this example, we start with a logit model that captures impact of each driver using a single linear term. Here is the model specification and the results:

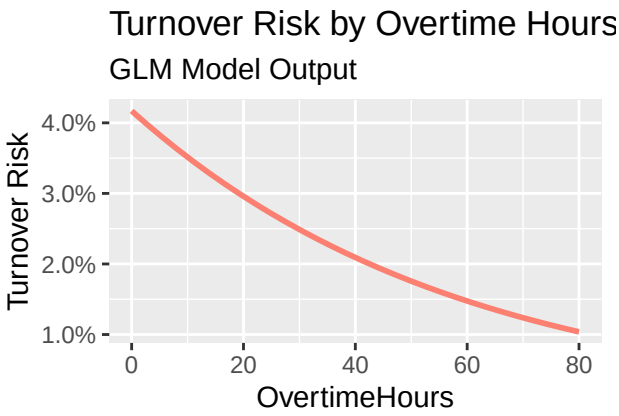
```
glmModel <- glm(Turnover ~
                  Tenure+
                  PERFORMANCE_DESCR+
                  JobTenure+
                  Age+
                  Salary+
                  OvertimeHours+
                  State+
                  JobGrouping,
                family=binomial, data=filter(data,test_train=='train'))
summary(glmModel)
```

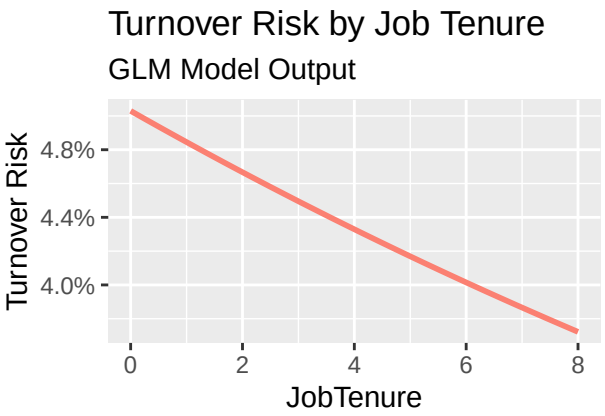
```
##
## Call:
## glm(formula = Turnover ~ Tenure + PERFORMANCE_DESCR + JobTenure +
##     Age + Salary + OvertimeHours + State + JobGrouping, family = binomial,
##     data = filter(data, test_train == "train"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4382  -0.1187  -0.0607  -0.0319   3.8596
##
## Coefficients:
##                                              Estimate    Std. Error z value
## (Intercept)                                -5.3176489649  0.3778310156 -14.074
## Tenure                                     -0.0998465245  0.0117780035  -8.477
## PERFORMANCE_DESCRExceptional               -0.2265991762  1.0525215500  -0.215
## PERFORMANCE_DESCRImprovement Required       4.2153794570  0.5813261108   7.251
## PERFORMANCE_DESCRMeets Expectations        -0.3318604209  0.3793116552  -0.875
## PERFORMANCE_DESCRMeets Some Expectations    1.3978459183  0.5035422010   2.776
## PERFORMANCE_DESCRNew in Position           -0.9295611856  1.0519843738  -0.884
## PERFORMANCE_DESCRNot Assigned               4.1526349219  0.3260144269  12.738
## JobTenure                                  -0.0393384120  0.0236872573  -1.661
## Age                                        -0.0020204490  0.0054384971  -0.372
## Salary                                      0.0000023598  0.0000007102   3.323
## OvertimeHours                              -0.0177819890  0.0019283351  -9.221
## StateFL                                    -0.4719049292  0.3779827440  -1.248
## StateNY                                    -0.1171711823  0.2383158608  -0.492
## StateTX                                     1.3155773548  0.1668448856   7.885
## JobGroupingJob B                            1.4994693192  0.2117935674   7.080
## JobGroupingJob C                           -0.4832856773  0.1753191385  -2.757
## JobGroupingJob D                            1.7070960733  0.3151937277   5.416
##                                                        Pr(>|z|)
```

