UNIFIED MENTOR DATA SCIENCE INTERNSHIP PROJECT

GREEN DESTINATION PROJECT

PANKAJ VARSHNEY

UNID: UMIP8438

In [4]:
```
from PIL import Image

image = Image.open("C:/Users/himan/Downloads/greendestination+logo.png")
image.show()
```

In [3]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the data from the provided CSV file
data = pd.read_csv("C:/Users/himan/Downloads/greendestination.csv")
```

In [10]:
```
data
```

Out[10]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Ed
0	41	Yes	Travel_Rarely	1102	Sales	1	2	
1	49	No	Travel_Frequently	279	Research & Development	8	1	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	
4	27	No	Travel_Rarely	591	Research & Development	2	1	
...	...	...	...	...	...	...	...	
1465	36	No	Travel_Frequently	884	Research & Development	23	2	
1466	39	No	Travel_Rarely	613	Research & Development	6	1	
1467	27	No	Travel_Rarely	155	Research & Development	4	3	
1468	49	No	Travel_Frequently	1023	Sales	2	3	
1469	34	No	Travel_Rarely	628	Research & Development	8	3	

1470 rows × 36 columns

In [11]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 36 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 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 
 35  Attrition_numeric         1470 non-null   int64 
dtypes: int64(27), object(9)
memory usage: 413.6+ KB
```

In [12]:

```
data.isnull()
```

Out[12]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	Edu
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
1465	False	False	False	False	False	False	False	
1466	False	False	False	False	False	False	False	
1467	False	False	False	False	False	False	False	
1468	False	False	False	False	False	False	False	
1469	False	False	False	False	False	False	False	

1470 rows × 36 columns

In [13]:
```
data.describe()
```

Out[13]:

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNuml
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000

8 rows × 27 columns

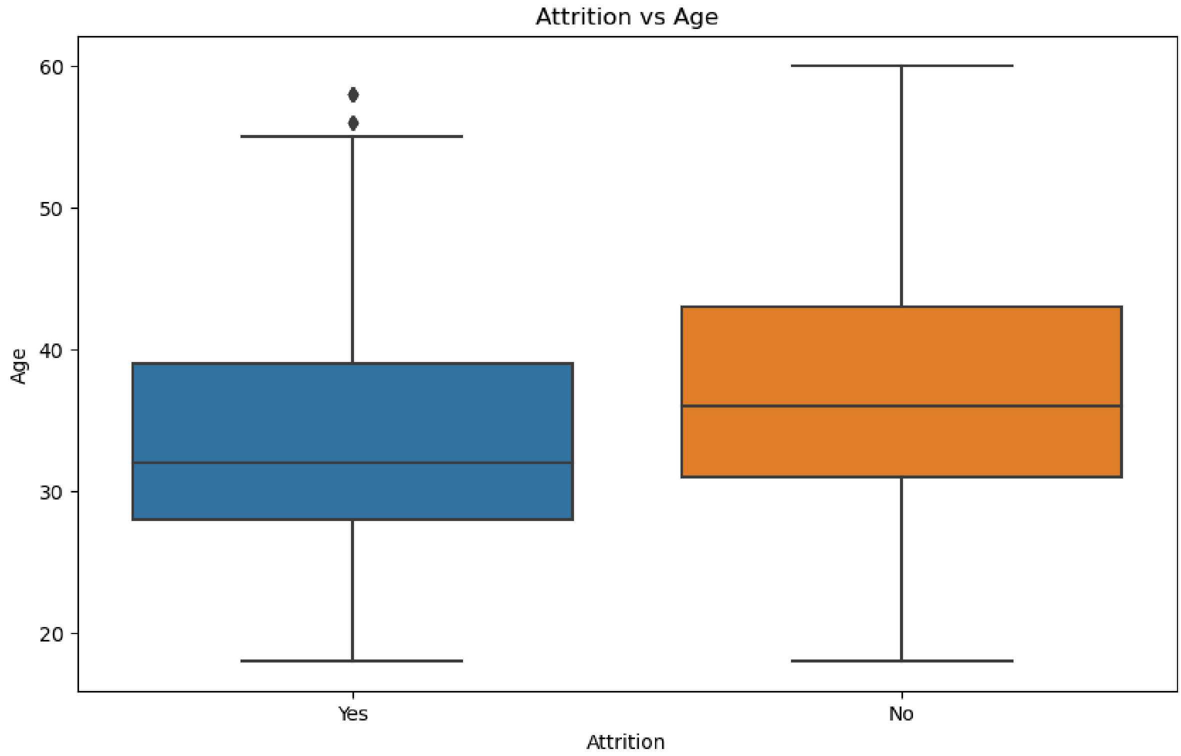
In [14]:
```
# Convert 'Attrition' to numerical values for correlation analysis
data['Attrition_numeric'] = data['Attrition'].apply(lambda x: 1 if x == 'Yes' else
# Calculate the overall attrition rate
attrition_rate = (data['Attrition'] == 'Yes').mean() * 100
print(f"Attrition Rate: {attrition_rate:.2f}%")
```

```
Attrition Rate: 16.12%
```

Result: The overall attrition rate at Green Destinations is approximately 16.12%.

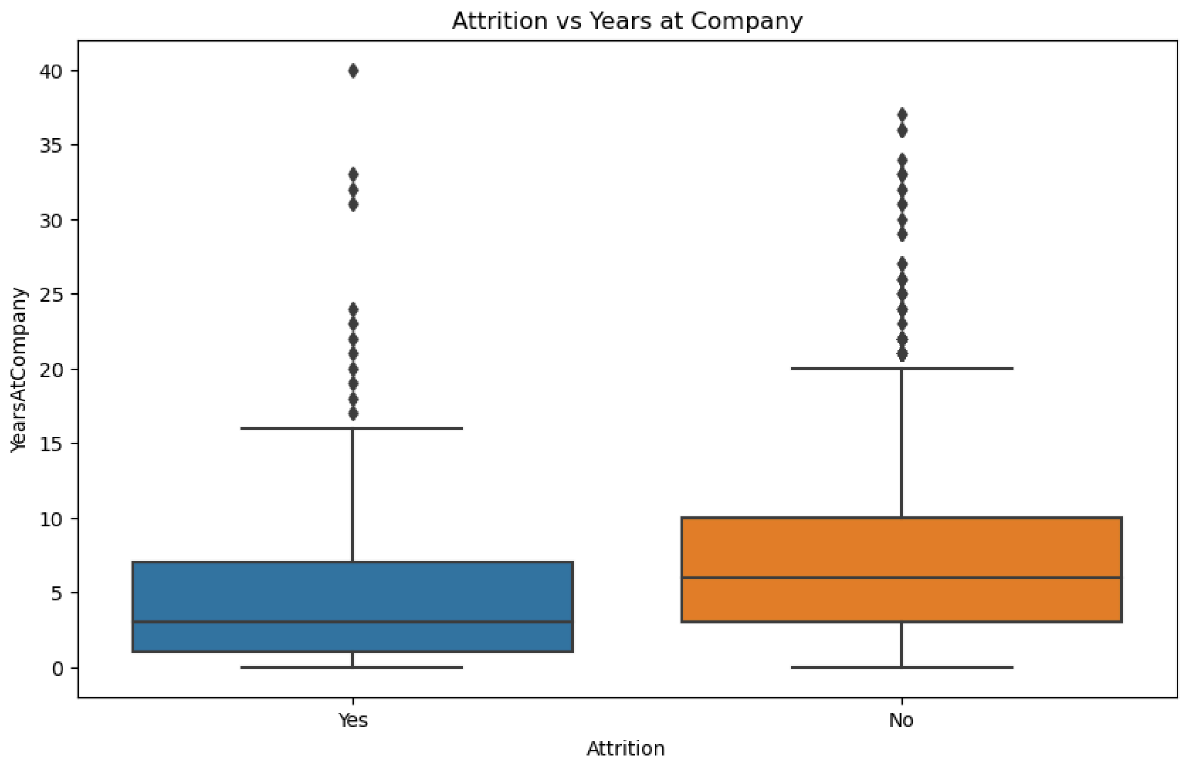
In [7]:
```
# Plot the distribution of age with respect to attrition
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='Age', data=data)
```

```
plt.title('Attrition vs Age')
plt.show()
```



