PROJECT 3 - EMPLOYEE ATTRITION ANALYSIS (HR ANALYTICS)

Ninad Agrawal

CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score ,
confusion_matrix


def load_data(filepath):
    try:
        data = pd.read_csv(filepath)
        print("Data loaded successfully!")
        print(f"Dataset shape: {data.shape}")
        print("The first 5 rowns of the dataset:\n")
        print(data.head())
        return data
    except FileNotFoundError:
        print(f"Error: The file '{filepath}' was not found.")
        return None

def initial_cleaning(df):
    if df is None:
        return None

    print("\n--- Starting Initial Cleaning ---")
    # Dropping columns that have only one unique value (constants) ,
these columns will provide no information
    cols_to_drop = [col for col in df.columns if df[col].nunique() ==
1]
    if cols_to_drop:
        df = df.drop(columns=cols_to_drop)
        print(f"Dropped constant columns: {cols_to_drop}")
```

```
    df['Attrition_Numeric'] = df['Attrition'].map({'Yes': 1, 'No': 0})
#mapped 'Attrition' to numeric values (Yes=1, No=0)

    print("Initial Cleaning Complete")
    return df

def preprocess_and_encode(df):
    if df is None:
        return
    le=LabelEncoder()
    df['Gender_Encoded'] = le.fit_transform(df['Gender'])

    print("\n--- Starting Preprocessing anf One-Hot-Encoding ---")
    categorical_cols = ['BusinessTravel', 'Department',
'EducationField', 'JobRole', 'MaritalStatus', 'OverTime']

    cols_to_encode = [col for col in categorical_cols if col in
df.columns]

    print(f"Original Columns to be encoded: {cols_to_encode} ")
    original_cols = set(df.columns)
    df_encoded = pd.get_dummies(df,columns=cols_to_encode ,
drop_first=True,dtype=int)
    cols_to_drop = ['Gender'] + cols_to_encode
    new_cols = set(df_encoded.columns)
    new_one_hot_cols = list(new_cols - original_cols)
    print(f"\n New one-hot columns created ({len(new_one_hot_cols)}):")
    if len(new_one_hot_cols) > 20:
        print(new_one_hot_cols)
    else:
        print(new_one_hot_cols)

    print('\n--- One-Hot Enconding Complete! ---')
    df_encoded = df_encoded.drop(columns=cols_to_drop ,
errors='ignore')
    return df_encoded

def perform_univariate_analysis(df):
    if df is None:
        return
```

```
    print("\n--- Starting Univariate Analysis ---")
    sns.set_style("whitegrid")

    # Attrition distribution
    plt.figure(figsize=(7, 5))
    sns.countplot(x='Attrition', data=df, palette='pastel' ,
hue='Attrition' , legend=False)
    plt.title('Overall Employee Attrition Distribution', fontsize=16)
    plt.xlabel('Attrition', fontsize=12)
    plt.ylabel('Count', fontsize=12)
    plt.savefig('univariate_attrition_distribution.png',
bbox_inches='tight')
    print("Saved 'univariate_attrition_distribution.png'")

    # Satisfaction Level Distribution
    plt.figure(figsize=(10, 6))
    sns.histplot(df['JobSatisfaction'], kde=True, bins=4,
color='skyblue')
    plt.title('Job Satisfaction Level Distribution', fontsize=16)
    plt.xlabel('Job Satisfaction (1=Low, 4=Very High)', fontsize=12)
    plt.ylabel('Frequency', fontsize=12)
    plt.savefig('univariate_job_satisfaction.png', bbox_inches='tight')
    print("Saved 'univariate_job_satisfaction.png'")

    # Department Distribution
    plt.figure(figsize=(10, 6))
    sns.countplot(y='Department', data=df, palette='viridis',
order=df['Department'].value_counts().index , hue='Department' ,
legend=False)
    plt.title('Employee Distribution by Department', fontsize=16)
    plt.xlabel('Count', fontsize=12)
    plt.ylabel('Department', fontsize=12)
    plt.savefig('univariate_department_distribution.png',
bbox_inches='tight')
    print("Saved 'univariate_department_distribution.png'")

    print("--- Univariate Analysis Complete ---")

def perform_bivariate_analysis(df):
    if df is None:
        return

    print("\n--- Starting Bivariate Analysis (vs. Attrition) ---")
```

```
    # Attrition vs. Salary
    plt.figure(figsize=(12, 7))
    sns.boxplot(x='Attrition', y='MonthlyIncome', data=df,
palette='coolwarm' , hue='Attrition' , legend=False)
    plt.title('Monthly Income vs. Attrition', fontsize=16)
    plt.xlabel('Attrition', fontsize=12)
    plt.ylabel('Monthly Income', fontsize=12)
    plt.savefig('bivariate_attrition_vs_income.png',
bbox_inches='tight')
    print("Saved 'bivariate_attrition_vs_income.png'")

    # Attrition vs. Department
    plt.figure(figsize=(10, 6))

    dept_attrition =
df.groupby('Department')['Attrition_Numeric'].mean().reset_index().sort
_values(by='Attrition_Numeric', ascending=False)
    sns.barplot(x='Attrition_Numeric', y='Department',
data=dept_attrition, palette='plasma' , hue='Department' ,
legend=False)
    plt.title('Attrition Rate by Department', fontsize=16)
    plt.xlabel('Attrition Rate', fontsize=12)
    plt.ylabel('Department', fontsize=12)
    plt.savefig('bivariate_attrition_vs_department.png',
bbox_inches='tight')
    print("Saved 'bivariate_attrition_vs_department.png'")

    # Bivariate: Attrition vs. YearsAtCompany
    plt.figure(figsize=(12, 7))
    sns.kdeplot(df[df['Attrition'] == 'Yes']['YearsAtCompany'],
label='Attrition: Yes', fill=True, color='red')
    sns.kdeplot(df[df['Attrition'] == 'No']['YearsAtCompany'],
label='Attrition: No', fill=True, color='blue')
    plt.title('Years at Company vs. Attrition', fontsize=16)
    plt.xlabel('Years at Company', fontsize=12)
    plt.ylabel('Density', fontsize=12)
    plt.legend()
    plt.savefig('bivariate_attrition_vs_years_at_company.png',
bbox_inches='tight')
    print("Saved 'bivariate_attrition_vs_years_at_company.png'")

    print("--- Bivariate Analysis Complete ---")
```

```
def generate_correlation_heatmap(df):
    if df is None:
        return

    print("\n--- Generating Correlation Heatmap ---")

    df_corr = df.copy()
    if 'Attribution' in df_corr.columns:
        df_corr = df_corr.drop(columns=['Attrition'])

    numeric_df = df_corr.select_dtypes(include=np.number)

    if numeric_df.empty:
        print("Error: No Numeric Data found for correlation Heatmap")

    plt.figure(figsize=(20, 16))
    correlation_matrix = numeric_df.corr()
    sns.heatmap(correlation_matrix, annot=True, fmt='.2f',
cmap='coolwarm', annot_kws={"size": 8} )
    plt.title('Correlation Heatmap of All Features', fontsize=20)
    plt.xticks(rotation=90)
    plt.yticks(rotation=0)
    plt.savefig('correlation_heatmap.png', bbox_inches='tight')
    print("Saved 'correlation_heatmap.png'")

    # Show top correlations with Attrition
    attrition_corr =
correlation_matrix['Attrition_Numeric'].sort_values(ascending=False)
    print("\nTop features correlated with Attrition (Positive):")
    print(attrition_corr[attrition_corr > 0][1:].head(10)) # Skip
itself
    print("\nTop features correlated with Attrition (Negative):")
    print(attrition_corr[attrition_corr <
0][1:].sort_values(ascending=True).head(10))

    print("--- Correlation Analysis Complete ---")

def derive_specific_insight(df):
    if df is None:
        return

    print("\n--- Deriving Specific Insight ---")
```

```
    # using 1st quartile (bottom 25%) as the definition for "low
income"
    low_income_threshold = df['MonthlyIncome'].quantile(0.25)
    print(f"Defining 'Low Income' as <= ${low_income_threshold:.2f}
(1st Quartile)")

    # 2. Create the segments
    df['Experience_Group'] = pd.cut(df['YearsAtCompany'], bins=[-1, 2,
float('inf')], labels=['<3 Years', '3+ Years'])
    df['Income_Group'] = pd.cut(df['MonthlyIncome'], bins=[-1,
low_income_threshold, float('inf')], labels=['Low Income', 'Not Low
Income'])

    # 3. Analyze the target group
    target_group = df[
        (df['Experience_Group'] == '<3 Years') &
        (df['Income_Group'] == 'Low Income')
    ]

    if not target_group.empty:
        overall_attrition_rate = df['Attrition_Numeric'].mean()
        target_group_attrition_rate =
target_group['Attrition_Numeric'].mean()

        print(f"\nOverall Attrition Rate:
{overall_attrition_rate:.2%}")
        print(f"Target Group (<3 Yrs Exp, Low Income) Attrition Rate:
{target_group_attrition_rate:.2%}")

        if target_group_attrition_rate > overall_attrition_rate:
            print("\nInsight Confirmed: Attrition is significantly
higher among employees with < 3 years of experience and low income.")
        else:
            print("\nInsight Not Confirmed: The target group does not
have a higher attrition rate than the overall average.")

        insight_data = df.groupby(['Experience_Group', 'Income_Group']
, observed=False)['Attrition_Numeric'].mean().reset_index()
        plt.figure(figsize=(10, 6))
        sns.barplot(x='Experience_Group', y='Attrition_Numeric',
hue='Income_Group', data=insight_data, palette='Set1')
```

```
        plt.title('Attrition Rate by Experience and Income',
fontsize=16)
        plt.ylabel('Attrition Rate', fontsize=12)
        plt.xlabel('Experience Group', fontsize=12)
        plt.axhline(y=overall_attrition_rate, color='blue',
linestyle='--', label=f'Overall Avg ({overall_attrition_rate:.2%})')
        plt.legend()
        plt.savefig('insight_experience_vs_income.png',
bbox_inches='tight')
        print("Saved 'insight_experience_vs_income.png'")

    else:
        print("Could not analyze the target group (no data found for
this segment).")

    print("--- Specific Insight Analysis Complete ---")

def logistic_regression_model(df):
    if df is None:
        return

    print("\n--- Building a Logistic Regression Model ---")

    y = df['Attrition_Numeric']
    X = df.drop(columns=['Attrition_Numeric','Attition'
,'EmployeeNumber'],errors='ignore')
    X = X.select_dtypes(include=np.number)
    X = X.fillna(0)

    print(f"\nTraining Model with {len(X.columns)} features.")