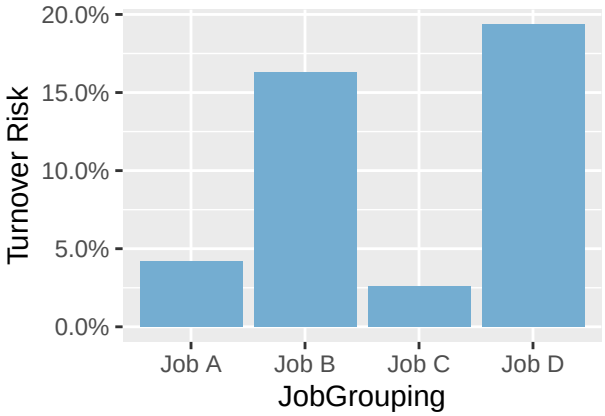
```
## (Intercept)                                < 0.0000000000000002 ***
## Tenure                                     < 0.0000000000000002 ***
## PERFORMANCE_DESCRExceptional                             0.829540
## PERFORMANCE_DESCRImprovement Required     0.00000000000041274 ***
## PERFORMANCE_DESCRMeets Expectations                      0.381627
## PERFORMANCE_DESCRMeets Some Expectations                 0.005503 **
## PERFORMANCE_DESCRNew in Position                         0.376898
## PERFORMANCE_DESCRNot Assigned              < 0.0000000000000002 ***
## JobTenure                                                0.096765 .
## Age                                                      0.710259
## Salary                                                   0.000892 ***
## OvertimeHours                              < 0.0000000000000002 ***
## StateFL                                                  0.211854
## StateNY                                                  0.622957
## StateTX                                   0.00000000000000314 ***
## JobGroupingJob B                          0.00000000000144298 ***
## JobGroupingJob C                                         0.005840 **
## JobGroupingJob D                          0.00000006093970345 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4329.1  on 21433  degrees of freedom
## Residual deviance: 2771.6  on 21416  degrees of freedom
## AIC: 2807.6
##
## Number of Fisher Scoring iterations: 9
```

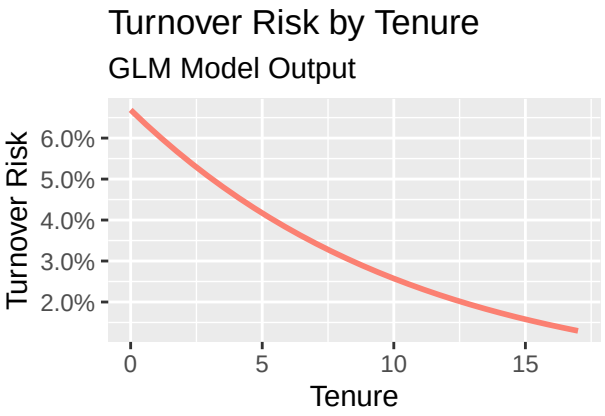
Driver Relationships

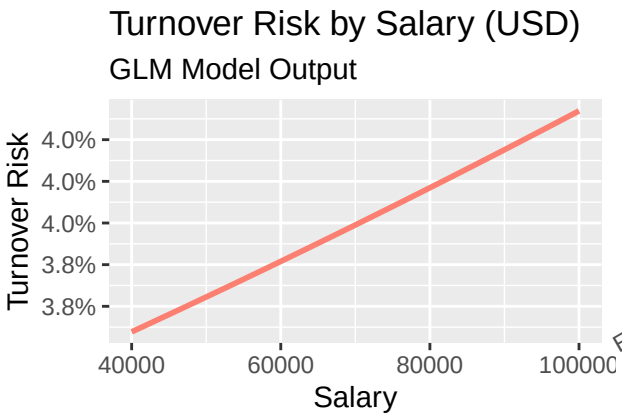
The model results above show how the likelihood of turnover varies by each predictor in the model, but the results can be difficult for one's audience to digest. It's standard practice to convert these results into more intuitive visuals, so that stakeholders can better understand the model. Here are graphics for each driver:

