OPTIMIZING EMPLOYEE ATTRITION PREDICTION THROUGH ADVANCED ANALYTICS

Objectives

The overarching goal of this project was to employ the HR Analytics data set to uncover the determinants of employee attrition within a company. Through meticulous data processing, feature engineering, and the application of multiple predictive modeling techniques, the project aimed to:

Predict Employee Attrition: Utilize advanced machine learning algorithms, including Logistic Regression, Random Forest, and XGBoost, to create robust models capable of predicting the likelihood of employee turnover. This objective was pursued with the intention of enabling preemptive interventions for at-risk employees.

Identify Key Attrition Factors: Through exploratory data analysis, correlation matrix inspection, and the interpretation of model feature importances, pinpoint the most influential factors contributing to employee attrition. Special attention was given to variables such as OverTime, JobLevel, TotalWorkingYears, and StockOptionLevel, which were hypothesized and later confirmed to have significant impacts on attrition rates.

Optimize Model Performance: Methodically tune hyperparameters for each model to optimize their performance. This included experimenting with different split ratios to balance training and testing data, regularizing logistic regression to prevent overfitting, and adjusting parameters for the Random Forest and XGBoost models to enhance their predictive accuracy and generalization to unseen data.

Cross-Model Comparisons: Compare the effectiveness of different modeling approaches in predicting attrition, utilizing metrics such as accuracy, precision, recall, and the F1 score. This comparison was aimed at identifying the most suitable model or models for deployment in real-world HR analytics applications.

Provide Actionable Insights: Translate the analytical findings into actionable recommendations for HR practices, including targeted retention strategies, tailored employee engagement programs, and personalized career development plans for employees identified as high risk for attrition.

By fulfilling these objectives, the project contributes valuable insights into the mechanics of employee turnover, offering a data-driven foundation upon which companies can build more cohesive, enduring, and satisfying work environments for their employees.

Data Preparation

Initial Assessment: The dataset was first examined for completeness, identifying any missing values or anomalies. It was found that the 'YearsWithCurrManager' column had

some missing values, which was handled by replacing the null values with the value of ‘YearSinceLastPromotion’, this when not ideal provides a reasonable estimation for the field ‘YearsWithCurrManager’.

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   EmpID                     1480 non-null   object
 1   Age                       1480 non-null   int64
 2   AgeGroup                  1480 non-null   object
 3   Attrition                 1480 non-null   object
 4   BusinessTravel            1480 non-null   object
 5   DailyRate                 1480 non-null   int64
 6   Department                1480 non-null   object
 7   DistanceFromHome          1480 non-null   int64
 8   Education                 1480 non-null   int64
 9   EducationField            1480 non-null   object
 10  EmployeeCount             1480 non-null   int64
 11  EmployeeNumber            1480 non-null   int64
 12  EnvironmentSatisfaction   1480 non-null   int64
 13  Gender                    1480 non-null   object
 14  HourlyRate                1480 non-null   int64
 15  JobInvolvement            1480 non-null   int64
 16  JobLevel                  1480 non-null   int64
 17  JobRole                   1480 non-null   object
 18  JobSatisfaction           1480 non-null   int64
 19  MaritalStatus             1480 non-null   object
 20  MonthlyIncome             1480 non-null   int64
 21  SalarySlab                1480 non-null   object
 22  MonthlyRate               1480 non-null   int64
 23  NumCompaniesWorked        1480 non-null   int64
 24  Over18                    1480 non-null   object
 25  OverTime                  1480 non-null   object
 26  PercentSalaryHike         1480 non-null   int64
 27  PerformanceRating         1480 non-null   int64
 28  RelationshipSatisfaction  1480 non-null   int64
 29  StandardHours             1480 non-null   int64
 30  StockOptionLevel          1480 non-null   int64
 31  TotalWorkingYears         1480 non-null   int64
 32  TrainingTimesLastYear     1480 non-null   int64
 33  WorkLifeBalance           1480 non-null   int64
 34  YearsAtCompany            1480 non-null   int64
 35  YearsInCurrentRole        1480 non-null   int64
 36  YearsSinceLastPromotion   1480 non-null   int64
 37  YearsWithCurrManager      1423 non-null   float64
dtypes: float64(1), int64(25), object(12)
memory usage: 439.5+ KB
```

Preliminary Feature Selection: The dataset includes a wide range of features, some of which are highly relevant to the analysis, while others may not contribute significantly to the outcomes of interest. Features were evaluated for their importance and relevance to

the analysis goals. Unnecessary columns were identified and removed to streamline the dataset for analysis.

Feature	Reason for removal
YearsInCurrentRole	Same as YearsSinceLastPromotion
Age	AgeGroup will be considered
MonthlyIncome	HourlyRate, StockOptionLevel is already considered
SalarySlab	HourlyRate, StockOptionLevel is already considered
DailyRate	HourlyRate, StockOptionLevel is already considered
MonthlyIncome	HourlyRate, StockOptionLevel is already considered
StandardHours	Constant for all entries
EmployeeCount	Constant for all entries
EmployeeNumber	Not significant for analysis
Over18	Constant for all entries
EmpID	Not significant for analysis

Data Transformation and Encoding: Several columns in the dataset contain categorical data that need to be encoded into a numerical format suitable for machine learning algorithms. For instance, 'BusinessTravel', 'AgeGroup' are transformed using label encoding, considering their ordinal nature while 'OverTime', 'Attrition', 'Gender', 'Department', 'EducationField', 'JobRole', 'MaritalStatus' were encoded one-hot encoding.

Feature Engineering: 'TotalWorkingYears' was binned into categories such as '0-5', '6-10',... to simplify its distribution and enhance its predictive power regarding attrition.

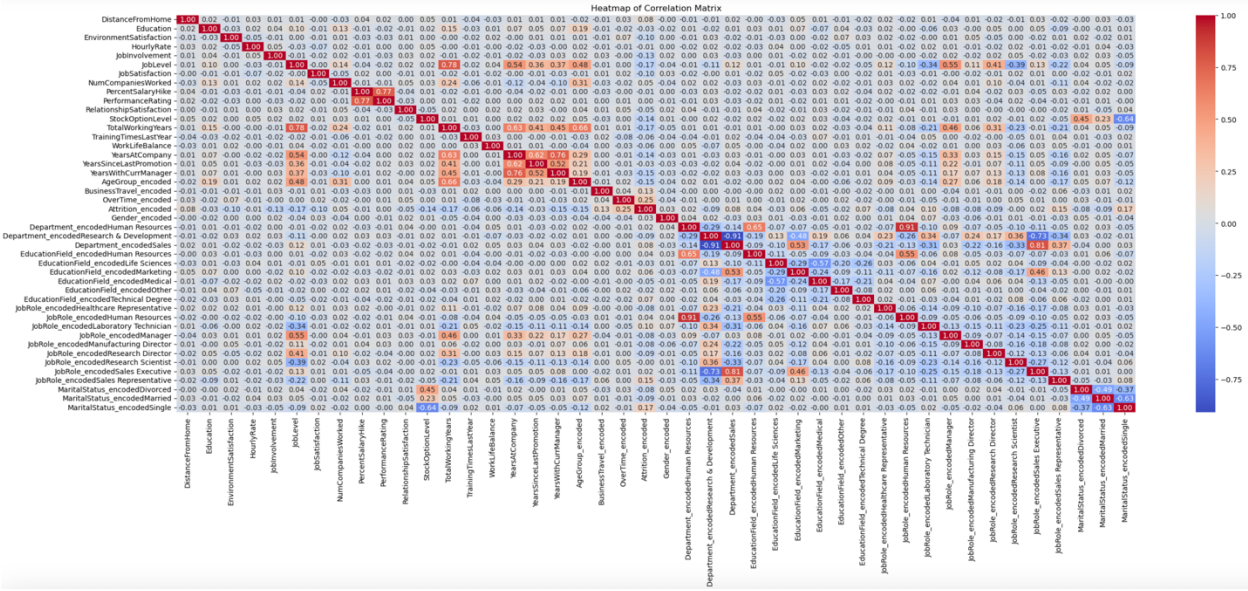
Normalization and Scaling: All Features were scaled to ensure that no single feature dominates the model due to its scale since models like logistic regression can be sensitive to the scale of the data MinMaxScaler() was used to ensure all datapoints was within 0-1.

Correlation Matrix Analysis

The correlation matrix served as an incipient tool for visualizing the linear relationships among the numerical variables within the HR dataset. Through the application of the corr() function to the processed data, a comprehensive matrix was generated, encapsulating the correlations between variables.

Visualization and Interpretation

The visualization of this matrix was achieved using a heatmap, facilitated by the seaborn library. This color-coded heatmap not only enhanced interpretability but also allowed for the immediate identification of variables that exhibit strong positive or negative correlations. Variables closely correlated with each other were highlighted, indicating potential predictors of attrition that warrant further examination.

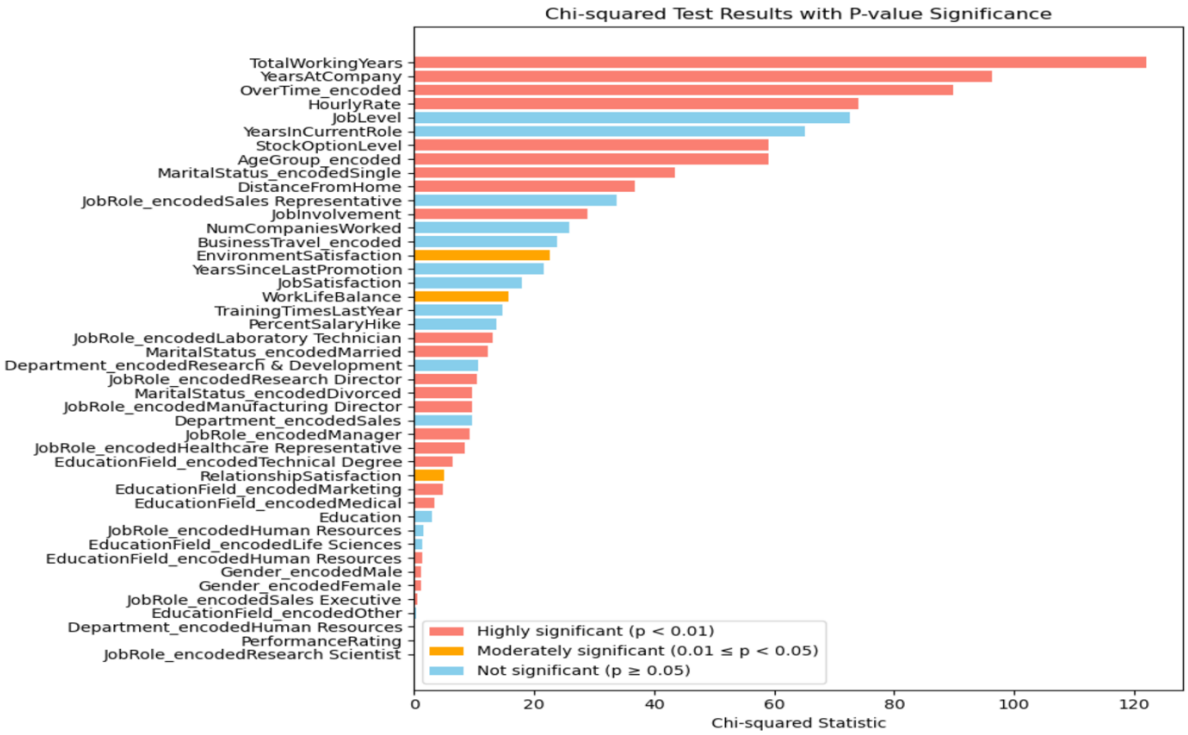


Chi-squared Test Application

Concurrently, Chi-squared tests were conducted to evaluate the association between categorical variables and employee attrition. This statistical test is crucial for identifying significant differences in the distribution of categorical variables, thereby shedding light on potential factors influencing attrition.

Visualization and Interpretation

The analysis involved creating contingency tables for categorical variables against 'Attrition_encoded' and calculating the Chi-squared statistic and p-values. The findings were systematically organized, enabling a clear distinction between variables based on their level of significance in relation to attrition. The visualization of these results through a color-coded bar chart further facilitated an intuitive understanding of the statistical significance of these relationships.



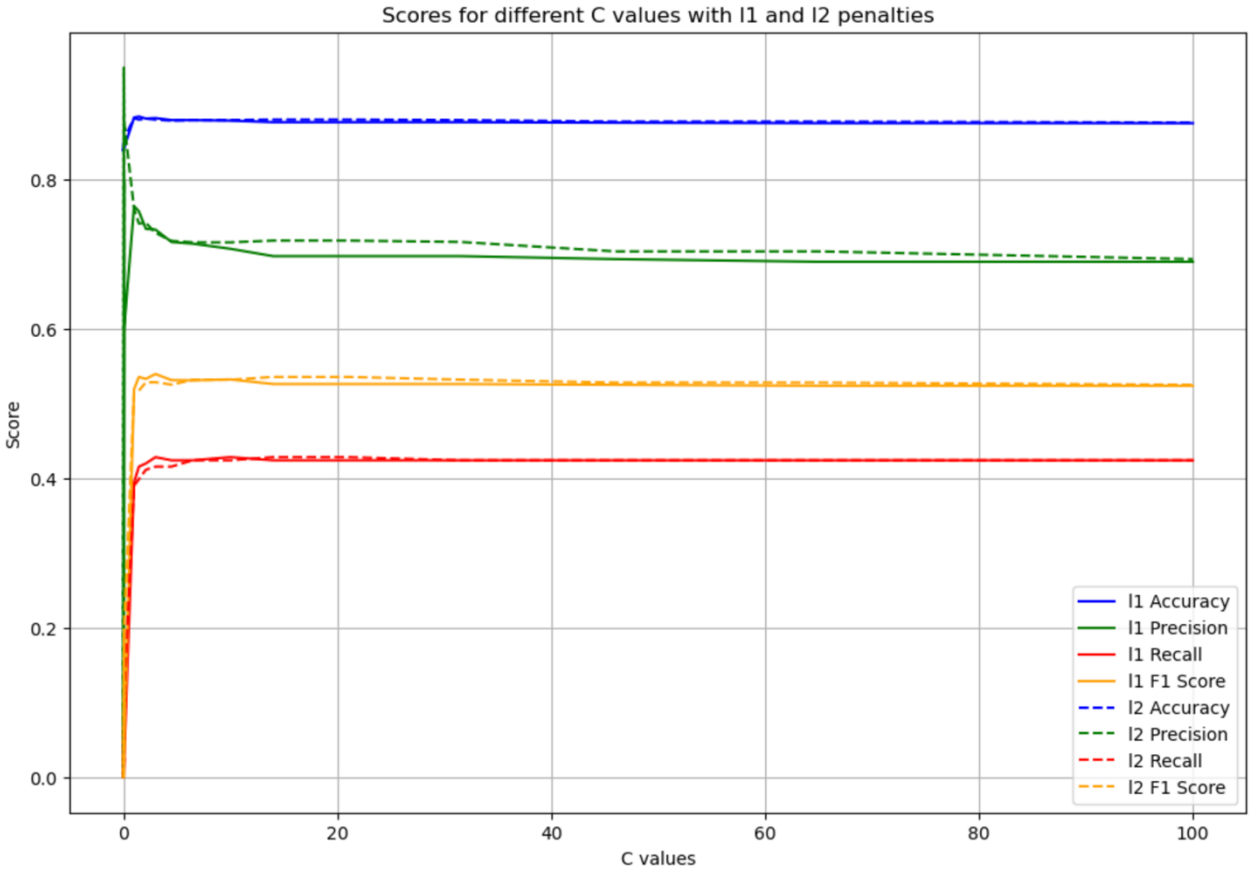
Logistic Regression Modeling

Model Training and Testing: The dataset was split into training and testing sets to evaluate the model's performance. Different split ratios were experimented with (e.g., 70-30, 80-20) to find the optimal balance between training and validation data. The logistic regression model was then trained on the training set, which involves fitting the model to the data by adjusting the coefficients to minimize the difference between the predicted and actual values of the target variable.

Cross-Validation: Stratified K-Fold cross-validation was utilized to assess the model's robustness and generalizability. This technique divides the data into K folds, ensuring that each fold is a good representative of the whole. The model is trained on K-1 folds and validated on the remaining fold, and this process is repeated K times. It helps in mitigating the variance and ensures that the model's performance is not dependent on a particular division of the train and test sets.

Performance Metrics: The performance of the logistic regression model was evaluated using various metrics, including accuracy, precision, recall, and the F1 score. These metrics provide a comprehensive view of the model's predictive capabilities, highlighting its strengths and areas for improvement. Accuracy measures the proportion of total correct predictions, precision measures the accuracy of positive predictions, recall (sensitivity) measures the ability of the model to find all the relevant cases, and the F1 score provides a balance between precision and recall.

Hyperparameter Tuning: The logistic regression model's hyperparameters, such as the regularization strength (C) and the type of penalty (L1 or L2 regularization), were fine-tuned to optimize the model's performance. This involves running the model multiple times with different hyperparameters to find the combination that yields the best performance based on cross-validated metrics.



Effect of Regularization Strength (C): For both 'l1' and 'l2' penalties, there are instances where very low values of C (e.g., 0.010, 0.015, 0.020, 0.030, 0.045) lead to zero precision, recall, and F1 score. This suggests that the models with these low values of C might be underfitting or failing to correctly classify positive instances.

As C increases, the performance metrics generally improve for both penalties, reaching peak values at certain points before plateauing or slightly declining. This indicates that moderate regularization strength tends to yield better performance than excessively weak or strong regularization.

The highest values of precision, recall, and F1 score across both penalty types are observed at intermediate values of C, such as 1.000 and 1.450 for 'l1', and 0.065 and 0.068 for 'l2'. This suggests that these regularization strengths strike a good balance between bias and variance, leading to optimal model performance.

Performance Discrepancies Between Penalty Types: Overall, the 'l2' penalty tends to yield higher precision, recall, and F1 score values compared to the 'l1' penalty across

different values of C. This suggests that the 'l2' penalty might be more effective in capturing the underlying patterns in the data.
However, there are instances where the 'l1' penalty achieves comparable or slightly higher performance than the 'l2' penalty, particularly at higher values of C.

Feature Importance Extraction: A logistic regression model with L2 regularization was fitted using the provided dataset. The regularization strength (C) was set to 1, and the solver used was 'liblinear'. The model was trained on the training data (X_train and y_train). After fitting the model, the coefficients associated with each feature were retrieved using the ridge_model.coef_ attribute.

Random Forest Classifier

Data Splitting: The dataset was split into training and testing sets using the train_test_split function from scikit-learn. The test set size was set to 20% of the total dataset, and stratification was applied to ensure balanced class distribution in both sets.

Random Forest Tuning: GridSearchCV was utilized to tune hyperparameters for the Random Forest classifier. Various combinations of hyperparameters including the number of estimators, maximum depth, minimum samples split, and minimum samples leaf were explored.

The metric for optimization was set to F1 score, a harmonic mean of precision and recall, which is suitable for imbalanced classification tasks.
Model Fitting and Prediction:

The Random Forest model was fitted on the training data using the best parameters obtained from grid search.

Predictions were made on the test data, and a classification report was generated to evaluate model performance, including precision, recall, F1 score, and support for each class.

Feature Importance Extraction: Feature importance were extracted from the trained Random Forest model using the feature_importance attribute.

A DataFrame named features_df was created to organize the features and their corresponding importance for visualization.

Sorting and Scaling Feature Importance: The DataFrame of feature importance was sorted based on importance values in descending order to identify the most important features.

MinMaxScaler was applied to scale the importance to the range [0, 1], facilitating comparison of relative feature importance.

Feature Importance Extraction: Feature importances were extracted from the Random Forest, XGBoost, and Logistic Regression models. Each model provided a ranking of features based on their importance scores.

The importance scores were scaled using MinMaxScaler to ensure comparability across models.

Merging Feature Importance DataFrames: The feature importance DataFrames from Random Forest, XGBoost, and Logistic Regression were merged based on the feature names.

The resulting merged DataFrame contained columns representing the importance scores from each model for each feature.

Comparative Analysis: The merged DataFrame facilitated the comparison of feature importance scores across different models.

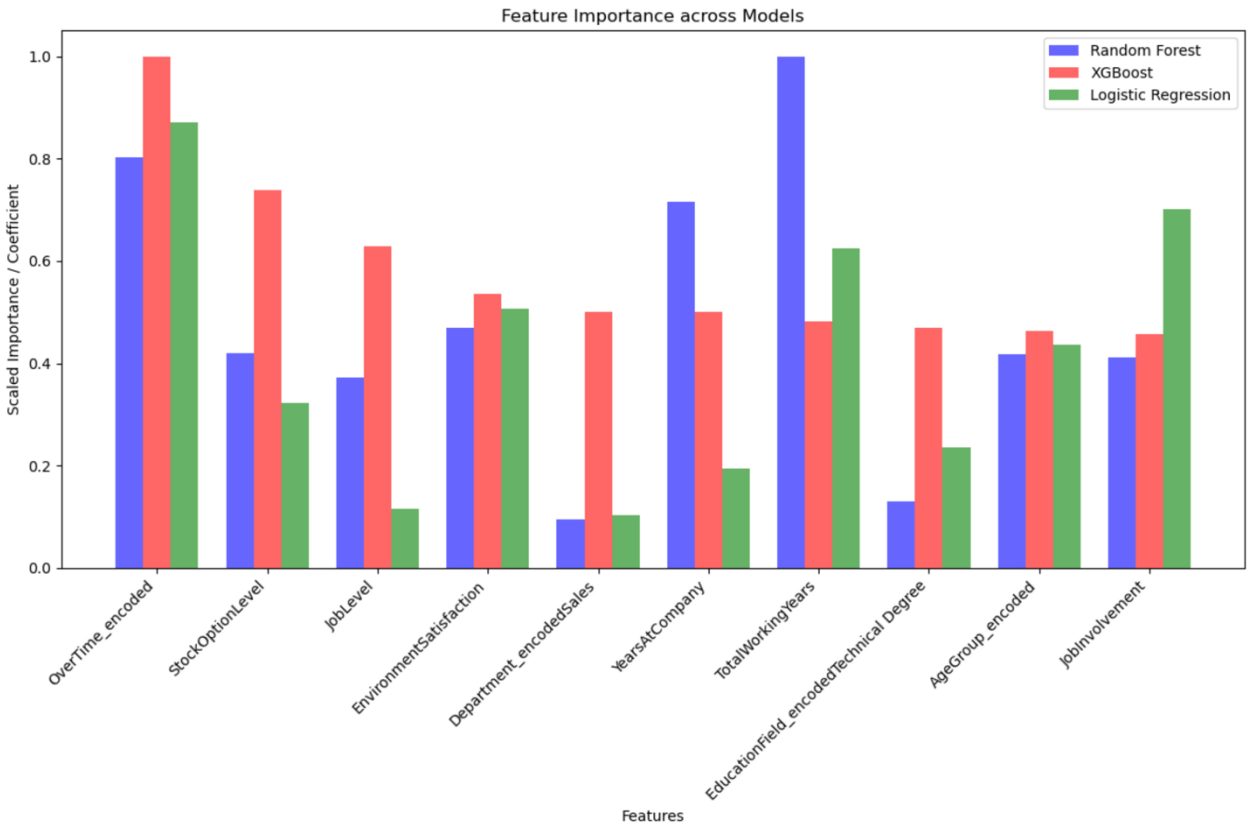
Features were sorted based on their importance scores from XGBoost in descending order to identify the most influential predictors.

Comparative Analysis of Feature Importance across Models

Feature Importance Extraction: Feature importances were extracted from the Random Forest, XGBoost, and Logistic Regression models. The importance scores were scaled to ensure comparability across models.

Top N Features Selection: The top N important features were selected from each model based on their scaled importance or coefficients. These features were chosen to represent the most influential predictors in each model.

Plotting Feature Importance: A bar plot was created to visualize the importance of top features across different models. Each model's importance scores or coefficients were represented by distinct colors in the plot.



Consistency in Important Features: Some features may consistently appear as important across multiple models, indicating their robust predictive power.

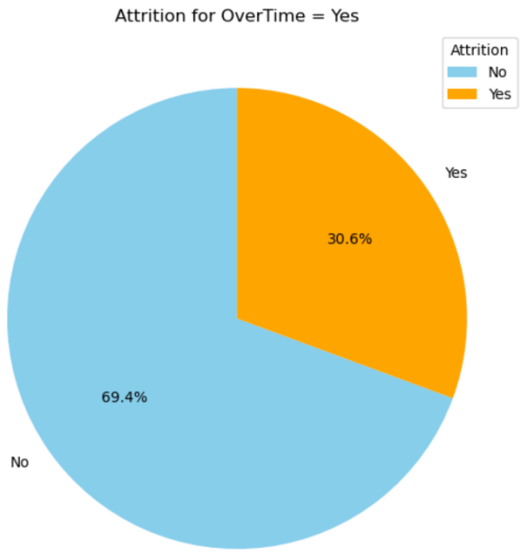
Discrepancies in Importance Rankings: Differences in feature importance rankings across models may highlight variations in model assumptions, algorithms, or data characteristics.

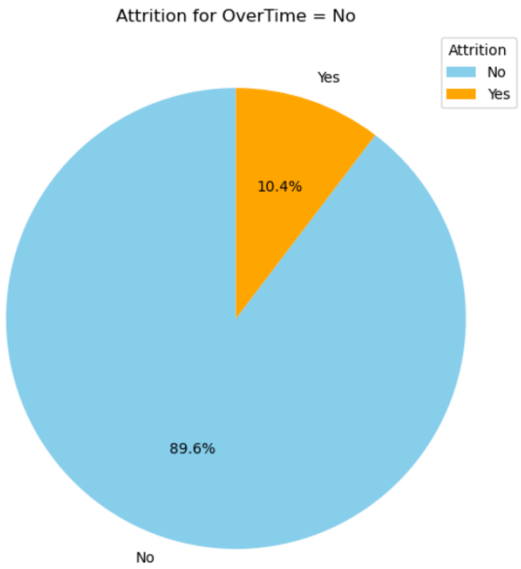
Relative Importance of Features: The height of the bars in the plot represents the relative importance or coefficients of features in each model. Features with taller bars exert a greater influence on model predictions.

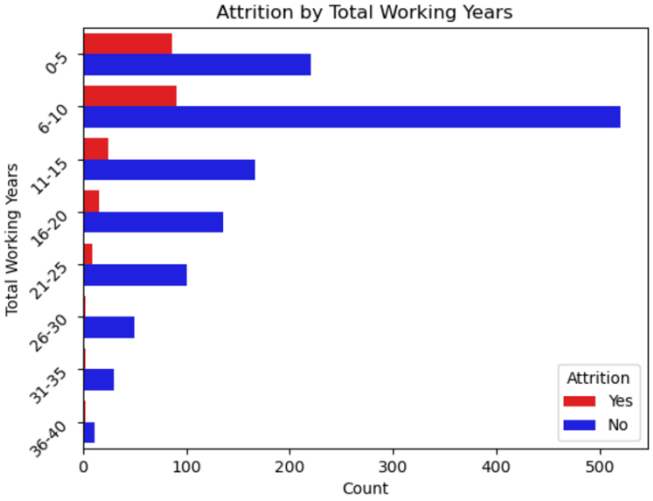
Visualizations:

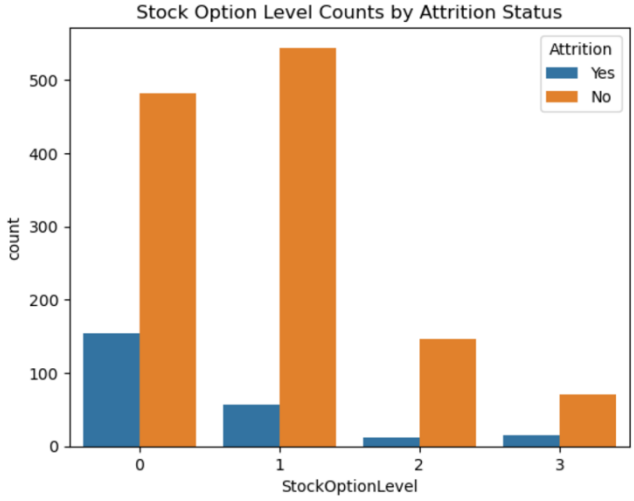
Once important features are identified, visualizations of attrition for various features were plot.

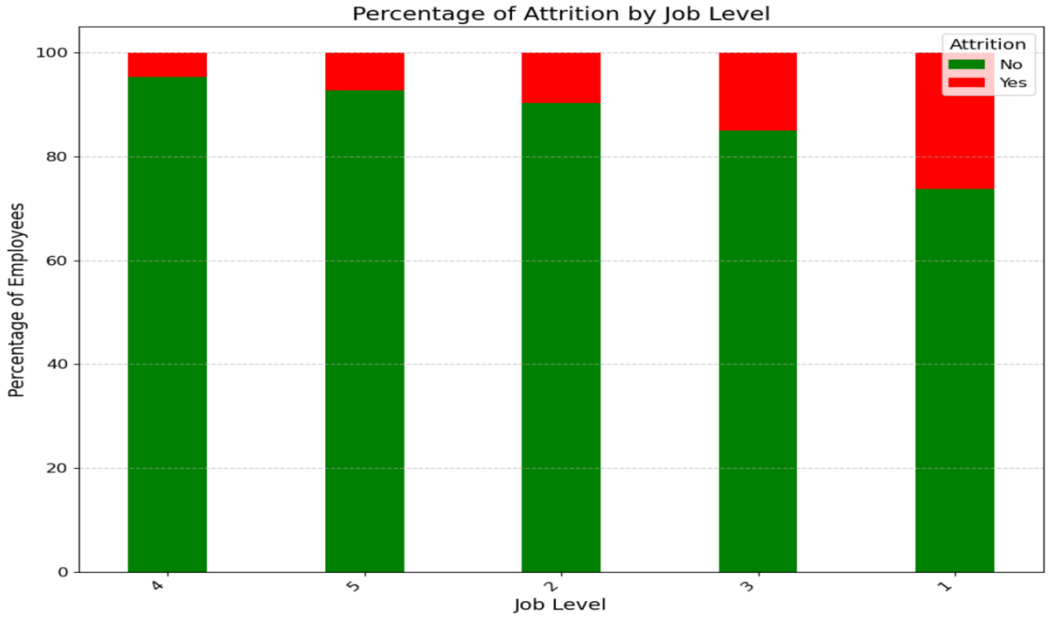
Features like Overtime shows significantly high values of attrition when an employee worked overtime, attrition reduced consistently with increase in total working years, stock option level and job level.


Attrition for OverTime = Yes
Attrition
No
Yes
30.6%
69.4%
Yes
No


Attrition for OverTime = No
Attrition
No
Yes
Yes
10.4%
89.6%
No


Attrition by Total Working Years
Total Working Years
0-5
6-10
11-15
16-20
21-25
26-30
31-35
36-40
0
100
200
300
400
500
Count
Attrition
Yes
No


Stock Option Level Counts by Attrition Status
count
0
100
200
300
400
500
0
1
2
3
StockOptionLevel
Attrition
Yes
No


Percentage of Attrition by Job Level
Percentage of Employees
0
20
40
60
80
100
4
5
2
3
1
Job Level
Attrition
No
Yes

Future steps and Conclusion

Further Investigation: It is recommended to further investigate discrepancies in feature importance rankings to gain deeper insights into model behavior and guide feature engineering or model selection processes.

Continuous Monitoring: Continuous monitoring and validation of feature importance can help ensure model robustness and reliability in real-world applications.
Exploration of Ensemble Techniques:

Future research could explore ensemble techniques or meta-learning approaches to combine insights from multiple models and improve overall predictive performance.

Conclusion: the comprehensive analysis of feature importance provided valuable insights into the factors influencing attrition and highlighted variations in model behavior. By leveraging multiple machine learning algorithms and visualization techniques, we have gained a deeper understanding of the relative importance of features and identified the best predictors of attrition. This knowledge can inform strategic decision-making processes aimed at mitigating attrition and improving organizational outcomes.

Resources:
Code: https://github.com/PranavKuramkoteSudhir/OPTIMIZING-EMPLOYEE-ATTRITION-PREDICTION-THROUGH-ADVANCED-ANALYTICS
Data source: https://www.kaggle.com/datasets/rishikeshkonapure/hr-analytics-prediction