    X_train , X_test, y_train , y_test =
train_test_split(X,y,test_size=0.2,random_state=42)

    scaler=StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    model=LogisticRegression(random_state=42,class_weight='balanced' ,
max_iter=1000)
    model.fit(X_train_scaled , y_train)

    y_pred = model.predict_proba(X_test_scaled)[: , 1] # for 'Yes'
```

```
    # print(len(y_pred))
    # print(y_test)
    boolarr = (y_pred > 0.66)
    y_pred = boolarr.astype(int)


    print("\nModel Evaluation Results:")


    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy:.2%}")


    print("\nConfusion Matrix:")
    cm = confusion_matrix(y_test, y_pred)

    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
                xticklabels=['Predicted No', 'Predicted Yes'],
                yticklabels=['Actual No', 'Actual Yes'])
    plt.title('Confusion Matrix', fontsize=14)
    plt.savefig('confusion_matrix.png', bbox_inches='tight')
    print("Saved 'confusion_matrix.png'")
    print(cm)


    print("\nClassification Report:")
    report = classification_report(y_test, y_pred, target_names=['No
(0)', 'Yes (1)'])
    print(report)

    print("--- Model Building Complete ---")




def main():
    FILEPATH = './WA_Fn-UseC_-HR-Employee-Attrition.csv'
    data = load_data(FILEPATH)

    if data is not None:

        data = initial_cleaning(data.copy())
```

```
        perform_univariate_analysis(data.copy())
        perform_bivariate_analysis(data.copy())
        derive_specific_insight(data.copy())
        df_encoded = preprocess_and_encode(data)
        generate_correlation_heatmap(df_encoded.copy())
        logistic_regression_model(df_encoded.copy())


        print("\n\nAnalysis complete!!")

if __name__ == "__main__":
    main()
```

TERMINAL OUTPUT -

"C:/Program Files/Python312/python.exe"
c:/Users/Administrator/Desktop/Training/Milestone1_17112025/milestone1.py
Data loaded successfully!
Dataset shape: (1470, 35)
The first 5 rowns of the dataset:

Age Attrition BusinessTravel DailyRate Department ... WorkLifeBalance YearsAtCompany YearsInCurrentRole YearsSinceLastPromotion YearsWithCurrManager
0 41 Yes Travel_Rarely 1102 Sales ... 1 6 4 0 5
1 49 No Travel_Frequently 279 Research & Development ... 3 10 7 1 7
2 37 Yes Travel_Rarely 1373 Research & Development ... 3 0 0 0 0
3 33 No Travel_Frequently 1392 Research & Development ... 3 8 7 3 0
4 27 No Travel_Rarely 591 Research & Development ... 3 2 2 2 2

[5 rows x 35 columns]

--- Starting Initial Cleaning ---
Dropped constant columns: ['EmployeeCount', 'Over18', 'StandardHours']

Initial Cleaning Complete

--- Starting Univariate Analysis ---
Saved 'univariate_attrition_distribution.png'
Saved 'univariate_job_satisfaction.png'
Saved 'univariate_department_distribution.png'
--- Univariate Analysis Complete ---

--- Starting Bivariate Analysis (vs. Attrition) ---
Saved 'bivariate_attrition_vs_income.png'
Saved 'bivariate_attrition_vs_department.png'
Saved 'bivariate_attrition_vs_years_at_company.png'
--- Bivariate Analysis Complete ---

--- Deriving Specific Insight ---
Defining 'Low Income' as <= $2911.00 (1st Quartile)

Overall Attrition Rate: 16.12%
Target Group (<3 Yrs Exp, Low Income) Attrition Rate: 38.60%

Insight Confirmed: Attrition is significantly higher among employees with < 3 years of experience and low income.
Saved 'insight_experience_vs_income.png'
--- Specific Insight Analysis Complete ---

--- Starting Preprocessing anf One-Hot-Encoding ---
Original Columns to be encoded: ['BusinessTravel', 'Department', 'EducationField', 'JobRole', 'MaritalStatus', 'OverTime']

New one-hot columns created (20):
['EducationField_Marketing', 'MaritalStatus_Married', 'JobRole_Laboratory Technician', 'JobRole_Sales Executive', 'Department_Research & Development', 'EducationField_Technical Degree', 'Department_Sales', 'JobRole_Sales Representative', 'JobRole_Manager', 'EducationField_Other', 'JobRole_Research Director', 'JobRole_Manufacturing Director', 'EducationField_Medical', 'MaritalStatus_Single', 'OverTime_Yes', 'JobRole_Research Scientist', 'EducationField_Life Sciences', 'BusinessTravel_Travel_Frequently', 'JobRole_Human Resources', 'BusinessTravel_Travel_Rarely']

--- One-Hot Enconding Complete! ---

--- Generating Correlation Heatmap ---
Saved 'correlation_heatmap.png'

Top features correlated with Attrition (Positive):
OverTime_Yes 0.246118
MaritalStatus_Single 0.175419
JobRole_Sales Representative 0.157234

```
BusinessTravel_Travel_Frequently    0.115143
JobRole_Laboratory Technician       0.098290
Department_Sales                    0.080855
DistanceFromHome                    0.077924
EducationField_Technical Degree     0.069355
EducationField_Marketing            0.055781
NumCompaniesWorked                  0.043494
Name: Attrition_Numeric, dtype: float64

Top features correlated with Attrition (Negative):
TotalWorkingYears      -0.171063
JobLevel               -0.169105
YearsInCurrentRole     -0.160545
MonthlyIncome          -0.159840
Age                    -0.159205
YearsWithCurrManager   -0.156199
StockOptionLevel       -0.137145
YearsAtCompany         -0.134392
JobInvolvement         -0.130016
JobSatisfaction        -0.103481
Name: Attrition_Numeric, dtype: float64
--- Correlation Analysis Complete ---

--- Building a Logistic Regression Model ---

Training Model with 44 features.