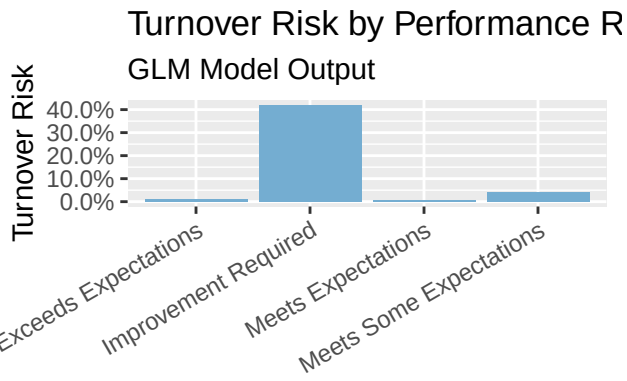

Turnover Risk by Overtime Hours
GLM Model Output
Turnover Risk
4.0%
3.0%
2.0%
1.0%
0
20
40
60
80
OvertimeHours

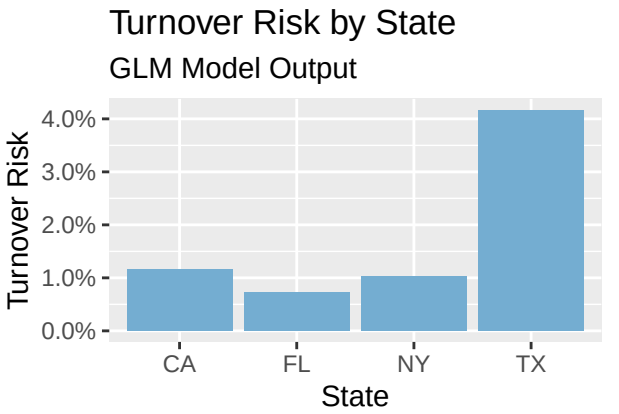

Turnover Risk by Job Tenure
GLM Model Output
Turnover Risk
4.8%
4.4%
4.0%
0
2
4
6
8
JobTenure


Turnover Risk
20.0%
15.0%
10.0%
5.0%
0.0%
Job A
Job B
Job C
Job D
JobGrouping


Turnover Risk by Tenure
GLM Model Output
Turnover Risk
6.0%
5.0%
4.0%
3.0%
2.0%
0
5
10
15
Tenure







The previous model captures each numeric predictor (Age, Tenure, Job Tenure and Overtime) using a linear relationship. However, it's not always realistic to expect the relationship to resemble a straight line.

Let's try using the R's Generalized Additive Modeling (GAM) feature, which allows more complex relationships to emerge.

