Predicting Employee Attrition:

Final Report

Zulykath Lucero

INST 414

Spring 2025

Professor Kem

GitHub Repository: https://github.com/luceroz/predicting-employee-attrition

Executive Summary

High attrition rates lead to greater recruitment and training expenses, as well as a decrease in business productivity. To alleviate this, I created a predictive model that flags employees with a high probability of attrition. This allows HR to form targeted and effective retention strategies that will help decrease attrition rates in a company. I used the "IBM HR Analytics Employee Attrition & Performance" dataset from Kaggle. The models I created were Logistic Regression, Random Forest, and XGBoost. After using hyperparameter tuning on each of these models, I used the best version of each to create a soft voting ensemble model. To improve performance, I addressed the data's class imbalance by using class weighting and SMOTE. Recall was prioritized as the performance metric for evaluation and comparison. After comparison, the tuned Logistic Regression model was chosen as the best implementation because of its high recall value of 0.68. While this model is limited by the absence of richer data, it provides businesses with an accurate tool for decreasing turnover.

Problem Statement & Business Context

Employee attrition, or the rate at which employees are leaving the organization, is a measure used by businesses and applicants when making decisions. If applicants see that an organization has high attrition rates, they might assume that there is something wrong with your organization, causing employees to leave very quickly. As a company's HR department runs analytics on its staff, noticing high attrition rates could signal an issue in how the company is treating employees. There are also costs associated with recruiting, hiring, and training new employees to replace those employees who are leaving. A predictive model to flag employees in

a company who are likely to leave the company within the next couple of months could allow the company to target those individuals with retention tactics.

Creating this model is significant because it can be used by businesses to lower employee attrition rates. The benefits of a lower turnover rate are numerous. The cost reduction of not having to recruit new employees is significant. Some sources even suggest that larger US companies spend up to $1 trillion annually[1] on the process of hiring new employees every year; therefore, lowering this number can free up the budget for development in other areas. Another benefit businesses see when lowering high turnover rates is a change in organizational culture. From the additional responsibilities placed on remaining employees after other employees leave to the disruption of losing connections and having to make new ones, a high turnover rate lowers morale and increases stress on employees. The quality of the work being done at a business is also deeply impacted by high attrition rates. Senior employees have valuable institutional knowledge, while new hires take an average of a year or two to reach the productivity level of the rest of the team[2].

Attrition is influenced by various factors, but one study claims that 42% of employees who choose to leave their current employment do so for preventable reasons[3]. By the time managers find out about an employee's decision to leave the company, it is almost impossible to change their minds. This is because most employees make this decision before talking to their employers. This means predicting who is at risk of leaving is incredibly important in order to make a dent in high turnover rates.

---

[1] https://www.netsuite.com/portal/resource/articles/human-resources/employee-retention-benefits.shtml

[2] https://www.netsuite.com/portal/resource/articles/human-resources/employee-retention-benefits.shtml

[3] https://www.gallup.com/workplace/646538/employee-turnover-preventable-often-ignored.aspx

Many companies' HR departments already collect valuable data from their employees, which could be used to create an accurate and tailored model. Performance evaluations, exit interviews, and employee satisfaction surveys could be a good source of information for training a model predicting employee turnover. Through the use of techniques learned in class and on my own, I believe creating this attrition prediction model was technically feasible and valuable to my understanding of machine learning.