Model Evaluation Results:
Accuracy: 81.29%

Confusion Matrix:
Saved 'confusion_matrix.png'
[[219  36]
 [ 19  20]]

Classification Report:
              precision    recall  f1-score   support

      No (0)       0.92      0.86      0.89       255
     Yes (1)       0.36      0.51      0.42        39

    accuracy                           0.81       294
   macro avg       0.64      0.69      0.65       294
weighted avg       0.85      0.81      0.83       294

--- Model Building Complete ---
```

Analysis complete!!

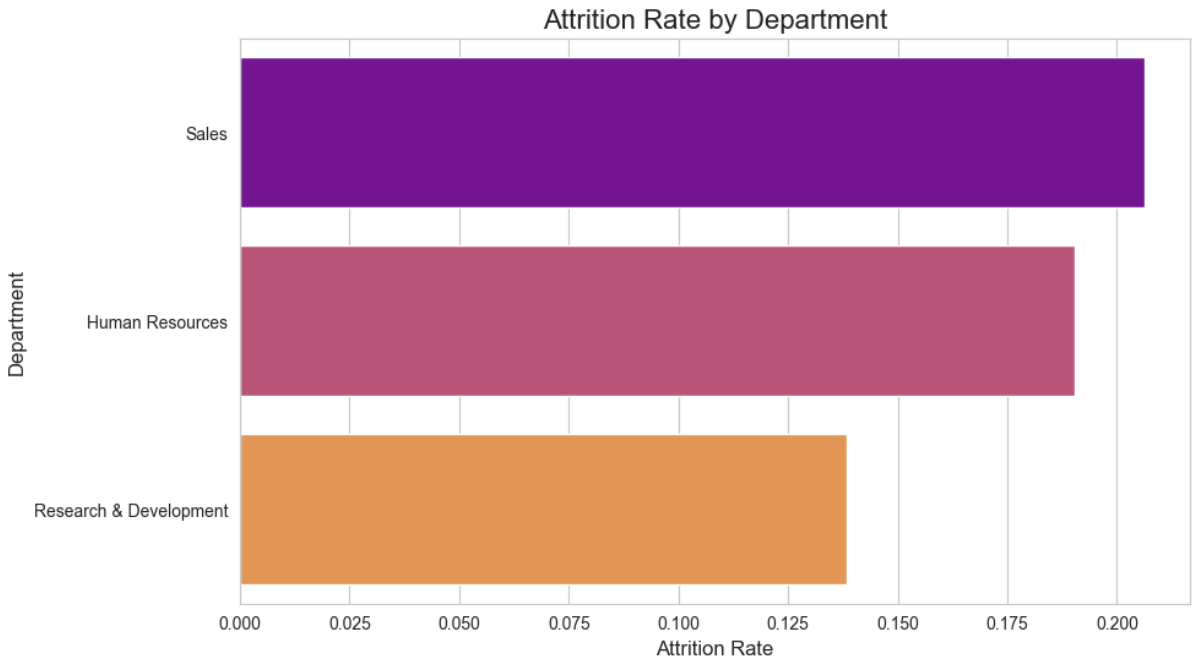
INSIGHTS DERIVED -

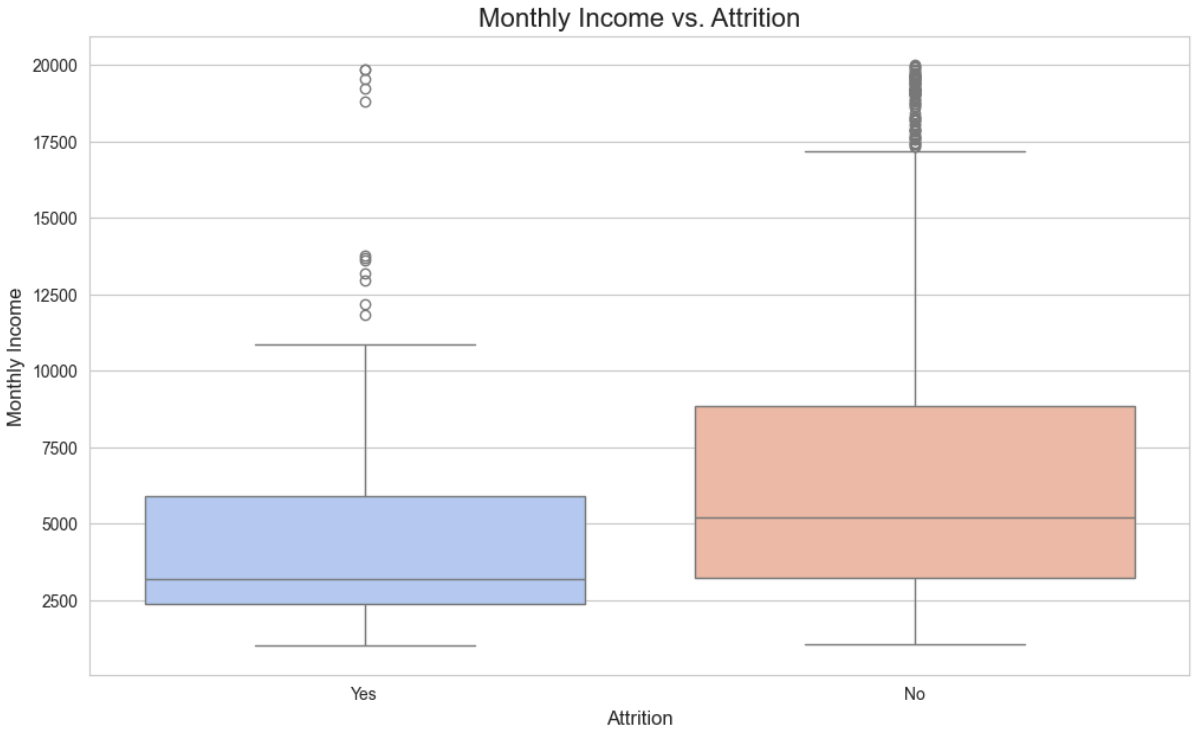