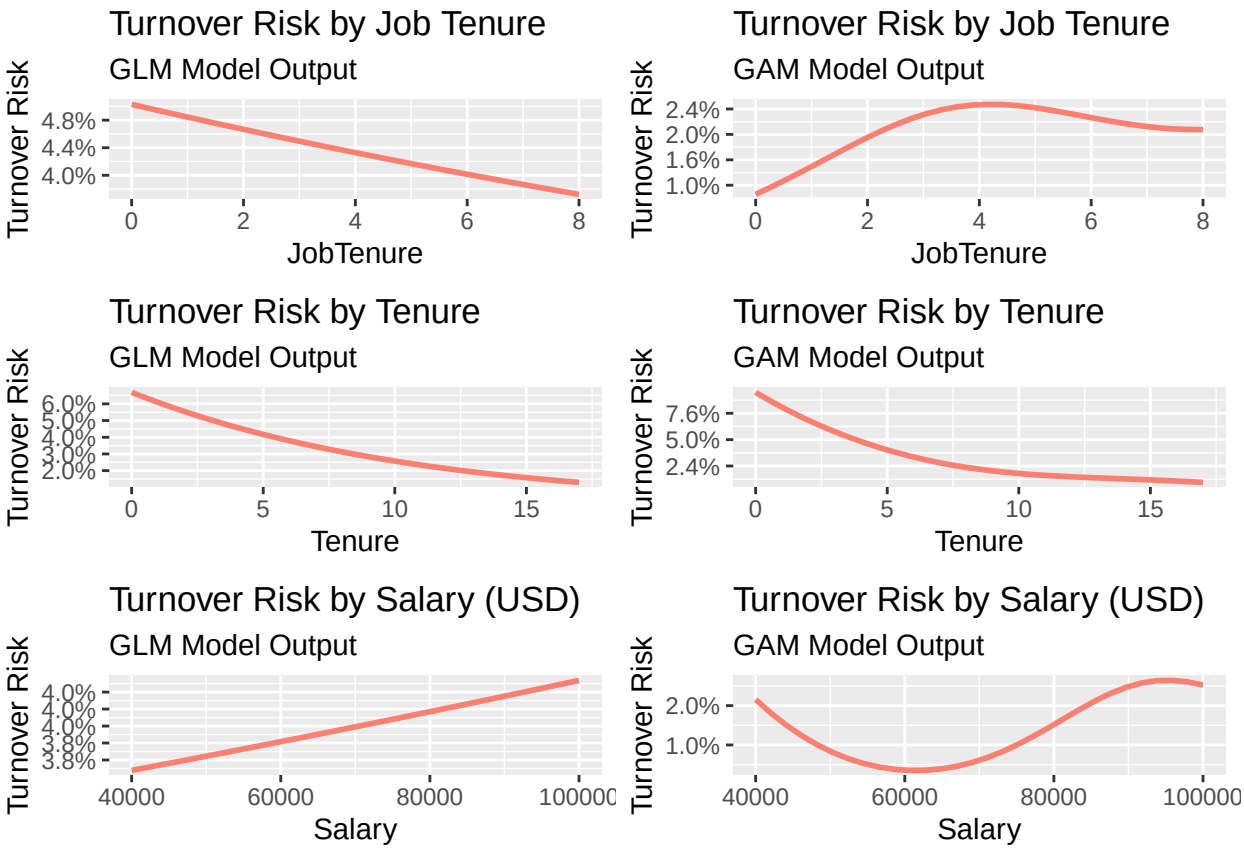
```
glmModel <- gam(Turnover ~
                  s(Tenure)+
                  Performance+
                  s(JobTenure)+
                  Age+
                  s(Salary)+
                  s(OvertimeHours)+
                  State+
                  JobGrouping,
                family=binomial, data=filter(data,test_train=='train'))
summary(glmModel)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Turnover ~ s(Tenure) + Performance + s(JobTenure) + Age + s(Salary) +
##     s(OvertimeHours) + State + JobGrouping
##
```

```
## Parametric coefficients:
##                    Estimate Std. Error z value   Pr(>|z|)
## (Intercept)      -2.417721   0.588845  -4.106 0.00004028 ***
## PerformanceOther -2.142132   0.474338  -4.516 0.00000630 ***
## Age              -0.016176   0.005639  -2.869    0.00412 **
## StateFL           0.632854   0.374321   1.691    0.09090 .
## StateNY           0.415408   0.236066   1.760    0.07846 .
## StateTX           0.624045   0.140369   4.446 0.00000876 ***
## JobGroupingJob B  0.488146   0.175092   2.788    0.00530 **
## JobGroupingJob C  0.966629   0.222722   4.340 0.00001424 ***
## JobGroupingJob D  0.782075   0.260634   3.001    0.00269 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq             p-value
## s(Tenure)         3.619  4.373 102.55 <0.0000000000000002 ***
## s(JobTenure)      3.450  4.258  10.43               0.0391 *
## s(Salary)         8.790  8.969  52.72 <0.0000000000000002 ***
## s(OvertimeHours) 3.130  3.879  46.18 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.0279   Deviance explained = 11.4%
## UBRE = -0.81848  Scale est. = 1         n = 21434
```

```
# glmModel <- gam(Turnover ~
#                   Tenure+
#                   Performance+
#                   JobTenure+
#                   Age+
#                   Salary+
#                   OvertimeHours+
#                   State+
#                   JobGrouping,
#                 family=binomial, data=filter(data,test_train=='train'))
# summary(glmModel)
```

