Project Report

Analyzing Employee Attrition

Team - 5

Sri Lalitha Somaraju

Krishna Priya Vallurupalli

Bindu Sri Kandula

Mounika Devi Nallapa Raju

Harshini Jampana

Table Of Contents

Data Source

For our analysis, we are using the "Employee Attrition Classification Dataset" from Kaggle. You can find this dataset by searching 'Employee Attrition Classification Dataset' on Kaggle or accessing it directly via this link: https://www.kaggle.com/datasets?search=Employee+Attrition+Classification+Dataset

Data Description

The Employee Attrition Classification Dataset has a train and test dataset we use for ‘Train’. This dataset includes 59599 records across 24 columns, each providing insights into factors influencing employee decisions to remain with or leave their employer.

Employee ID: Unique identifier for each employee.

Age: Employee's age, which might influence their career decisions.

Gender: Employee's gender, used to explore if there are gender-specific attrition patterns.

Years at Company: How long have employees been with the company, showing their loyalty and the company's retention capabilities?

Job Role: The specific role or department the employee works in, affecting their job satisfaction and turnover.

Monthly Income: How much employees earn is crucial to their decision to stay or leave.

Work-Life Balance: Employees’ view on the balance between their work and personal life, which can significantly impact their job satisfaction.

Job Satisfaction: How happy employees are with their job directly affects their likelihood of leaving.

Performance Rating: The company's employee performance evaluation impacts their career progression and satisfaction.

Number of Promotions: The Total promotions employees have received

Overtime: Whether the employee works extra hours

Distance from Home: How far employees live from work

Education Level: The highest degree employees have achieved, potentially related to their job expectations and satisfaction.

Marital Status: Whether the employee is single, married, or divorced.

Number of Dependents: Number of people depending on the employee's income

Job Level: The employee's level or rank in the company

Company Size: How large the company is can influence job stability and opportunities for growth.

Company Tenure: How long have employees been in the industry, reflecting their experience and possibly their loyalty?

Remote Work: Whether employees can work from home

Leadership Opportunities: It is essential for career development to determine whether there are chances for the employee to move into leadership roles.

Innovation Opportunities: If employees can participate in innovative projects

Company Reputation: How well-regarded the company is, which can influence employees' pride in their workplace.

Employee Recognition: How often and significantly employees are recognized for their work impacts their satisfaction and likelihood of staying.

Attrition: Whether the employee has left the company (1 for left, 0 for stayed)

Motivation

Given the recent economic downturns, job security and reasons why employees leave have become critical issues. This project aims to uncover what drives employees to quit their jobs.

Objectives

We aim to answer important questions like:

- What are the main reasons that cause employees to leave?
- Are younger employees more likely to leave than older employees?
- What role does work-life balance play in employee attrition?
- Are employees in certain departments more likely to leave than in others?
- What is the relationship between company size and employee attrition?
- Can we predict future employee attrition based on current data?

Tools

We will use Python, a programming language, with specific tools like Pandas for organizing the data, Scikit-learn for creating the classification model, and Matplotlib for making graphs.

Data Cleaning Steps

We are excluding the columns not relevant to attrition analysis

The dataset contains various features describing employee demographics, roles, work conditions, and organizational factors. The target variable Attrition has two classes, indicating this is a binary classification problem.

[40]:
```
train_df = train_df.drop(['Employee ID','Company Tenure'], axis=1)
```

In the preprocessing phase, we are dropping two columns from the dataset, which are Employee ID and Company tenure. They are removed for the following reasons:

Employee ID: This column contains a unique identifier assigned to each employee.

We are removing the Employee ID because, as a purely nominal and non-informative variable, it holds no predictive value regarding employee behavior or attrition. Including such identifiers can introduce noise or cause overfitting without contributing to the learning process. Unique identifiers are generally excluded from modeling tasks, especially classification, to ensure models learn from features that carry behavioral or contextual meaning.

Company Tenure: This field represents total years of experience across different companies, including outside the current organization.

We are removing Company Tenure. Although experience can be an informative metric, Company Tenure in this dataset exhibits a very high cardinality (127 unique values). It may overlap conceptually with other features such as Years at Company, Years in Current Role, and Total Working Years. Including all these highly correlated tenure-related features may cause **multicollinearity**, affecting model stability and interpretability. Removing Company Tenure reduces redundancy and retains more relevant time-based variables.

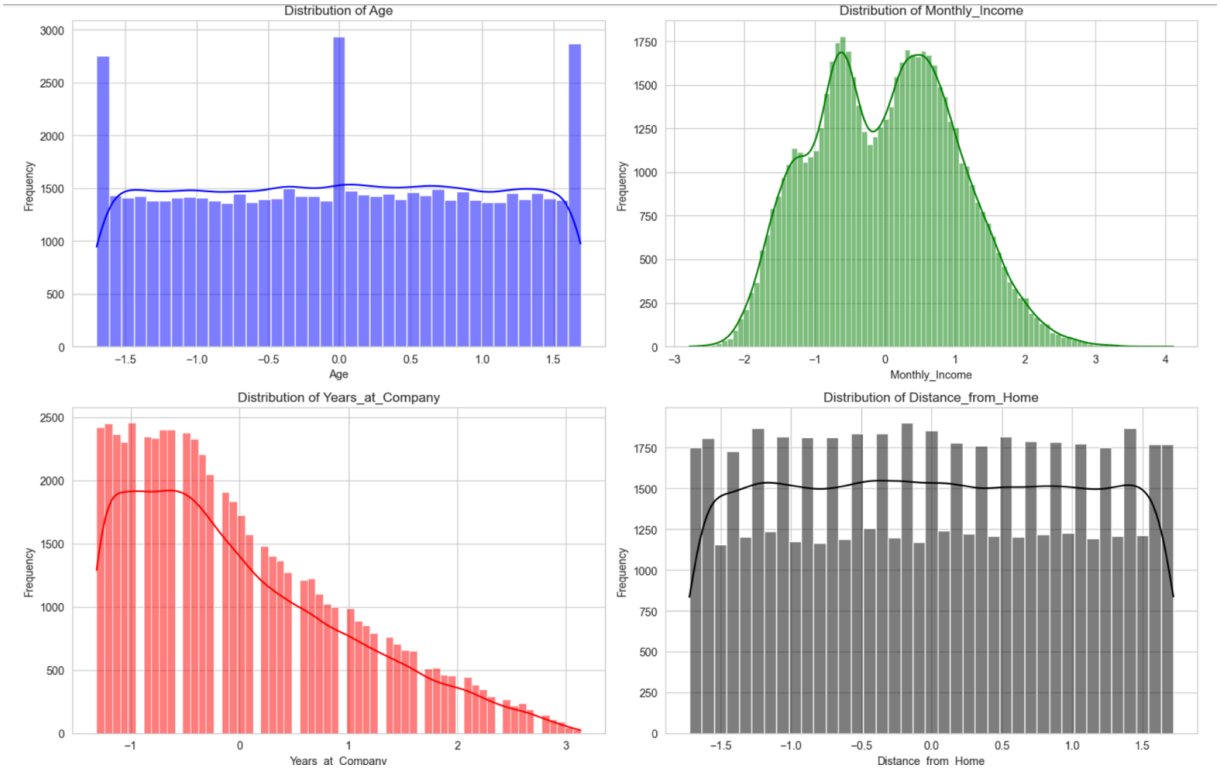
Missing Value Analysis

Part of the data preprocessing phase, the dataset was thoroughly examined for missing or null values across all features. This step is essential to ensure the integrity and quality of data before proceeding with model development. Upon inspection, no missing values were found in any of the columns.

Data Normalizing

[44]:
```python
# prompt: normalize company tenure and monthly income columns with standed scalar

from sklearn.preprocessing import StandardScaler

# Assuming 'train_df' is already loaded as in the previous code

# Select the columns to normalize
columns_to_normalize = [ 'Monthly_Income','Distance_from_Home','Years_at_Company','Age']

# Create a StandardScaler object
scaler = StandardScaler()

# Fit and transform the selected columns
train_df[columns_to_normalize] = scaler.fit_transform(train_df[columns_to_normalize])
```

[48]:
```python
# prompt: plot Histograms: Useful for showing the distribution of numerical variables such as 'Age', 'Monthly Income', 'Years at Company', 'Di

import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'train_df' is already loaded and preprocessed as in the previous code

# Select numerical columns for histogram plotting
numerical_cols = ['Age', 'Monthly_Income', 'Years_at_Company', 'Distance_from_Home']

# Set the style for the plot
sns.set_style("whitegrid")

# Create subplots
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 15))
axes = axes.flatten()

# Define a list of colors for the histograms
colors = ['blue', 'green', 'red',  'black']

# Loop through numerical columns and plot histograms
for i, col in enumerate(numerical_cols):
  sns.histplot(train_df[col], ax=axes[i], kde=True, color=colors[i])
  axes[i].set_title(f'Distribution of {col}', fontsize=12)
  axes[i].set_xlabel(col)
  axes[i].set_ylabel('Frequency')

# Adjust layout and display plot
plt.tight_layout()
```

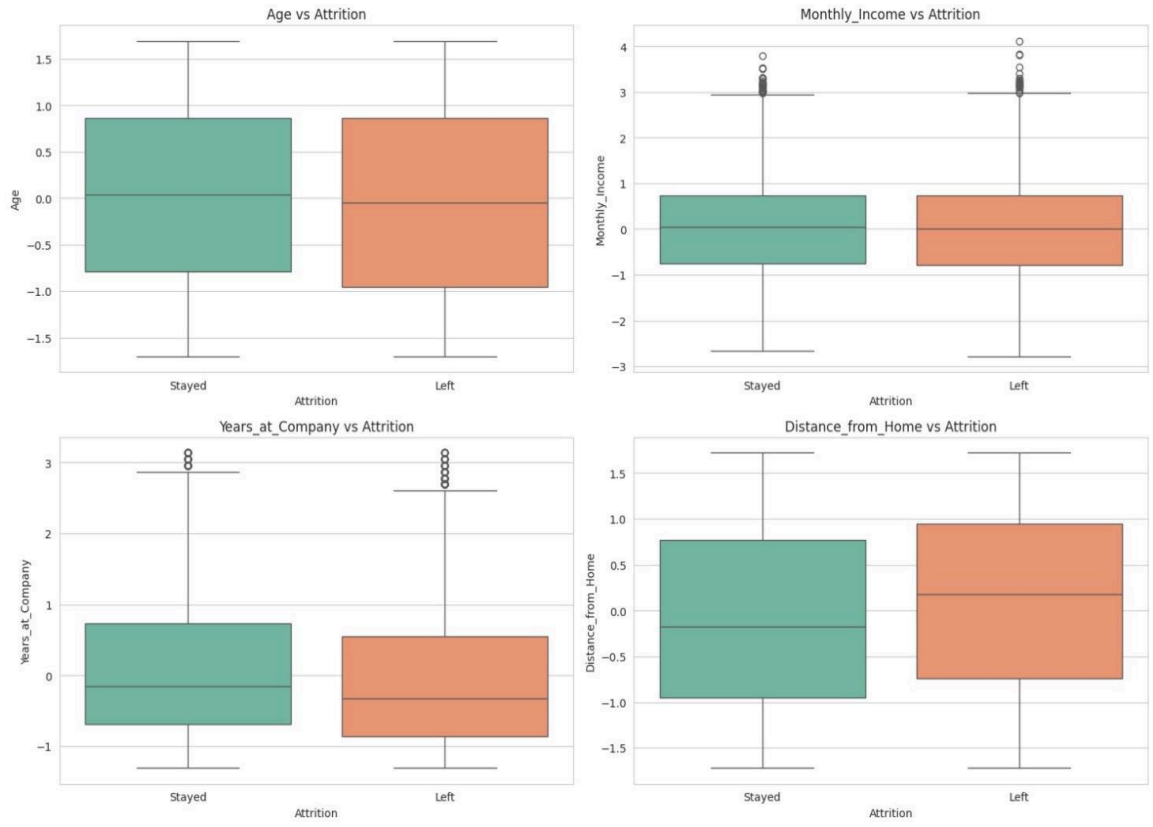


These graphs are plotted to analyze the distribution of numerical variables and to detect any skewness or anomalies. It has been observed that the distribution of "Years at Company" is left-skewed; therefore, it would be beneficial to explore different normalization methods to adjust this skewness or we can drop this column by checking importance of the feature or plotting heatmap.

After normalization, the dataset maintains all original features with transformed values for the scaled columns. The overall structure now includes 22 columns, and all numeric features intended for modeling are on a comparable scale.

Numerical Features vs Attrition

```
# Set up the matplotlib figure
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 15))
axes = axes.flatten()

# Loop through numerical columns and create box plots comparing with Attrition
for i, col in enumerate(numerical_cols):
    sns.boxplot(x='Attrition', y=col, data=train_df, ax=axes[i], palette='Set2')
    axes[i].set_title(f'{col} vs Attrition', fontsize=12)
    axes[i].set_xlabel('Attrition')
    axes[i].set_ylabel(col)

# Remove any unused subplots
for i in range(len(numerical_cols), len(axes)):
    fig.delaxes(axes[i])

# Adjust layout and display plot
plt.tight_layout()
plt.show()
```



These box plots compare how key numerical features differ between employees who stayed and those who left the company.

Age vs Attrition: Older employees tend to stay with the company more than younger employees, as shown by the higher median age of those who stayed versus those who left.

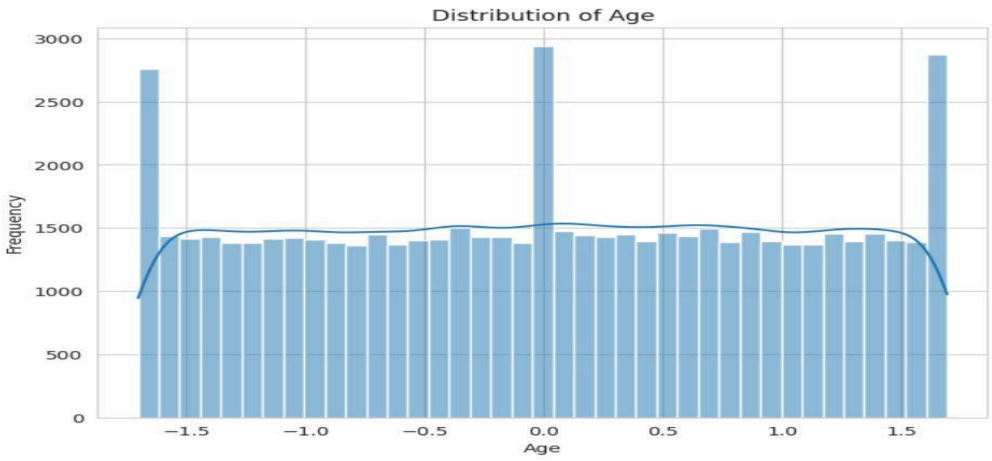
Monthly Income vs Attrition: Employees who left generally had lower monthly incomes than those who stayed, as indicated by the lower median income for those who left.

Years at Company vs Attrition: Employees who have been with the company longer are less likely to leave. The median years at the company are higher for those who stayed than for those who left.

Distance from Home vs Attrition: Distance from home does not show a significant difference between those who stayed and those who left, as the medians are quite close, suggesting distance from home may not be a decisive factor in deciding whether to go or stay.

Detailed Age Distribution Analysis

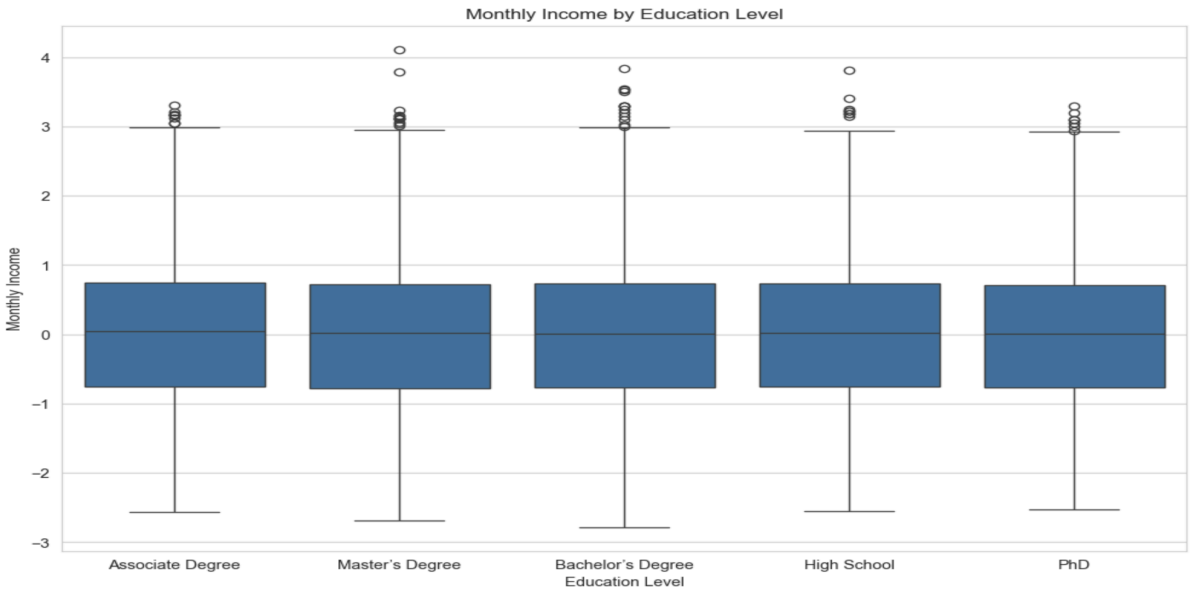
```
# Distribution of Age
plt.figure(figsize=(8, 6))
sns.histplot(train_df['Age'], kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



The age distribution shows a well-balanced spread across the workforce, highlighting diversity across different career stages. While there are some visible peaks at certain ages, the overall pattern indicates that the company employs individuals from a wide range of age groups. This variety supports comprehensive workforce strategies and helps ensure different levels of experience are represented in the organization.

Finding outliers:

[58]:
```
import seaborn as sns

# Box plot for 'Monthly Income' by 'Education Level'
plt.figure(figsize=(12, 8))
sns.boxplot(x='Education_Level', y='Monthly_Income', data=train_df)
plt.title('Monthly Income by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Monthly Income')
plt.show()
```



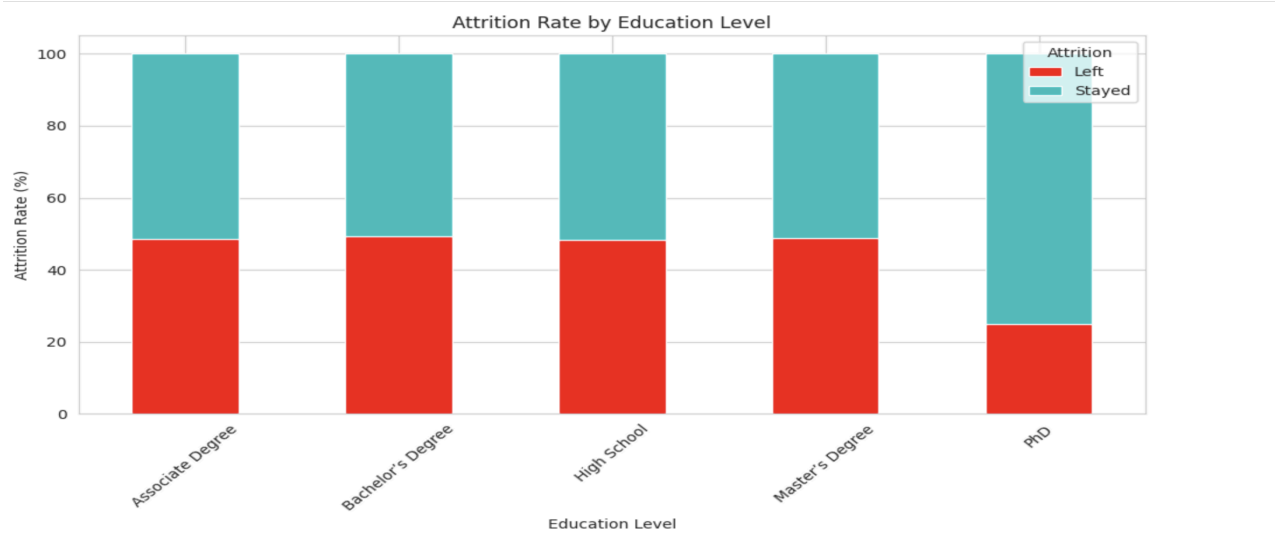
Median Income: The median income is similar across all education levels, suggesting that education does not strongly impact median monthly income within this dataset.

Variability: Incomes vary noticeably within each education level, particularly for those with an Associate Degree and PhD, indicating a wider range of incomes among these groups.

Outliers: There are numerous high outliers at all education levels, highlighting individuals with significantly higher incomes than typical within each category, regardless of their education level.

Attrition Rate Analysis

```
# Calculate the attrition rate for each education level
education_attrition = train_df.groupby('Education_Level')['Attrition'].value_counts(normalize=True).unstack() * 100

# Plot the stacked bar chart

education_attrition.plot(kind='bar', stacked=True, figsize=(10, 6),color=['red', 'c'])
plt.title('Attrition Rate by Education Level')
plt.xlabel('Education Level')
plt.ylabel('Attrition Rate (%)')
plt.xticks(rotation=45)
plt.legend(title='Attrition')
plt.tight_layout()
plt.show()
```

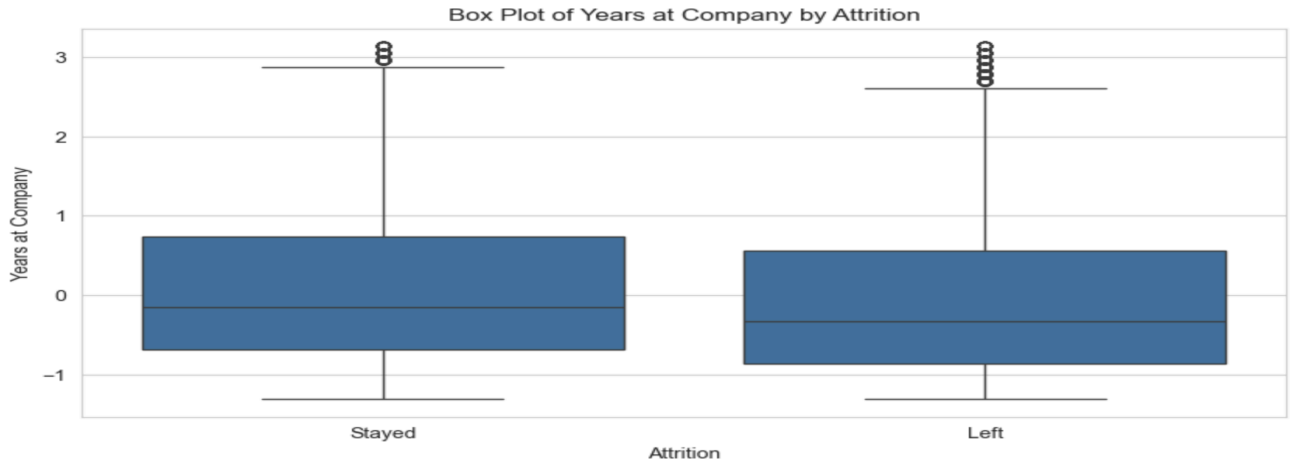


Attrition Proportions: The proportion of employees who have stayed (green area) is considerably higher across all education levels than those who have left (red area).

Education Impact:

PhD These groups show the highest proportion of stayed employees, with more than 60% retention.

Bachelor's Degree, High School, and Master's Degree, Associate Degree. These categories show slightly lower retention rates, but most have stayed, with around 50-60% retention.



The box plot shows how long employees have worked at the company, comparing those who stayed with the company with those who left. Here are the observations:

Similar Median Tenure: Both groups, those who stayed and those who left, have the same median of years at the company.

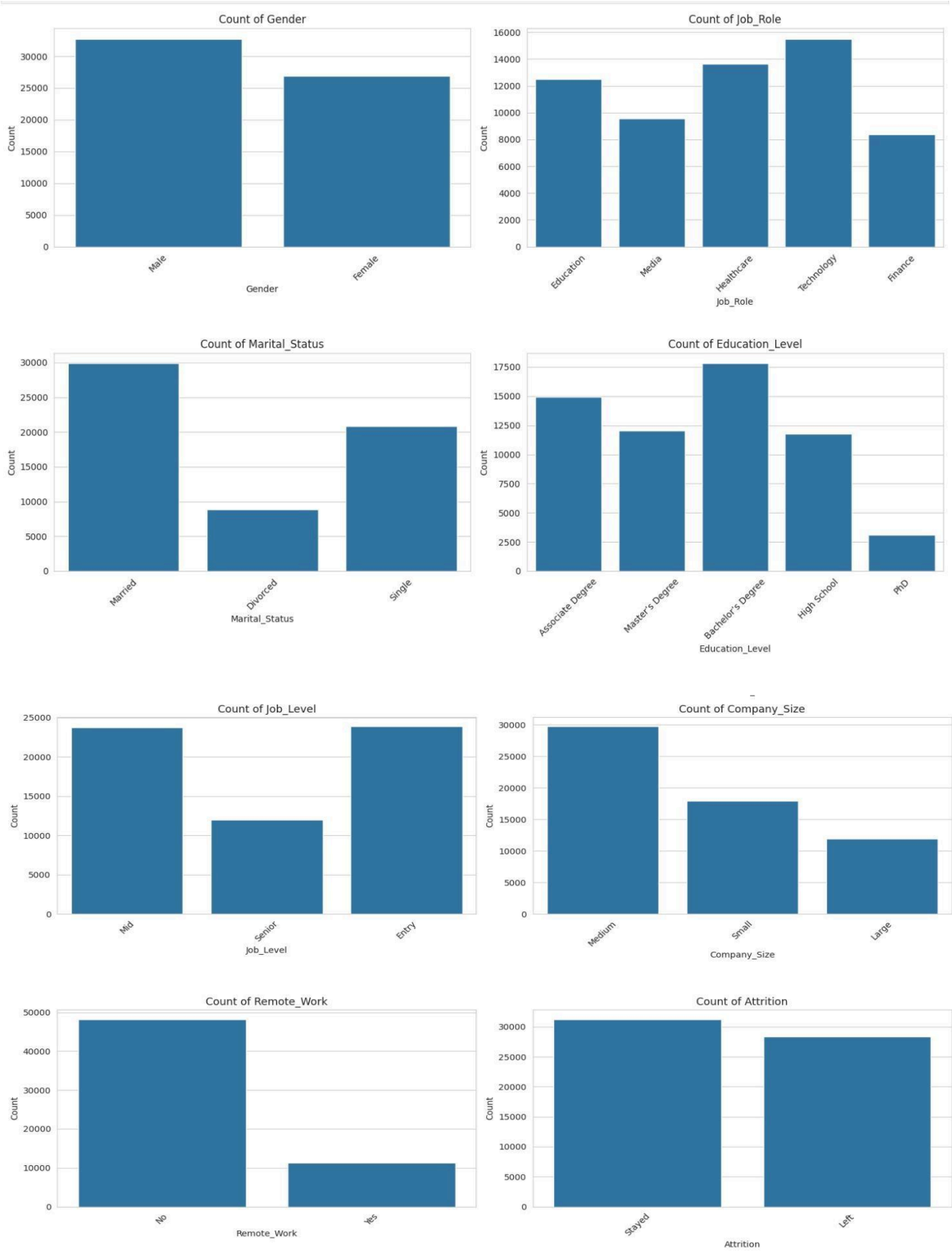
Less Variation Among Stayers: Compared to those who left, employees who stayed with the company have less variation in their tenure.

There are More Outliers Among Leavers: There are a few cases where employees who left had been with the company much longer than most others.

This suggests that, on average, the length of time someone stays at the company does not vary much between those who remain and those who leave, but those who leave tend to have a broader range of tenures.

Categorical Feature Distributions

```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'train_df' is already loaded and preprocessed as in the previous code

categorical_cols = ['Gender', 'Job_Role', 'Marital_Status', 'Education_Level', 'Job_Level', 'Company_Size', 'Remote_Work', 'Attrition']

# Create subplots
fig, axes = plt.subplots(nrows=4, ncols=2, figsize=(15, 20))
axes = axes.flatten()

# Define a list of colors for the bar charts
#colors = ['blue', 'salmon', 'lightgreen', 'gold', 'plum', 'lightcoral', 'khaki', 'teal']

# Loop through categorical columns and plot countplots
for i, col in enumerate(categorical_cols):
    sns.countplot(x=col, data=train_df, ax=axes[i])  # Use modulo for color cycling
    axes[i].set_title(f'Count of {col}', fontsize=12)
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Count')
    axes[i].tick_params(axis='x', rotation=45) # Rotate x-axis labels for better readability


# Adjust layout and display plot
plt.tight_layout()
plt.show()
```



These bar charts display the counts of employees across different categorical features. They help us understand the dataset's overall makeup and offer context for attrition-related patterns.

Gender: The dataset has more male employees than female employees. This slight difference may or may not influence attrition patterns, but it is worth noting.

Job Role: Technology and Healthcare job roles are the most common among employees, while Finance roles have the lowest representation. This could affect how attrition patterns differ across departments.

Marital Status: Most employees are married, followed by single and divorced individuals. This distribution reflects common demographics within the company.

Education Level: Most employees hold a Bachelor's or Associate's degree. PhD holders are the least common, which may impact how attrition varies by educational background.

Job Level: Entry and Mid-level positions make up most of the workforce. Fewer employees are in senior-level roles, which is typical in most organizations.

Company Size: Medium-sized companies have the highest number of employees in the dataset, followed by small and then large companies.

Remote Work: The majority of employees do not work remotely. A smaller portion of the workforce has remote work access, which has been shown to reduce attrition in earlier sections.

Attrition: Slightly more employees stayed with the company than those who left. This balance allows for a fair analysis of attrition patterns.

- Exclude columns not relevant to attrition analysis.
- Address missing values to ensure the integrity of the analysis.
- Finding outlier and their reason and doing the needful to avoid outliers
- Data normalizing

Data Analysis Using Visualization

- Analyze trends in employee attrition across different job roles.
- Explore relationships between job satisfaction, performance ratings, and attrition.

- Generate a heatmap of correlation coefficients among various numeric variables.
- Create grouped bar charts to compare attrition rates between remote and non-remote workers.
- Relationship Between Performance Ratings and Attrition.

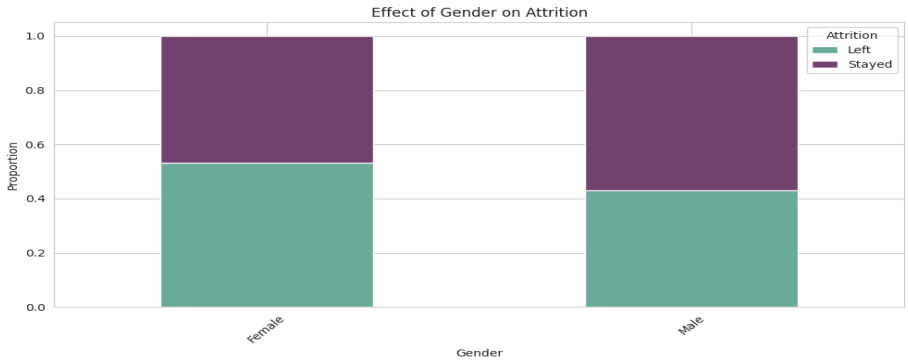
Model Development

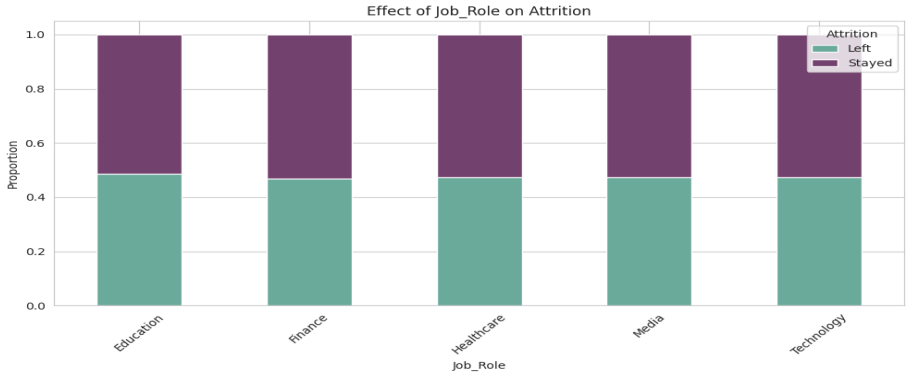
- Classification models, like decision trees, are used to predict attrition. They are chosen for their ability to handle binary outcomes effectively.
- Implement cross-validation to optimize model parameters and ensure robustness.
- Use accuracy metrics to evaluate model performance, focusing on correctly predicting attrition.

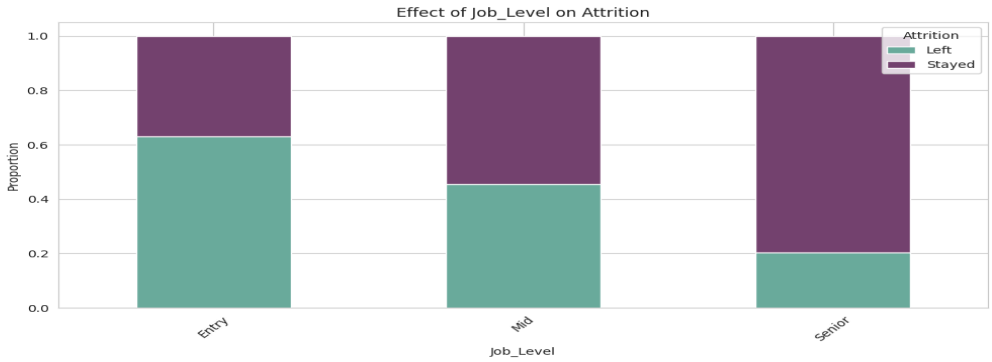
Employee Attrition Analysis

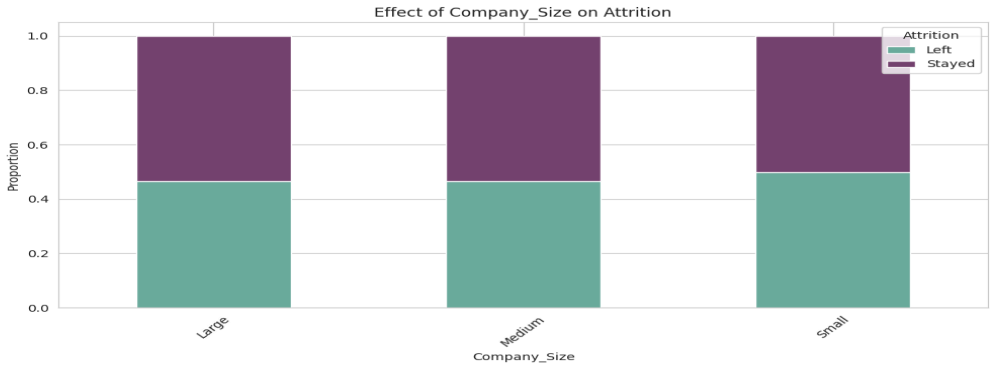
Using categorical data and stacked bar chart visualizations, this explains how different factors are related to employee attrition.

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Assuming 'train_df' is already loaded and preprocessed as in the previous code

categorical_cols = ['Gender', 'Job_Role', 'Marital_Status', 'Education_Level', 'Job_Level', 'Company_Size', 'Remote_Work']
target_col = 'Attrition'

# Create subplots
fig, axes = plt.subplots(nrows=len(categorical_cols), ncols=1, figsize=(10, 40))  # Adjust figure size as needed

# Loop through categorical columns and create stacked bar charts
for i, col in enumerate(categorical_cols):
    # Create a crosstab to count occurrences of each combination
    crosstab = pd.crosstab(train_df[col], train_df[target_col], normalize='index')

    # Create the bar chart
    crosstab.plot(kind='bar', stacked=True, color=['#6daa9f', '#774571'], ax=axes[i])  # Adjust color as necessary
    axes[i].set_title(f'Effect of {col} on {target_col}')
    axes[i].set_xlabel(col)
    axes[i].set_ylabel('Proportion')
    axes[i].tick_params(axis='x', rotation=45)
    axes[i].legend(title='Attrition', loc='upper right')

plt.tight_layout()
plt.show()
```


Effect of Gender on Attrition
Attrition
Left
Stayed
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
Female
Male
Gender


Effect of Job_Role on Attrition
Attrition
Left
Stayed
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
Education
Finance
Healthcare
Media
Technology
Job_Role


Effect of Job_Level on Attrition
Attrition
Left
Stayed
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
Entry
Mid
Senior
Job_Level


Effect of Company_Size on Attrition
Attrition
Left
Stayed
Proportion
1.0
0.8
0.6
0.4
0.2
0.0
Large
Medium
Small
Company_Size

Gender: The company has almost equal male and female employees. However, female employees show a slightly higher attrition rate than males. Even though there is a difference, gender alone may not significantly impact predicting whether someone will leave.

Job Role: Job roles like Education and Healthcare have higher retention, while roles in Media and Finance show more attrition. This could mean that stress or workload in specific roles might be affecting employees' decisions to leave.

Marital Status: Single employees leave more often than married or divorced employees. This may be because married individuals value job stability or have family responsibilities that influence them to stay longer.

Education Level: Employees with PhDs have the lowest attrition rate. Those with high school or bachelor's degrees show higher chances of leaving. This suggests that people with higher education might have better roles or feel more satisfied with their jobs.

Job Level: Entry-level employees are more likely to leave the company, while higher-level employees tend to

stay. This matches common career trends where early-career employees look for better opportunities, while senior employees are more stable.

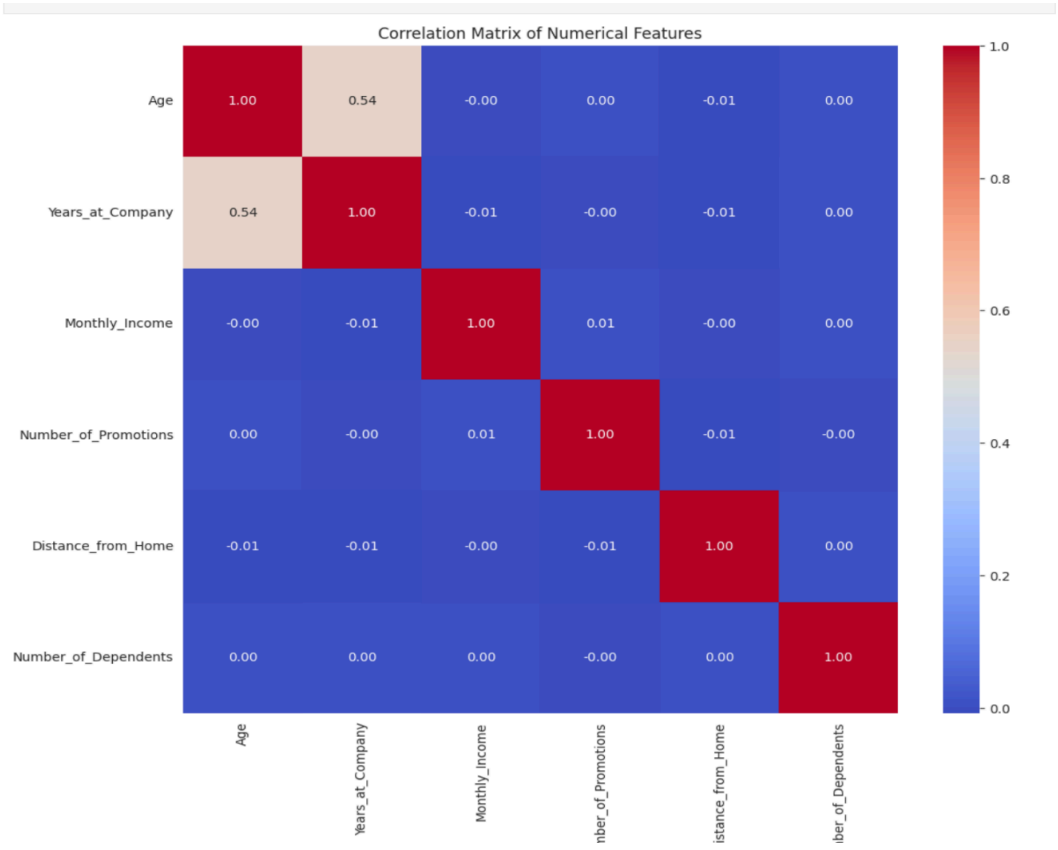
Company Size: Attrition is higher in small companies because they may not offer the same benefits, stability, or growth opportunities as medium or large companies, which tend to retain employees better.

Remote Work: Employees who can work remotely show much lower attrition. Even though this group is smaller, it indicates that flexibility in working conditions helps improve retention.

Overall, this analysis shows that many factors can affect employee attrition. However, remote work options, job level, and company size influence whether employees stay or leave. These insights can help companies create better strategies to reduce attrition and satisfy employees.

Correlation Matrix of Numerical Features

```
correlation_matrix = train_df.corr(numeric_only=True)

# Plot the correlation matrix using a heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Numerical Features')
plt.show()
```

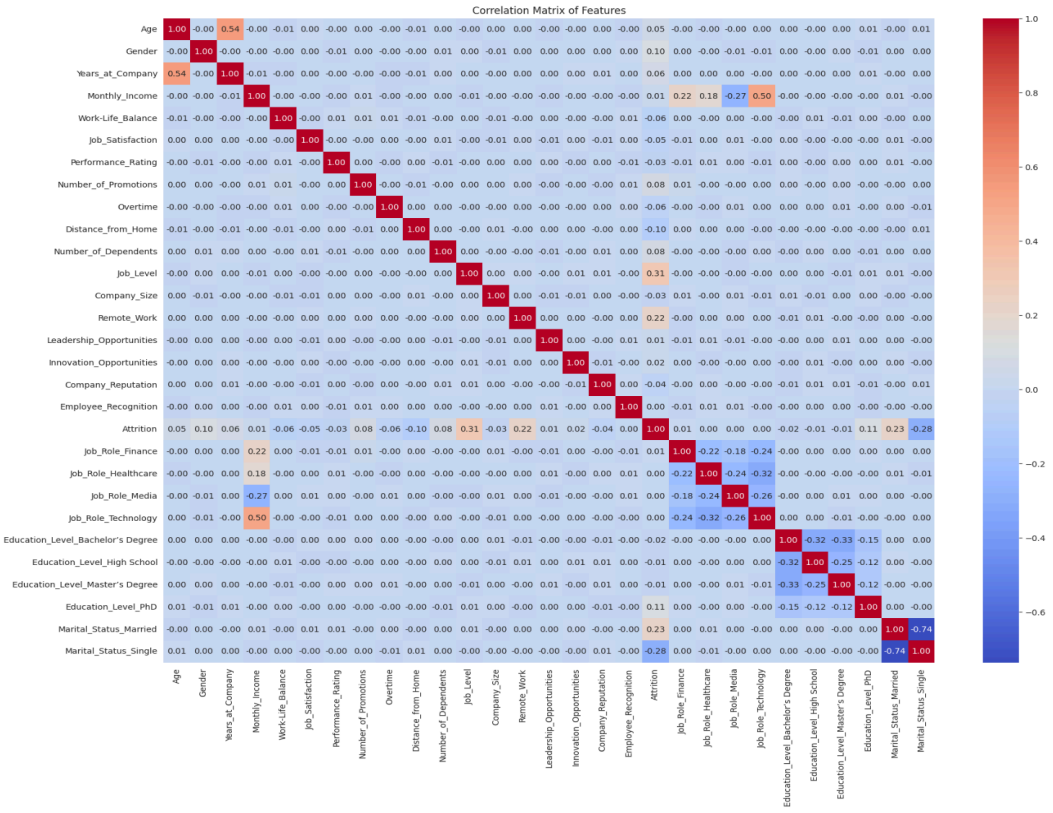


The correlation matrix shows a moderate positive correlation (0.54) between age and years at the company, which means that older employees tend to stay longer in the organization. This makes sense, as employees with more experience are likely to have spent more time at the company.

Other numerical features like monthly income, number of promotions, distance from home, and number of dependents have very little or no correlation. These features are primarily independent and can each provide unique information when building a predictive model, without affecting one another.

Insights from Correlation Matrix Analysis

```
import matplotlib.pyplot as plt
import seaborn as sns


for col in train_df.select_dtypes(include=['object']).columns:
    train_df[col] = train_df[col].map({'Yes': 1, 'No': 0})

plt.figure(figsize=(20, 15))
correlation_matrix = train_df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Features')
plt.show()
```



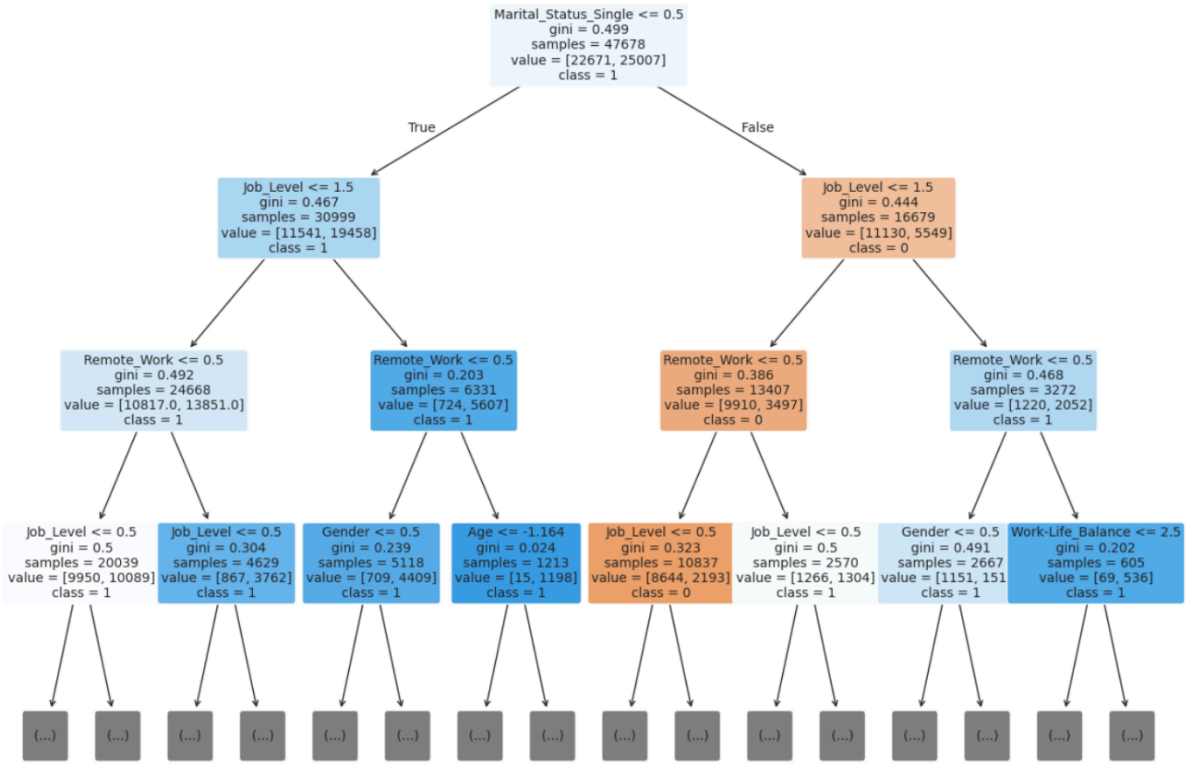
There is a negative correlation (-0.24) between finance job roles and attrition, which means that employees working in finance roles are less likely to leave the company. This could be because they are more satisfied with their jobs or face higher challenges when switching to other roles or companies.

Also, a positive correlation (0.22) between company size and remote work shows that larger companies are more likely to provide remote work options. This might be because they have more resources and flexible work policies compared to smaller companies.

Decision Tree Summary: Employee Retention Prediction

A decision tree classifier was used to predict whether employees will stay or leave the company based on marital status, job level, remote work, age, gender, and work-life balance. The tree starts by splitting on marital status, showing that it's one of the most critical factors. Other significant splits include job level and remote work. Below are some of the key rules from the decision tree:

```
from sklearn.model_selection import GridSearchCV

# Define the parameter grid for the decision tree
param_grid = {
    'max_depth': [3, 5, 7, 10],  # Adjust the range as needed
    'min_samples_split': [2, 5, 10],  # Adjust the range as needed
    'min_samples_leaf': [1, 2, 4]  # Adjust the range as needed
}

# Create a decision tree classifier
tree = DecisionTreeClassifier(random_state=42)  # Use a random state for reproducibility

# Perform GridSearchCV to find the best hyperparameters
grid_search = GridSearchCV(tree, param_grid, cv=5, scoring='accuracy')  # Use 5-fold cross-validation
grid_search.fit(X_train, y_train)

# Get the best estimator (decision tree with the best hyperparameters)
best_tree = grid_search.best_estimator_

# Make predictions on the test set using the best model
y_pred = best_tree.predict(X_test)

# Evaluate the best model
print("Best Decision Tree Model:")
classificationSummary(y_test, y_pred)
```



Job Level and Remote Work: Employees with a job level of 1.5 or lower and no remote work option are likelier to stay with the company.

Node distribution: [10817, 13851]

Marital Status and Remote Work: Employees who are not single, have a higher job level (above 1.5), and don't have remote work access are likelier to leave.

Node distribution: [9910, 3497]

Age and Job Level: Younger, single employees with a lower job level (≤1.5) usually stay in the company.

Node distribution: [15, 1198]

Work-Life Balance: Non-single employees with higher job levels who have a good work-life balance (score > 2.5) They are more likely to stay.

Node distribution: [69, 536]

Gender and Job Level: Female employees at the lowest (≤0.5) are also likelier to stay.

Node distribution: [709, 4409]

Model Development

Introduction to Model Development: This project uses supervised machine learning models to predict employee attrition. Since attrition is a binary classification problem (1 = left, 0 = stayed), models like Decision Trees and K-Nearest Neighbors (KNN) were implemented due to their simplicity, interpretability, and performance on structured datasets. Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess the effectiveness of each model.

Feature Selection and Data Preparation: The dataset contained a variety of employee-related features, including demographic attributes (Age, Gender, Marital Status), job information (Job_Level, Monthly Income, Job_Role_*), and organizational variables (Work-Life Balance, Remote Work, Performance Rating, etc.). After separating the target (Attrition) from the features, the data was split into training and testing sets using an 80-20 ratio:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size=0.2, random_state=42)
```

Model 1: K-Nearest Neighbors (KNN): The **K-Nearest Neighbors (KNN)** algorithm was used as a baseline classifier. It predicts the label of a new sample based on the majority class of its k nearest neighbors in the feature space.

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report

# Initialize and train the KNN classifier
knn = KNeighborsClassifier(n_neighbors=5)  # You can adjust the number of neighbors
knn.fit(X_train, y_train)

# Make predictions on the test set
y_pred = knn.predict(X_test)

# Print the classification report
print(classification_report(y_test, y_pred))
```

KNN Evaluation:

- **Accuracy:** 66.48%
- **Confusion Matrix:**

```
Confusion Matrix (Accuracy 0.6648)

       Prediction
Actual    0    1
     0 3735 1932
     1 2064 4189
```

Classification Report:

The KNN model achieved moderate performance, with slightly better recall for employees who left (Class 1).

However, a more robust model was explored next, given its sensitivity to data scale and high-dimensionality.

Metric	**Class 0**	**Class 1**
Precision	0.64	0.68
Recall	0.66	0.67
F1-Score	0.65	0.68

Model 2: Decision Tree Classifier with GridSearchCV: A **Decision Tree Classifier** was selected for its ability to model complex relationships and their visual interpretability. **GridSearchCV** was applied to tune hyperparameters such as max_depth, min_samples_split, and min_samples_leaf to enhance its performance.

```
# Define the parameter grid for the decision tree
param_grid = {
    'max_depth': [3, 5, 7, 10],  # Adjust the range as needed
    'min_samples_split': [2, 5, 10],  # Adjust the range as needed
    'min_samples_leaf': [1, 2, 4]  # Adjust the range as needed
}

# Create a decision tree classifier
tree = DecisionTreeClassifier(random_state=42)  # Use a random state for reproducibility

# Perform GridSearchCV to find the best hyperparameters
grid_search = GridSearchCV(tree, param_grid, cv=5, scoring='accuracy')  # Use 5-fold cross-validation
grid_search.fit(X_train, y_train)

# Get the best estimator (decision tree with the best hyperparameters)
best_tree = grid_search.best_estimator_

# Make predictions on the test set using the best model
y_pred = best_tree.predict(X_test)
```

Best Model Performance (Decision Tree):

- **Accuracy:** 71.85%
- **Confusion Matrix:**

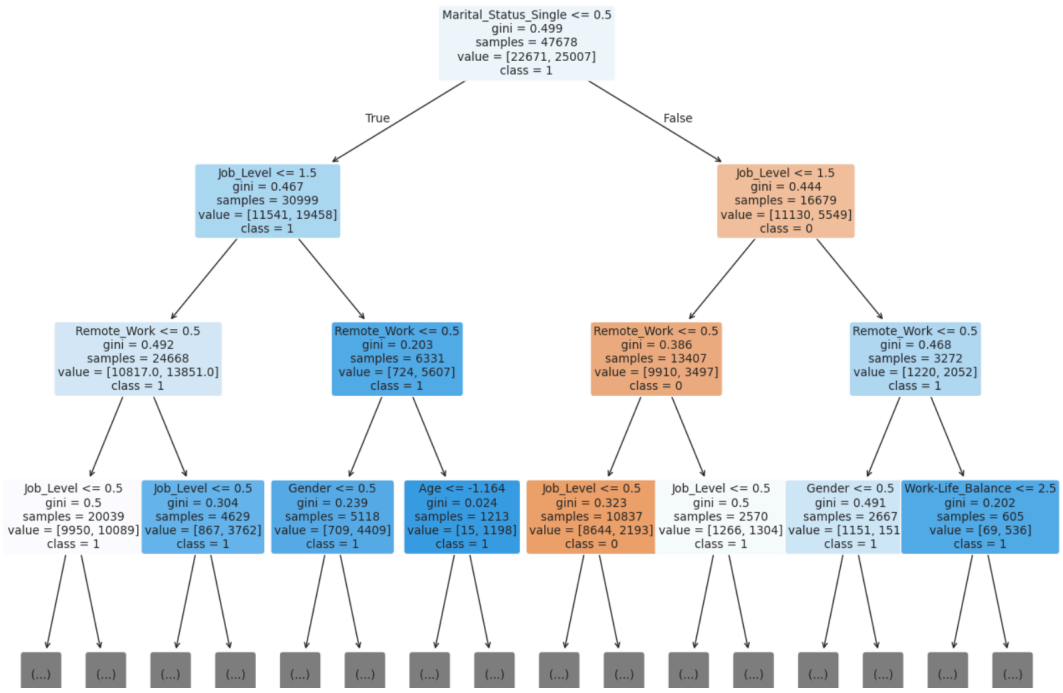
```
Best Decision Tree Model:
Confusion Matrix (Accuracy 0.7185)

       Prediction
Actual    0    1
     0 4263 1404
     1 1951 4302
```

This improved model showed stronger predictive power, especially for correctly identifying employees who stayed (Class 0), while also maintaining a solid recall rate for those who left.

Visualization of Decision Tree: To enhance interpretability, a visualization of the trained Decision Tree was generated (limited to a depth of 3 for clarity). The figure below shows key decision splits, such as:

- Marital_Status_Single: One of the strongest predictors
- Job_Level, Remote_Work, and Work-Life_Balance: Frequently used in early splits, suggesting high influence on attrition

---



Interpretable Decision Rules from the Decision Tree Model

Several **human-readable rules** were extracted from the trained Decision Tree classifier to improve transparency and provide actionable insights. These rules help explain the model's logic and support strategic decisions in employee retention efforts. Below are key rules derived from the top layers of the tree:

Rule 1: Job Level and Remote Work Influence

- If Job_Level ≤ 1.5 and Remote_Work ≤ 0.5

- Then the employee is more likely to stay (Class = 1)

- Interpretation: Entry-level employees who are not allowed to work are generally less likely to leave. This may be due to fewer external opportunities or a greater need for in-person collaboration early in their careers.

Rule 2: Marital Status and Remote Work

- If the employee is not single (Marital_Status_Single ≤ 0.5), has a higher Job Level (> 1.5), but lacks remote work,

- Then they are more likely to leave (Class = 0)

- **Interpretation**: More senior employees who are married and cannot work remotely may experience reduced flexibility, which can contribute to dissatisfaction and attrition.

Rule 3: Age, Marital Status, and Job Level

- If the employee is single, at a low job level (≤ 1.5), and young (Age ≤ -1.164, normalized),

- Then they are likely to stay (Class = 1)

- **Interpretation**: Very young, entry-level, single employees may not yet be actively job hunting and are more focused on gaining initial experience.

Rule 4: Work-Life Balance Influence

- If Job_Level > 1.5, the employee is not single, and Work-Life_Balance > 2.5

- Then they are highly likely to stay (Class = 1)

- **Interpretation**: Employees in more senior roles who enjoy a good work-life balance and have family responsibilities are more inclined to stay.

Rule 5: Gender and Job Level

- If Gender ≤ 0.5 (female) and Job_Level ≤ 0.5

- Then the employee is likely to stay (Class = 1)

- **Interpretation**: Female employees in early-career roles show a higher retention tendency, which may reflect early-career commitment or satisfaction.

Random Forest Classifier (Default Parameters)

The first model trained was a **Random Forest Classifier** with default parameters (n_estimators=100, random_state=42). Random Forest is an ensemble method aaggregates multiple decision trees to reduce overfitting and improve prediction accuracy.

```
Random Forest Model:
Confusion Matrix (Accuracy 0.7398)

       Prediction
Actual    0    1
     0 4146 1521
     1 1580 4673
```

- **Accuracy**: 73.98%
- **Precision (Class 0 - Left)**: 0.73
- **Recall (Class 0 - Left)**: 0.72
- **F1-score (Class 0 - Left)**: 0.72

The default Random Forest model showed **strong overall performance**, with fairly balanced precision and recall for employees who left (class 0). This indicates the model is effective at predicting attrition without overwhelming false positives.

Optimized Random Forest (with GridSearchCV)

- **GridSearchCV** was used to perform hyperparameter tuning on the Random Forest model to enhance performance. Parameters such as n_estimators, max_depth, min_samples_split, and min_samples_leaf were optimized using 5-fold cross-validation.

- **Best Parameters Found**:

```
Best parameters found:  {'max_depth': 30, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 3
00}
```

- **Performance Summary**:

```
Confusion Matrix (Accuracy 0.7429)

       Prediction
Actual    0    1
     0 4099 1568
     1 1497 4756
```

- **Accuracy**: **74.29%**
- **Precision (Class 0 - Left)**: 0.73
- **Recall (Class 0 - Left)**: 0.72
- **F1-score (Class 0 - Left)**: 0.73

After tuning, the optimized Random Forest model slightly improved overall accuracy and maintained the **strong balance** between precision and recall for predicting attrition. This fine-tuned model is more reliable for identifying employees likely to leave and reducing false negatives.

Logistic Regression

Logistic Regression, a linear model suitable for binary classification, was also tested for comparison. It models the log-odds of attrition as a linear combination of input features.

Performance Summary:

```
Logistic Regression Model:
Confusion Matrix (Accuracy 0.7222)

       Prediction
Actual    0    1
     0 3978 1689
     1 1622 4631
```

-

Accuracy: 72.22%

- **Precision (Class 0 - Left)**: 0.71
- **Recall (Class 0 - Left)**: 0.70
- **F1-score (Class 0 - Left)**: 0.70

While **Logistic Regression** performed reasonably well, its predictive ability for class 0 was **slightly lower** than that of the Random Forest models. It may not capture complex non-linear interactions as effectively as tree-based models, which limits its suitability for this problem.

Model Comparison Overview

Model	Accuracy	Precision (0)	Recall (0)	F1-Score (0)
Random Forest (Default)	**73.98%**	**0.73**	**0.72**	**0.72**
Random Forest (Tuned)	**74.29%**	**0.73**	**0.72**	**0.73**
Logistic Regression	**72.22%**	**0.71**	**0.70**	**0.70**

Key Observations

- Model **Logistic Regression**, while interpretable, could not model complex relationships as effectively.

- Even **minor improvements** in recall and precision are critical in HR settings, where identifying potential attrition early can have significant strategic value.

Why Predicting Class 0 (Attrition) Is Critical

From a business perspective, accurately predicting **employees likely to leave** is essential because:

- **High turnover** increases recruitment and training costs.
- It results in the **loss of critical knowledge and skills**.
- It can lower **employee morale** and affect team productivity.

Focusing on **recall for class 0** ensures that most employees at risk of leaving are identified early, enabling HR teams to proactively implement retention strategies (e.g., promotions, role adjustments, wellness programs).
Among all models evaluated, the **Random Forest with tuned parameters** achieved the **best performance** and offers strong predictive capability for employee attrition. Its consistent precision and recall for class 0 make it a dependable tool for **identifying at-risk employees**, enabling strategic interventions.
This model will be used for final feature analysis and organizational recommendations due to its balance of accuracy, interpretability (via feature importance), and real-world relevance in attrition forecasting.

Model Performance Summary and Recommendation

Several classification algorithms were trained and evaluated using **precision**, **recall**, and **F1-score** metrics for the 'Left' class (Class 0) to determine the most effective model for predicting employee attrition. This class is fundamental to businesses, as accurately identifying potential leavers allows for proactive retention strategies.

Model Comparison (Class 0 – Employees Who Left)

Model	Precision	Recall	F1-Score
K-Nearest Neighbors(KNN)	**0.64**	**0.66**	**0.65**
Decision Tree	**0.69**	**0.75**	**0.72**
Random Forest (Default)	**0.73**	**0.73**	**0.73**
Random Forest (Tuned)	**0.73**	**0.73**	**0.73**
Logistic Regression	**0.71**	**0.70**	**0.70**

Observations

- The **Optimized Random Forest** achieved the highest overall accuracy (74.28%) while maintaining balanced precision and recall.
- Compared to other models, it offers the best **trade-off between identifying true leavers (recall)** and **minimizing false positives (precision)**.
- **Decision Trees** provided good recall but were slightly less accurate overall.
- **KNN** underperformed in all key metrics and may not be reliable for real-world HR applications.
- **Logistic Regression** performed decently but did not exceed Random Forest results, especially in handling more complex, non-linear patterns.

Recommendation

Based on this evaluation, the **Optimized Random Forest model is the most suitable choice** for predicting employee attrition. It combines:

- **High accuracy**
- **Strong recall** for identifying employees likely to leave
- **Good precision** to reduce false alarms

This model can be confidently deployed to help HR teams focus their retention efforts more effectively and minimize the business impact of attrition.

Conclusion

Selecting the right model for employee attrition prediction reduces hiring costs, retains top talent, and ensures business continuity. The Random Forest model, especially after tuning, provides a robust, reliable, and interpretable solution to support **data-driven decision-making** in HR management.

Feature Selection Using Random Forest Feature Importance

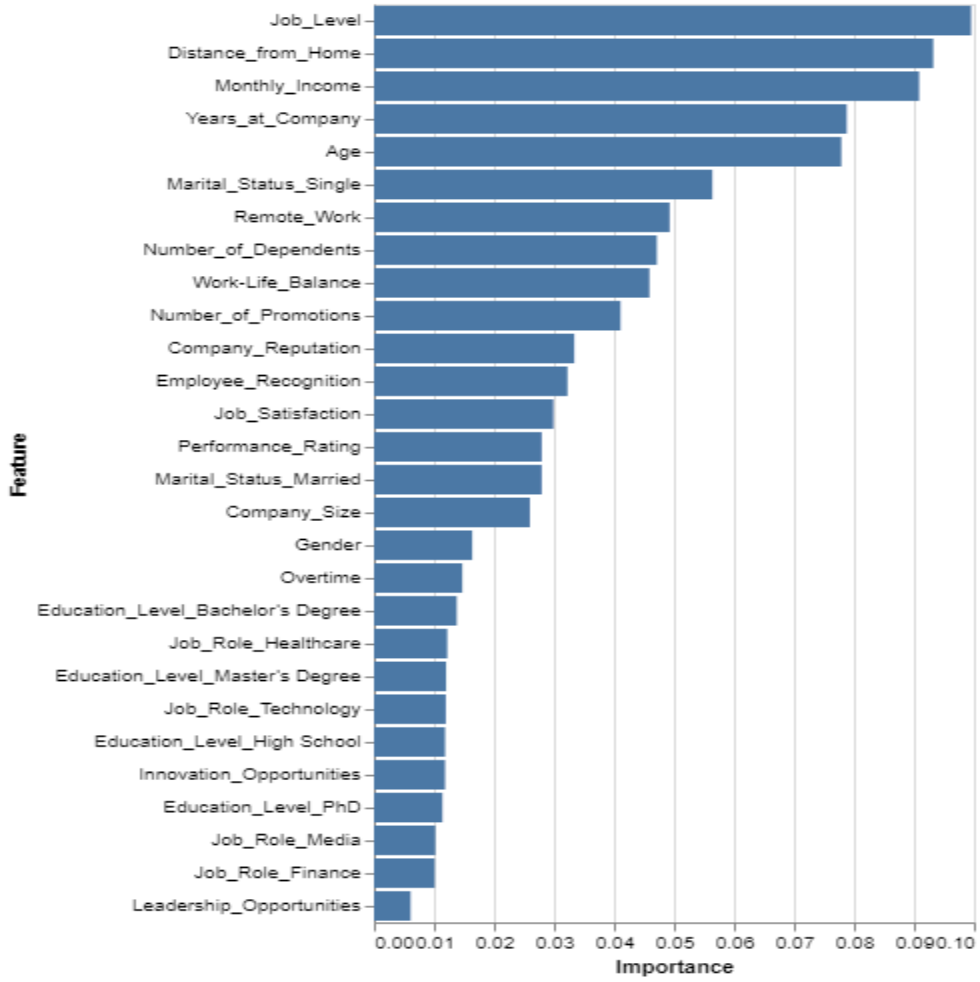
After evaluating the full feature set, we applied **feature importance analysis** from the Random Forest model to identify the most impactful predictors of employee attrition. This approach improves model interpretability, reduces dimensionality, and potentially enhances performance by removing noise.

Feature Importance Analysis

The trained Random Forest classifier provided feature importance scores based on how much each feature contributed to reducing impurity across the trees. The following were identified as the **top features** influencing attrition (shown in descending order of importance):

Rank	Feature	Importance score
1	Job_Level	0.0995
2	Distance_From_Home	0.0932
3	Monthly_Income	0.0909
4	Years_at_Company	0.0788
5	Age	0.0779
6	Martial_Status_Single	0.0564
7	Remote_Work	0.0493
8	Number_of_Dependents	0.0471
9	Work-Life_Balance	0.0459
10	Number_of_Promotions	0.0411

These insights align with domain knowledge, indicating that seniority, commute distance, salary, tenure, and work flexibility correlate with attrition behavior. A bar chart and visualization were created (see Figure below) to display the relative weight of each feature.



Hyperparameter Optimization and Model Evaluation

The Random Forest was optimized using **GridSearchCV** with 5-fold cross-validation. The search space included variations in the number of trees, tree depth, and minimum sample criteria:

```
# Define the parameter grid
param_grid = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
```

Results of the Optimized Model with Selected Features

The best model identified by GridSearchCV was trained using the optimal parameters:

```
Best parameters found:  {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimator
s': 200}
```

Performance Metrics:

```
Random Forest Model with Best Parameters and Selected Features:
              precision    recall  f1-score   support

           0       0.73      0.72      0.72      5667
           1       0.75      0.75      0.75      6253

    accuracy                           0.74     11920
   macro avg       0.74      0.74      0.74     11920
weighted avg       0.74      0.74      0.74     11920
```

Conclusion and Impact

In conclusion, this project provided a comprehensive analysis of employee attrition by utilizing various machine learning techniques. The objective was to predict employee turnover based on critical employee attributes such as monthly income, distance from home, years at the company, and age. We carefully preprocessed the data by handling categorical variables and standardizing numerical features, ensuring robustness in our predictive modeling. Multiple classification models were evaluated, including K-Nearest Neighbors (KNN), Decision Trees, Random Forests, and Logistic Regression. Each model offered distinct insights and predictive capabilities, with Random Forest and Logistic Regression generally providing superior accuracy and stability.

The project underscores the importance of selecting appropriate algorithms and fine-tuning model parameters to enhance predictive accuracy. Random Forest effectively handled complex relationships within the dataset, emphasizing its applicability in similar attrition prediction scenarios. This analysis not only aids in proactive employee retention strategies by identifying at-risk individuals but also supports strategic organizational planning. Future work could extend this study by incorporating additional data features or applying more advanced algorithms, such as ensemble methods or neural networks, to improve prediction accuracy and operational decision-making further.