--- Deriving Specific Insight ---
Defining 'Low Income' as <= $2911.00 (1st Quartile)

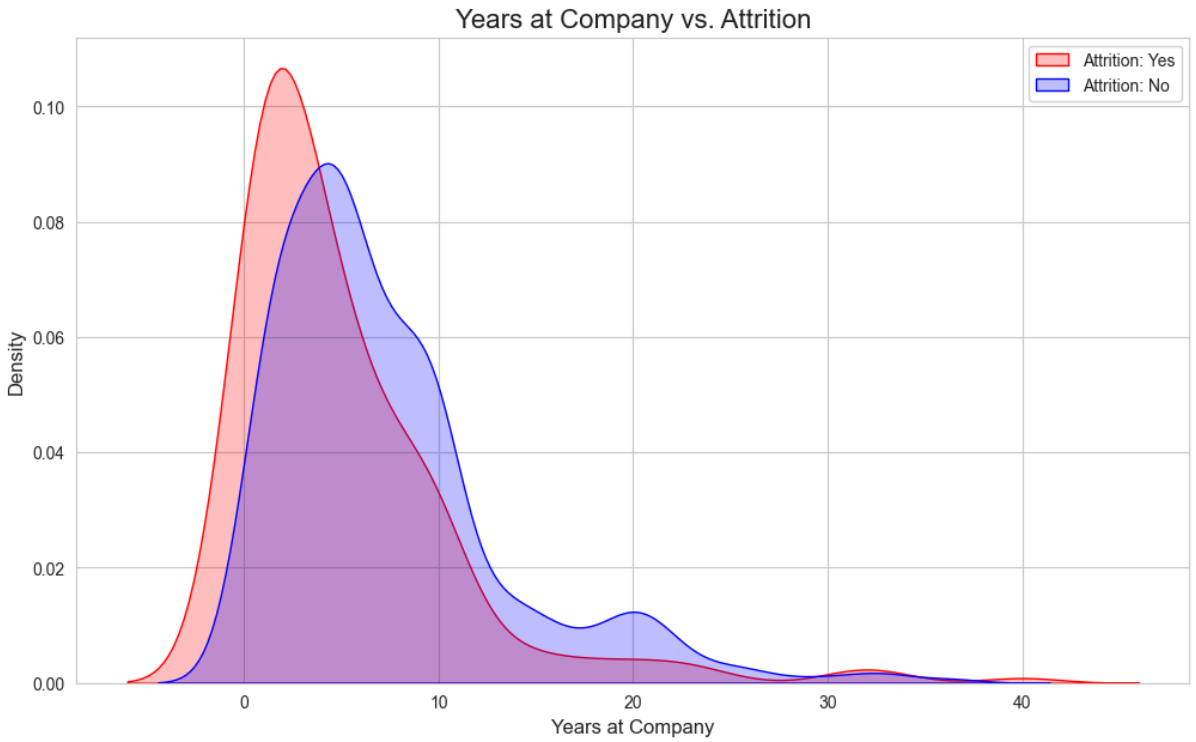
Overall Attrition Rate: 16.12%
Target Group (<3 Yrs Exp, Low Income) Attrition Rate: 38.60%

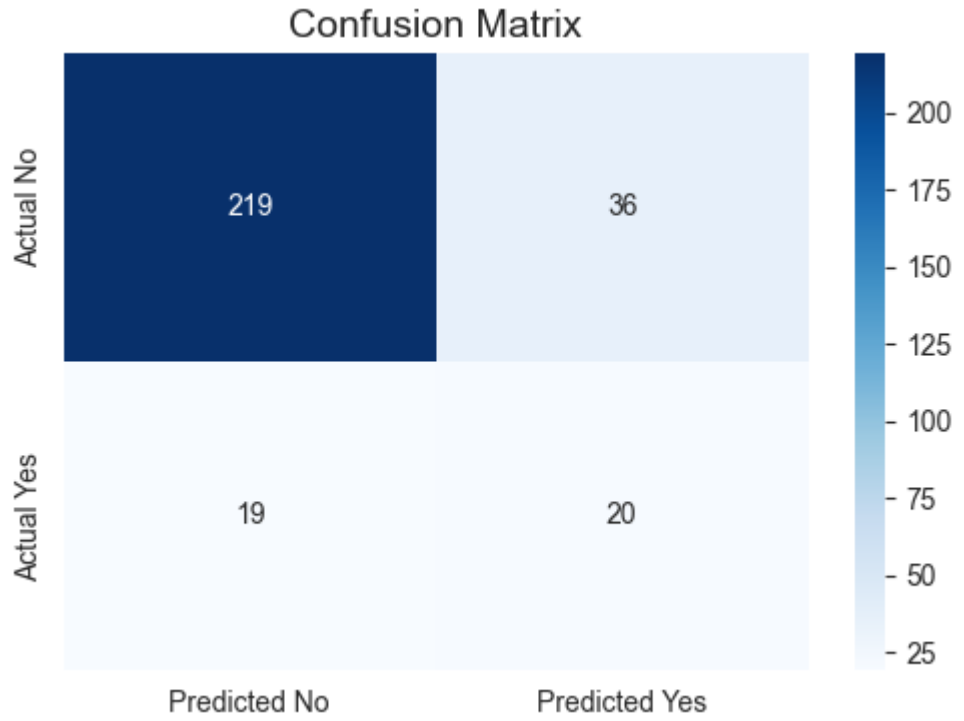
Insight Confirmed: Attrition is significantly higher among employees with < 3 years of experience and low income.
Saved 'insight_experience_vs_income.png'
--- Specific Insight Analysis Complete —

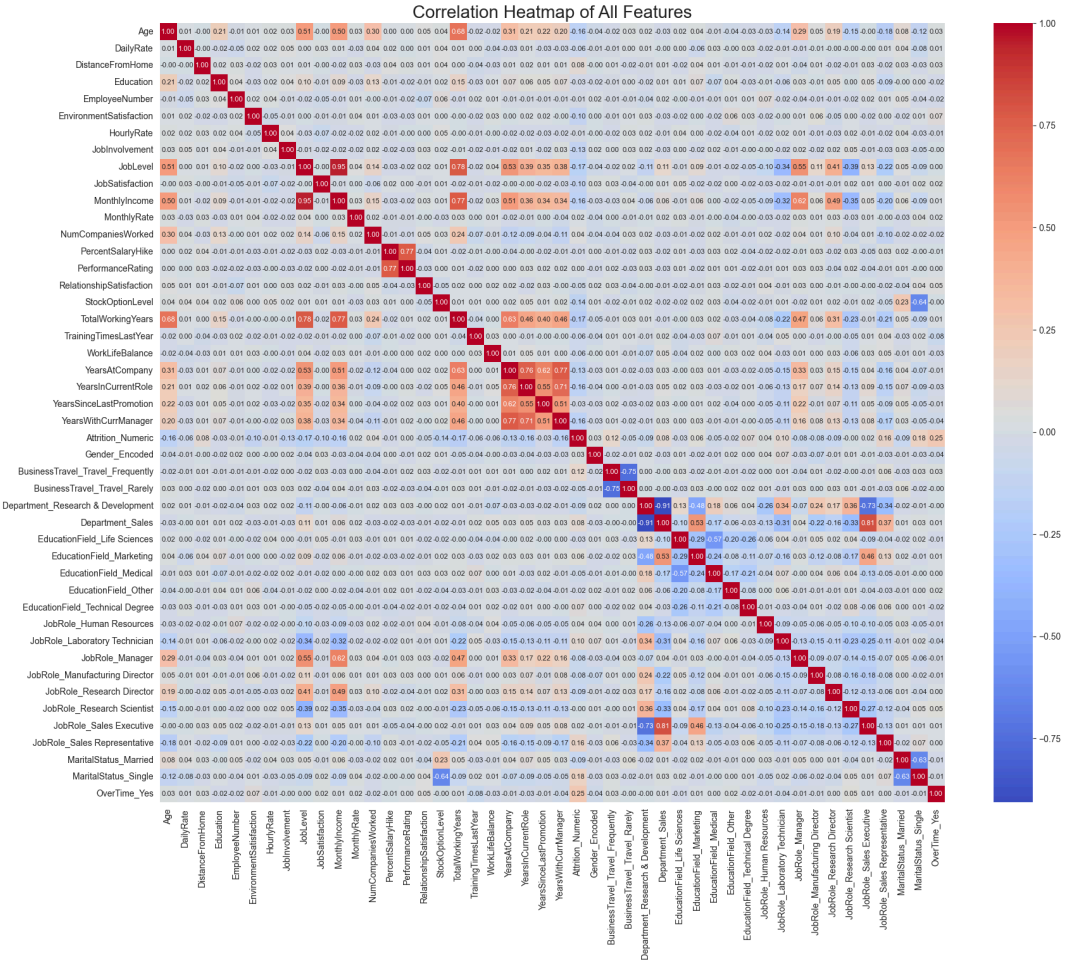
OUTPUT VISUALIZATIONS -

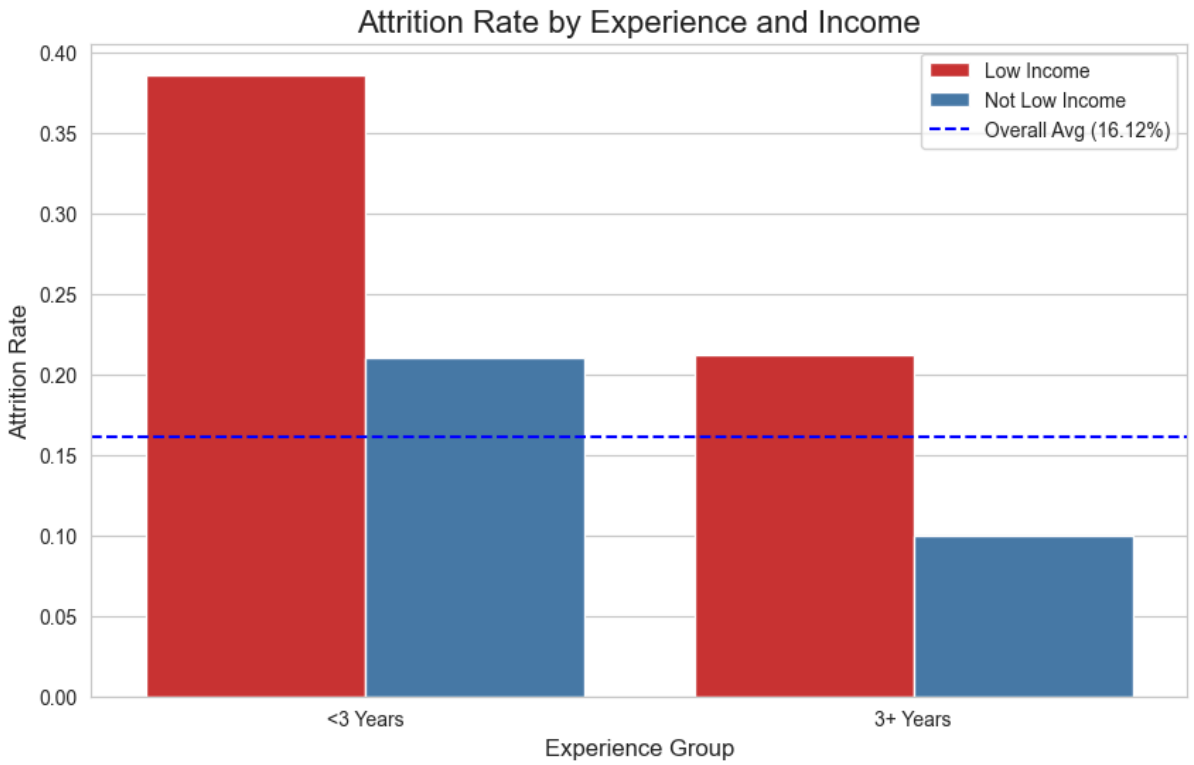

Attrition Rate by Department
Sales
Human Resources
Research & Development
