Report Employee Attrition

Machine Learning Problem:
Classification - predict employee attrition.

Advised Key Performance Indicators:

- Accuracy: The proportion of correctly classified instances (employees who will leave or stay).
- F1-score: The harmonic mean of precision and recall, providing a balanced measure of both.
- AUC-ROC: The area under the receiver operating characteristic curve, indicating the model's ability to distinguish between leaving and staying employees.

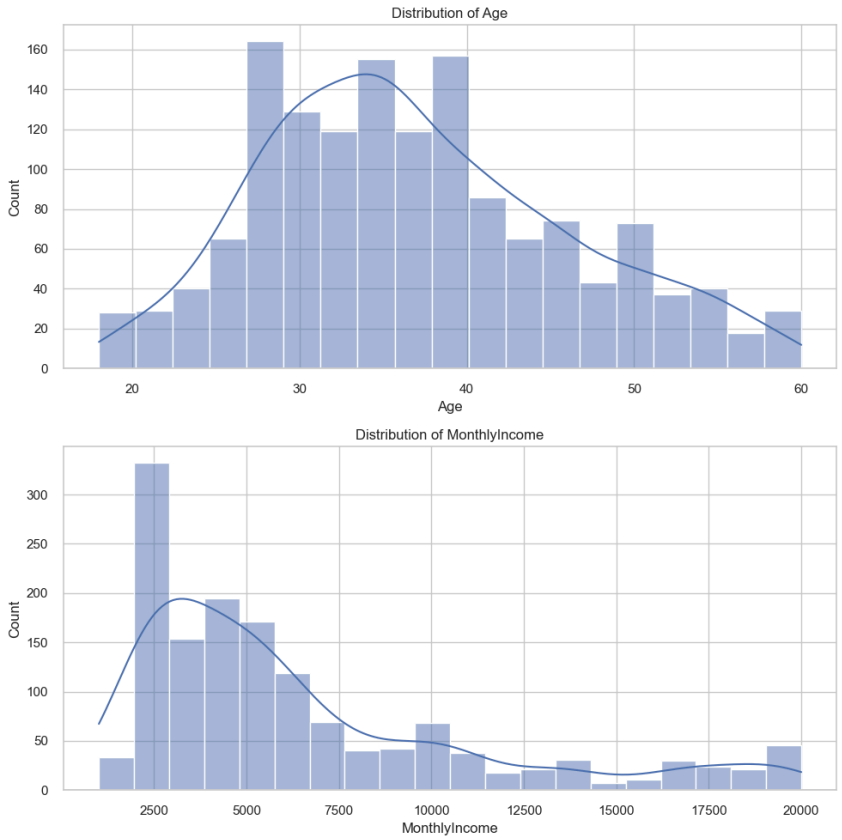
Advised Key control indicators:

- Data quality: The completeness and accuracy of the data in the `employee_attrition.csv` file.
- Feature relevance: The importance of each feature in the dataset in predicting employee attrition.
- Model complexity: The complexity of the chosen classification algorithm and its potential for overfitting.

Initial Insights:

- The `Age` column has a mean of 36.92 and a standard deviation of 9.14, indicating that the ages of the employees are relatively normally distributed.
- The `DailyRate` column has a mean of 802.49 and a standard deviation of 403.51, indicating a wide range of daily rates.
- The `MonthlyIncome` column has a mean of 6502.93 and a standard deviation of 4707.96, indicating a wide range of monthly incomes.

Distribution of Age and Monthly Income



Top correlations between features columns:

1. **JobLevel-MonthlyIncome = 0.950300**: This shows a very strong positive correlation, indicating that as the job level increases, the monthly income also tends to increase significantly.

2. **JobLevel-TotalWorkingYears = 0.782208**: This shows a strong positive correlation, suggesting that higher job levels are often associated with more total working years.

3. **PercentSalaryHike-PerformanceRating = 0.773550**: This also shows a strong positive correlation, implying that employees with higher performance ratings tend to receive higher percent salary hikes.

4. **MonthlyIncome-TotalWorkingYears = 0.772893**: This shows a strong positive correlation, indicating that employees with more total working years tend to have a higher monthly income.

5. **YearsAtCompany-YearsWithCurrManager = 0.769212**: This shows a strong positive correlation, suggesting that employees who have been at the company longer also tend to have been with their current manager for longer.

6. **YearsAtCompany-YearsInCurrentRole = 0.758754**: This shows a strong positive correlation, indicating that employees who have been at the company longer also tend to have been in their current role for longer.

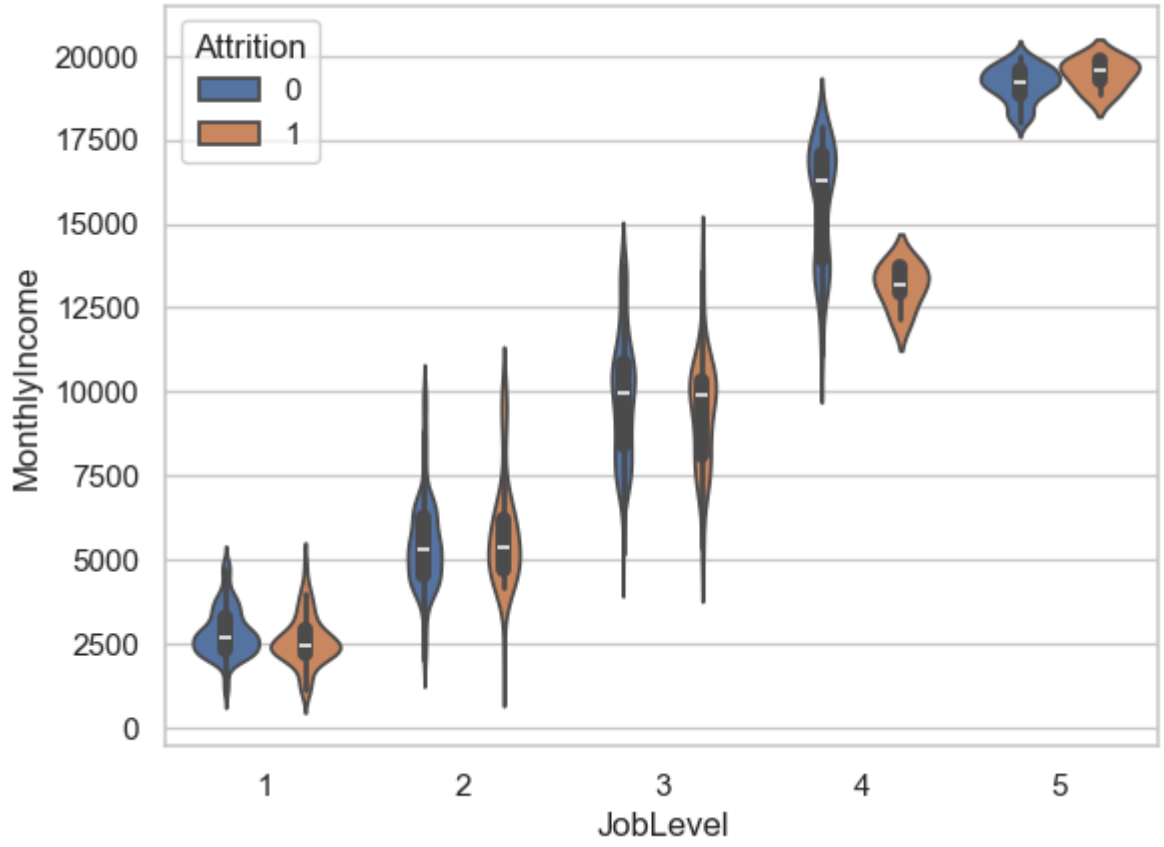
7. **YearsInCurrentRole-YearsWithCurrManager = 0.714365**: This shows a strong positive correlation, suggesting that employees who have been in their current role for longer also tend to have been with their current manager for longer.

8. **Age-TotalWorkingYears = 0.680381**: This shows a moderate to strong positive correlation, indicating that older employees tend to have more total working years.

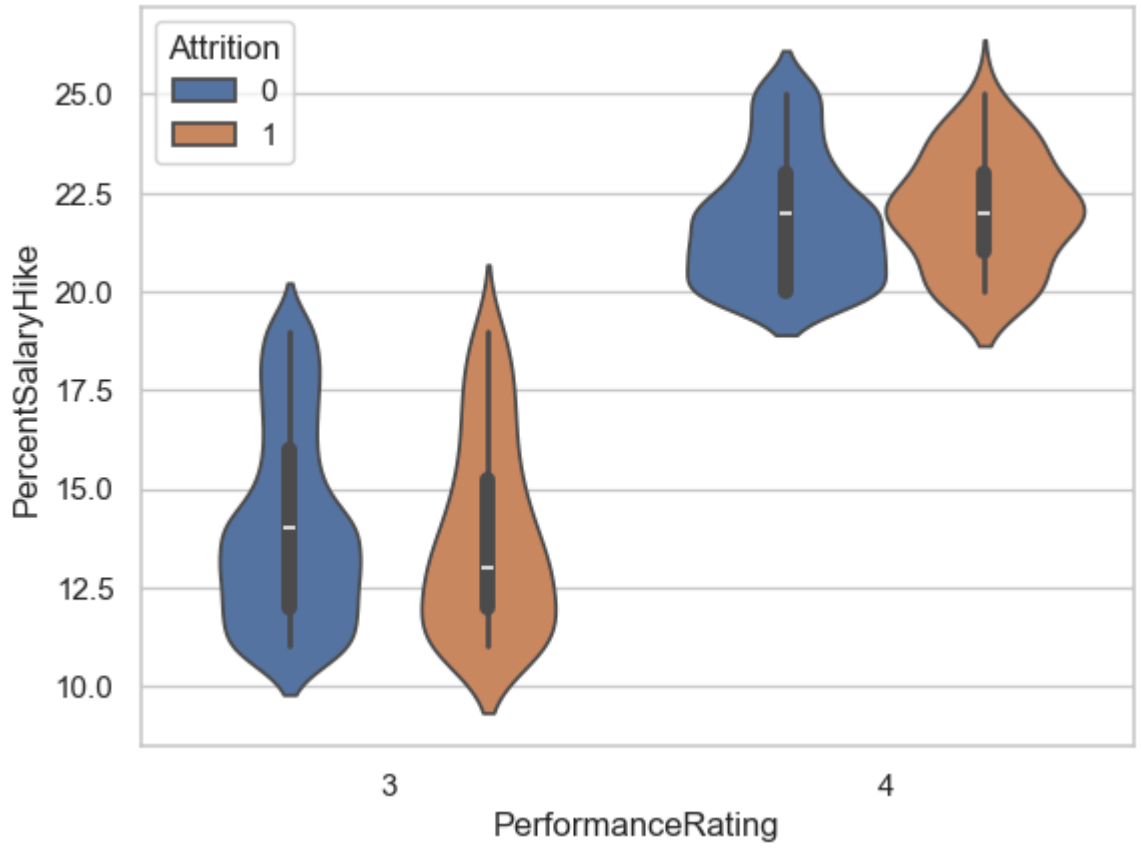
9. **TotalWorkingYears-YearsAtCompany = 0.628133**: This shows a moderate positive correlation, suggesting that employees who have more total working years also tend to have been at the company for longer.

10. **YearsAtCompany-YearsSinceLastPromotion = 0.618409**: This shows a moderate positive correlation, indicating that employees who have been at the company longer also tend to have gone longer since their last promotion.

Visualization with violin (**JobLevel-MonthlyIncome)**

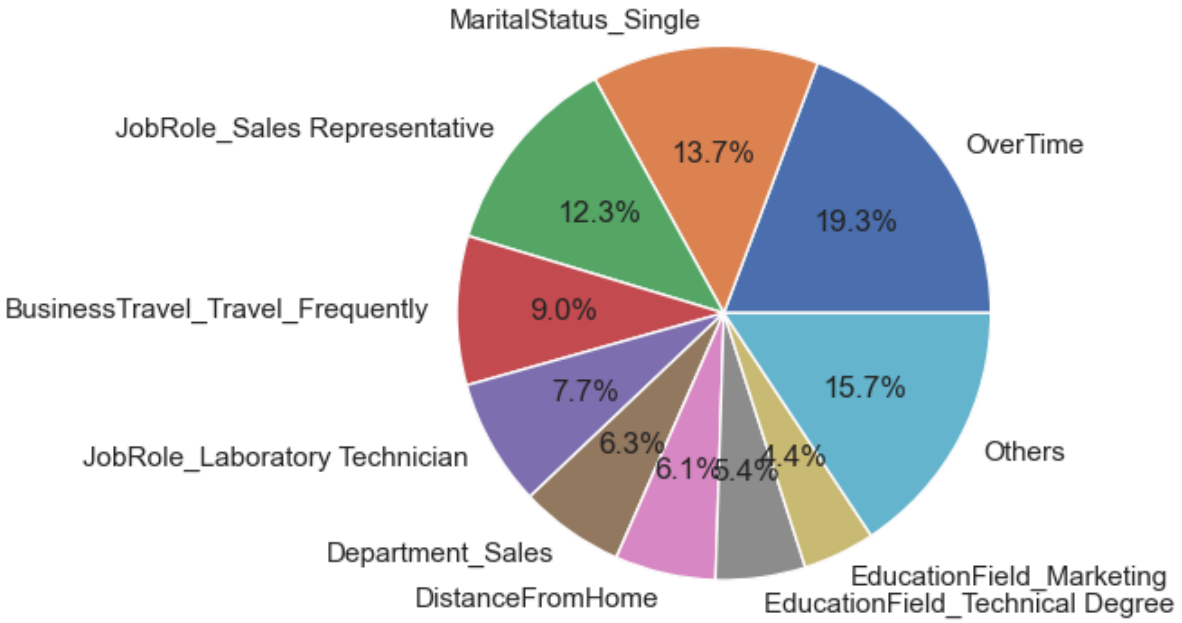


Visualization with violin (**PercentSalaryHike-PerformanceRating)**



Highest Correlations between features and attrition:
1. `OverTime` (0.246118) - This suggests that employees who work overtime are more likely to attrite.
2. `TotalWorkingYears` (0.171063) - This implies that employees with more years of service are more likely to leave the company.
3. `JobLevel` (0.169105) - This indicates that employees at higher job levels are more likely to attrite.
4. `MaritalStatus` (0.162070) - This suggests that marital status may play a role in employee attrition.
5. `YearsInCurrentRole` (0.160545) - This implies that employees who have been in their current role for a longer period are more likely to leave.

Pie visualization representing the highest correlations with encoded variables:



Machine Learning Model Adviced to use:
Random Forest Classifier

Explanation and Alternatives for the Machine Learning Model:
Random Forest Classifier is a robust and widely used algorithm for classification tasks. It can effectively handle correlated features, non-linear relationships between features and the target variable, and can handle a large number of features. Although it can handle class imbalance to some extent, it can be combined with techniques like oversampling the minority class, undersampling the majority class, or using class weights to handle class imbalance. Alternative models that can be considered are Gradient Boosting Classifier, Support Vector Machines, and K-Nearest Neighbors.

Transformations Made to the data:
Separated the data into features (X) and target (y), then split the dataset into training and test sets.
Splitting:
Split the dataset into training and test sets with a test size of 0.2, and saved the data to X_train.csv, y_train.csv, X_test.csv, and y_test.csv

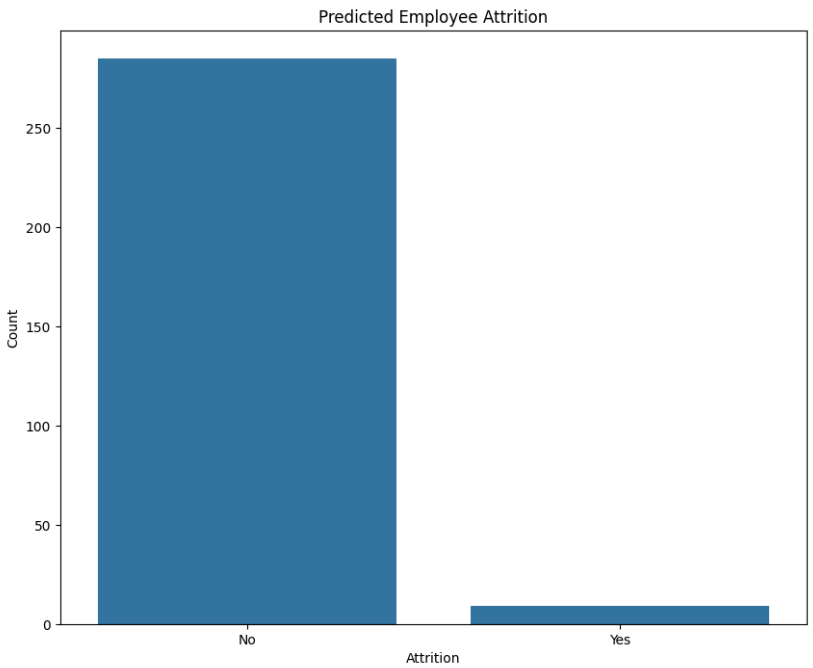
Results of the evaluations:
The accuracy of the model is 0.8775, which indicates that the model is able to correctly classify approximately 87.75% of the samples.

The classification report provides more insights into the model's performance:

- The **precision** for the "**No**" class is 0.88, indicating that 88% of the samples predicted as "No" are actually "**No**".
- The **recall** for the "**No**" class is 1.00, indicating that the model is able to detect all the "No" samples.
- The **F1-score** for the "**No**" class is 0.93, which is a harmonic mean of precision and recall.
- The **precision** for the "**Yes**" class is 0.80, indicating that 80% of the samples predicted as "Yes" are actually "**Yes**".
- The **recall** for the "**Yes**" class is 0.10, indicating that the model is only able to detect 10% of the "Yes" samples.
- The **F1-score** for the "**Yes**" class is 0.18, which is relatively low.

The **confusion matrix** provides a more detailed view of the model's performance:

* The model correctly classified 254 "No" samples and 4 "Yes" samples.
* The model misclassified 1 "No" sample as "Yes" and 35 "Yes" samples as "No".



Feature importance for the trained model: