Final Report: Predicting Employee Attrition

By Kyle Rodriguez

Problem Identification Overview

How can we identify employees at-risk for voluntary resignation based on their traits, performance, supervisor's behaviors, etc.?

Context:

It's estimated that losing an employee can cost a company 1.5-2x the employee's salary. Depending on the individual's level of seniority, the financial burden fluctuates. For hourly workers, it costs an average of $1,500 per employee, while for technical positions, the cost jumps to 100-150% of salary. Finally, at the high end, C-suite turnover can cost up to 213% of the employee's salary.

Not only are you forced to dedicate time and resources to recruiting, onboarding and training a new hire after an employee leaves, but it's estimated that two thirds of all sunk costs due to turnover are intangible, including lost productivity and knowledge, and internal strain on employees while the roles remain unfilled.

Being able to predict, and possibly minimize, employee turnover by identifying employees that may be at-risk for leaving voluntarily could help to reduce the cost of replacement hire due to turnover and increase employee retention. Specifically, the question I sought to answer was, "**What contributing factors increase the likelihood of an employee leaving voluntarily?**", and can we use those to determine when, and if, an employee might resign?

Data and Method:

This synthetic dataset contains 18 months' worth of daily performance and attrition data for a factory whose organizational structure comprises 508 workers. Due to employee turnover, a total of 687 persons appears in the dataset.

The dataset contained ~412,000 rows of data and 42 distinct features. Each record represents a type of **event that occurred on a particular day in relation to a particular worker**. It's possible for a given worker to have more than one event (and row) for the same day.

The dataset's observations cover both regular daily events (like workers' attendance and daily level of Efficacy) and special one-time events (like accidents, an employee's termination, or the onboarding of a new employee). A unique feature of the dataset is diverse causal relationships "hidden" within the data that are waiting to be uncovered through machine learning. See the supplemental information section for more information on event types.

After some data wrangling and feature engineering, the dataset and problem statement were approached like a binary classification supervising learning problem, with a Resignation designated as the positive class, 1, and non-resignations designated as the negative class, 0.

Dataset: https://www.kaggle.com/datasets/gladdenme/factory-workers-daily-performance-attrition-s?resource=download

Exploratory Data Analysis (EDA)

Some initial exploration of the dataset indicated that over the 18-month period that the data covered, there was 77 total voluntary resignations due to seven distinct reasons. Those distinct reasons being: **Poor teammates, Under-recorded Efficacy, Recruited away, Low commitment, and ethically inferior supervisor.** Notably, low commitment, under-recorded efficacy, and ethically inferior supervisors were the top 3 reasons.

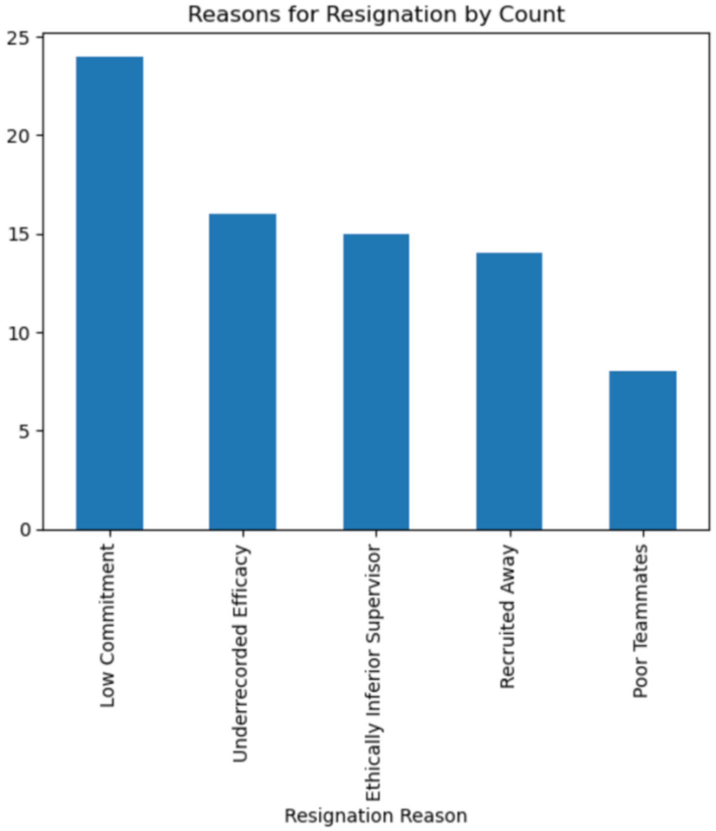
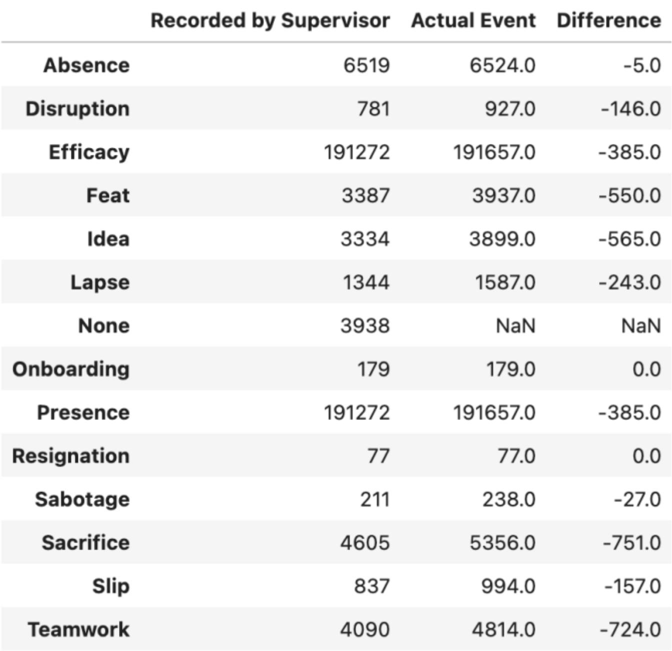


Figure 1: Bar chart showing voluntary resignation counts by reason.

Taking a closer look at the events performed by employees, two seemingly important features were noted: the **actual event** and the **events recorded by supervisor**. I became interested in whether there were any discrepancies in what the supervisor may have recorded, since having under-recorded efficacies and an ethically inferior supervisor were among the top 3 reasons for resignations.



	Recorded by Supervisor	Actual Event	Difference
Absence	6519	6524.0	-5.0
Disruption	781	927.0	-146.0
Efficacy	191272	191657.0	-385.0
Feat	3387	3937.0	-550.0
Idea	3334	3899.0	-565.0
Lapse	1344	1587.0	-243.0
None	3938	NaN	NaN
Onboarding	179	179.0	0.0
Presence	191272	191657.0	-385.0
Resignation	77	77.0	0.0
Sabotage	211	238.0	-27.0
Sacrifice	4605	5356.0	-751.0
Slip	837	994.0	-157.0
Teamwork	4090	4814.0	-724.0

Figure 2: Table showing mismatched events performed by employees, grouped by event type.

Interestingly, there was a very high number of mismatches for not just Efficacy events but **positive events** like Feats, Ideas, Sacrifices and Teamwork. In total, there were **3938 records** that had mismatched events; where the supervisor recorded something different than was actually performed for that given record.

Lastly, it occurred to me that of those employees that voluntarily resigned, they may have had some traits in common, perhaps like a high commitment score. Since these hidden traits directly affects what kind of events an employee might exhibit (see supplemental information), it became apparent that these trait scores may be significant.

From the boxplots below, there is a common trend among employees that voluntarily resigned to have scores for these hidden traits that fall between 0.7 and 0.9, with a median hovering around 0.75 across the board. Interestingly, the majority of resigned employees did not have exceptionally high scores, nor exceptionally low ones.

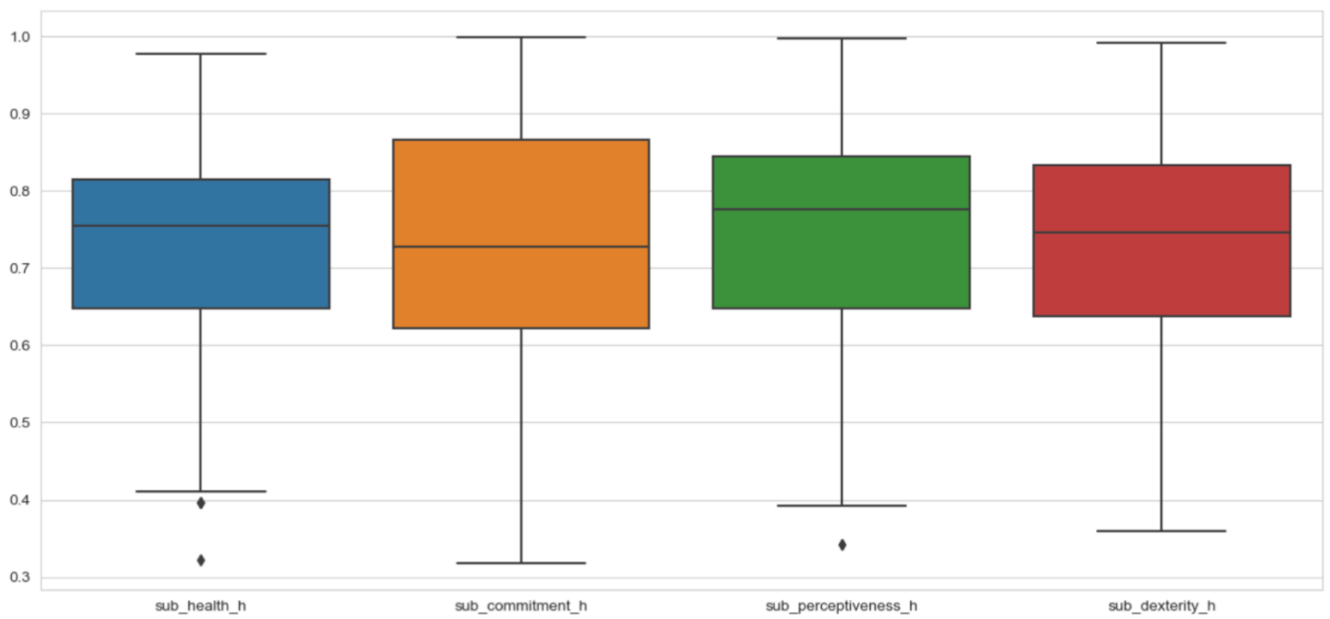


Figure 3: Box plot showing scores for hidden employee traits: Health, Commitment, Perceptiveness and Dexterity (L-R).

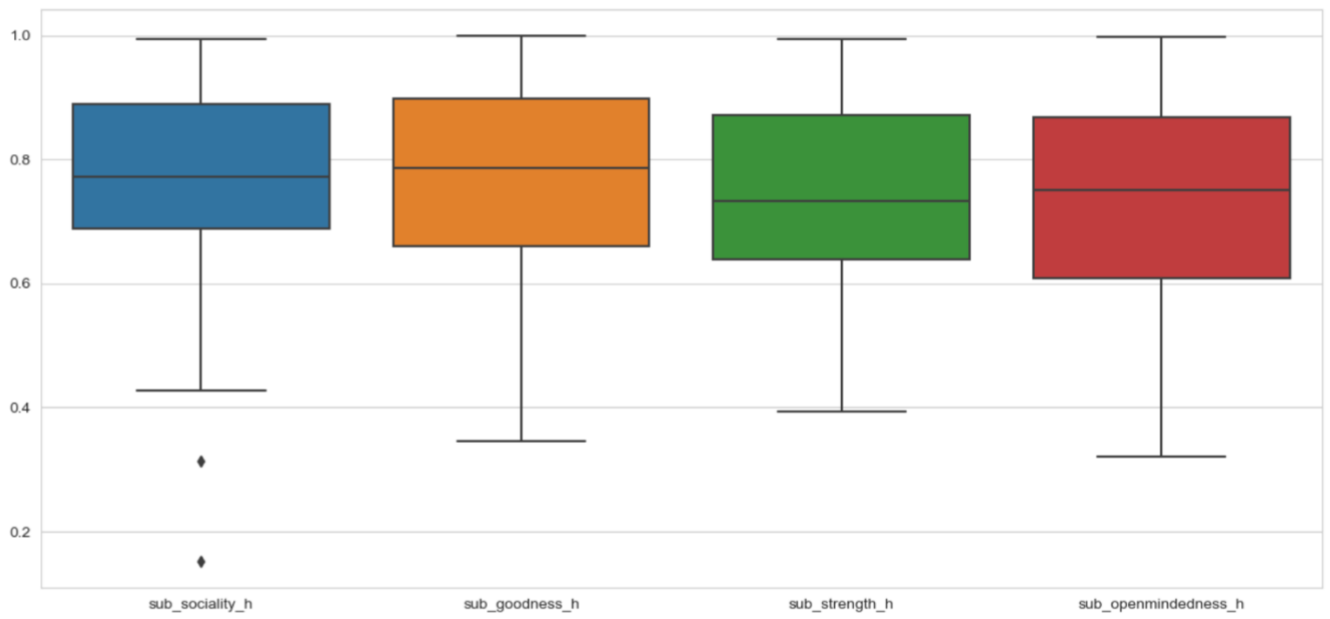


Figure 4: Box plot showing scores for hidden employee traits: Sociality, Goodness, Strength, and Open-mindedness (L-R).

Data Preprocessing Steps of Note

The dataset was relatively tidy and had little to no missing values for the features of the dataset. However, there were still some preprocessing steps needed to be done before moving on to the modelling step.

Target Feature

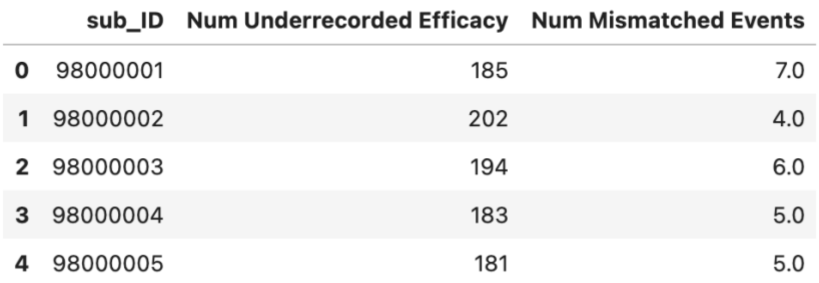
Firstly, the dataset was missing a target feature designating the binary classification of the records as to whether they were a resignation or not, so a column was generated as such.

One-hot encoding categorical variables

Secondly, the features of the dataset exhibited a mixture of categorical and numerical data types. Several categorical features like employee sex, supervisor sex, employee shift etc. were encoded via pandas' get_dummies function for better processing by the model. As a result, this increased the number of features to greater than 70 in total.

Feature Engineering Mismatches

After seeing how significant the number of mismatched events were recorded, I decided to count the number of times each employee had a mismatched positive event and add an additional column to hold that information. In addition to that, in a similar way, I decided to count the number of times an employee had specifically an under-recorded efficacy score. This was another feature added to the dataset.



	sub_ID	Num Underrecorded Efficacy	Num Mismatched Events
0	98000001	185	7.0
1	98000002	202	4.0
2	98000003	194	6.0
3	98000004	183	5.0
4	98000005	181	5.0

Figure 5: Table showing a subset of employee records and counts of Mismatched Events and Under-recorded Efficacy scores by supervisors.

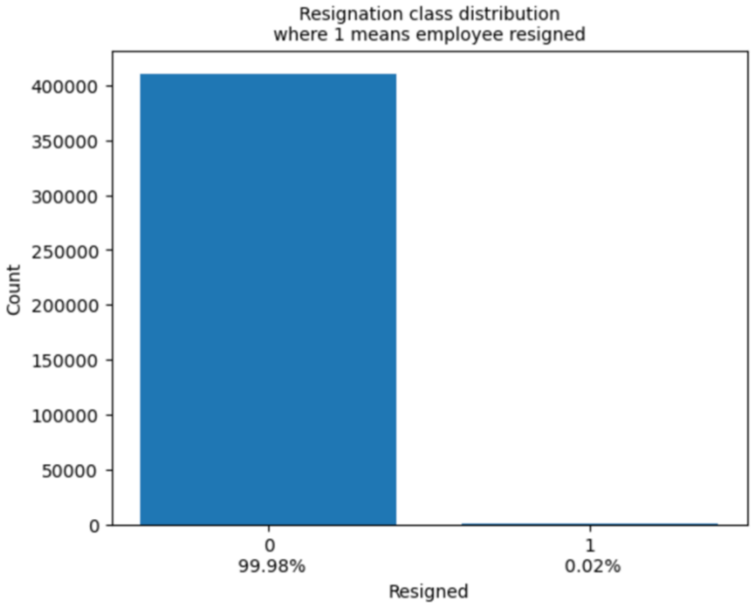
Dropping Duplicate Records & Unnecessary Features

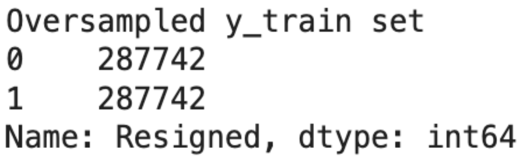
Due to the increased number of features, areas around the dataset where the fat could be trimmed included dropping some unnecessary features like employee/supervisors' names, date of events, day of the week the event happened, etc. In the end, the number of features were reduced to about ~62 in total.

The dataset had a substantial number of repetitive records and initially dropping duplicate rows seemed to be a good approach to reducing this redundancy. After careful consideration, however, it was decided that each row would simply be treated as an individual employee record. This was due to the fact that the positive class (resignations) was significantly small.

Oversampled Minority Class

In the EDA stage, it was noted that there were only 77 resignations of ~412k records. This corresponded to a positive class of just 0.02% and as a result the dataset was highly imbalanced. Any subsequent modelling would skew predictions in favor of the majority negative class – which is not of particular interest. Subsequently, the positive class was randomly oversampled using the imblearn library to a more 50:50 split.





```
Oversampled y_train set
0    287742
1    287742
Name: Resigned, dtype: int64
```

Figure 6: Bar chart and code snippet showing the percentage of positive class (1) vs. negative class (0) before and after Random Oversampling.

Model Description

Due to the problem being a binary classification one, the first of three models I decided to use was a Logistic Regression model. In addition to this, Random Forest and Catboost models were also chosen. From the dataset, the feature matrix and target feature were first identified as the X and y variables then the data was split into training and test sets, with a test size of 0.3.

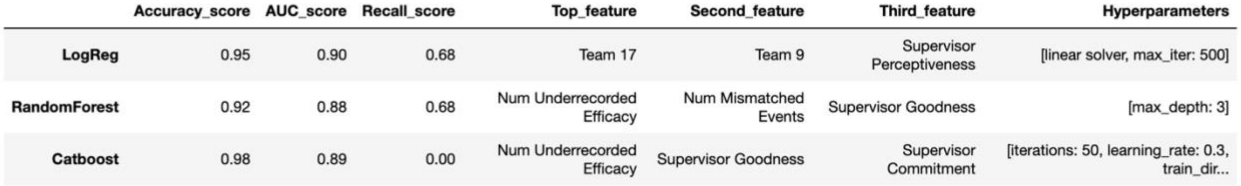
Beginning with the logistic regression model, there were several iterations done to first test the effect of some preprocessing steps such as: a baseline model, a model with oversampled minority class, a model with scaled numeric features only and finally a model with both an oversampled minority class and scaled numeric features. After the iterations, it was determined that the best performing model was the one with the oversampled minority class only, and thus the scaling step was left out in subsequent model types.

The remaining model types, Random Forest and Catboost models, were trained using the oversampled data only. Important to note, prior to oversampling the minority class, iterations were done on each of these model types after some hyperparameter tuning to accommodate the highly imbalanced data, and in the end, the oversampling was deemed to best route to go for any meaningful results.

Model Performance

After some initial testing of the Logistic Regression model's performance with oversampled data, scaled data or a combination of the two, it was decided that the iteration with oversampled data was the best performing baseline model.

Therefore, comparing the three models out of the box on accuracy, AUC score, and recall, the top performing model was the Logistic Regression model with 0.95, 0.90 and 0,68, respectively. This was followed by the Random Forest model with an accuracy score of 0.98, an AUC score of 0.88 and a recall score of 0.68. Lastly, the CatBoost model performed the poorest in terms of recall, but substantially well for AUC and accuracy, having an AUC score of 0.89 and 0.98, respectively (Figure 7).



	Accuracy_score	AUC_score	Recall_score	Top_feature	Second_feature	Third_feature	Hyperparameters
LogReg	0.95	0.90	0.68	Team 17	Team 9	Supervisor Perceptiveness	[linear solver, max_iter: 500]
RandomForest	0.92	0.88	0.68	Num Underrecorded Efficacy	Num Mismatched Events	Supervisor Goodness	[max_depth: 3]
Catboost	0.98	0.89	0.00	Num Underrecorded Efficacy	Supervisor Goodness	Supervisor Commitment	[iterations: 50, learning_rate: 0.3, train_dir...

Figure 7: Table comparing the final three models' metrics, important features and hyperparameters used for tuning.

It is important to note that these results were simply from the models out-of-box performance with very little hyperparameter tuning.

Model Findings

Each model had different features that were listed as the top contributing factors in terms of predictive power, though there was some overlap. For the logistic regression model, the most important predictors for voluntary resignation included being on Team 17, followed by being on Team 9 and lastly, having a supervisor with a low perceptiveness score (Figure 8).

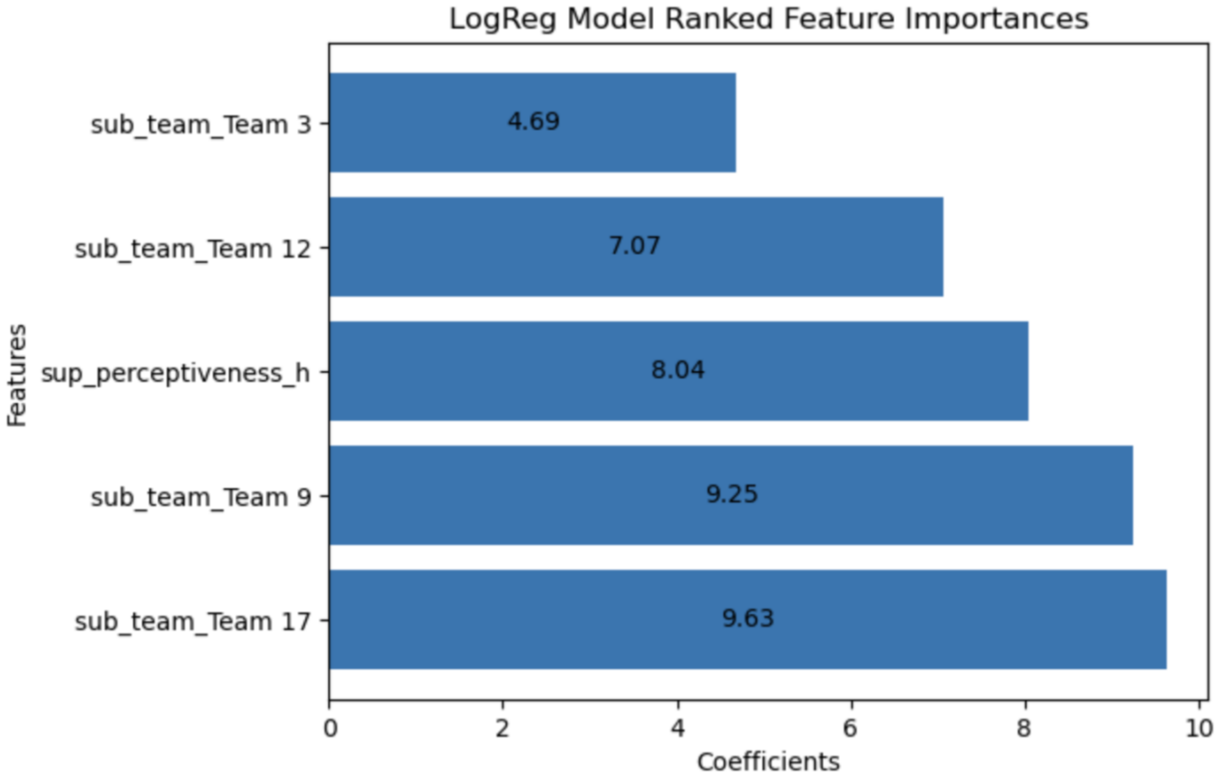


Figure 8: Top 5 features ranked by importance for the Logistic Regression model.

For the CatBoost model, the most important predictors for voluntary resignation included having a high number of under-recorded efficacies, followed by having a supervisor with a low goodness score and lastly, having a supervisor with a low commitment score (Figure 9).

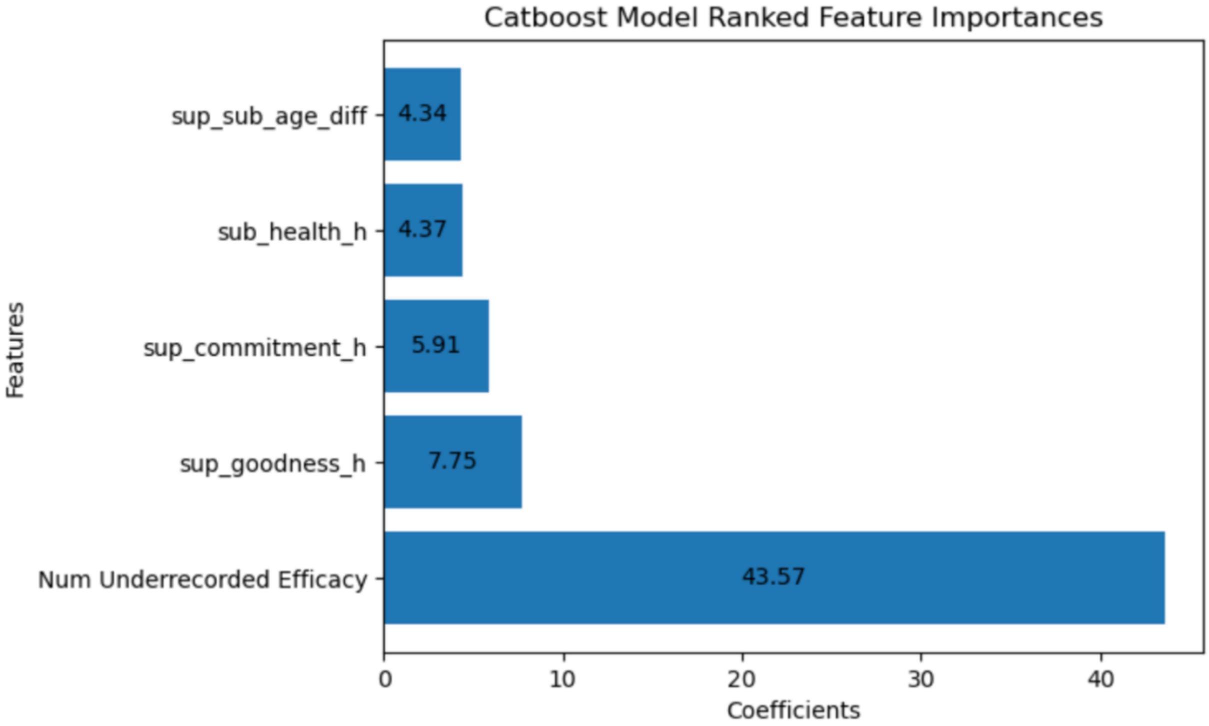


Figure 9: Top 5 features ranked by importance for the CatBoost model.

Lastly, for the Random Forest model, the most important predictors for voluntary resignation included having a high number of under-recorded efficacies, followed by having a high number of mismatched events recorded and lastly, having a supervisor with a low goodness score (Figure 9).

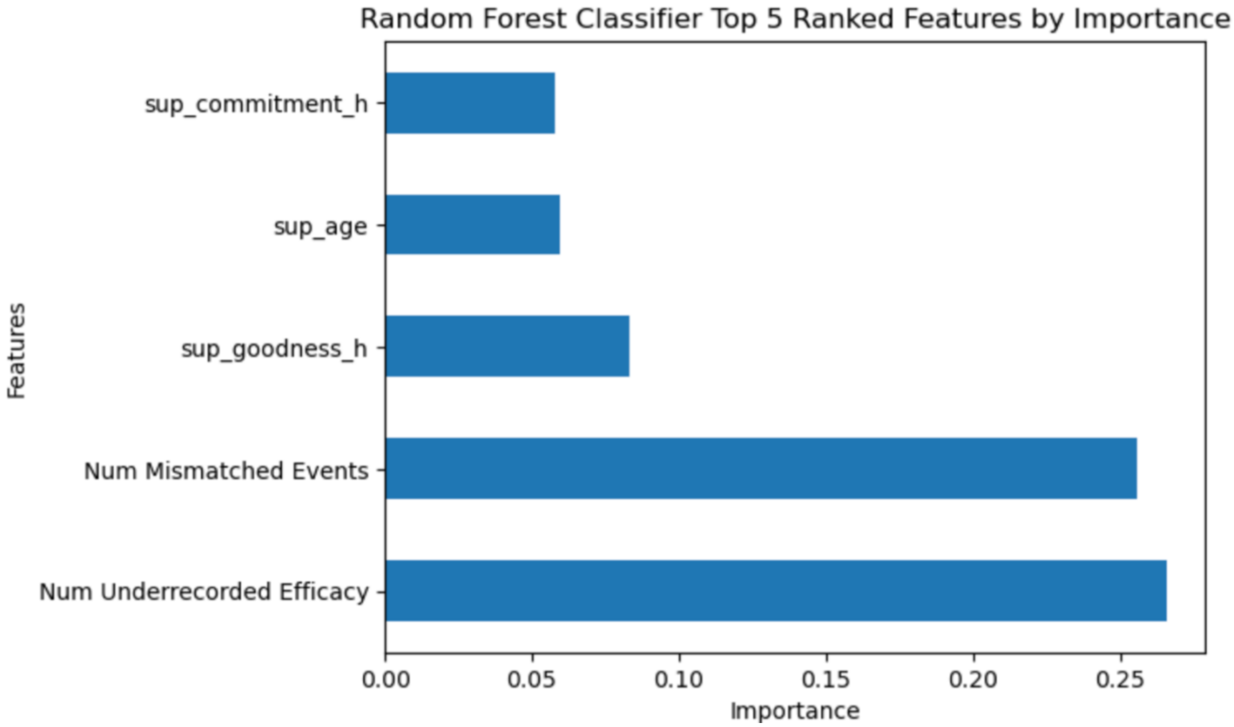


Figure 10: Top 5 features ranked by importance for the Random Forest model.

Based on the results, the following recommendations seem appropriate for the upper-level management at this factory:

1. A proper evaluation on how events and efficacies are recorded by supervisors so as to minimize as much as possible any possible mismatches. This recommendation is aimed at removing the possible frustration that employees may feel as a result of the mismatches.
2. Secondly, the company may consider employing better behavioral-based interview questions aimed at evaluating more accurately a managerial role's Perceptiveness, Goodness and Commitment, in however they may deem fit.

Next Steps

Due to the models not being well-tuned, the results from these findings should be taken with a grain of salt. The next step in this endeavor would be to tune the parameters of these models to better optimize performance and achieve higher scores in the specific metrics.

In addition to this, seeing that the hidden traits from this dataset is not a realistic feature to obtain in the real world, perhaps incorporating data that may serve as a proxy for these trait scores may be useful such aptitude scores, or behavioral strengths.

Lastly, some more useful features that employees may actually care about such as commute time, salary, years employed and work-life balance may add even more value to the performance of the models.

Supplemental Information

Events by Type:

Presence: worker showed up to work

Absence: worker did not show up to work

Efficacy: reflects the degree of efficiency and productivity that an employee demonstrated over the course of the given workday. There are two related elements: the actual Efficacy that the employee generated on the given day, and the recorded Efficacy that the worker's supervisor entered into the factory's HRM/ERP system for the employee for that day.

Resignation: on the given date, an employee quit his or her job (i.e., the employee experienced a voluntary separation). After resigning, the employee was no longer a part of the workforce and did not generate any future behaviors. Only Laborers and Team Leaders are liable to experience a Resignation event

Termination: the employee was fired by the organization (i.e., the employee experienced an involuntary separation).

Onboarding: subject is a newly hired employee who began work on the given date. In order to maintain a stable size for the factory's workforce, a new employee is hired whenever an existing employee has resigned or been terminated.

Idea: when an employee imagines some innovative new approach to solving a long-standing problem or otherwise improving the factory's functioning. Workers with a high Perceptiveness stat are more likely to have an Idea.

Lapse: when a worker makes some severe mental mistake that negatively impacts the factory's functioning (e.g., by omitting some step from a business process or ordering the wrong parts). Workers with a low Perceptiveness stat are more likely to have a lapse.

Feat: when an employee performs some exceptional act dependent on physical skill (e.g., expertly manipulating some high-precision tool or sorting objects into bins with remarkable speed). Workers with a high Dexterity stat are more likely to generate a Feat.

Slip: when an employee experiences some accident or misstep resulting from a lack of physical adroitness (e.g., knocking over a storage rack or dropping some delicate instrument). Workers with a low Dexterity stat are more likely to generate a Slip behavior.

Teamwork: when an employee goes to exceptional lengths to encourage, teach, or otherwise support his or her colleagues in some way. Workers with a high Sociality stat are more likely to generate this event

Disruption: occurs when an employee quarrels with colleagues, belligerently rejects a supervisor’s request, or otherwise displays a hostile attitude toward his or her coworkers. Workers with a low Sociality stat are more likely to generate this event.

Sacrifice: when a worker performs – of his or her own initiative – some action that’s unpleasant or inconvenient for the employee but which spares others from suffering the same hardship (e.g., volunteering to take on some grueling or unpleasant assignment that no one else wants to do). Workers with a high Goodness stat are more likely to generate a Sacrifice behavior.

Sabotage: when an employee knowingly performs some act (typically, with the hope that he or she won’t be observed while doing so) that will damage equipment, spoil products, or otherwise lessen the organization’s productivity or harm its brand image. Workers with a low Goodness stat are more likely to generate a Sabotage behavior