Department
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200
Attrition Rate

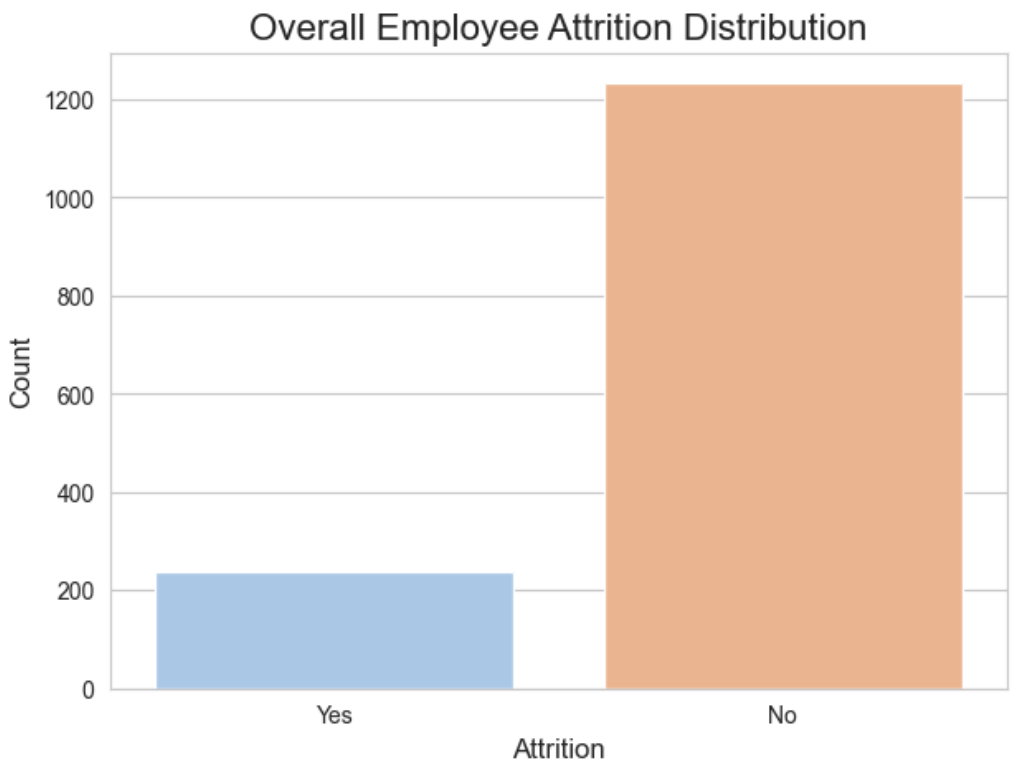

Monthly Income vs. Attrition
20000
17500
15000
12500
10000
7500
5000
2500
Monthly Income
Yes
No
Attrition

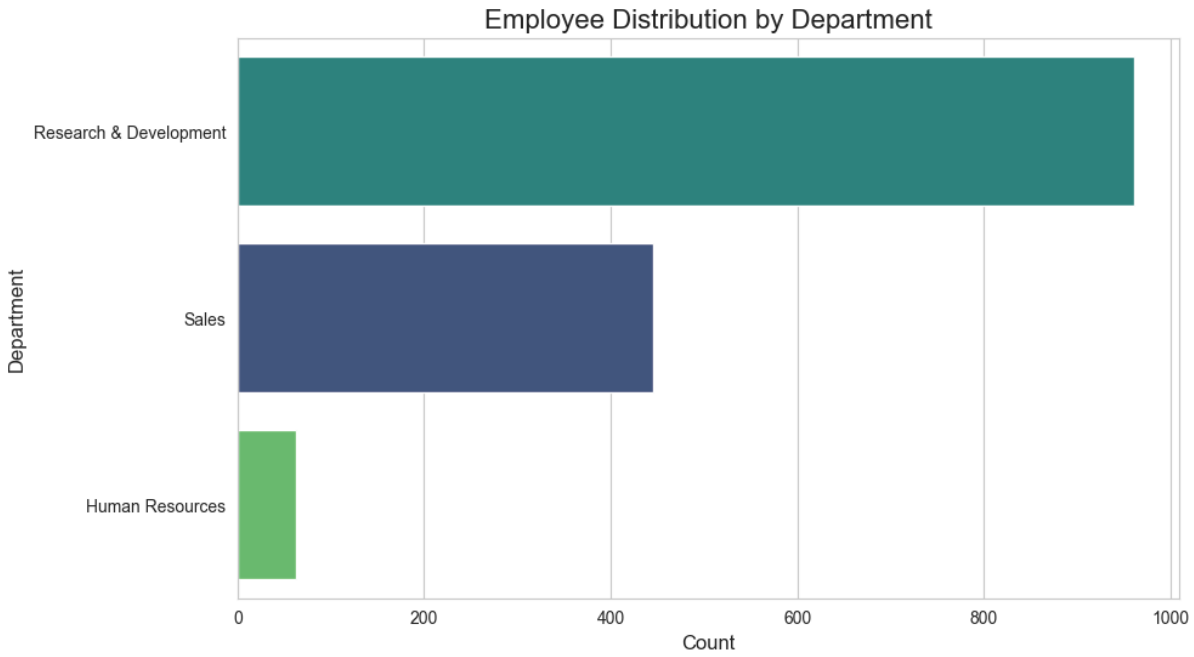

Years at Company vs. Attrition
Attrition: Yes
Attrition: No
Density
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
Years at Company

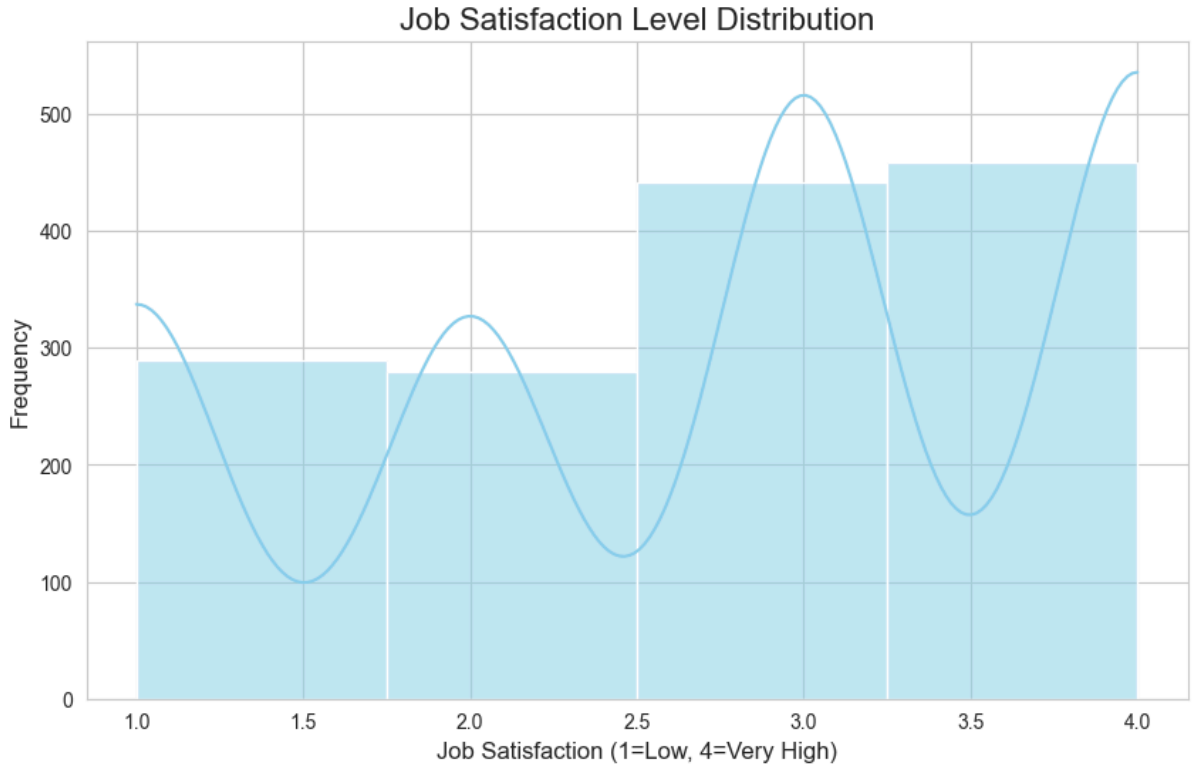

Confusion Matrix
Actual No
Actual Yes
219
36
19
20
Predicted No
Predicted Yes
200
175
150
125
100
75
50
25


Correlation Heatmap of All Features
Age
DailyRate
DistanceFromHome
Education
EmployeeNumber
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
Attrition_Numeric
Gender_Encoded
BusinessTravel_Travel_Frequently
BusinessTravel_Travel_Rarely
Department_Research & Development
Department_Sales
EducationField_Life Sciences
EducationField_Marketing
EducationField_Medical
EducationField_Other
EducationField_Technical Degree
JobRole_Human Resources
JobRole_Laboratory Technician
JobRole_Manager
JobRole_Manufacturing Director
JobRole_Research Director
JobRole_Research Scientist
JobRole_Sales Executive
JobRole_Sales Representative
MaritalStatus_Married
MaritalStatus_Single
OverTime_Yes
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75


Attrition Rate by Experience and Income
Attrition Rate
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Low Income
Not Low Income
Overall Avg (16.12%)
<3 Years
3+ Years
Experience Group


Overall Employee Attrition Distribution
1200
1000
800
600
400
200
0
Count
Yes
No
Attrition


Employee Distribution by Department
Research & Development
Sales
Human Resources
Department
0
200
400
600
800
1000
Count


Job Satisfaction Level Distribution
Frequency
0
100
200
300
400
500
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Job Satisfaction (1=Low, 4=Very High)