Six Reasons Your Employee Attrition Model Needs a Rebuild

Employee attrition modeling is one of the most widely touted applications of predictive modeling. The process of building and deploying an attrition model can seem deceptively simple: prepare the data, apply a predictive algorithm, and voila, you have a reliable indication of which high-value employees are most likely to quit.

While building a basic employee attrition model might seem easy, building one that works *in action* – that truly and effectively serves your business needs -- is another matter entirely. Just because a model looks good in an executive slide deck doesn't mean it will get the job done; high-level summary statistics can look great while dysfunction reigns under the hood.

Read on to discover the various ways in which an attrition model can be fraught with poor construction, even if it looks good on the surface.

1. You spent too much time on window dressing

An attrition model uses employee attributes, such as age, tenure, and salary, to estimate an employee's propensity to quit. When you're building a model, you generally have a long list of potential attributes to consider. Modern machine learning methods can handle many attributes, and data wrangling techniques can be used to build additional attributes that don't initially appear in the data. Since employee attrition is such a relatable use case – most people have left a job at some point – you might find that your stakeholders are eager to share their theories about what attributes they think should be considered. All these factors add to the ever-growing list of potential predictors to choose from.

When done correctly, using many attributes can bring nuance and sophistication to a model, but there is typically a point of diminishing returns. Inexperienced practitioners often get wrapped up in chasing the ever-smaller improvements in accuracy that come with adding more and more attributes. At some point, adding more predictors amounts to mere window dressing, and your energy is better spent focusing on foundational issues that help your stakeholders more directly.

For example, you might build a basic three-factor model which reveals that young, low-tenure, low-salary employees are the most likely to quit. You may then find that risk within this group can be further differentiated using other factors, such as union membership and performance history. At some point, however, the gains in accuracy from adding new predictors becomes quite small, and your efforts are better spent elsewhere. In our experience, this typically occurs somewhere between the seventh and tenth predictor.

One way to avoid this problem is to feed all your potential predictors into an algorithm that can rank their potential usefulness. The random forests package in R has a variable importance plot designed for this purpose, or in the linear regression context, you can try forward or backward selection. We recommend doing this as an initial step *before* you've spent too much time exploring and preparing each predictor, so you can get an initial sense of which predictors deserve your attention.

With the energy and time that you've saved by focusing only on the most useful predictors, you can focus on more foundational issues that we'll discuss in the following sections.

2. You built for the wrong purpose

Form should follow function. A model can be optimized to meet several different goals, so your model should be designed with your specific use case in mind. Three of the most common uses of attrition modeling are **Risk Segmentation, Forecasting,** and **Policy Evaluation.** Below are some tips on how to optimize for each of these goals.

Risk Segmentation is perhaps the most familiar application of attrition modeling. A segmentation model seeks to differentiate employees based on their turnover risk. Typically, the goal is to identify the highest-risk employees so that these employees can be targeted for potential retention efforts. However, in many cases, the stakeholder does not actually intend to use the model for this purpose, so you should be clear that this is your stakeholder's intention. Otherwise, optimizing for a different purpose may be preferable.

If you determine that segmentation is indeed your stakeholder's desired goal, you should focus on including a wide variety of predictors in your model. As discussed above, additional predictors will yield diminishing returns at some point, but if segmentation is the goal, this point will come later than it would otherwise. If the model needs to be optimized for segmentation, even small refinements in risk can prove worthwhile.

While risk segmentation may be the most familiar application of attrition modeling, **Forecasting** is more widely desired by real-world stakeholders. Forecasting is less about estimating the risk of individual employees, and more about producing accurate attrition forecasts across larger groups. Forecasting tends to be a more desired application because it is more actionable. Organizations accept that individual employees will come and go, but by having a reliable forecast of churn at the aggregate level of interest, they can develop plans to proactively develop and recruit talent to fill anticipated talent gaps.