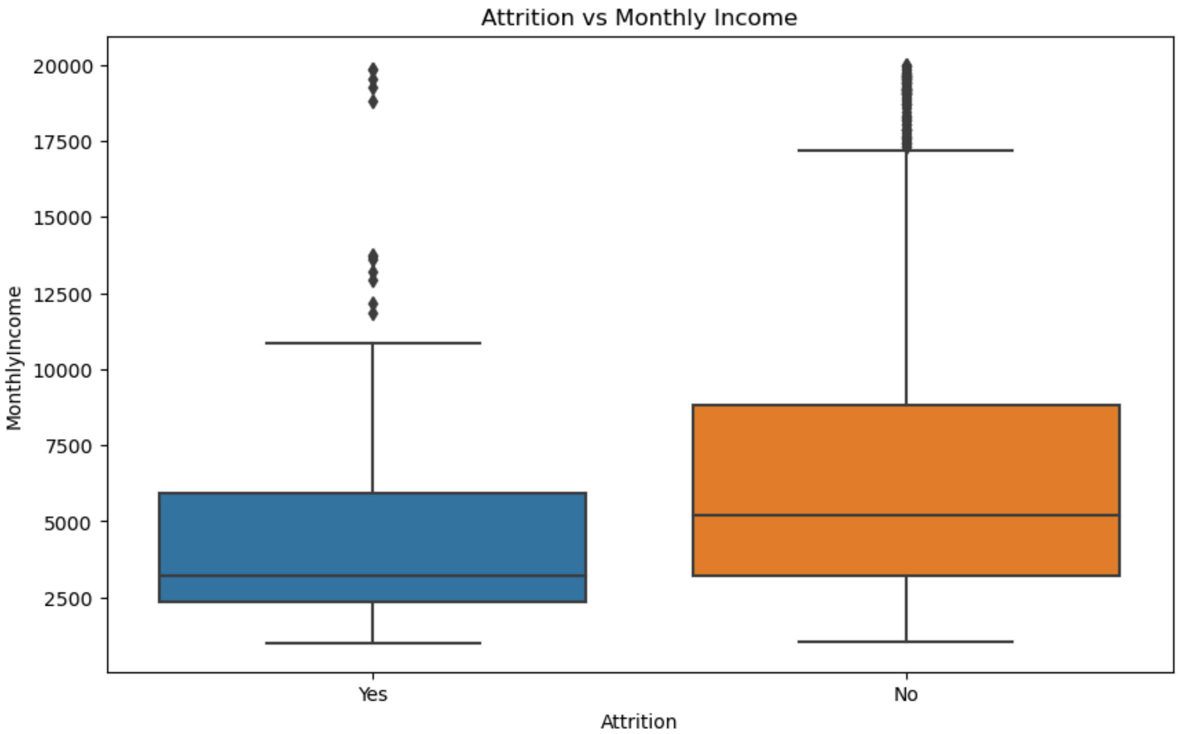
Observation: Employees who left the company tend to be younger on average compared to those who stayed

In [6]:
```
# Plot the distribution of years at the company with respect to attrition
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='YearsAtCompany', data=data)
plt.title('Attrition vs Years at Company')
plt.show()
```



Observation: Employees with fewer years at the company are more likely to leave

In [7]:
```
# Plot the distribution of monthly income with respect to attrition
plt.figure(figsize=(10, 6))
sns.boxplot(x='Attrition', y='MonthlyIncome', data=data)
plt.title('Attrition vs Monthly Income')
plt.show()
```



Observation: Employees with lower monthly income are more likely to leave.