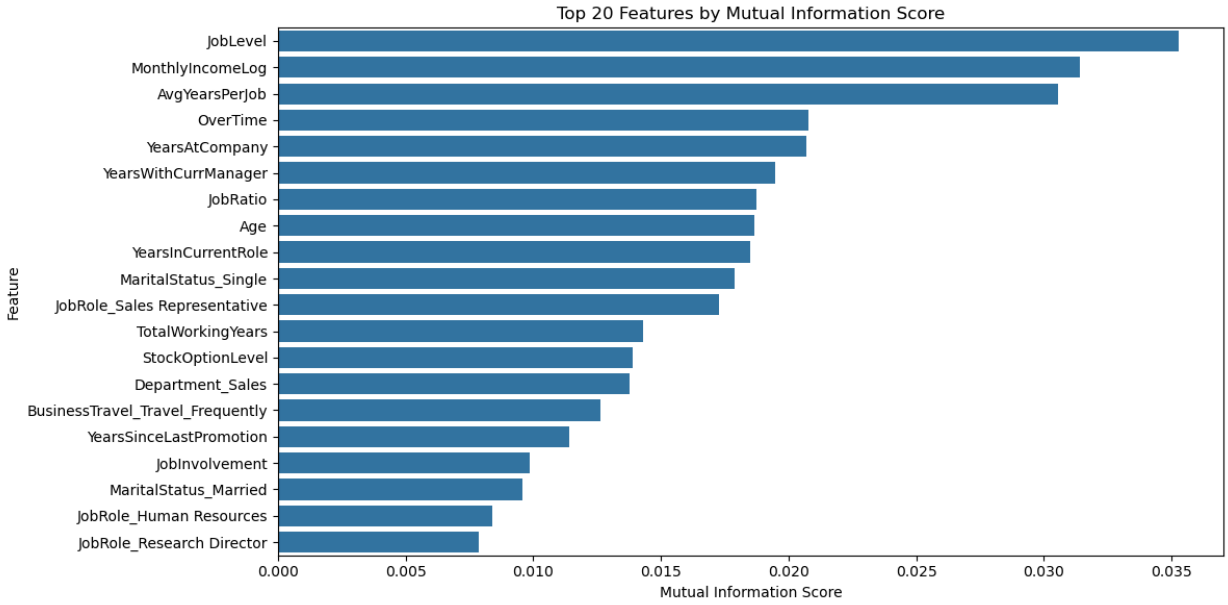
Methodology

Data Cleaning and Preprocessing

The first step I took to clean the dataset was to drop the columns I found were not usable for the training process. This included *EmployeeCount*, *Over18*, *StandardHours*, *EmployeeNumber*, and *PerformanceRating*. I then confirmed that no missing values were present in the data. Some binary variables were represented using words, but since models prefer numbers, I encoded them using 1 and 0. I chose this instead of one-hot coding because adding columns will lead to unnecessary complexity when tree models already handle 1/0 natively. In the dataset, ordinal variables were already encoded as integers, so they were kept that way. This ensures that the models can interpret the variables as ranked levels. Nominal variables, however, were one-hot encoded to represent the values numerically but not ordinally. Lastly, I decided to use log transformation to fix the skewed distribution of *MonthlyIncome.* Through these changes, I hope to minimize the noise in the dataset and increase the reliability of my predictive model.

Feature Engineering and Selection

To optimize the predictive power of the data we have, I used feature engineering techniques. I started the process by creating new features. I created *JobRatio*, which is used to measure the proportion of an employee's career spent at the company. This variable could help represent what stage in their career each employee is at, which could be used to predict if they are likely to leave the company. The other variable I created is *AvgYearsPerJob,* which measures the average number of years spent per company they have previously worked at. This can help identify "job-hoppers". Next, I scaled the features using z-score normalization. I did this with the intention that the numerical features are all treated with the same weight, and to avoid larger values being treated with more importance. Lastly, I used the information gain method of calculating feature importance. I ranked the features on a graph and created an alternate dataset with only the top 20 most influential features.

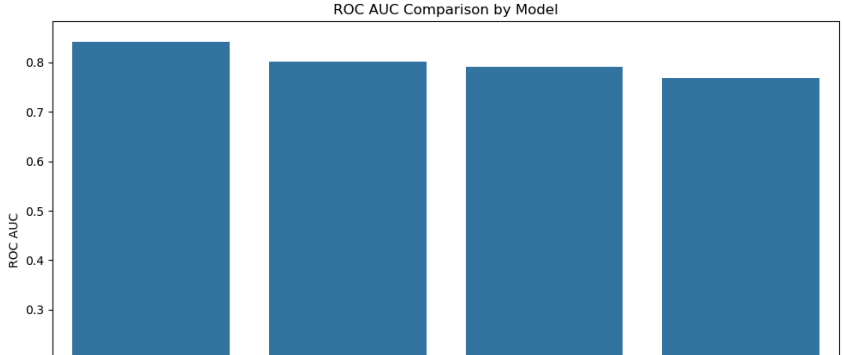


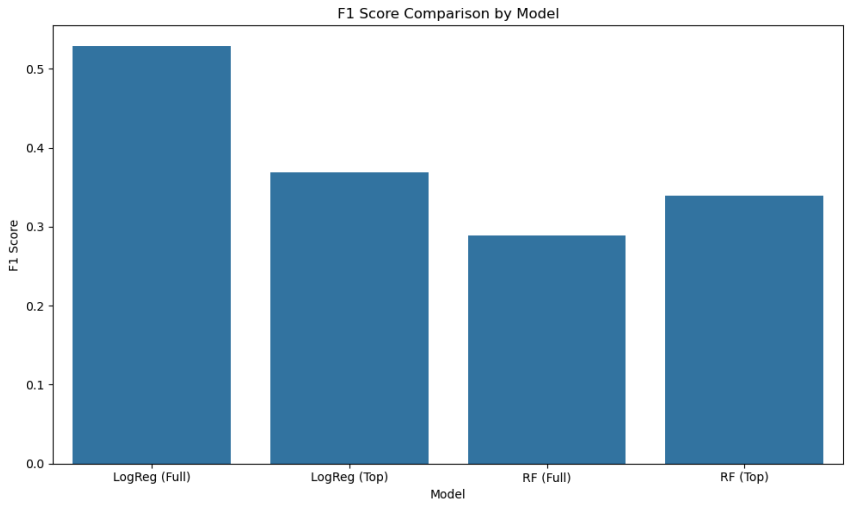
Baseline Model Implementation

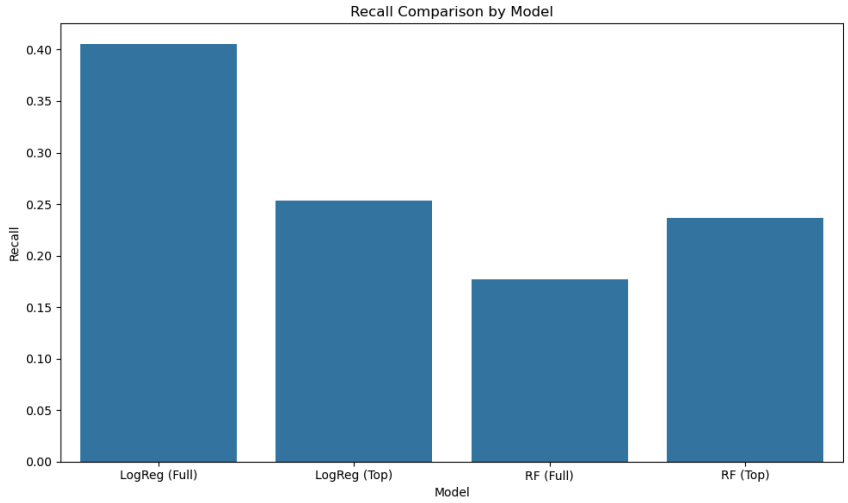
After preparing two clean and optimized datasets, I started the process of creating the models. I began by splitting both datasets into training and testing sets. I chose a 70/30 split in order to accommodate the size of the dataset. I then created the logistic regression model and trained it on the dataset with all the features in it. I generated the predicted values and printed out the accuracy, recall, F1 score, and ROC AUC. I printed out the cross-validated evaluation metrics as well. I repeated this process but trained the logistic regression on the dataset with the top 20 most influential features. Then, the same process was performed for the random forest model trained on the full dataset and the random forest model trained on the optimized dataset. One issue I ran into while training the first logistic regression model was that I was getting a convergence warning. I tried increasing the value of max_iter from 1000 to 5000, but the same warning appeared. Changing the default solver to "liblinear", however, fixed the issue.

Model Evaluation and Comparison

In order to compare the four models, I created a bar chart comparing the cross-validated F1, recall, and ROC AUC values. From the visualizations, I found that the logistic regression trained on the full dataset performed the best. This model had the highest cross-validated F1 score (0.53), recall (0.41), and ROC AUC (0.84). The logistic regression trained on the optimized dataset had the next highest scores for all three metrics. As for the Random Forest models, the version trained on the full dataset had a lower F1 score and lower recall but slightly greater ROC AUC than the version trained on the filtered dataset.







Model Optimization and Tuning

Using the visualizations created for the initial models, I picked the best version of each kind of model. I chose to train the logistic regression on the full dataset because that resulted in the highest values in each performance metric. For the random forest model, I decided to use the dataset with the top 20 most influential features because it resulted in higher recall. The initial performance evaluation also revealed that all models were underperforming in recall. To improve

model performance, I decided to address the class imbalance in the target variable, hypertune the models, and implement different kinds of models from the initial two.

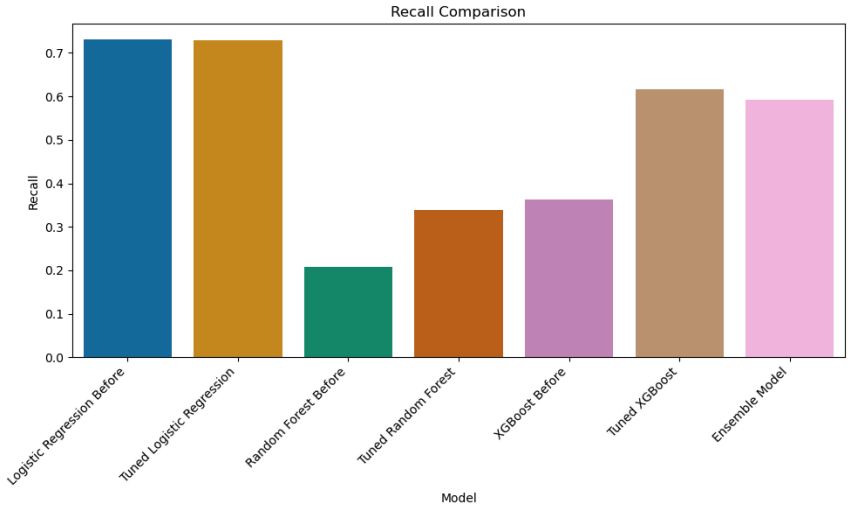
The extreme class imbalance was causing the recall to be low. To address this, I created an alternate training set using SMOTE, which created balance in a dataset by creating more instances of the minority class. I also adjusted each model to pay more attention to the minority class. For the logistic regression and random forest models, I added "class_weight='balanced'" to the parameters, and for the XGBoost models, I calculated the weight ratio of the *Attrition* variable and set the "scale_pos_weight" parameter. I also made sure to set the scoring metric to "recall" in the cross-validated evaluations.

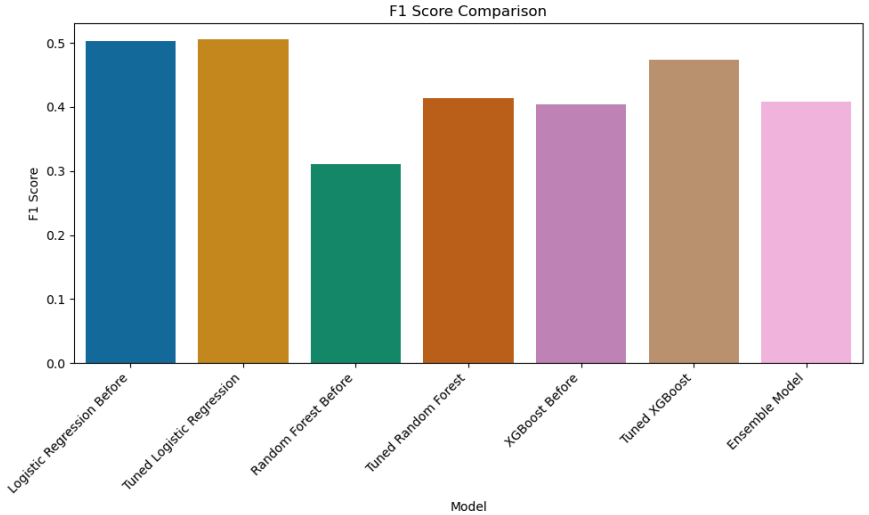
I created two versions of the linear regression, random forest, and XGBoost models, before and after hyperparameter tuning. I used GridSearchCV as my hyperparameter tuning method for linear regression and random forest models because the smaller parameter space in these models allows for such an exhaustive method. XGBoost, however, has a much larger parameter space, so I used RandomizedSearchCV.

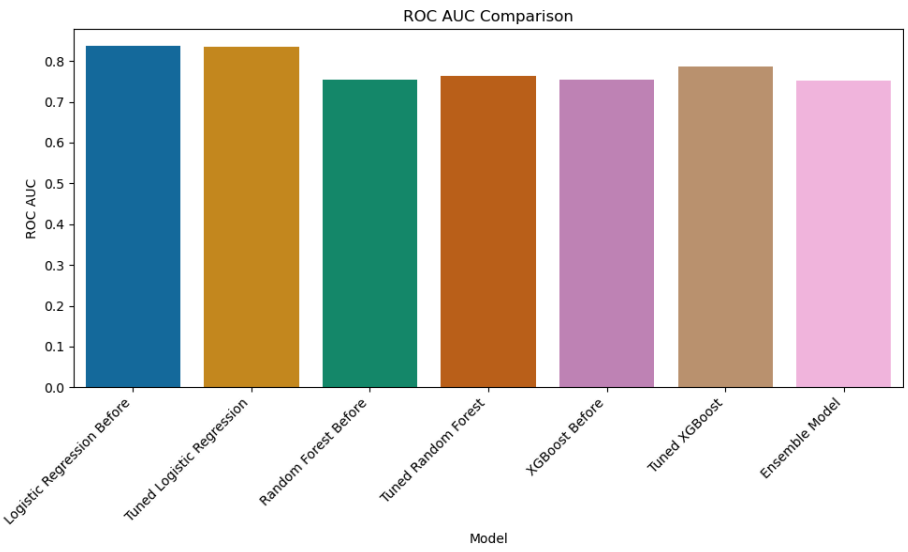
Lastly, I created an ensemble model using the tuned versions of the three previous models. In theory, combining all these weak learners will create a much stronger model. I trained this ensemble model on the SMOTE version of the training data. I evaluated all the models I created using recall, F1 score, and ROC AUC.

Model	**Recall**	**F1 Score**	**ROC AUC**
Logistic Regression Before	0.730053	0.503427	0.836139

Tuned Logistic Regression	0.729965	0.504974	0.835040
Random Forest Before	0.206915	0.310652	0.755397
Tuned Random Forest	0.337943	0.413554	0.763678
XGBoost Before	0.363032	0.403903	0.753953
Tuned XGBoost	0.616223	0.473692	0.786950
Ensemble Model	0.591549	0.407767	0.752189





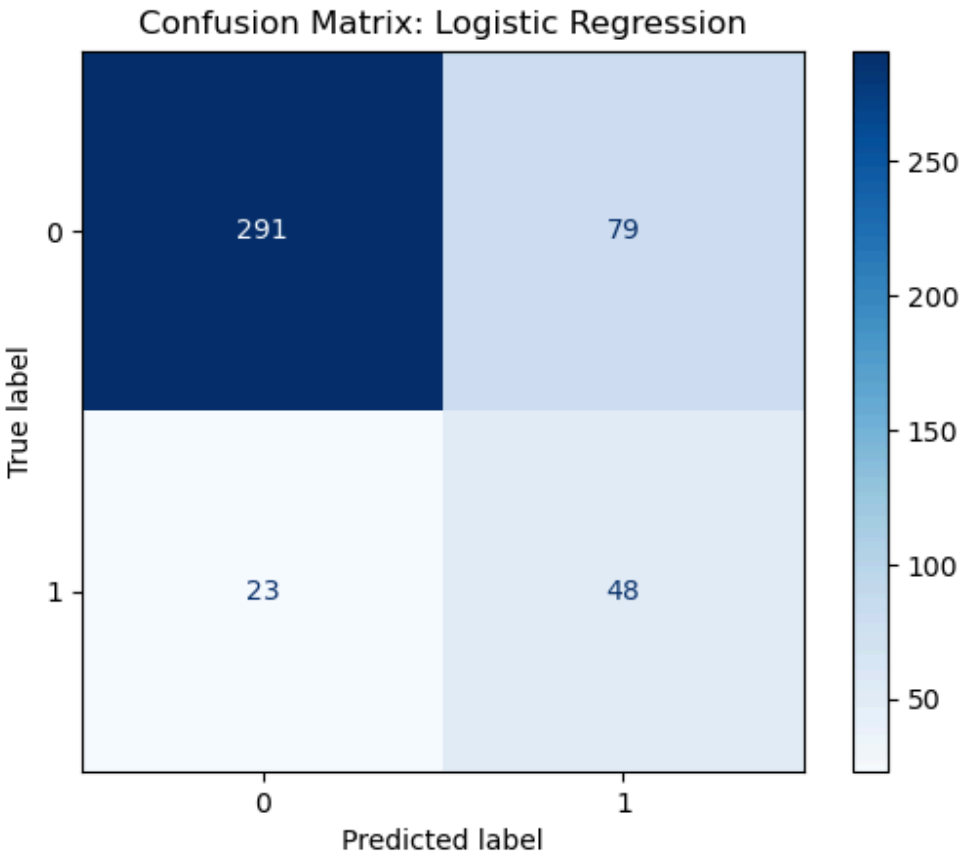


Model Evaluation & Selection

The table and charts show that we successfully improved recall across all models. The best recall obtained in the first round of modeling was 0.405, and in this round, we were able to get a max recall of 0.73. We can also observe that the hypertuning managed to improve each model. The greatest jump in performance after tuning was in recall for the XGBoost model. Recall was 0.363 for the model before tuning, and it jumped to 0.616 after tuning. Unfortunately, we were unable to raise F1 scores in any model to the desired 0.65.

Across all performance metrics, the logistic regression models outperformed the other models. I would consider the tuned logistic regression model the best model to proceed with further evaluation because of its high recall value and ease of interpretation. The classification report and ROC curve showed that the model's positive class had a recall value of 0.68, an F1 score of 0.48, and a value of 0.81 for ROC AUC. This shows success in 2 out of the 3 chosen performance metrics, with our model underperforming in terms of F1 score. Additionally, a confusion matrix shows that our model successfully minimized the number of false negatives,

which was our main objective.



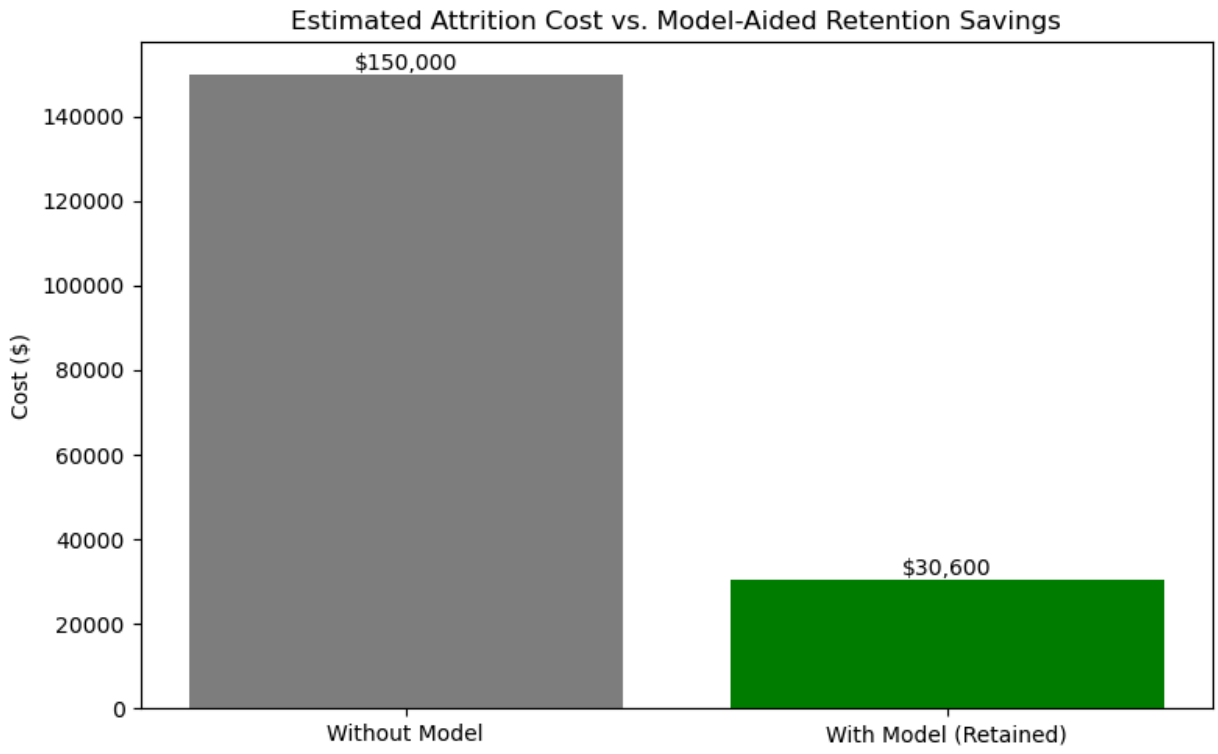
Implementation Strategy & Business Value

This model is valuable to businesses in two main ways: it gives the business insights as to what contributes to employees' decisions to leave the company, and it allows the HR department to intervene with retention techniques to get at-risk employees to stay at the company. Using the coefficients assigned to each feature, I ranked all the features based on importance. We can see from the visualization that employees with the job roles of "sales representative", "sales executive", and "human resources" are more likely to be flagged as at risk for leaving the company. This insight could prompt the HR department to investigate why these roles are

leaving at higher rates. Similarly, seeing the model links overtime with higher chances of attrition, the business might choose to cut down on the amount of overtime employees are doing. Making data-backed business decisions based on this model's attrition pattern insights could lower attrition rates all on their own.

Using this model to target at-risk employees with retention methods to get them to stay will reduce spending by saving businesses on recruitment, onboarding, and training costs. According to peoplekeep.com, some studies quantify the cost of losing an hourly worker as $1,500.[4] Given that our model achieved a recall of 0.68, it can correctly identify about 68% of employees likely to leave. In a large company, with 100 employees at risk of leaving, HR would only have to successfully intervene 30% of the time to save over $100,000. This number only rises when considering that the HR department will be more successful in their intervention with the insights provided by the model. If the intervention techniques are tailored to the reasons why the employee is choosing to leave, retention rates will skyrocket.

[4] https://www.peoplekeep.com/blog/causes-of-employee-turnover#:~:text=This%20leads%20to%20increased%20workloads.an%20hourly%20employee%20is%20%241%2C500.

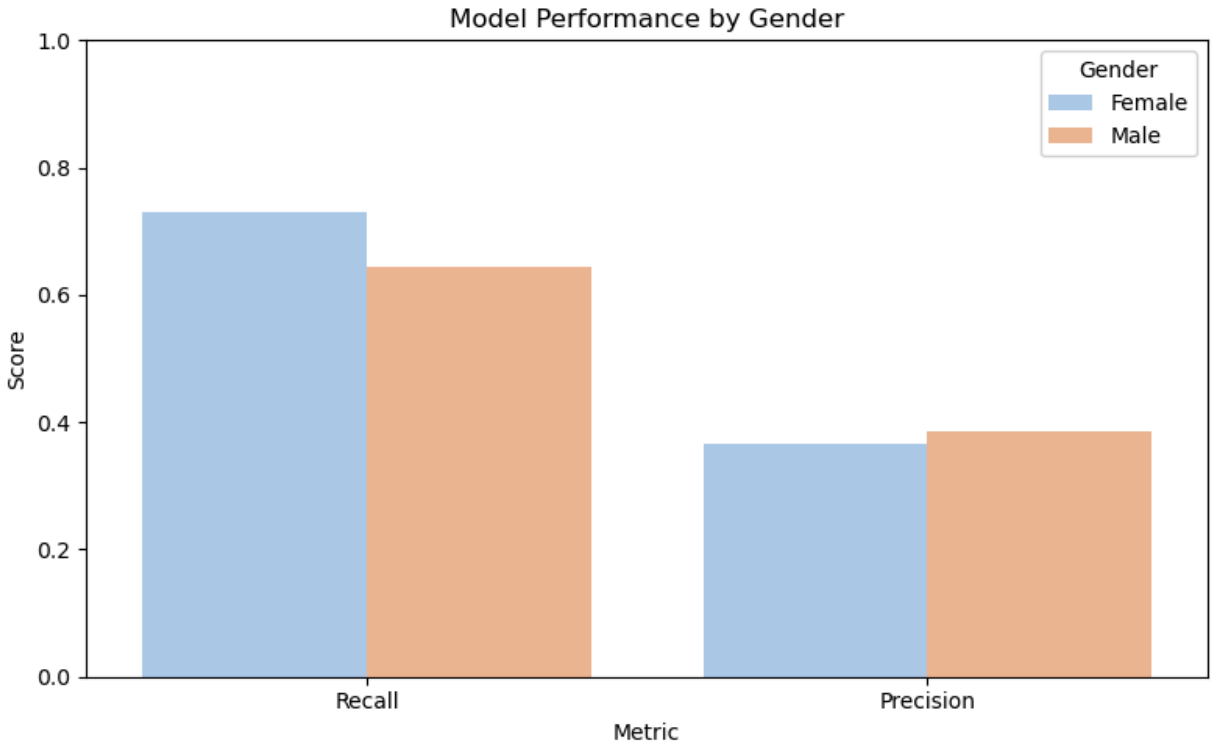


Some issues that businesses might face when implementing this model include technical training and the need to tune the model to fit the specific business's requirements. This implementation of the model does not include easy-to-use graphical user interfaces, meaning HR employees would have to be trained to use the model to extract insights. This period requires a reduction in productivity and a trade-off between training and other priorities. Additionally, every industry has different attrition rates and patterns, and every company has its own needs. Therefore, for the model to be useful, each company would have to collect large amounts of data from each employee. This can be challenging for businesses lacking the appropriate data infrastructure.

Ethical Considerations & Limitations

Although this model was made to be used for targeting time and resources towards employees who might need it, there is a possibility that it could be used for discriminatory hiring practices. HR employees may directly or indirectly discriminate against employees with similar demographics to those flagged by the model as “leavers”. The company might want to avoid applicants who might leave sooner or require more attention. This presents potential ethical challenges for implementation.

It is an unfortunate reality that racial and gender discrimination exist in the workplace, therefore, we must account for this by inspecting the model for potential bias. The training data does not collect racial information and is free of proxy variables (such as name and geographic location). The data does, however, include a gender variable. To get an idea of how gender affects the performance of the model, I created the following visualization.



The model predicts attrition in women with higher recall relative to men. This may potentially overlook male workers who are at risk of attrition. In order to mitigate this, we might consider dropping the gender column from the training data. Although this would have to be balanced with the reality that women have higher turnover rates than men, potentially because of institutional discrimination.

Although this model is effective within the context of this project, real-world applications will reveal limitations and potential improvements. The model could benefit from more data, since the total dataset contains only 1470 records, which could have led to overfitting. Other factors, such as macroeconomic indicators, also affect attrition rates, but my data does not include them. These additions would help improve the robustness of this model.