Employee Attrition Model Analysis

A comparison between Logistic Regression, Random Forest, and Neural Network based on employee attrition data.

Contributors: Michael Li, Jackson Zilles, Marco Ma, Elliot Hong, Ronan Loberg

B Engineering 178, Gabriel Gomez, 12/11/2024

Introduction

Employee attrition is a critical concern for organizations, impacting workforce stability, operational efficiency, and financial performance. In our scenario we will define employee attrition as whether or not the employee quits. Understanding the factors driving attrition and predicting employees at risk of leaving can help organizations develop proactive retention strategies, and provide more stable workplaces for employees and employees alike. This research explores employee attrition prediction using machine learning models applied to a comprehensive dataset of employee attributes. Leveraging logistic regression, random forest, and neural network models, this study compares their performance in accurately forecasting attrition. Key features, including demographic details, job satisfaction, work-life balance, and organizational tenure, are processed and analyzed to uncover patterns and predictors of employee turnover. By combining traditional statistical methods with advanced neural networks, this research aims to identify optimal modeling techniques for enhancing predictive accuracy and offering actionable insights to human resource management.

Description of Data

The dataset used for employee attrition prediction includes a variety of features representing demographic, employment, and job satisfaction details of employees. We take our dataset from Kaggle, a data science platform for machine learning (Arthurtok, 2018). This is a fictional dataset developed by IBM scientists. The dataset includes a total of 35 features, including but not limited to age, business travel, hourly rate, distance from home, education, education field, gender, job level, job satisfaction, marital status, etc. There are a total of 1471 individuals, all with complete data, represented in the set.

Age provides a basic demographic factor, often correlated with career stage and stability. Business travel indicates whether an employee travels frequently for work, which may impact work-life balance and job satisfaction. Hourly rate is a financial factor, reflecting the compensation offered, which could affect an employee's retention. Similarly, job satisfaction and

work-life balance are critical non-monetary aspects of employee experience, influencing whether they stay with the company. Job satisfaction and work-life balance are rated on a scale of 1-4, with four being greatest satisfaction and greatest work-life balance. Gender, education (scale of 1-5, from below college to doctorate), and job level help capture diversity and career progression, which are important for understanding attrition in different employee groups. Job level is defined as the level of involvement in the workplace, ranked 1-5, from very low to very high. Years at the company and years since last promotion are particularly useful for assessing how long employees have been with the company and whether career growth influences their decision to leave. By combining these diverse data points, the model can identify patterns and correlations between employee characteristics and attrition risk.

There are several variables that are categorical instead of numerical, such as business travel and marital status, which are separated into frequently, rarely, or non-travel, and single, married, or divorced, respectively. Gender is male or female, and the education field has a variety of possibilities representing the individual's choice of study.

Note that one limitation of this dataset is that data points are not taken from real people, and certain points may be more or less likely to exist in a real world scenario. This fact is left out, with the knowledge that real world scenarios are likely to have non-uniform data, meaning two individuals with the same age may not have the same marital status. The randomness of the data is meant to reflect this fact. Another limitation is whether or not data is dependent on other values. Meaning whether or not years since last promotion is affected by years at the company is not considered, although in a real life scenario that would probably have a significant effect. Lastly, real world scenarios depend heavily on personal opinion, which varies greatly and is impossible to predict.

The biggest challenge with this dataset is that there is an imbalance over ‘yes’ and ‘no’ responses, which is typical of attrition datasets. We have approximately 83% ‘no’ and 17% ‘yes’, which makes the baseline accuracy (accuracy from blindly guessing ‘no’) 83%. This creates a big challenge when we train our models because we need to aim to perform better than blindly guessing. This is the main obstacle we need to tackle.

Methodology

The primary goal of this research is to predict employee attrition—whether an employee will quit their job—using machine learning models. To achieve this, we used a combination of logistic regression, random forest, and neural network models, applying them to a structured dataset of employee attributes. The methodology is designed to not only identify key drivers of attrition but also to evaluate the performance of different models in predicting employee turnover. Our code works in several python files, which will be explained in this section.