In [8]:
```
# Calculate the correlation matrix to see the numerical relationships
correlation_matrix = data[['Attrition_numeric', 'Age', 'YearsAtCompany', 'MonthlyIr
print(correlation_matrix)
```

```
                   Attrition_numeric       Age  YearsAtCompany  MonthlyIncome
Attrition_numeric           1.000000 -0.159205       -0.134392      -0.159840
Age                        -0.159205  1.000000        0.311309       0.497855
YearsAtCompany             -0.134392  0.311309        1.000000       0.514285
MonthlyIncome              -0.159840  0.497855        0.514285       1.000000
```

Observation: Age, years at the company, and monthly income have weak negative correlations with attrition (-0.159, -0.134, and -0.160 respectively)

In [9]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
```

In [10]:
```
data['Attrition_numeric'] = data['Attrition'].apply(lambda x: 1 if x == 'Yes' else

# Select relevant features and target variable
features = ['Age', 'YearsAtCompany', 'MonthlyIncome']
target = 'Attrition_numeric'

X = data[features]
y = data[target]
```

In [11]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Initialize and fit the logistic regression model
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predict on the test set
y_pred = model.predict(X_test)

# Evaluate the model
print("Classification Report:")
print(classification_report(y_test, y_pred))

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

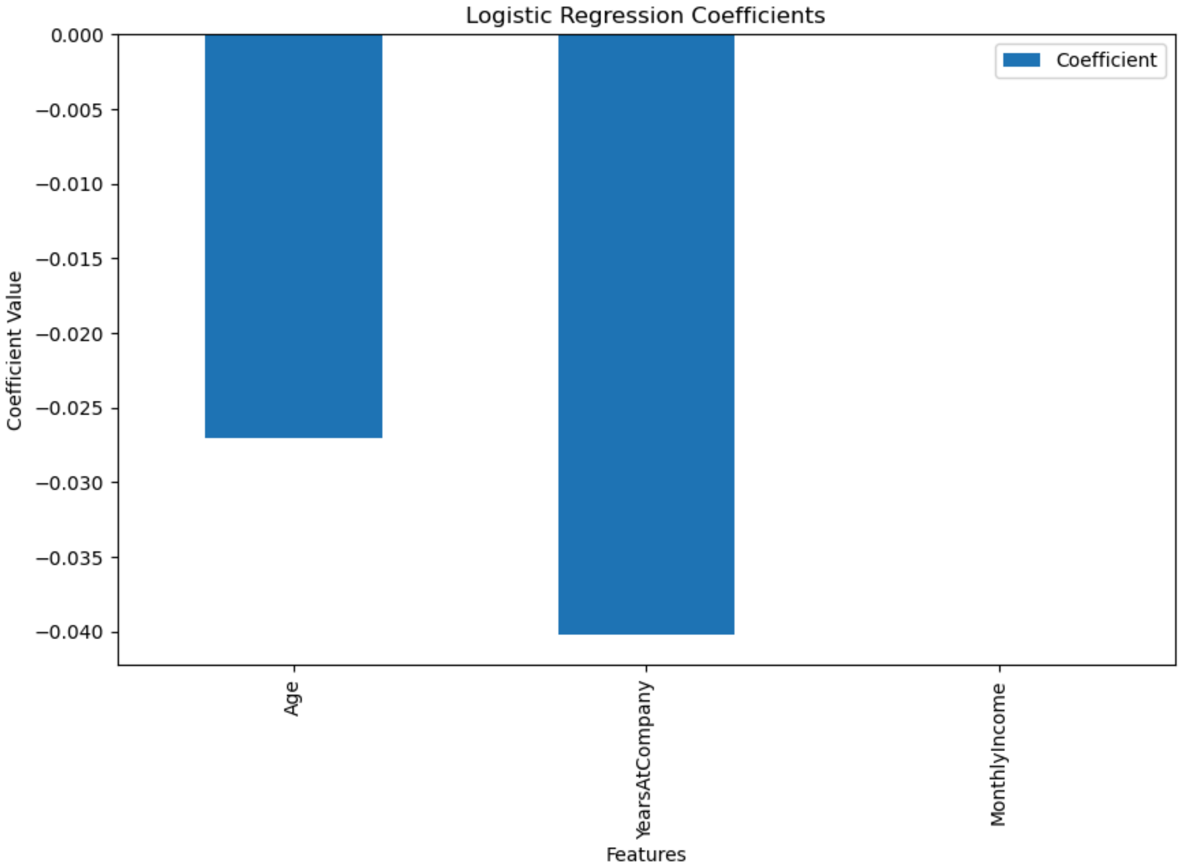
```
Classification Report:
              precision    recall  f1-score   support