To optimize a model for forecasting accuracy, some segmentation is desirable, but trend detection becomes equally important. Furthermore, it is critical to know which talent groupings the organization is most interested in forecasting. A workforce can be divided and studied in many ways, so the model builder must know the most relevant groupings from the perspective of the stakeholder. For example, a public utility might have trouble filling jobs for senior engineers in Austin, mid-level linesmen in Dallas, and junior electricians in Houston. The organization needs a reliable three-year forecast for each of these groups. Armed with this understanding of the business case, the model builder can then study the most effective way to segment the data into the relevant groupings, use trend detection algorithms to see how turnover has trended for each group, and provide accurate forecasts of where turnover is headed.

Finally, some organizations use attrition modeling for **Policy Evaluation**. The organization might want to know whether a certain professional development program improves retention, or it might want to know the optimal wage to pay different types of employees to minimize attrition risk. Because these questions center around *causal* relationships, they require a careful statistical approach to isolate the causal impact of the policy being investigated. The importance of distinguishing correlation from causation is particularly relevant for this application.

In a forecasting or segmentation model, distinguishing correlation from causation is less important. If a high salary is correlated with a low turnover risk, it doesn't matter whether the low salary is the *cause* of the low risk, or if this relationship is a mere correlation that can be explained by some third factor. As long as salary can be used to reliably predict attrition, the matter of causality isn't so important. But in the realm of policy evaluation, this issue becomes critical. In the salary example, the organization may be looking to make wage decisions based on the analysis, which means it wants to be sure that the higher wage is in fact the *cause* of the lower risk.

Therefore, a policy evaluation model should key in on the policy in question and study how the estimated impact of this policy changes with the addition and subtraction of other factors from the model. The modeler should pay special attention to the statistical significance of the policy's relationship to turnover risk. Furthermore, the model builder should study statistical interactions with respect to the policy – does the effectiveness of the intervention vary based on other predictors? For example, perhaps a higher salary substantially reduces turnover risk for one critical talent group, but not for another. A finding like this can help the organization decide where to allocate scarce resources.

3. You forgot to anticipate future weight loads

To forecast turnover for a group of employees, you need to know the likelihood that each individual in the active workforce will turn over, but you also need to account for turnover among the employees *who aren't in the workforce yet*. Over the upcoming three years, some portion of an organization's turnover will occur among employees who will be hired during this future time horizon. From a forecasting perspective, this presents a challenge – we don't yet know who these individuals are, how many there will be, and for which jobs they'll be hired.

This requires a new hire forecast to be layered on top of the turnover forecast for existing employees. We can study the behavior of past new hires and use trend methods to simulate the composition and behavior of future new hire cohorts, providing an estimate of how many additional turnover cases will come from new hires. This ensures that a critical source of turnover will be accounted for in the forecast.

4. Your blueprint is impossible to read

The past decade has seen a surge in enthusiasm for complex 'black box' style predictive algorithms, such as random forests, support-vector machines, and gradient boosting techniques. These complex algorithms have found many exciting uses in the realms of image processing, speech recognition, bioinformatics and more. They are well-suited to these use cases, but in the realm of employee attrition modeling, their complexity does more harm than good.

The biggest drawback of black box algorithms is that they are much less interpretable than more traditional approaches like logistic regression. A regression-based turnover model is quite transparent – the data scientist can create visualizations to show how the model uses each predictor to adjust an employee's turnover risk. If interactions are present in the model, she can explore how this relationship changes across different employee groups. This allows the data scientist to check whether the relationships in the model conform to intuition. In some cases, counter-intuitive relationships are discovered and allowed to stand, providing new insights into employee behavior; in other cases, such relationships are deemed to be statistical noise and removed from the model. In employee attrition modeling, the best modeling outcomes arise from a combination of machine power and human judgement.

The lack of interpretability of black-box style algorithms prevents human judgement from adjusting the model. Some consider this to be a strength, but our experience suggests otherwise – black-box style models tend to be less accurate than those which have been skillfully adjusted by an experienced analyst. Furthermore, uninterpretable models are less accountable to stakeholders. In order to trust a model, most stakeholders want to know how the model arrives at its predictions. A more interpretable model will instill trust and facilitate wider use on the part of stakeholders.

5. You hired the wrong inspector

Most model builders understand that diagnostics are important, but many fail to choose the right sort of diagnostic for the job. One popular diagnostic for attrition modeling is the ROC curve, which provides a visual illustration of the true positive rate (the probability of correctly predicting a turnover event, given that a turnover event occurred) against the false positive rate (the probability that a turnover event is predicted, given that a turnover event did not occur). Calculating the area under the ROC curve (AUC) can provide a rough gauge of accuracy, and therefore it can be a useful metric for comparing different models during the model-building process. However, this diagnostic is not particularly useful for presenting to stakeholders, nor for truly understanding the model's effectiveness in action.

One problem with the ROC diagnostic is that thinking of turnover models in terms of false positive rates or false negative rates makes little intuitive sense. Turnover is a rare event and - for most job roles of interest - nearly all individual employees will have a turnover risk well below 50%, so on an individual level, you're almost always better off predicting that the individual does not turnover. This renders the binary classification schema of the ROC curve counterintuitive for turnover modeling.

Which diagnostic is chosen in place of the ROC curve will depend on the use case. For segmentation models, the *decile analysis* is a great approach. To use this approach, a test dataset must be set aside during the modeling phase, and this set must not be used for training the model. Once the model is trained, it should be used to assign a turnover probability to each employee in the test set. The employees are then divided into ten deciles of equal size, with group 1 having the lowest model-assigned risk, group 10 having the highest risk, and groups 2-9 being ordered accordingly. The analyst should then calculate the actual turnover rates across these ten groups, checking for a clean ordering of turnover rates across the groups (with group 1 having the lowest *observed* turnover rate in addition to the lowest predicted rate), and a close concordance of predicted and observed turnover rates within each group. This provides stakeholders with a sense of confidence that the model has correctly identified high- and low-risk employees, and that predicted turnover rates will be accurate at the aggregate level.

For forecasting models, the decile analysis can be useful, but measuring forecasting accuracy across key talent groups is the priority, and the decile analysis is not adequate for this purpose. Rather, an effective diagnostic must capture the historical turnover trends at both the overall headcount level and across those groupings and subgroupings that the stakeholder deems important. The diagnostic should then compare these historical rates to the rates predicted by the model. Where turnover predictions show a clear departure from historical behavior, the model builder should be able to explain what factors account for this departure.

6. You neglect upkeep and repairs

Many model builders make the mistake of building a model, putting it into production, and letting it be. However, where employee attrition is concerned, patterns are always changing, and these changes must be detected and accounted for if the model is to remain accurate. This has been particularly true during the COVID-19 pandemic, which saw a dramatic reduction in turnover behavior during the early stages of the pandemic, followed by a dramatic increase in turnover during the second half of 2021.

In order to keep a model current, parameters should be re-estimated each time new data become available, ideally on a monthly or quarterly basis. This ensures that temporal shifts in employee behavior are detected and built into the model. Ideally, then, the model should not be hard-coded. Rather, the model fitting process should be automated and built into the data pipeline, with automated diagnostics that allow the analyst to review any changes that occur and ensure that model accuracy holds up over time.

Summary

Machine learning algorithms have revolutionized the way organizations make decisions, but their ease of construction is often overstated. A good predictive model can help your organization run smoother, but only if it has been crafted with expertise and care. In the initial stages of model deployment, it can be difficult to tell the difference between a good model and a bad one, so don’t let promising summary statistics make you overconfident. Be sure that your model is built for the right purpose, holds up to the right diagnostic tests, and has the flexibility to adapt.

Have Questions or Ready to Get Started?

[Contact Us Today]