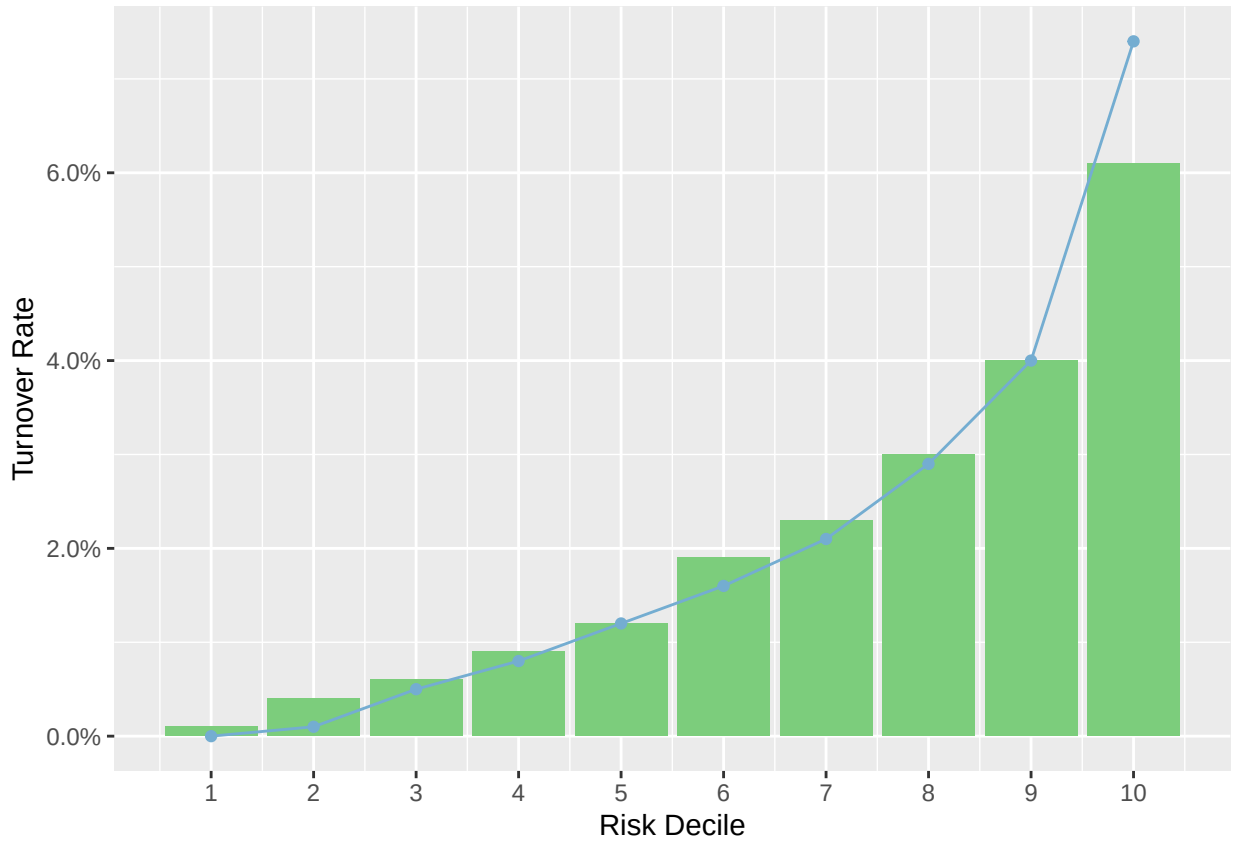
Here is a comparison between the relationships found using the two different modeling techniques, for four of the predictors. Note the more complex patterns that emerge when the GAM Model is used:



Segmentation Accuracy

Next, it's important to examine the accuracy of the model by testing it on a subset of the data. In this example, we used 80% of the data to train the model, and here we use the remaining 20% to test the accuracy. This 20% holdout sample allows us to see how the model will perform in the future.

There are many different methods of measuring accuracy; in this example, we use the decile lift approach. In this approach, the model above is used to assign a turnover risk to each employee in the test dataset (without the model actually knowing whether the employee turned over). The employees in the test set are then divided into ten groups, with group one containing the lowest-risk employees and group ten containing the highest risk. We then calculate the observed turnover rate of each group and compare it to the model's prediction. The green bars show the observed turnover rate, while the blue line shows the predicted rate. The model shows excellent segmentation, with the lowest-risk group also having the lowest observed turnover rate, vice-versa for the highest risk group, and a clean ordering of observed turnover rates across the groups.



Even when a lift diagnostic shows excellent accuracy, it's important to understand what this says specifically about what the model can and can't do. The diagnostic suggests that the model does a good job segmenting the overall workforce into high- and low-risk groups. If the model designates an employee as high-risk, it's safe to assume that employee does in fact have an elevated risk of turning over within the next year.

But segmentation accuracy is not the same as forecasting accuracy. Just because a model is able distinguish between high- and low-risk employees across the workforce, this does not necessarily mean it will provide an accurate forecast of turnover rates for key employee groups, which is often the primary intended purpose of an attrition model.

How to Optimize a Model for Forecasting Accuracy

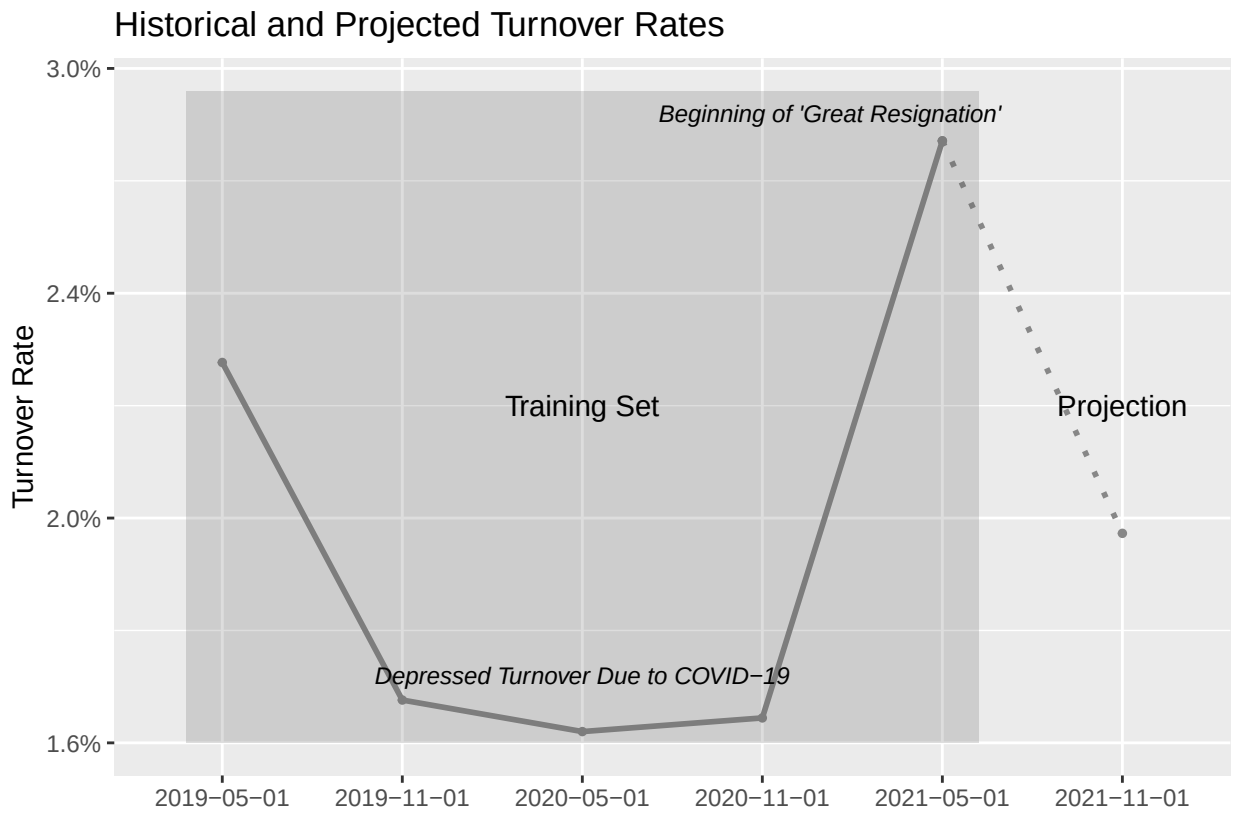
In order to evaluate the accuracy of our model's forecasts, we first need to determine how to group the workforce in a way that is relevant to the organization. For this particular example, let's assume that the organization is interested in predicting headcount needs at the business level, and therefore requires reliable attrition forecasts for each distinct business unit within the organization.

Table 2 shows the forecast accuracy for each business unit.

Table 2 reveals that the model is significantly under-forecasting for all four business units. For instance, the model has predicted a turnover rate of 2% for Business B, while the actual observed rate is nearly twice that at 3.8%. This raises the question: why is the model under-forecasting turnover across the board? To answer this, we need to examine the historical data that was used to train the model.

Table 2: Forecast Turnover by Business

Business	Headcount	Predicted Turnover Rate	Observed Turnover Rate	Error
Business A	3824	2.2%	3.1%	-1.0%
Business B	3425	2.0%	3.8%	-1.8%
Business C	1667	1.6%	2.4%	-0.8%
Business D	673	2.9%	4.5%	-1.5%



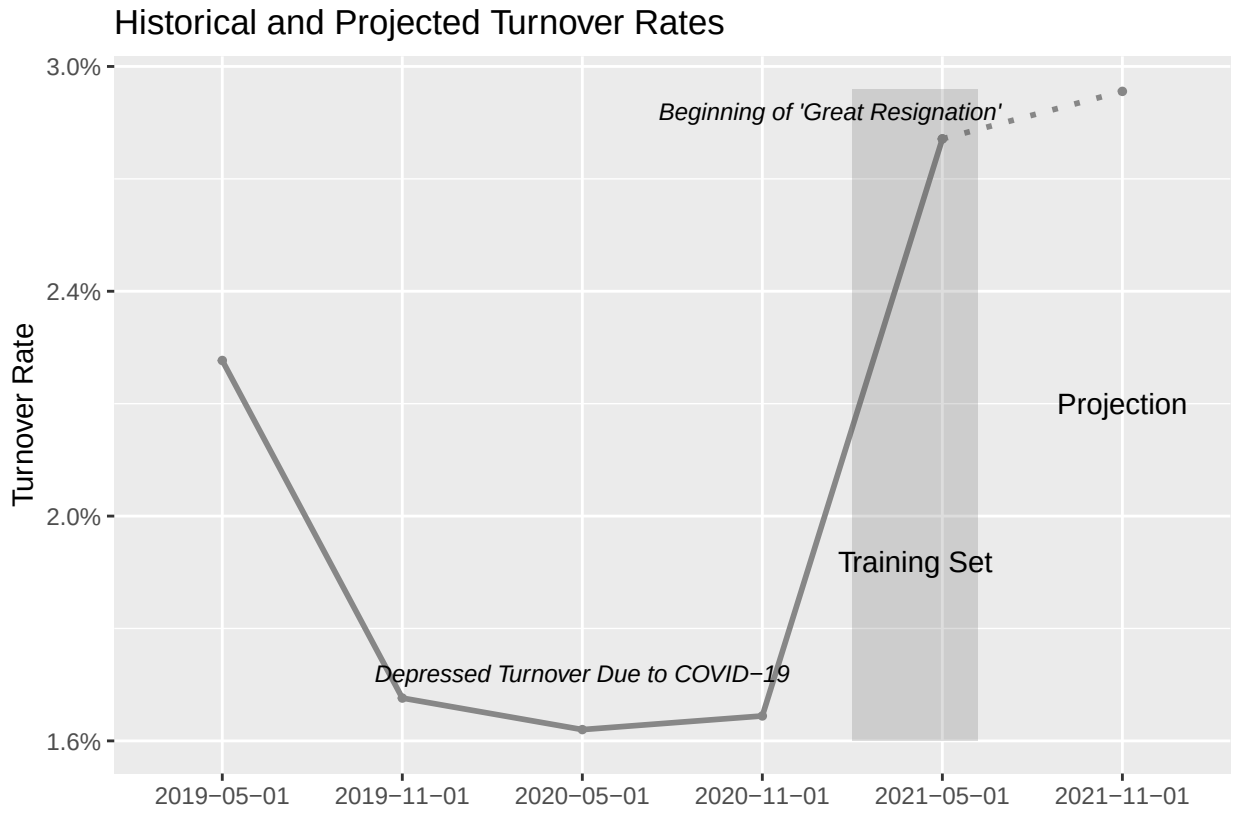
Our model was constructed using data from a period of three years, divided into discrete six-month intervals. The turnover rate for each period is displayed in this chart, providing insight into changes in turnover over time. The chart highlights the impact of the COVID-19 pandemic, with turnover artificially low at around 1.6% during the early stages of the pandemic, when economic uncertainty prevailed. During the later stages of the pandemic, we see a sharp increase in turnover, reflecting the so-called 'Great Resignation' that took hold in the latter half of 2021. The dotted line represents the period that the model is used to forecast. The model assumes a regression-to-the-mean across this historical period. The peaks and valleys in attrition behavior are treated as random fluctuations, and the model takes the average of these fluctuations.

However, in this instance, we are aware that the fluctuations in turnover behavior are not random; they are a result of specific historical circumstances that are unlikely to be repeated. As a result, including data from the early stages of the pandemic leads to an artificially low forecast. Instead, it may be more appropriate to train the model on only the most recent time period, assuming that the near future will resemble the recent past.

Table three and the graphic below it show the result of this change. Note that in the historical graphic, we see that the low turnover during the early phase of the pandemic is no longer exerting a downward pull on the forecast. Furthermore, Table 3 shows a significant improvement in accuracy.

Table 3: Forecast Accuracy By Business, Revised

Business	Headcount	Predicted Turnover Rate	Observed Turnover Rate	Error
Business A	3824	3.0%	3.1%	-0.2%
Business B	3425	3.0%	3.8%	-0.8%
Business C	1667	2.4%	2.4%	0.0%
Business D	673	4.7%	4.5%	0.2%



One important caveat: in this exercise, we have tuned our model with the help of a diagnostic that knows the actual turnover rates for each business during the period that we are attempting to forecast. In a real-life forecasting situation, we would not have the benefit of this information, since the future would remain unknown. The point of this exercise is to show the importance of considering our training dataset carefully, and making sure that we weight each historical period appropriately, removing those periods that may contain anomalous behavior that threatens the accuracy of the forecast.