           0       0.87      1.00      0.93       255
           1       0.00      0.00      0.00        39

    accuracy                           0.87       294
   macro avg       0.43      0.50      0.46       294
weighted avg       0.75      0.87      0.81       294

Confusion Matrix:
[[255   0]
 [ 39   0]]
```

```
C:\Users\himan\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:146
9: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\himan\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:146
9: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\himan\anaconda3\Lib\site-packages\sklearn\metrics\_classification.py:146
9: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [12]:
```
# Display the coefficients
coefficients = pd.DataFrame(model.coef_.flatten(), index=features, columns=['Coeffi
print("\nLogistic Regression Coefficients:")
print(coefficients)
```

```
# Plot the coefficients for better visualization
coefficients.plot(kind='bar', figsize=(10, 6))
plt.title('Logistic Regression Coefficients')
plt.xlabel('Features')
plt.ylabel('Coefficient Value')
plt.show()
```

```
Logistic Regression Coefficients:
                 Coefficient
Age                -0.027077
YearsAtCompany     -0.040219
MonthlyIncome      -0.000070
```



Result: Logistic regression model is trained to predict attrition. The coefficients indicate the impact of each feature on attrition probability:

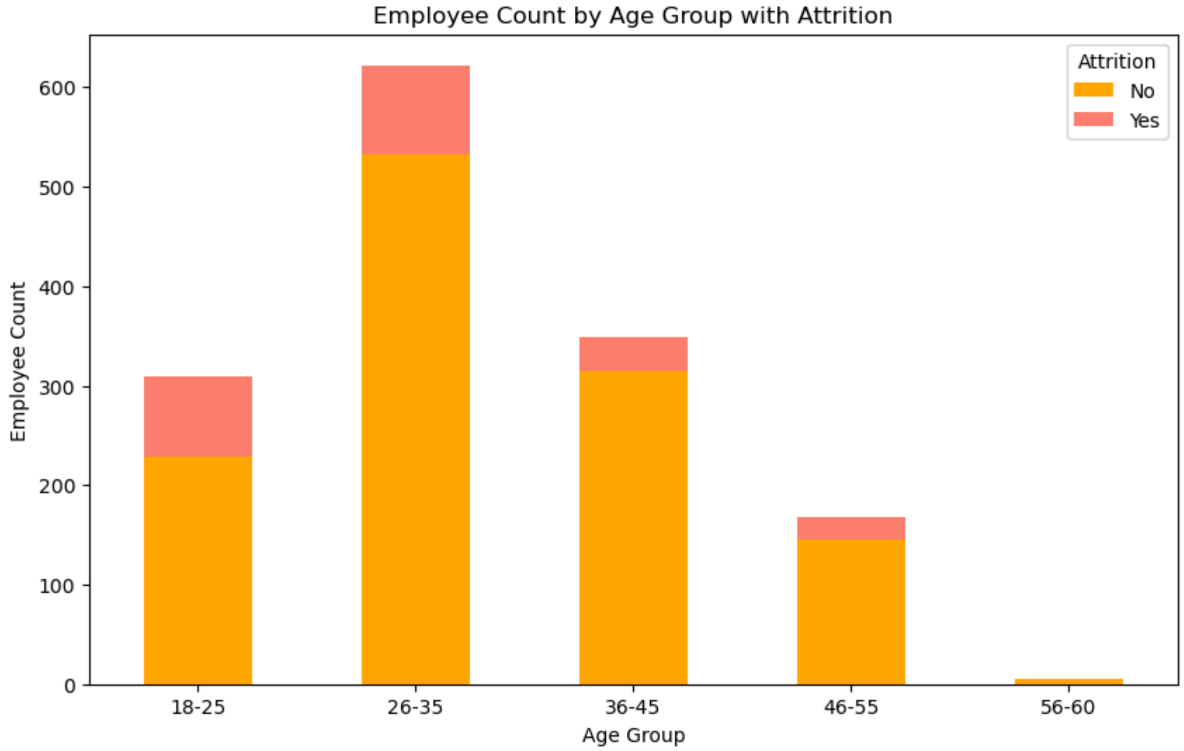
Age: Negative coefficient, implying younger employees are more likely to leave.

Years at Company: Negative coefficient, implying employees with fewer years at the company are more likely to leave.

Monthly Income: Negative coefficient, implying employees with lower income are more likely to leave.

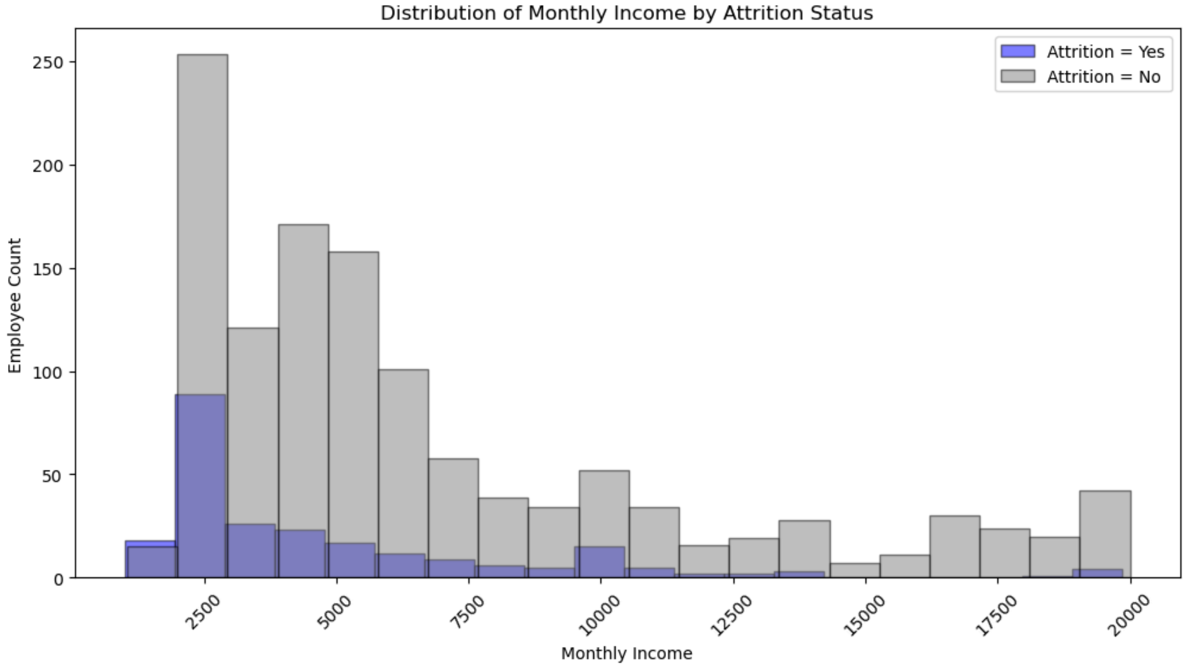
In [18]:
```
# Define age groups
bins = [20, 30, 40, 50, 60, 70]
labels = ['18-25', '26-35', '36-45', '46-55', '56-60']
data['AgeGroup'] = pd.cut(data['Age'], bins=bins, labels=labels, right=False)

# Count the number of employees in each age group by attrition status
age_group_attrition_counts = data.groupby(['AgeGroup', 'Attrition']).size().unstack
```

```
# Plot the bar plot
age_group_attrition_counts.plot(kind='bar', stacked=True, figsize=(10, 6), color=['
plt.title('Employee Count by Age Group with Attrition')
plt.xlabel('Age Group')
plt.ylabel('Employee Count')
plt.xticks(rotation=0)
plt.legend(title='Attrition')
plt.show()
```



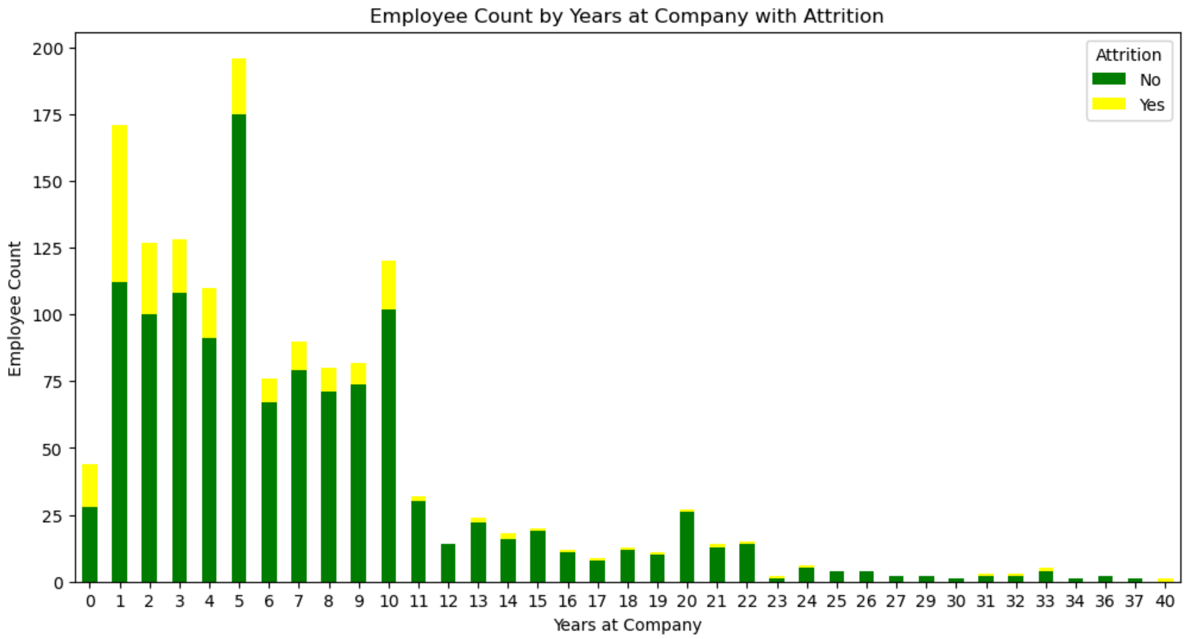
In [26]:

```
# Filter data based on attrition
data_yes = data[data['Attrition'] == 'Yes']
data_no = data[data['Attrition'] == 'No']

# Plot histograms for Monthly Income with attrition information
plt.figure(figsize=(12, 6))

# Histogram for employees who left
plt.hist(data_yes['MonthlyIncome'], bins=20, alpha=0.5, label='Attrition = Yes', co

# Histogram for employees who stayed
plt.hist(data_no['MonthlyIncome'], bins=20, alpha=0.5, label='Attrition = No', colo

plt.title('Distribution of Monthly Income by Attrition Status')
plt.xlabel('Monthly Income')
plt.ylabel('Employee Count')
plt.legend()
plt.xticks(rotation=45)
plt.show()
```



In [29]:

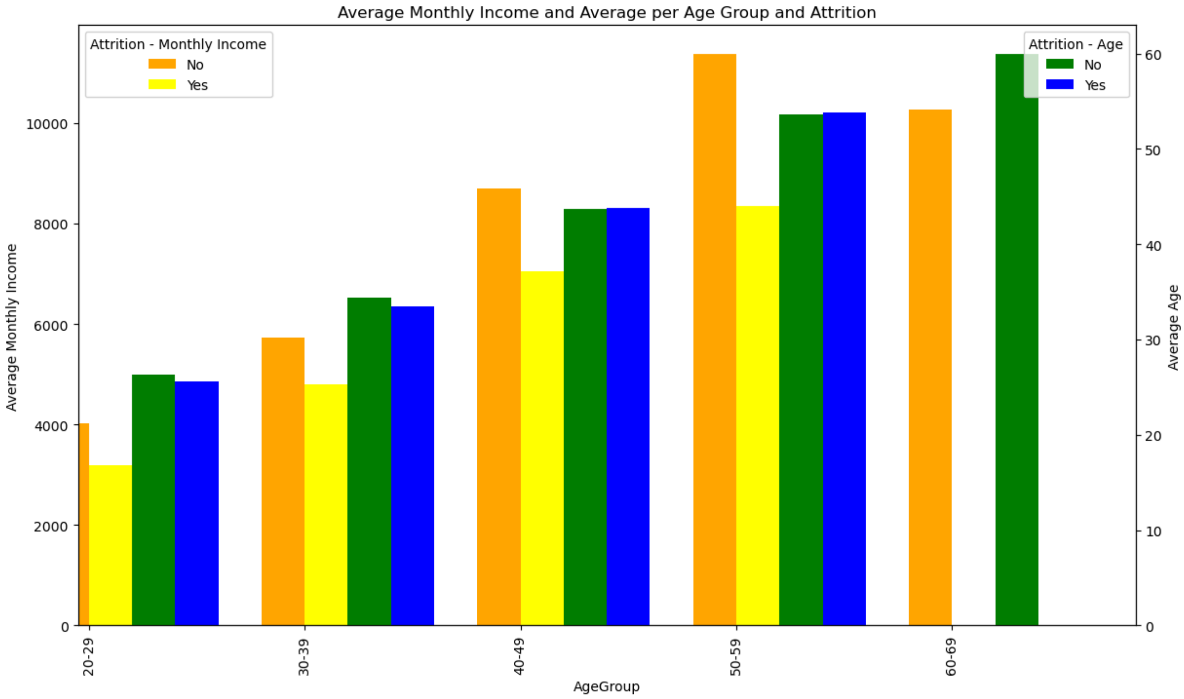
```
# Count the number of employees in each years-at-company group by attrition status
years_at_company_attrition_counts = data.groupby(['YearsAtCompany', 'Attrition']).s

# Plot the bar plot
years_at_company_attrition_counts.plot(kind='bar', stacked=True, figsize=(12, 6), c
plt.title('Employee Count by Years at Company with Attrition')
plt.xlabel('Years at Company')
plt.ylabel('Employee Count')
plt.legend(title='Attrition')
plt.xticks(rotation=0)
plt.show()
```



In [18]:

```
# Define age groups
bins = [20, 30, 40, 50, 60, 70]
labels = ['20-29', '30-39', '40-49', '50-59', '60-69']
data['AgeGroup'] = pd.cut(data['Age'], bins=bins, labels=labels, right=False)

# Group by AgeGroup and Attrition and calculate average Monthly Income and Age
grouped_data = data.groupby(['AgeGroup', 'Attrition']).agg({'MonthlyIncome': 'mean'

# Plotting
```

```
fig, ax1 = plt.subplots(figsize=(14, 8))

# Bar plot for average Monthly Income
grouped_data['MonthlyIncome'].plot(kind='bar', ax=ax1, position=0.5, width=0.4, col
ax1.set_ylabel('Average Monthly Income')
ax1.set_title('Average Monthly Income and Average per Age Group and Attrition')
ax1.legend(title='Attrition - Monthly Income', loc='upper left')

# Create a second y-axis for average Age
ax2 = ax1.twinx()

# Bar plot for average Age
grouped_data['Age'].plot(kind='bar', ax=ax2, position=-0.5, width=0.4, color=['gree
ax2.set_ylabel('Average Age')
ax2.legend(title='Attrition - Age', loc='upper right')

plt.xlabel('Age Group')
plt.xticks(rotation=0)
plt.show()
```



Conclusion

The analysis reveals that younger employees, those with fewer years at the company, and those with lower monthly incomes are more likely to leave Green Destinations.

The logistic regression model supports these findings, indicating that these factors negatively impact retention.

Visualizations such as bar plots and histograms help in understanding the distribution of employees and the impact of attrition across different demographics.

In [ ]: