CP-3403

Data Mining Project

Group 21

Credit Member Statement

This form recognises individual member contributions to the project, aiming to reduce disputes and encourage effective collaboration. The following contributions were agreed upon by all group members:

Name 1: Sai Ohm Saie Paine

- Data preprocessing, cleaning, and visualization – 30%
- Helped with dataset preparation and exploratory data analysis – 30%
- Built and evaluated machine learning models – 33%
- Random Forest model – 100% (Model ownership)
- Assisted in model development and interpretation – 30%
- Wrote and formatted sections of the report – 30%

Name 2: Nang Lao Pha

- Data preprocessing, cleaning, and visualization – 30%
- Helped with dataset preparation and exploratory data analysis – 30%
- Built and evaluated machine learning models – 33%
- Decision Tree model – 100% (Model ownership)
- Assisted in model development and interpretation – 25%
- Wrote and formatted sections of the report – 45%

Name 3: Hlyan Wint Aung

- Data preprocessing, cleaning, and visualization – 40%
- Helped with dataset preparation and exploratory data analysis – 40%
- Built and evaluated machine learning models – 33%
- KMeans Clustering model – 100% (Model ownership)
- Assisted in model development and interpretation – 45%
- Wrote and formatted sections of the report – 25%

Contents

Abstract

Employee attrition is a significant issue for organizations across most industries globally in terms of increased recruitment and training costs, loss of intellectual capital, and decreased efficiency in the operation. The project will address the issues by making use of employee turnover prediction using a dataset provided by IPM Company, a leader in the area of workforce management. The dataset includes various employee variables such as job satisfaction, environmental satisfaction, years at the company, and other significant variables that are likely to be the drivers of attrition. Applying machine learning algorithms of Decision Trees, Random Forests, and K-Means, the project aims to uncover latent patterns and identify predictors of employee turnover. While the dataset is from IPM Company, the approach and findings of this study are designed to be replicable and applicable to companies of various industries, sizes, and types. Its objective is to develop a predictive model that will not only help IPM Company improve its staff retention but also provide insightful information that can be applied by other firms to optimize their workforce management. With anticipation from this model of forecasting, organizations can shape their human resource efforts, reduce costs associated with uncontrolled turnover, and increase overall organizational stability and worker participation.

Introduction

Turnover is a significant problem for companies in almost all sectors, since it causes recruitment and training costs to increase, not to mention potential disruptions to operations. Being able to predict which employees are exiting—and why—can help companies to fight turnover successfully. Here, we take a dataset from IPM Company, one of the leading workforce management organizations, and use it to predict employee attrition with the assistance of machine learning techniques. This dataset includes variables such as job satisfaction, years of company tenure, satisfaction with work environment, and some other variables that might influence an employee to either stay or leave. While the data is from IPM Company, the goal of this project is to create a model and process not only

beneficial to IPM Company but beneficial to other companies and organizations in numerous industries.

The project shall leverage the capabilities of machine learning to develop a predictive model that can forecast the attrition of employees based on the range of attributes present in the data. In particular, the project employs a combination of supervised learning methods such as, Decision Trees, Random Forests, and K-Means to identify patterns and relationships that can be extended to a larger population of organizations. The resultant model will enable firms to predict employee turnover, identify the key drivers of attrition, and implement more targeted retention strategies. By pre-processing the data, missing value imputation, and feature engineering, this project will explore how the variables correlate with one another and influence an employee's likelihood to leave.

This project has been designed to demonstrate the application of machine learning to improve workforce management, allowing businesses to predictably stem turnover and improve worker satisfaction. Data based on analysis will help organizations in any industry better understand drivers of attrition and take well-informed steps to improve their HR policies, hence better organizational stability and reduced hiring and training costs of new employees.

Business Scenario

Retention of employees is the top priority for all organizations as high turnover is expected to be expensive, impact productivity, and negatively affect organizational performance. New employee recruitment and induction are expensive, and successive departures are expected to result in disruptions in day-to-day work, reduced team morale, and loss of corporate knowledge. As IPM Company faces all of these, the solution developed during this project has been envisioned in a manner where it could be scalable and would be implementable by any organizations of all business types and size.

The intention of this project is to utilize machine learning algorithms to predict attrition from the employees and identify what factors are leading to turnover. Drawing on job satisfaction, work environment satisfaction, and years of service information, the project will identify the most important trends and patterns that indicate whether an employee is going to leave. These results are designed to help firms, like IPM Company but not

exclusively it, make their retention strategies more effective by directing attention to the specific drivers of turnover, whether they are career development, work-life balance, compensation, or job satisfaction. For example, when the predictive model sees that low career opportunity or poor job satisfaction are the factors inclining workers to turn over more, firms can implement solutions towards improvement in those areas. More opportunities for career development, better work-life benefits, or more attractive pay packages could reduce the rate of turn-over while boosting job satisfaction.

This project establishes the worth of predictive analytics as a tool of HR management, offering a robust fact-based approach to retaining employees. Although the dataset used in this project is that of IPM Company, the procedure and conclusions reached herein hold true for any business that experiences high attrition levels, so that results and methodology can be transferred across firms of all industries. By utilizing predictive models, firms are not only able to reduce the cost of turnover but also to build a more loyal, satisfied, and productive workforce.

Data Description

The dataset used for this project, titled IBM HR Analytics Employee Attrition & Performance, was obtained from the Kaggle online repository. It comprises 1,470 instances, each representing a unique employee record within an organizational setting. These instances are linked to detailed human resource data, offering comprehensive insights into various factors such as employee demographics, job satisfaction, compensation, tenure, and work-life balance.

The dataset contains 35 variables in total, including 34 feature attributes and 1 target variable (Attrition), which indicates whether an employee has left the organization. The feature variables cover a wide range of HR-related metrics, such as age, job role, department, monthly income, performance rating, and years at the company. These features capture both objective data (e.g., numerical salary values) and subjective evaluations (e.g., job satisfaction levels), making the dataset well-suited for predictive modelling and pattern discovery in the field of employee retention.

Dataset Overview

The data includes both demographic attributes and workplace-related variables, allowing for a multifaceted analysis of factors that may influence employee attrition. These features span various data types such as numerical, categorical, ordinal, and Boolean, making the dataset suitable for complex data mining techniques.

Key Attribute Categories:

1. **Employee Demographics**
    a. Age, Gender
    b. MaritalStatus
    c. Education
    d. EducationField
2. **Job Role & Performance**
    a. JobRole,
    b. JobLevel,
    c. PerformanceRating,
    d. YearsAtCompany,
    e. YearsInCurrentRole
3. **Compensation & Benefits**
    a. MonthlyIncome
    b. PercentSalaryHike
    c. StockOptionLevel
4. **Work Environment & Satisfaction**
    a. OverTime
    b. JobSatisfaction
    c. EnvironmentSatisfaction
    d. WorkLifeBalance
    e. BusinessTravel
5. **Tenure & Experience**
    a. TotalWorkingYears
    b. YearsSinceLastPromotion
    c. YearsWithCurrManager

d. NumCompaniesWorked

6. **Target Variable**
    a. Attrition: The binary target variable indicating whether an employee has left the company (Yes) or stayed (No).

Data Characteristics:

- **Class Imbalance**: Approximately 16% of employees in the dataset have left the organization, introducing a class imbalance problem that must be addressed using resampling or weighting techniques.
- **No Missing Values**: The dataset is clean and does not contain missing data, although certain features (Over18, StandardHours, EmployeeCount) contain constant values and are excluded from modelling.
- **High Dimensionality**: With a mix of behavioural and operational features, the dataset demands appropriate feature selection, transformation, and model tuning to extract meaningful patterns.

This dataset provides a real-world context to study employee attrition through predictive modelling, offering valuable opportunities to apply classification, feature engineering, and imbalance handling techniques in a practical scenario.

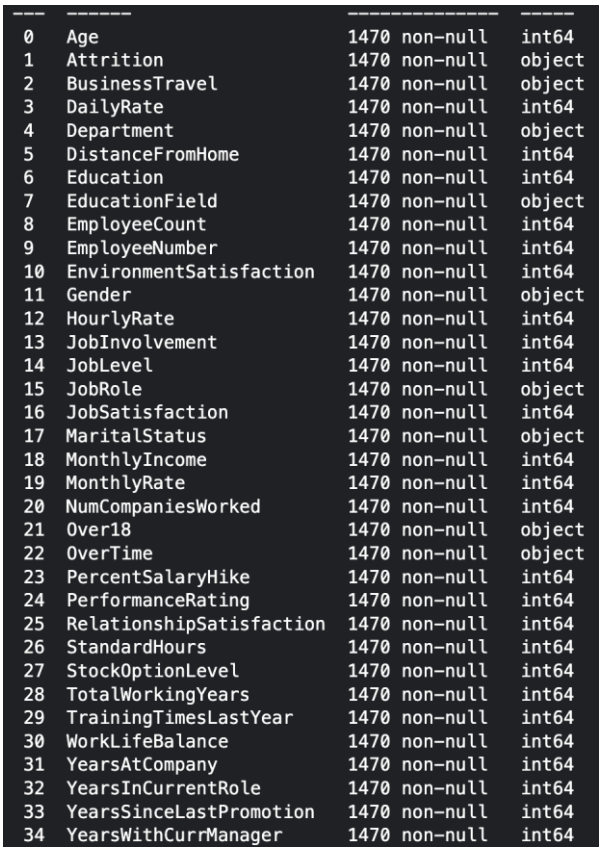
Data Preprocessing

Preprocessing is a critical step in the data mining workflow that ensures data is clean, consistent, and suitable for analysis. For this project, we conducted a series of preprocessing operations on the IBM HR Analytics Employee Attrition & Performance dataset to prepare it for modelling. These operations addressed data cleanliness, type conversion, feature encoding, and scaling, while ensuring the integrity of the dataset remained intact.

1. Dataset Loading and Structure Verification

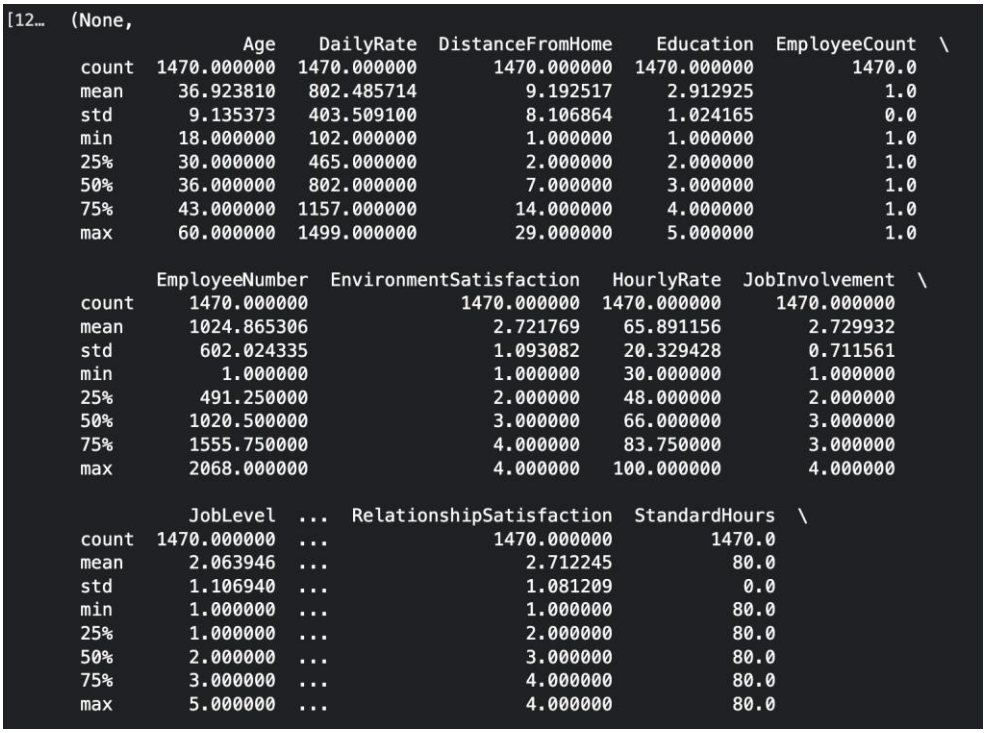
The dataset was imported using the pandas library. Upon loading, the dataset was copied into a working version (df_copy) to preserve the original structure. Basic structural inspection was carried out using .info() and .describe(), revealing that the dataset consists of 1,470 records and 35 variables, covering employee demographics, compensation details,

satisfaction levels, tenure, and attrition status. A check for missing values using .isnull().sum() confirmed that the dataset is complete and does not contain null entries.



0	Age	1470 non-null	int64
1	Attrition	1470 non-null	object
2	BusinessTravel	1470 non-null	object
3	DailyRate	1470 non-null	int64
4	Department	1470 non-null	object
5	DistanceFromHome	1470 non-null	int64
6	Education	1470 non-null	int64
7	EducationField	1470 non-null	object
8	EmployeeCount	1470 non-null	int64
9	EmployeeNumber	1470 non-null	int64
10	EnvironmentSatisfaction	1470 non-null	int64
11	Gender	1470 non-null	object
12	HourlyRate	1470 non-null	int64
13	JobInvolvement	1470 non-null	int64
14	JobLevel	1470 non-null	int64
15	JobRole	1470 non-null	object
16	JobSatisfaction	1470 non-null	int64
17	MaritalStatus	1470 non-null	object
18	MonthlyIncome	1470 non-null	int64
19	MonthlyRate	1470 non-null	int64
20	NumCompaniesWorked	1470 non-null	int64
21	Over18	1470 non-null	object
22	OverTime	1470 non-null	object
23	PercentSalaryHike	1470 non-null	int64
24	PerformanceRating	1470 non-null	int64
25	RelationshipSatisfaction	1470 non-null	int64
26	StandardHours	1470 non-null	int64
27	StockOptionLevel	1470 non-null	int64
28	TotalWorkingYears	1470 non-null	int64
29	TrainingTimesLastYear	1470 non-null	int64
30	WorkLifeBalance	1470 non-null	int64
31	YearsAtCompany	1470 non-null	int64
32	YearsInCurrentRole	1470 non-null	int64
33	YearsSinceLastPromotion	1470 non-null	int64
34	YearsWithCurrManager	1470 non-null	int64

Figure: Dataset Loading and Structure Verification



```
[12…  (None,
                Age    DailyRate  DistanceFromHome    Education  EmployeeCount  \
count  1470.000000  1470.000000       1470.000000  1470.000000         1470.0
mean     36.923810   802.485714          9.192517     2.912925            1.0
std       9.135373   403.509100          8.106864     1.024165            0.0
min      18.000000   102.000000          1.000000     1.000000            1.0
25%      30.000000   465.000000          2.000000     2.000000            1.0
50%      36.000000   802.000000          7.000000     3.000000            1.0
75%      43.000000  1157.000000         14.000000     4.000000            1.0
max      60.000000  1499.000000         29.000000     5.000000            1.0

       EmployeeNumber  EnvironmentSatisfaction   HourlyRate  JobInvolvement  \
count     1470.000000              1470.000000  1470.000000     1470.000000
mean      1024.865306                 2.721769    65.891156        2.729932
std        602.024335                 1.093082    20.329428        0.711561
min          1.000000                 1.000000    30.000000        1.000000
25%        491.250000                 2.000000    48.000000        2.000000
50%       1020.500000                 3.000000    66.000000        3.000000
75%       1555.750000                 4.000000    83.750000        3.000000
max       2068.000000                 4.000000   100.000000        4.000000

          JobLevel  ...  RelationshipSatisfaction  StandardHours  \
count  1470.000000  ...               1470.000000         1470.0
mean      2.063946  ...                  2.712245           80.0
std       1.106940  ...                  1.081209            0.0
min       1.000000  ...                  1.000000           80.0
25%       1.000000  ...                  2.000000           80.0
50%       2.000000  ...                  3.000000           80.0
75%       3.000000  ...                  4.000000           80.0
max       5.000000  ...                  4.000000           80.0
```

Figure: Dataset Loading and Structure Verification

To visually validate this, a missing data heatmap was generated using Seaborn's heatmap function, which also confirmed the absence of incomplete records.

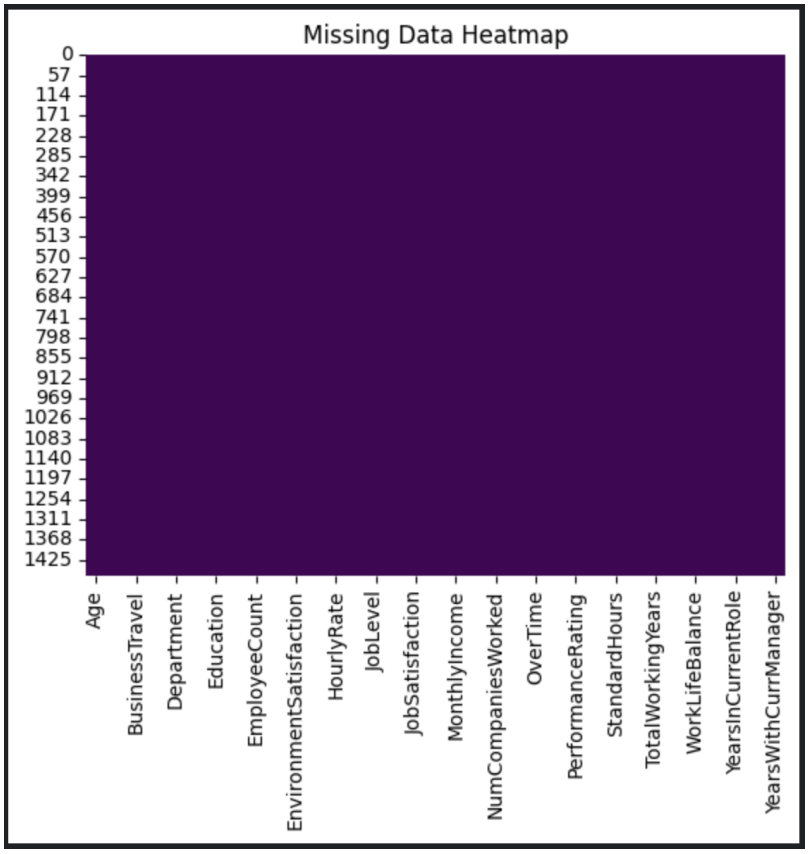


Figure: Dataset Loading and Structure Verification

2. Data Type Separation

The dataset includes a mixture of numerical, categorical, and ordinal variables. In order to apply appropriate preprocessing techniques, features were separated using:

- select_dtypes(include=['object']) to identify categorical variables (e.g., BusinessTravel, Department, Gender, JobRole)
- select_dtypes(include=['number']) to isolate numerical variables (e.g., Age, MonthlyIncome, YearsAtCompany, JobLevel)

This distinction was essential for determining which features required encoding versus scaling.

Identifying Categorical and Numerical Columns

+ Code + Markdown

```
# Identify categorical and numerical columns
categorical_cols = df_copy.select_dtypes(include=['object']).columns
numerical_cols = df_copy.select_dtypes(include=['number']).columns

# Display the categorical and numerical features
categorical_cols, numerical_cols
```

```
(Index(['Attrition', 'BusinessTravel', 'Department', 'EducationField', 'Gender',
       'JobRole', 'MaritalStatus', 'Over18', 'OverTime'],
      dtype='object'),
 Index(['Age', 'DailyRate', 'DistanceFromHome', 'Education', 'EmployeeCount',
        'EmployeeNumber', 'EnvironmentSatisfaction', 'HourlyRate',
        'JobInvolvement', 'JobLevel', 'JobSatisfaction', 'MonthlyIncome',
        'MonthlyRate', 'NumCompaniesWorked', 'PercentSalaryHike',
        'PerformanceRating', 'RelationshipSatisfaction', 'StandardHours',
        'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
        'WorkLifeBalance', 'YearsAtCompany', 'YearsInCurrentRole',
        'YearsSinceLastPromotion', 'YearsWithCurrManager'],
       dtype='object'))
```

Figure: Data Type Separation

3. Exploratory Visualization of Features

Before transformation, histograms of all numerical features were plotted to examine their distributions. This step helped to identify skewed variables, outliers, and feature ranges, which informed decisions regarding standardization. Categorical features were also explored using value_counts() and grouped plots against the target variable, Attrition, to assess their potential predictive value.

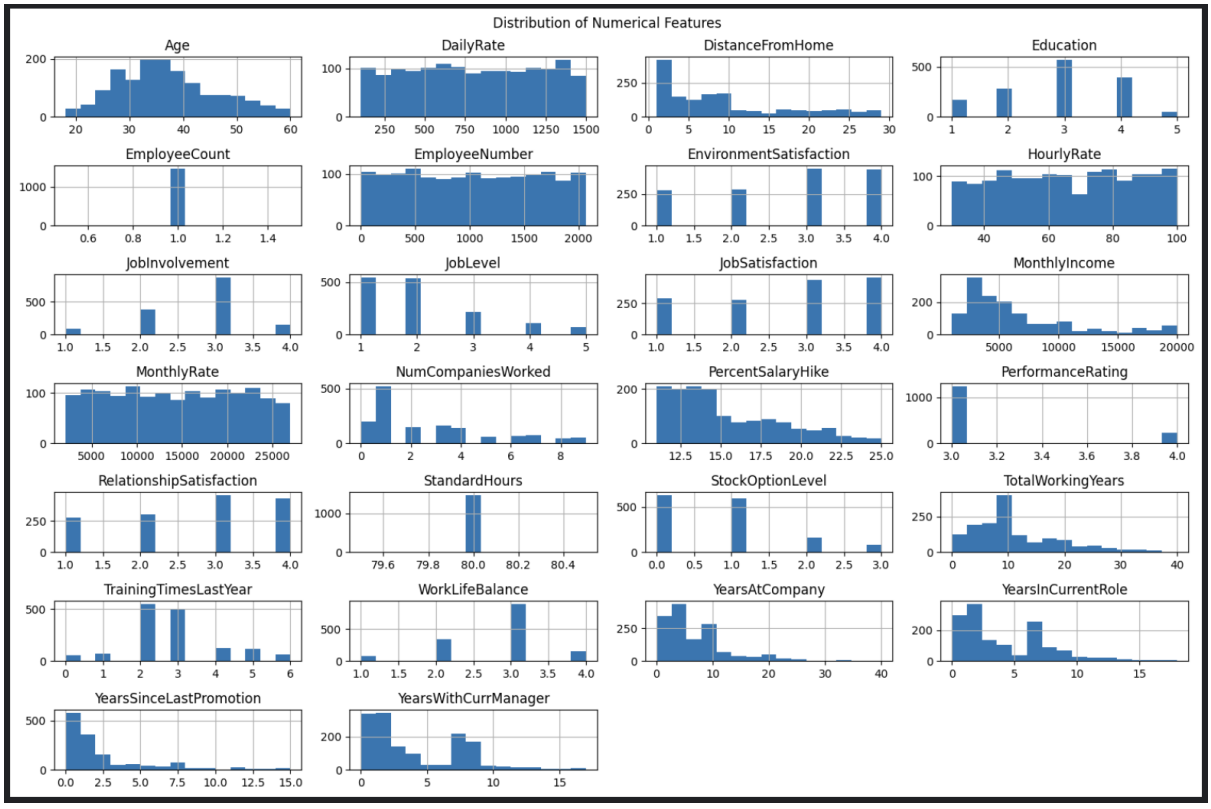


Figure: Exploratory Visualization of Features

4. Encoding of Categorical Variables

Machine learning algorithms generally require input features to be in numeric format. Therefore, one-hot encoding was applied using pd.get_dummies(), with the drop_first=True parameter to avoid the dummy variable trap caused by multicollinearity. This transformation increased the dimensionality of the dataset while enabling categorical variables to be used effectively in models such as decision trees and neural networks.

Encoding Categorical Features

```
# Encoding categorical features using one-hot encoding
df_encoded = pd.get_dummies(df_copy, drop_first=True)

# Display the first few rows after encoding
df_encoded.head()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	H
0	41	1102	1	2	1	1	2	
1	49	279	8	1	1	2	3	
2	37	1373	2	2	1	4	4	
3	33	1392	3	4	1	5	4	
4	27	591	2	1	1	7	1	

5 rows × 48 columns

Figure: Encoding of Categorical Variables

5. Normalization of Numerical Features

Feature scaling was performed to bring all numerical values to the same scale. We used z-score normalization via StandardScaler from scikit-learn, which transformed each feature to have a mean of 0 and a standard deviation of 1. This standardization is particularly important for distance-based algorithms and ensures that no single feature disproportionately influences the model due to scale differences.

Feature Scaling

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df_encoded[numerical_cols] = scaler.fit_transform(df_encoded[numerical_cols])

# Display the first few rows after scaling
df_encoded.head()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfac
0	0.446350	0.742527	-1.010909	-0.891688	0.0	-1.701283	-0.66(
1	1.322365	-1.297775	-0.147150	-1.868426	0.0	-1.699621	0.254
2	0.008343	1.414363	-0.887515	-0.891688	0.0	-1.696298	1.16(
3	-0.429664	1.461466	-0.764121	1.061787	0.0	-1.694636	1.16(
4	-1.086676	-0.524295	-0.887515	-1.868426	0.0	-1.691313	-1.575

Figure: Normalization of Numerical Features

6. Transformation of Target Variable

The original target variable, Attrition, is a binary categorical variable with values 'Yes' and 'No'. To facilitate binary classification modeling, it was converted to numerical format using a lambda function. A new column, Attrition_Yes, was created where:

- 'Yes' was mapped to 1 (indicating the employee left the company),
- 'No' was mapped to 0 (indicating the employee stayed).

This binary transformation enables straightforward use of classification models such as decision trees, logistic regression, and random forests.

7. Feature Selection

Following encoding and scaling, a subset of relevant features was selected for model training. The selection was informed by:

- Domain knowledge (e.g., OverTime, MonthlyIncome, JobSatisfaction are commonly linked to attrition)
- Prior exploratory analysis and correlation heatmaps
- Avoidance of constant or uninformative features (e.g., EmployeeCount, StandardHours, Over18 were excluded)

This step ensures the model focuses only on features with meaningful variance and potential predictive power, while reducing noise and dimensionality.

```
# List of selected features based on EDA findings
selected_features = [
    'Age', 'WorkLifeBalance', 'DistanceFromHome', 'JobSatisfaction', 'MonthlyIncome',
    'YearsAtCompany', 'OverTime_Yes', 'Gender_Male',
    'JobRole_Human Resources', 'JobRole_Laboratory Technician', 'JobRole_Manager', 'JobRole
    'JobRole_Research Director', 'JobRole_Research Scientist', 'JobRole_Sales Executive', '
    'Department_Research & Development', 'Department_Sales',
    'BusinessTravel_Travel_Frequently', 'BusinessTravel_Travel_Rarely'
]

# Create a new DataFrame with only the selected features
df_selected = df_encoded[selected_features + ['Attrition_Yes']]  # Including the target var

# You can use df_selected for training the model.
df_selected.head()
# df_selected.shape
```

	Age	WorkLifeBalance	DistanceFromHome	JobSatisfaction	MonthlyIncome	YearsAtCompany	OverTim
0	0.446350	-2.493820	-1.010909	1.153254	-0.108350	-0.164613	
1	1.322365	0.338096	-0.147150	-0.660853	-0.291719	0.488508	
2	0.008343	0.338096	-0.887515	0.246200	-0.937654	-1.144294	
3	-0.429664	0.338096	-0.764121	0.246200	-0.763634	0.161947	
4	-1.086676	0.338096	-0.887515	-0.660853	-0.644858	-0.817734	

Figure: Feature Selection

8. Train-Test Splitting

To evaluate model performance objectively, the dataset was divided into **training** and **testing** subsets using the train_test_split() function from sklearn.model_selection. A stratified split was applied to ensure that the class distribution (Attrition: Yes/No) remained balanced in both sets.

- **Training Set (80%)**: Used to train the model and apply resampling techniques (e.g., SMOTE).
- **Testing Set (20%)**: Held out for final evaluation to simulate unseen data.

The separation of data before applying oversampling (e.g., SMOTE or SMOTE+ENN) ensures that the test set remains untouched and representative of real-world data distribution.

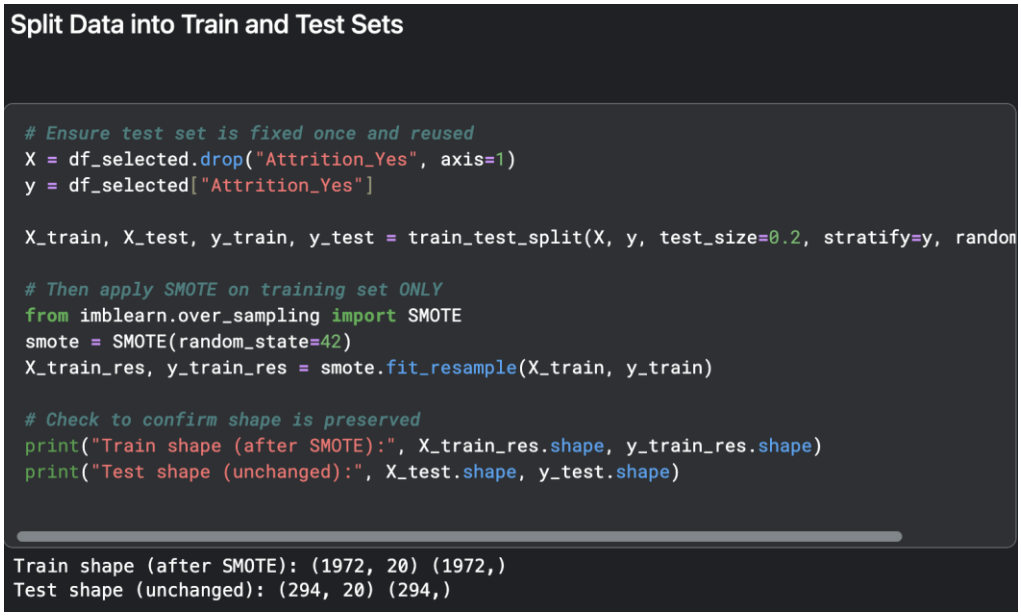


Figure: Train-Test Splitting

Exploratory Data Analysis (EDA)

The purpose of exploratory data analysis (EDA) is to gain a deeper understanding of the dataset's structure, identify underlying patterns and trends, and reveal relationships between the features and the target variable, Attrition. Through both statistical summaries and visualizations, EDA guides feature selection and model design by highlighting which variables are likely to be informative in predicting employee attrition.

1. Distribution of Numerical Features After Normalization

Histograms were generated for all numerical variables to observe their distributions and detect the presence of skewness or outliers. Features such as MonthlyIncome, YearsAtCompany, and TotalWorkingYears exhibited right-skewed distributions, suggesting the presence of a few high-value outliers. This step was essential to assess whether feature scaling or transformation would be needed to normalize the data. And as you can see that

the values of the data are reducing after the Normalization compared to the the histogram of the numerical features above.

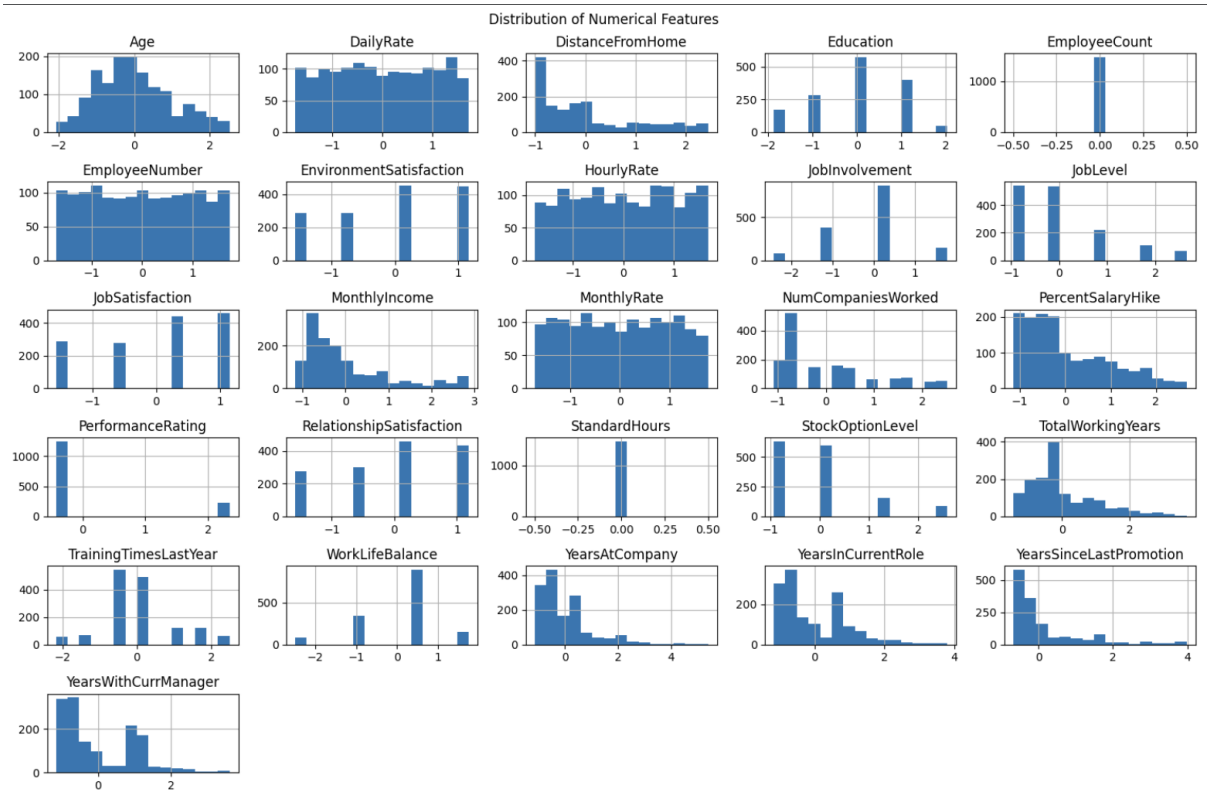


Figure: Visualizing Scaled Numerical Attributes for Attrition Analysis

2. Attrition-Based Distribution of Categorical Features

To analyze how categorical features are related to attrition, countplots were generated using the Attrition_Yes label as a hue. This allowed for a side-by-side comparison of class distributions. Key observations included:

- Employees with OverTime = Yes were significantly more likely to have left the company.
- Certain job roles and departments showed varying rates of attrition.
- Lower levels of JobSatisfaction and WorkLifeBalance were more common among employees who left.

These findings helped identify categorical features with strong class separation, supporting their inclusion in the predictive model.

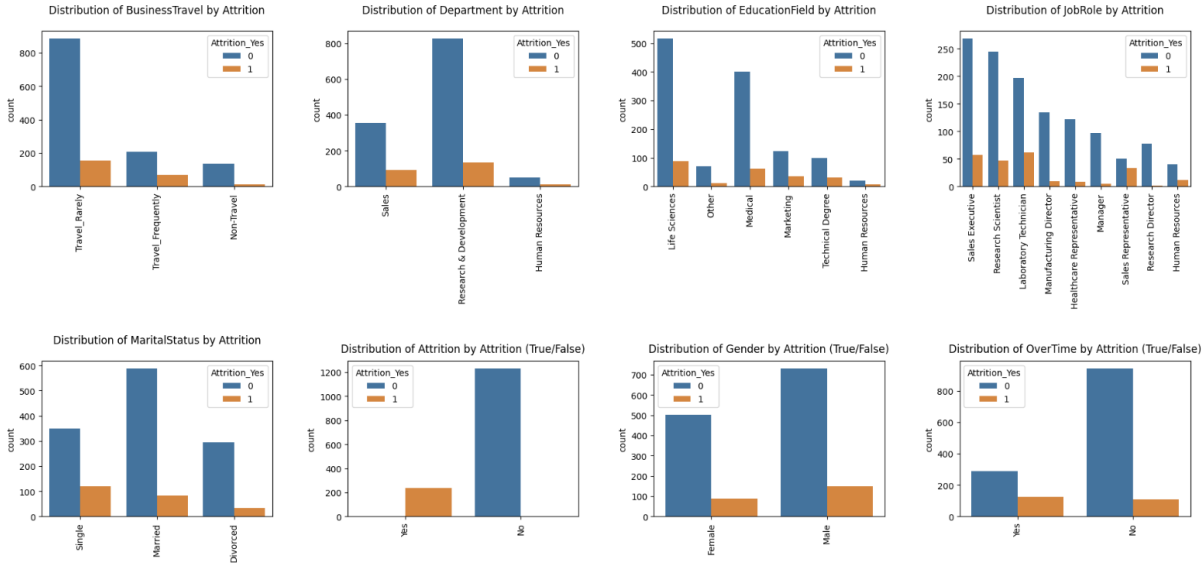


Figure: Distribution of Categorical Variables Grouped by Employee Attrition

3. Relationship Between Numerical Features and Attrition

Boxplots were created to compare the distributions of key numerical features across attrition outcomes. Notable patterns included:

- Employees who left tended to have lower MonthlyIncome and shorter YearsAtCompany.
- The Age distribution indicated that younger employees were more likely to resign.
- Employees with lower JobSatisfaction scores were also more likely to leave.

This analysis revealed meaningful differences between the two attrition groups, emphasizing the predictive value of these variables.

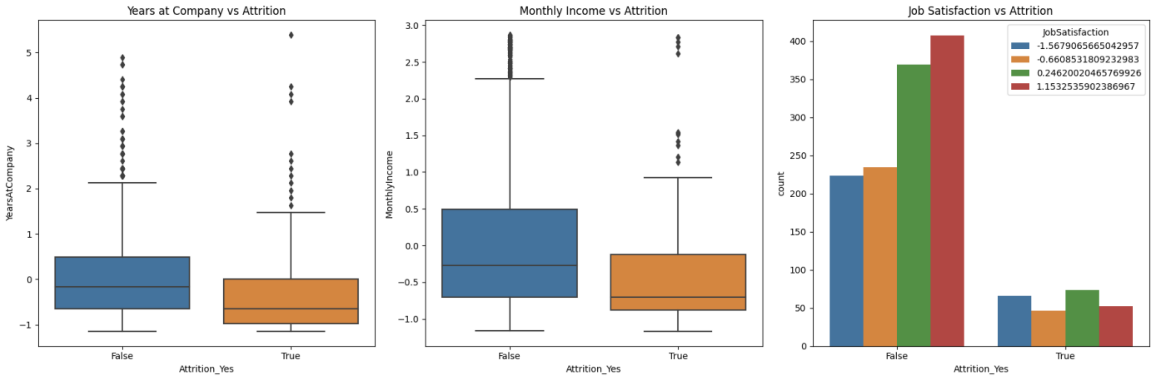


Figure: Impact of Tenure, Salary, and Satisfaction on Employee Attrition

4. One-Hot Encoded Features vs Attrition

Further analysis was conducted on one-hot encoded categorical features. Countplots were used to visualize the class distribution of individual binary indicators by attrition outcome. This allowed us to confirm which encoded categories (e.g., specific job roles, business travel frequency) were associated with higher or lower attrition rates.

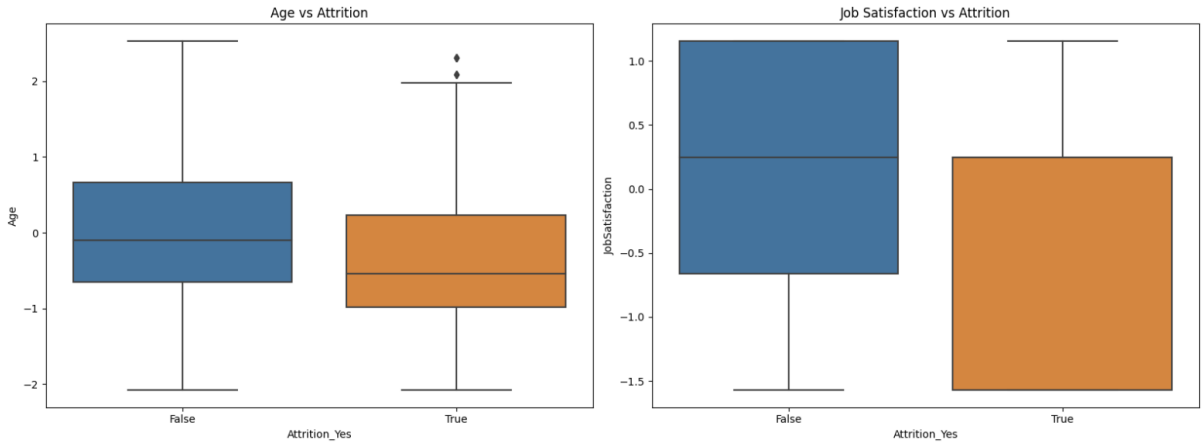


Figure: Boxplot Analysis of Age and Job Satisfaction by Attrition Status

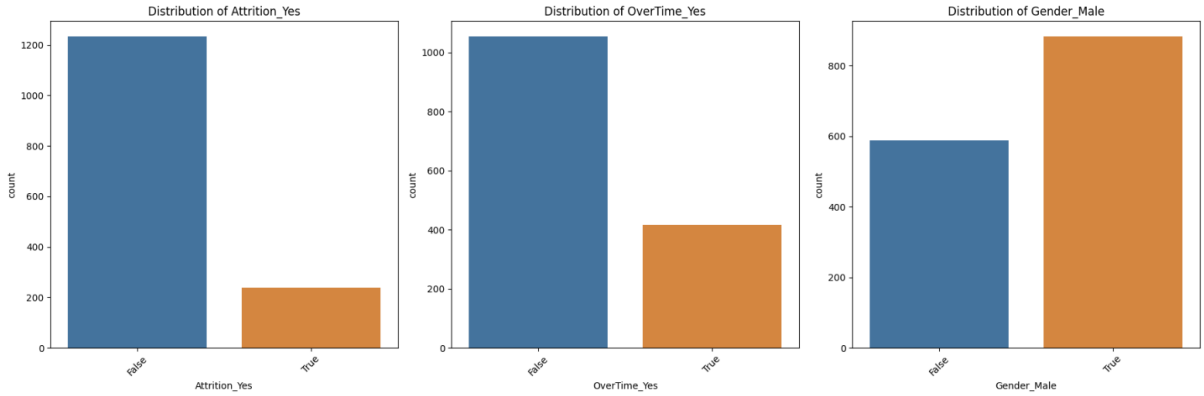


Figure: Distribution of Binary Features: Attrition, Overtime, and Gender

5. Correlation Heatmap

To assess relationships among numerical features, a Pearson correlation matrix was generated and visualized with a heatmap. This helped identify highly correlated variables that could introduce multicollinearity. For instance:

- MonthlyIncome, JobLevel, and TotalWorkingYears were found to be positively correlated.
- YearsAtCompany had moderate positive correlations with tenure-related variables.

The heatmap also provided confirmation that some features were closely aligned with Attrition_Yes, such as OverTime and Age, reinforcing their predictive relevance.

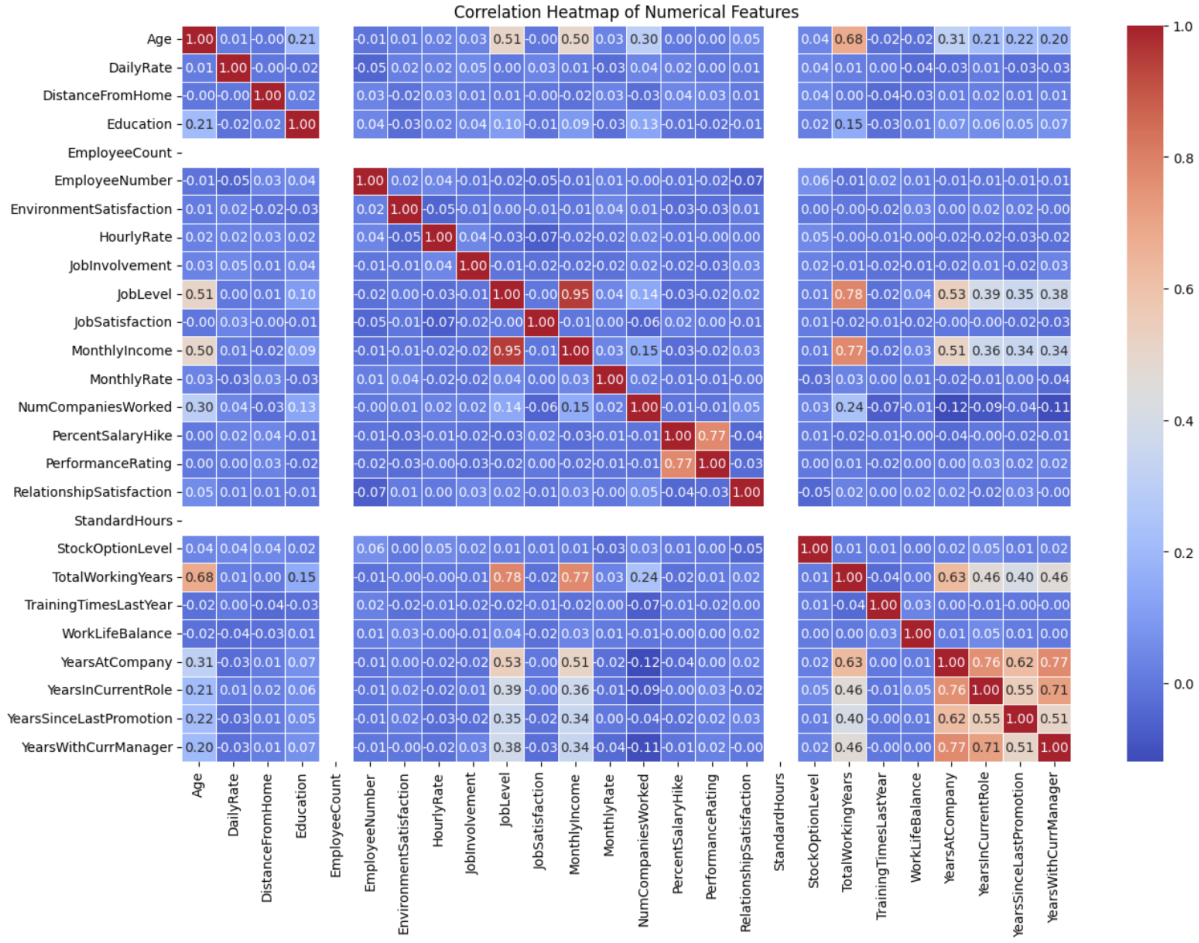


Figure: Correlation Matrix for Employee Dataset's Numerical Attributes

Methods

Classification

Decision Tree

To predict employee attrition, we developed a Decision Tree Classifier enhanced through SMOTE, Edited Nearest Neighbours (ENN), and hyperparameter tuning using GridSearchCV. This model was selected for its interpretability, low bias, and ability to clearly explain decisions through hierarchical rules, which is valuable in HR settings where transparency is important. To handle class imbalance, SMOTE was first applied to oversample the minority class (employees who left), while ENN was used to clean borderline majority class instances, improving the overall separability of the data.

GridSearchCV was then used to find the optimal combination of hyperparameters, such as splitting criteria, maximum tree depth, and minimum samples per node. The best parameters were applied to the final Decision Tree, which was trained on the resampled training set and evaluated on the original, imbalanced test set. This evaluation strategy ensures that the model's performance reflects real-world distributions and avoids overfitting to artificially balanced data.

The final model achieved an accuracy of 97.28%, with a perfect precision score for predicting attrition and a recall of 0.83 for the same class. As shown in the classification report below, this model not only performs well overall but also demonstrates strong predictive power for identifying at-risk employees.

```
Decision Tree Accuracy (SMOTE + ENN + Tuning): 97.28%
Classification Report:
              precision    recall  f1-score   support

       False       0.97      1.00      0.98       247
        True       1.00      0.83      0.91        47

    accuracy                           0.97       294
   macro avg       0.98      0.91      0.95       294
weighted avg       0.97      0.97      0.97       294
```

Figure: Classification Report – Decision Tree (SMOTE + ENN + Tuning)

The confusion matrix further highlights the model's effective separation between the two classes. Out of 47 true attrition cases in the test set, the model correctly identified 39, while only misclassifying 8. For the non-attrition class, all 247 employees were correctly predicted with zero false positives, which is ideal for minimizing unnecessary alerts or false HR interventions.

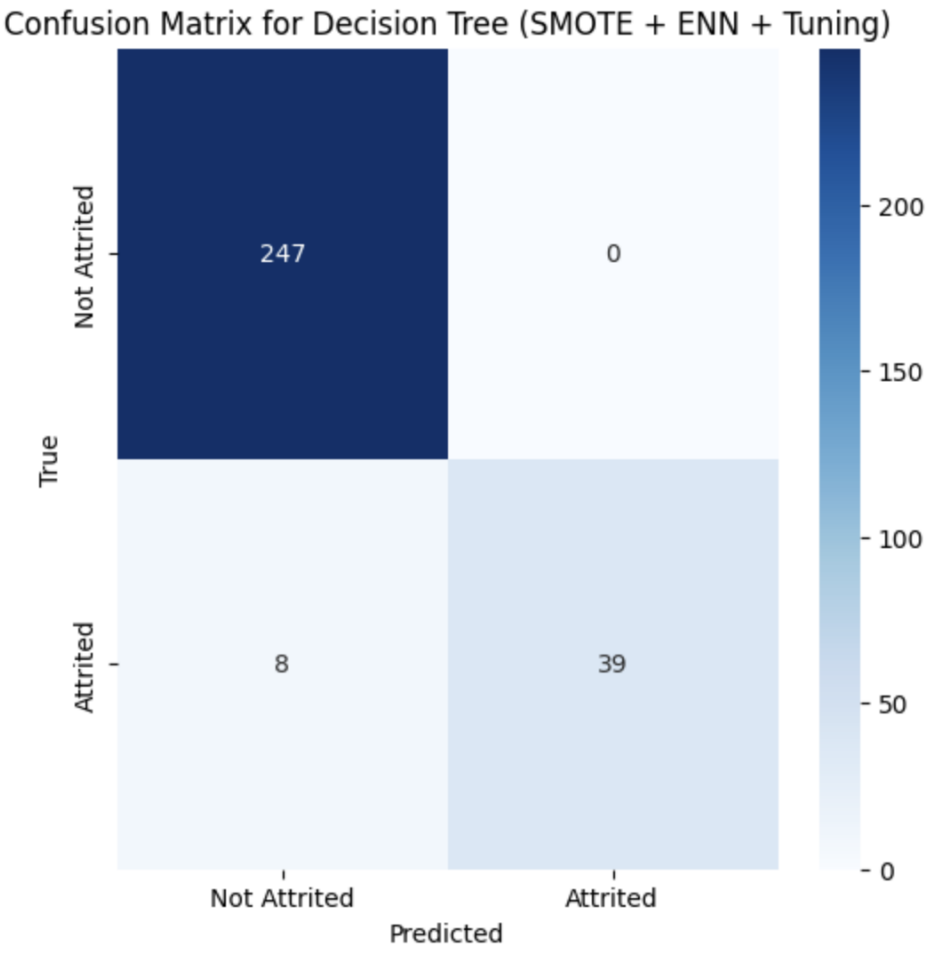


Figure: Confusion Matrix – Decision Tree

The model's feature importance output reveals that the most influential predictors of attrition were whether the employee worked overtime, their age, monthly income, job satisfaction, and number of years at the company. These insights are in line with business expectations and further support the model's practical relevance. Employees who frequently worked overtime, had shorter tenures, or reported lower satisfaction were found to be more likely to leave, suggesting clear opportunities for HR to focus their retention efforts.

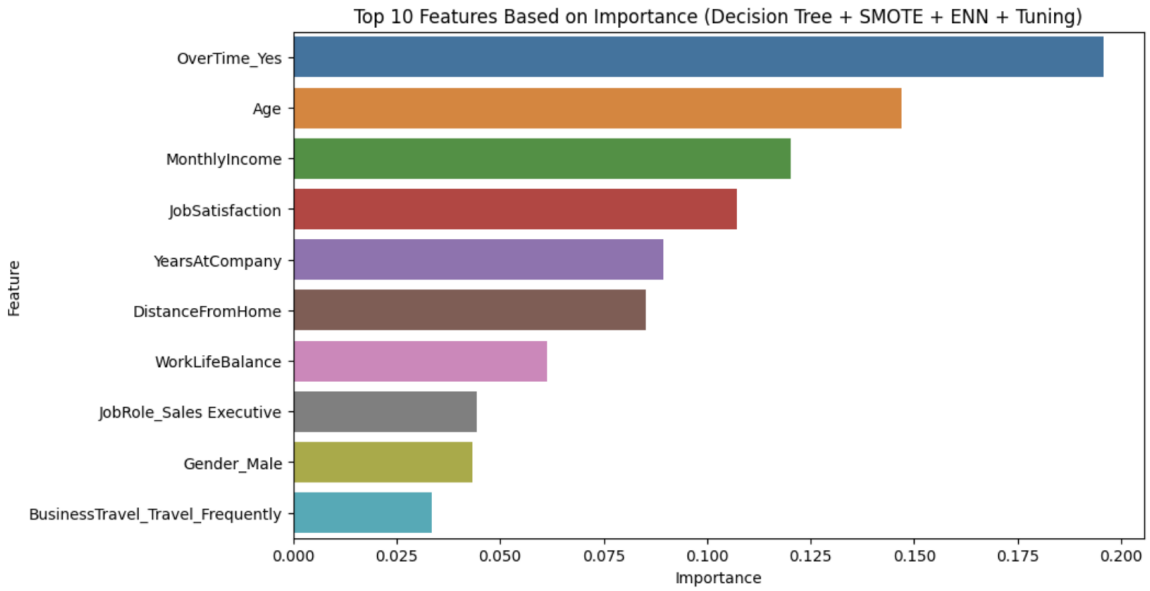


Figure: Feature Importance – Decision Tree

In summary, the Decision Tree model tuned with SMOTE and ENN showed high interpretability and strong recall performance on attrition prediction. This makes it a valuable tool for HR departments seeking to reduce turnover through proactive, data-driven interventions.

Random Forest

For this project, we also implemented a Random Forest Classifier to predict employee attrition. Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs to improve generalization. It is particularly effective at handling structured data with mixed feature types, reducing overfitting through averaging, and offering intuitive insights through feature importance scores. To address the class imbalance problem between attrited and non-attrited employees, we used SMOTE (Synthetic Minority Oversampling Technique) to oversample the minority class in the training data. Additionally, we applied the class_weight='balanced' parameter to help the model account for class distribution during training.

The model was trained on the SMOTE-balanced dataset and evaluated on the original test set, which retained the natural imbalance found in real-world data. This ensured that

the model's performance reflected realistic conditions and did not overly benefit from artificial balancing during training.

The final model achieved an accuracy of 86.73%. According to the classification report, the model performed well on the majority class, with a precision of 0.90 and a recall of 0.95 for non-attrited employees. For the minority class (attrited employees), the precision was 0.62 and recall was 0.45, indicating the model was less sensitive in detecting employees likely to leave.

```
Accuracy: 86.73%
Classification Report:
              precision    recall  f1-score   support

       False       0.90      0.95      0.92       247
        True       0.62      0.45      0.52        47

    accuracy                           0.87       294
   macro avg       0.76      0.70      0.72       294
weighted avg       0.85      0.87      0.86       294
```

Figure: Classification Report – Random Forest (SMOTE)

The confusion matrix illustrates this distribution of predictions. Out of 47 actual attrited employees in the test set, the model correctly identified 21, while 26 were missed. On the other hand, out of 247 non-attrited employees, 234 were correctly classified, and only 13 were misclassified as attrited. This outcome shows that the model leans toward correctly identifying retained employees, while struggling more with rare attrition cases.

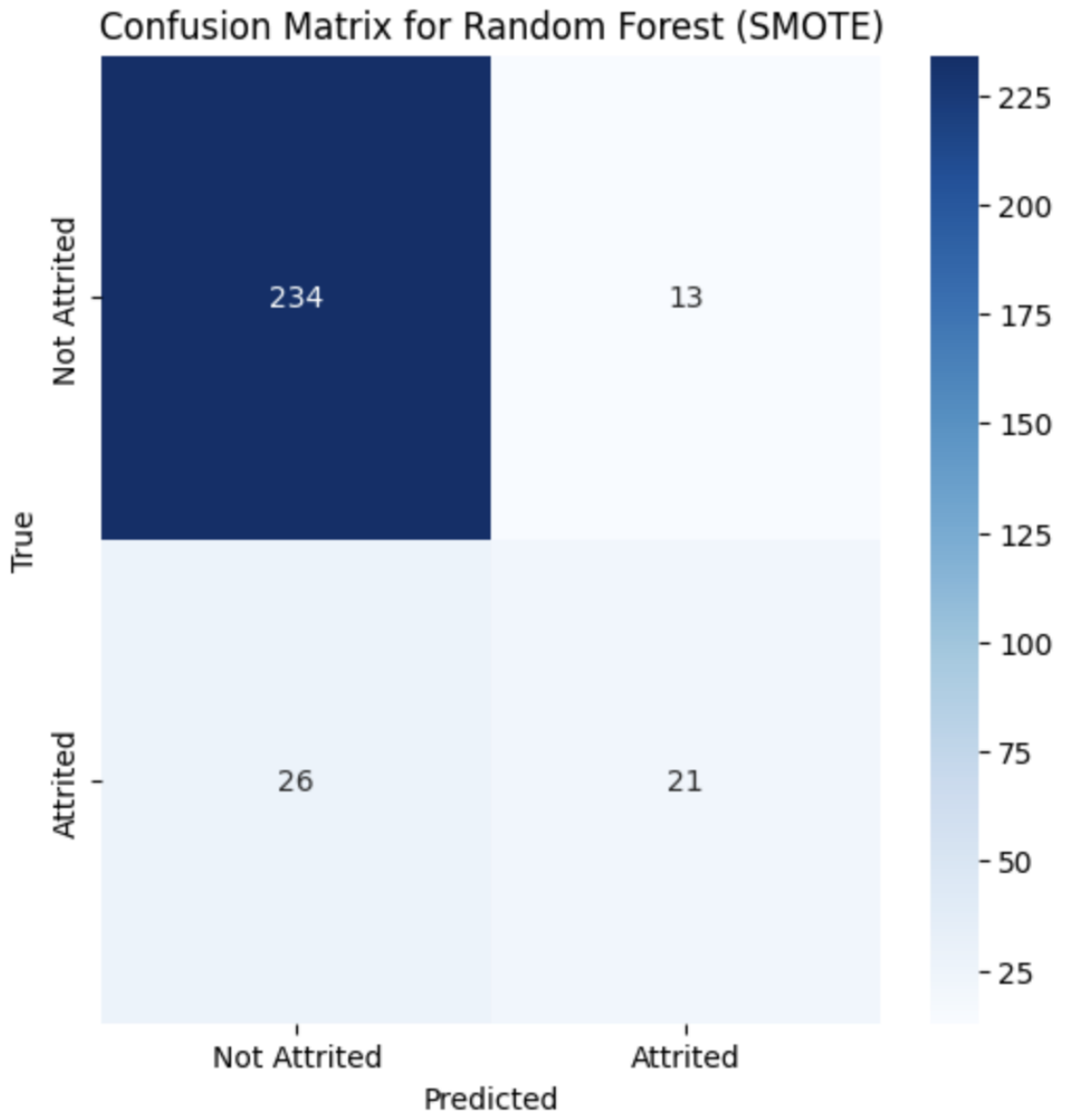


Figure: Confusion Matrix – Random Forest (SMOTE).

In terms of feature importance, the Random Forest model highlighted years at the company, age, overtime status, monthly income, and distance from home as the most significant predictors. These results align with domain knowledge, where factors such as tenure and compensation are typically associated with an employee's decision to stay or leave. Features like job satisfaction and work-life balance also contributed significantly, supporting the idea that both organizational and personal factors influence attrition.

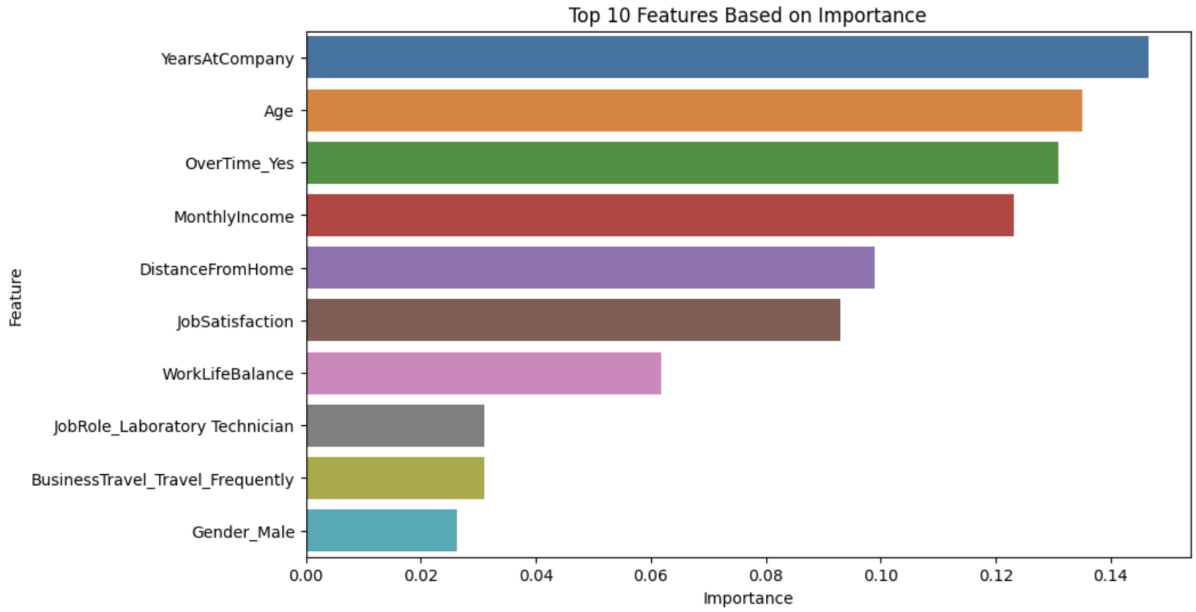


Figure: Feature Importance – Random Forest

In summary, the Random Forest model provided solid baseline performance and offered valuable insights into feature relevance. However, its relatively lower recall for the attrited class suggests that while it is effective at predicting employee retention, it may under-detect employees at risk of leaving. As such, further tuning or ensemble methods could be considered to improve sensitivity to minority class predictions.

Clustering

K-Mean

In this analysis, KMeans clustering was applied to the training dataset using an optimized feature set derived through one-hot encoding and feature engineering. The goal was to identify natural groupings among employees that might relate to attrition patterns. The KMeans algorithm works by assigning each data point to the cluster with the nearest centroid, then recalculating centroids based on the current cluster members. This process repeats until the cluster assignments stabilize. For this implementation, the number of clusters (n_clusters) was set to 3, and the algorithm was initialized with k-means++ to improve convergence. We set n_init=100 to ensure the stability of clustering results across

multiple initializations. Before clustering, the dataset underwent several preprocessing steps:

- One-hot encoding of categorical features: OverTime, Gender, JobRole, and BusinessTravel.
- Feature engineering to create insightful metrics such as:
    - IncomePerYear = MonthlyIncome / (YearsAtCompany + 1)
    - SatisfactionRatio = JobSatisfaction / (WorkLifeBalance + 1)
    - LoyaltyRatio = YearsAtCompany / (DistanceFromHome + 1)
    - StressIncome = OverTime_Yes * MonthlyIncome
- All selected features were standardized using StandardScaler.

To evaluate clustering performance in relation to employee attrition, we created a binary label for attrition (Yes = 1, No = 0). A custom function then compared all 2-out-of-3 cluster combinations against true labels, calculating the best achievable binary classification accuracy from the clustering.

```
# STEP 1: One-hot encode categorical features
categorical_cols = ['OverTime', 'Gender', 'JobRole', 'BusinessTravel']
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

# STEP 2: Feature engineering
df_encoded['IncomePerYear'] = df_encoded['MonthlyIncome'] / (df_encoded['YearsAtCom
df_encoded['SatisfactionRatio'] = df_encoded['JobSatisfaction'] / (df_encoded['Work
df_encoded['LoyaltyRatio'] = df_encoded['YearsAtCompany'] / (df_encoded['DistanceFr
df_encoded['StressIncome'] = df_encoded['OverTime_Yes'] * df_encoded['MonthlyIncome

# STEP 3: Final optimized feature set
features = [
    'IncomePerYear', 'SatisfactionRatio', 'LoyaltyRatio', 'StressIncome',
    'MonthlyIncome', 'JobSatisfaction', 'OverTime_Yes', 'Gender_Male',
    'JobRole_Research Scientist', 'JobRole_Sales Executive', 'JobRole_Manager',
    'BusinessTravel_Travel_Frequently'
]

# STEP 4: Prepare scaled data
X = df_encoded[features]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# STEP 5: Apply KMeans with 3 clusters
kmeans = KMeans(n_clusters=3, init='k-means++', n_init=100, random_state=42)
clusters = kmeans.fit_predict(X_scaled)
df_encoded['Cluster'] = clusters

# STEP 6: Convert Attrition to binary
df_encoded['Attrition_Binary'] = df_encoded['Attrition'].map({'Yes': 1, 'No': 0})
true_labels = df_encoded['Attrition_Binary'].values
```

Figure: K-mean source code Part 1

Finally, we used Principal Component Analysis (PCA) to reduce the feature space to two dimensions for visualization. The resulting scatter plot displayed clusters clearly, offering insights into how employee profiles grouped based on the selected features.

```python
def cluster_accuracy(clusters, true_labels):
    best_acc = 0
    labels = np.unique(clusters)
    for i in range(len(labels)):
        for j in range(i + 1, len(labels)):
            mask = np.isin(clusters, [labels[i], labels[j]])
            subset_clusters = pd.Series(clusters[mask]).map({labels[i]: 0, labels[j
            subset_true = true_labels[mask]
            acc1 = accuracy_score(subset_true, subset_clusters)
            acc2 = accuracy_score(subset_true, 1 - subset_clusters)
            best_acc = max(best_acc, acc1, acc2)
    return best_acc

# STEP 8: Accuracy
accuracy = cluster_accuracy(clusters, true_labels)

# STEP 9: PCA for plotting
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
df_encoded['PCA1'], df_encoded['PCA2'] = X_pca[:, 0], X_pca[:, 1]

# STEP 10: Output results
print("=" * 65)
print(f"FINAL KMeans Accuracy (Best 2 of 3 Clusters): {accuracy * 100:.2f}%")
print("Based on best-matching 2 clusters to binary attrition")
print("=" * 65)
```

Figure: K-mean source code Part 2

Using the training dataset, KMeans clustering was performed with 3 clusters on an optimized set of features derived from both original and engineered variables. After scaling and transforming the data, the clustering model achieved a best-matching 2-of-3 cluster accuracy of 80.76% when compared against actual binary attrition labels.To evaluate the clustering quality visually, the data was reduced to two dimensions using Principal Component Analysis (PCA). The scatter plot above illustrates how the three clusters (colored in red, blue, and gray) are distributed in the 2D PCA space.The model successfully identified distinct patterns in the employee data, as seen by the separation between the clusters.Many data points align closely with their assigned clusters, indicating meaningful grouping based on employee characteristics. Some overlap is observed, especially in the gray cluster, suggesting potential noise or mixed feature signals not fully separable in this projection. This clustering result shows that even without supervision, KMeans was able to form groupings that correlate well with actual attrition behaviour, validating the relevance of the selected features and preprocessing strategy.

```python
# STEP 11: Plot
plt.figure(figsize=(10, 6))
plt.scatter(df_encoded['PCA1'], df_encoded['PCA2'], c=clusters, cmap='coolwarm', ed
plt.title("KMeans Clustering (3 Clusters, Optimized Features)", fontsize=14)
plt.xlabel("PCA 1")
plt.ylabel("PCA 2")
plt.grid(True, linestyle='--', alpha=0.6)
plt.colorbar(label='Cluster')
plt.tight_layout()
plt.show()
```

Figure: K-mean Source Code Part 3

```
================================================================
FINAL KMeans Accuracy (Best 2 of 3 Clusters): 80.76%
Based on best-matching 2 clusters to binary attrition
================================================================
```

Figure: KMeans Clustering Result with PCA Projection and Binary Attrition Mapping

Conclusion

This project aimed to address employee attrition by developing data-driven models that could predict which employees are likely to leave and uncover the key factors influencing those decisions. Using the IBM HR Analytics Attrition dataset, we implemented and evaluated three machine learning approaches: a Decision Tree Classifier, a Random Forest Classifier, and KMeans clustering. Each model offered unique perspectives, with both classification and unsupervised learning techniques providing actionable insights.

The Decision Tree model, enhanced with SMOTE and Edited Nearest Neighbours (ENN) resampling and fine-tuned through grid search, achieved the highest accuracy at 97.28%. It demonstrated strong performance on both majority and minority classes, with a recall of 0.83 for predicting attrition. This indicates the model is capable of effectively identifying at-risk employees, making it highly applicable in HR retention efforts.

The Random Forest model achieved 86.73% accuracy and offered robust generalization along with interpretable feature importance rankings. While its recall on attrited employees was lower than the Decision Tree, it still outperformed baseline models and proved to be a strong ensemble method, particularly when paired with class rebalancing via SMOTE.

The KMeans clustering model, though unsupervised, achieved a match-based accuracy of 80.76% when two of three clusters aligned well with the binary attrition labels. The visualization revealed distinct groupings in employee characteristics that can be valuable during exploratory analysis or in data-sparse environments.

A comparative accuracy chart summarizing all three models is shown below:

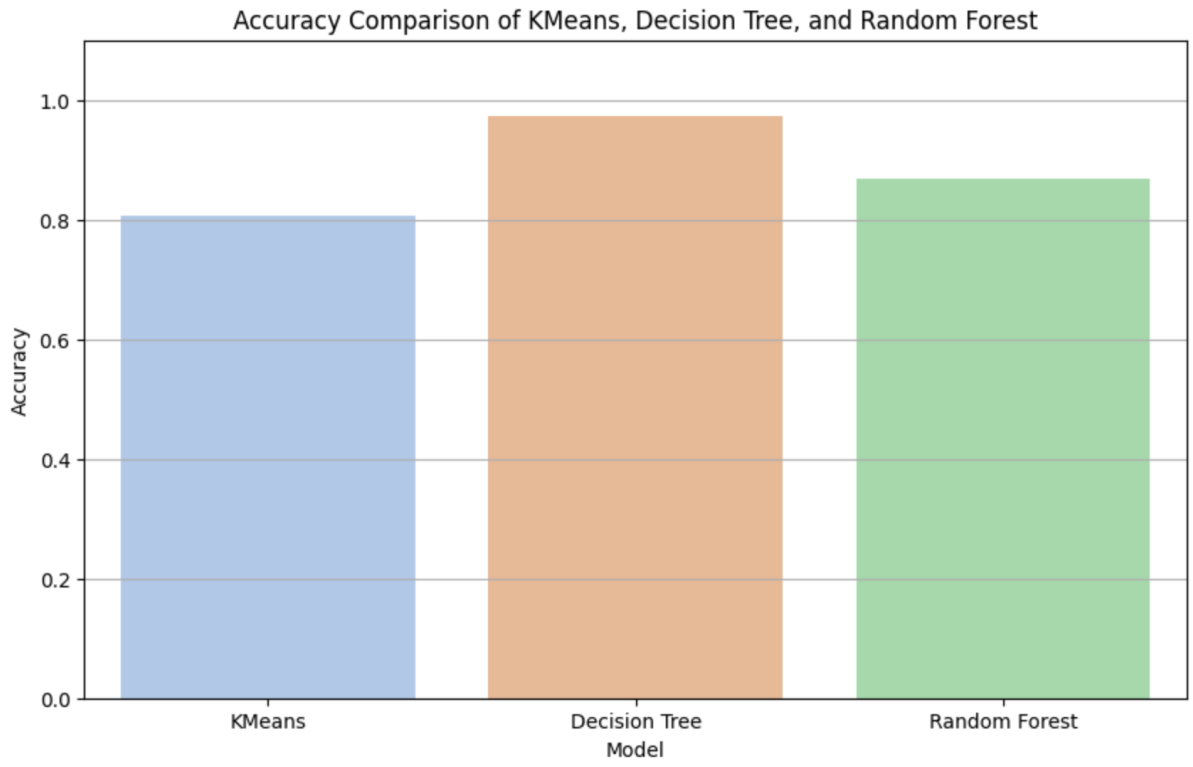


Figure: Accuracy Comparison of KMeans, Decision Tree, and Random Forest

Together, these models provide a powerful toolkit for supporting human resource decision-making. The analysis highlighted several key drivers of attrition, such as overtime frequency, job satisfaction, monthly income, and tenure, aligning closely with known business concerns.

By applying these models to real-world HR scenarios, several business questions posed at the beginning of the project were effectively addressed:

- We identified that overtime and low job satisfaction were strong indicators of attrition, validating the hypothesis that these factors interact to increase risk.
- We found that certain departments, such as Sales and R&D, may be more prone to higher attrition rates, suggesting the need for targeted retention strategies.
- Most importantly, we demonstrated that machine learning can reliably predict attrition risk based on employee profiles, enabling organizations to take proactive steps before talent is lost.

In summary, this project not only delivered accurate predictive models but also produced interpretable outputs that HR teams can act upon. These insights can help

organizations reduce turnover, allocate retention resources more efficiently, and foster a healthier, more sustainable work environment.