The approach starts with data preprocessing, where we first examine the dataset for any missing or invalid values. It turns out that four feature columns have constant values, so we dropped them. We then examined the coefficient of correlation between every two features, and dropped ones that are highly correlated to reduce redundancy in our data (we used a cut-off point of >0.7 in correlation).

After cleaning the data, one-hot encoding is applied to categorical variables like business travel and marital status to convert them into binary numerical formats suitable for machine learning models (code block 4). Additionally, numerical features such as job level and hourly rate are scaled using standard normalization techniques to ensure all features contribute equally to the model and prevent any one feature from disproportionately influencing the outcome due to differences in scale (code block 8). The mean of this result is 0, and the standard deviation 1. This is meant to help convergence and regulation processes. The data is then split into training and testing sets, with 80% allocated for training the models and 20% for evaluating model performance (code block 7).

Once the data is preprocessed, three models are used for prediction: logistic regression, random forest, and neural network. The logistic regression model is first trained on the data to establish a baseline, which we call simple_model (code block 9). This is a simple yet effective model for binary classification, where we aim to predict whether an employee will quit (1) or stay (0). The model is optimized using cross-validation and tested. Lasso regularization (code block 11) is then used to create a second model, and validated for accuracy, which is then fed to the next step to determine which model is the best fit, or highest accuracy (code block 12). Lasso regularization was implemented to mainly zero out unimportant features which helped to increase the accuracy.

The accuracy score of the two models are then compared in code block 13 and 14. There are two general parameters extracted: precision and recall, which measures how many predicted positives are actually positive, and how many actual positives are correctly predicted, respectively, which could also be seen as comparison for accuracy.

For the neural network, two main files are built: nn_main.py which establishes the bulk of the neural network computing, and parse.py, which is responsible for parsing the data to be used in nn_main.py. After the data is parsed by calling parse.py, a multi-layer perceptron (MLP) architecture is implemented (code block 17). The neural network model includes multiple layers with neurons that process the input data in a more complex manner, potentially capturing non-linear relationships between features. MLP makes and groups these neurons into perception layers, which are then stacked to create the network. We also apply SMOTE, synthetic minority oversampling technique, to address imbalance in the dataset (code block 16). A custom loss function, binary cross-entropy, is used to compute the model's error during training, and backpropagation is employed to adjust the model's weights based on this error. Finally, the model is evaluated, the AUC score taken, graphed (as ROC graph in code block 19, and the model is evaluated on the test set.

In the code implementation, both the logistic regression and the neural network model are trained using standard libraries from scikit-learn. We explored using Andrej Karpathy's micrograd engine as the foundation for neural networks, but training proved to be too time consuming. Hence we pivoted back to scikit-learn.

Random Forest is the model used in tandem with the training, and is an ensemble method that combines decision trees to make predictions. It is implemented in code block 8. It uses two key techniques; bootstrap aggregation and random feature selection, which enhance predictive accuracy and reduce overfitting compared to a single decision tree. This model trains each decision tree on a random subset of the training data, allowing each tree to learn different patterns. At each node in the tree, a random subset of features is considered for splitting, which reduces the correlation between trees. When a prediction is needed, each tree independently predicts the attrition result, and the final prediction is determined by a simple majority voting across all trees.

Each model's performance is assessed based on its accuracy, and the results are compared to identify the best-performing approach for predicting employee attrition. Through this methodology, we aim to understand how different factors contribute to attrition, while also evaluating the predictive power of three distinct modeling techniques - logistic regression, known for its simplicity and interpretability, the more complex neural network, which can capture intricate relationships in the data, and random forest, for its robustness and nonlinear relationship handling.

Results and Analysis

1. Logistic Regression Results

We evaluated the performance of two models — simple linear regression, Lasso-regularized regression — to predict classification outcomes. The simple linear regression model achieved an accuracy of 0.8810, serving as a baseline for comparison. The Lasso regression model, after hyperparameter tuning, achieved its best performance at index 9, yielding an accuracy of 0.8673. On a separate test dataset, the Lasso model achieved an improved accuracy of 0.8844, outperforming the simple linear regression (0.8810).

The Lasso model introduced regularization, reducing several feature weights to zero and selecting only the most influential features. Among the retained features, significant positive weights included 0.856, 0.459, and 0.447, while notable negative weights were -0.437, -0.357, and -0.331, indicating each feature's contribution to the prediction. This sparsity in the weight distribution highlighted Lasso's ability to reduce model complexity and improve interpretability.

Classification metrics further evaluated model performance. The AUC-ROC score of 0.7942 indicated a good ability to distinguish between the two classes. However, performance

varied across classes. For class 0, the model achieved high precision (0.91), recall (0.96), and F1-score (0.93), reflecting strong classification performance. In contrast, class 1 showed lower precision (0.59), recall (0.41), and F1-score (0.48), indicating challenges in identifying instances of this minority class. This discrepancy is likely due to class imbalance, with 255 samples in class 0 and only 39 in class 1.

2. Neural Network Results

Our neural network, implemented using sklearn's MLPClassifier, consisted of three hidden layers (512, 512, and 256 neurons) with logistic activation, early stopping, and a maximum of 3000 iterations to ensure convergence. The model's performance was evaluated using accuracy, classification reports, confusion matrices, and ROC AUC scores.

On the validation set, the neural network achieved an accuracy of 0.8299, lower than the baseline accuracy of 0.8673 from the simple linear regression model. However, given the dataset's class imbalance, accuracy alone is not a reliable metric. Instead, the ROC AUC score was prioritized, as it accounts for classification thresholds. The model achieved an ROC AUC score of 0.7962, indicating reasonable effectiveness in distinguishing between the "yes" and "no" classes.

The confusion matrix showed that the model correctly classified 223 out of 255 "no" instances (majority class) and 21 out of 39 "yes" instances (minority class). The classification report revealed that for class 0 ("no"), the model achieved a precision of 0.93, recall of 0.87, and F1-score of 0.90, demonstrating strong performance for the majority class. For class 1 ("yes"), the model had a precision of 0.40, recall of 0.54, and F1-score of 0.46, reflecting difficulty in identifying instances of the minority class. While SMOTE improved class 1 detection, class imbalance still affected overall performance.

Prioritizing the ROC AUC score (0.7962) over raw accuracy provided a more comprehensive assessment of the model's classification ability. Unlike accuracy, which is biased by the prevalence of "no" instances, the ROC AUC score captures performance across all classification thresholds, where 1.0 represents perfect classification and 0.5 indicates random guessing.

3. Random Forest Results

Our Random Forest model achieved a validation accuracy of 84.35%, outperforming the baseline accuracy. Due to the imbalanced dataset, accuracy alone was not sufficient, so the ROC AUC score of 0.8257 was prioritized, indicating the model's ability to distinguish between "yes" and "no" attrition.

The confusion matrix for the validation set revealed that the model correctly predicted 122 instances of "no" attrition (true negatives) and 2 instances of "yes" attrition (true positives), but misclassified 22 "yes" instances as "no" (false negatives) and 1 "no" instance as "yes" (false positive). The classification report showed that for employees who stayed (class 0), the model

achieved precision of 0.847, recall of 0.992, and F1-score of 0.913. For employees who left (class 1), precision was 0.667, recall was 0.083, and F1-score was 0.148, reflecting difficulties in identifying the minority class.

When tested on an independent set, the model achieved a test accuracy of 84.35%, and the ROC AUC score was 0.771, indicating good discriminatory ability. The confusion matrix for the test set showed similar results, with the model correctly predicting 121 "no" attrition instances and 3 "yes" instances, while misclassifying 21 "yes" instances and 2 "no" instances. The classification report for the test set showed precision of 0.852, recall of 0.984, and F1-score of 0.913 for employees who stayed, but precision of 0.600, recall of 0.125, and F1-score of 0.207 for employees who left.

The Random Forest model performed well at predicting employees who stayed but struggled to identify those who left. The recall for "yes" attrition was notably low (0.25 for validation, 0.08 for test), suggesting the need for improved handling of the class imbalance. Future work could focus on mitigating this imbalance through oversampling, undersampling, or cost-sensitive learning techniques.

On top of running the random forest model for fitting the data, we also computed the top features that contribute toward attrition data. The top three most significant features are Monthly Income, Total Working Years, and Age. The three least significant features are Standard Hours, Employee Count, and Job Role.

4. Comparative Analysis

The three models—Logistic Regression, Neural Network, and Random Forest—differ in performance and approach. Logistic Regression, after hyperparameter tuning and Lasso regularization, achieved the highest test accuracy (0.8844) and improved interpretability by selecting key features. However, it was affected by class imbalance, resulting in a lower F1-score for the minority class. The Neural Network performed well in ROC AUC (0.7962) but had a lower accuracy (0.8299) and struggled with predicting the minority class, despite using SMOTE. The Random Forest model showed a validation accuracy of 87.07%, excelling in predicting the majority class but underperforming with the minority class, reflected in a low recall for employees who left (0.25). While Logistic Regression was the most accurate and interpretable, the Random Forest model was more robust for majority class predictions, and the Neural Network had a solid ROC AUC but struggled with the minority class.

A reasonable interpretation we can draw from this result is that many features have a linear relationship with attrition rate. Logistic Regression inherently assumes a linear relationship between the input features and the log-odds of the outcome. The high accuracy and the model's ability to maintain interpretability even after applying Lasso regularization indicate that key features likely have a straightforward influence on the prediction. This linearity simplifies the decision boundary, making Logistic Regression a good fit for the data. However, its struggles with the minority class (as seen in the lower F1-score) also highlight limitations in fully

capturing more nuanced or non-linear patterns in the data. This performance suggests that while some aspects of employee attrition may follow a linear trend, other complex relationships, especially those related to minority class predictions, may require non-linear models to uncover.

Conclusion

This study aimed to predict employee attrition using machine learning models. The Logistic Regression model, with Lasso regularization, achieved the highest test accuracy (88.44%) but faced challenges with class imbalance, especially in predicting employees who left. The Neural Network showed strong performance in ROC AUC (0.7962) but had lower accuracy and difficulties with minority class detection. The Random Forest model performed well in predicting employees who stayed but struggled to identify those who left, as reflected in its low recall for the minority class.

These findings provide valuable insights into the factors influencing employee attrition and demonstrate the effectiveness of different machine learning models in predicting turnover. Key predictors of attrition identified in the models include demographic factors, job satisfaction, and career progression, which could inform HR strategies for retention. The results underscore the importance of addressing class imbalance in datasets to improve model performance, especially for predicting the minority class, which is often the primary focus in attrition prediction.

While the models showed promising results, several limitations must be considered. The dataset used in this study is fictional, which means it may not fully reflect the variability and complexities of real-world employee data. Factors such as personal opinion, organizational culture, and external economic conditions were not considered in the models, which could significantly affect employee decisions to leave or stay. Furthermore, the class imbalance in the dataset posed challenges for model performance, particularly in predicting the minority class. Future studies should focus on real-world datasets and explore techniques like oversampling, undersampling, or cost-sensitive learning to address these issues and improve predictive accuracy.

In conclusion, while logistic regression performed best in terms of accuracy and interpretability, each model presented strengths and weaknesses, highlighting the complexity of predicting employee attrition and the need for more sophisticated approaches to balance class distributions and capture nuanced relationships in the data.

References

Arthurtok. “Employee Attrition via Ensemble Tree-Based Methods.” *Kaggle*, Kaggle, 24 Sept. 2018,
www.kaggle.com/code/arthurtok/employee-attrition-via-ensemble-tree-based-methods/input.