## Perform Exploratory Data Analysis (EDA):

The initial stage involves data preprocessing, where we prioritize operations without scaling. Scaling may introduce interpretational challenges. The key activities in this stage include importing essential libraries, loading the dataset, eliminating constant columns, and encoding non-numeric features.

1. T-Tests for Attrition Analysis

Calculate T-stat and p-value for all features in the dataset to determine if the difference in means between employees who left (Attrition: Yes) and those who stayed (Attrition: No) is statistically significant.

```
from scipy.stats import ttest_ind

features_to_test = list(df.columns)

for feature in features_to_test:
    attrition_yes = df[df['Attrition'] == 1][feature]
    attrition_no = df[df['Attrition'] == 0][feature]

    t_stat, p_value = ttest_ind(attrition_yes, attrition_no)

    print(f"T-Test for {feature} and Attrition:")
    print(f"T-Stat: {t_stat}, P-Value: {p_value}")
    if p_value < 0.05:
        print(f"The difference in {feature} means between Attrition: Yes and Attrition: No is statistically significant.")
    else:
        print(f"The difference in {feature} means between Attrition: Yes and Attrition: No is not statistically significant.")
    print()
```

| Feature | T-Stat | P-Value | Interpretation |
|---|---|---|---|
| Age | -6.1786638 | 8.3563e-10 | Statistically significant |
| BusinessTravel | 4.9059 | 1.0335e-06 | Statistically significant |
| DailyRate | -2.1741 | 0.0299 | Statistically significant |
| Department | -2.9726 | 0.0030 | Statistically significant |
| DistanceFromHome | 2.9947 | 0.0028 | Statistically significant |
| Education | -1.2026 | 0.2293 | Not statistically significant |
| EducationField | 1.9784 | 0.0481 | Statistically significant |
| EmployeeNumber | -0.4053 | 0.6853 | Not statistically significant |
| EnvironmentSatisfaction | -3.9819 | 7.1723e-05 | Statistically significant |
| Gender | -1.1290 | 0.2591 | Not statistically significant |
| HourlyRate | -0.2623 | 0.7931 | Not statistically significant |
| JobInvolvement | -5.0241 | 5.6771e-07 | Statistically significant |
| JobLevel | -6.5738 | 6.7954e-11 | Statistically significant |
| JobRole | 0.1040 | 0.9171 | Not statistically significant |

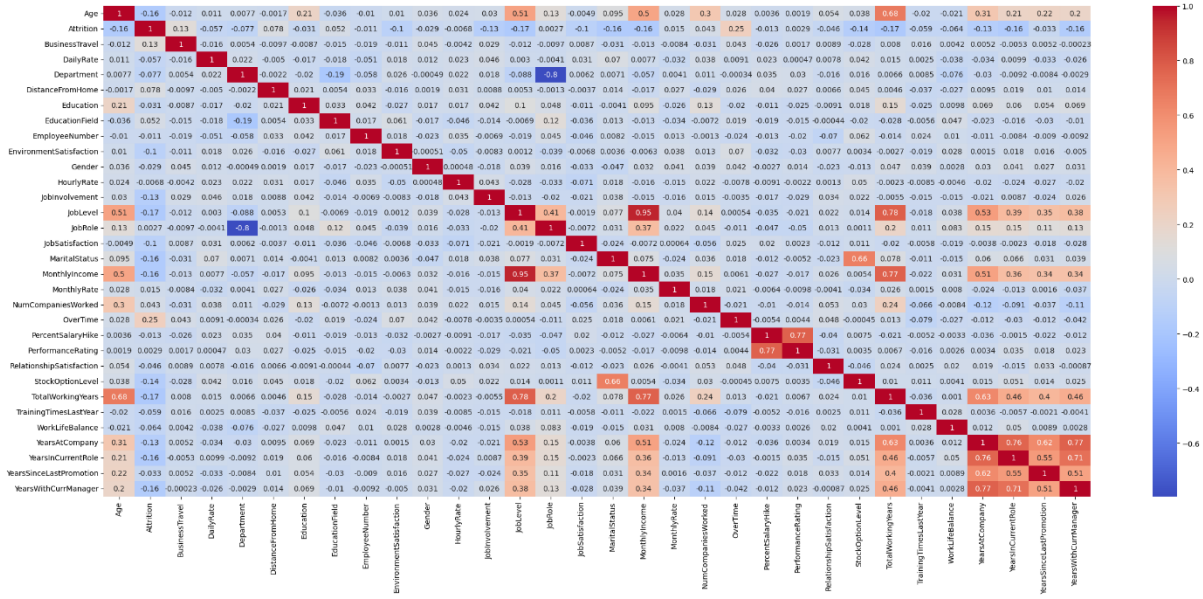
| | | | |
|---|---|---|---|
| JobSatisfaction | -3.9862 | 7.0431e-05 | Statistically significant |
| Feature | T-Stat | P-Value | Interpretation |
| MaritalStatus | -6.2928 | 4.1060e-10 | Statistically significant |
| MonthlyIncome | -6.2039 | 7.1474e-10 | Statistically significant |
| MonthlyRate | 0.5813 | 0.5611 | Not statistically significant |
| NumCompaniesWorked | 1.6680 | 0.0955 | Not statistically significant |
| OverTime | 9.7292 | 1.0093e-21 | Statistically significant |
| PercentSalaryHike | -0.5165 | 0.6056 | Not statistically significant |
| PerformanceRating | 0.1107 | 0.9119 | Not statistically significant |
| RelationshipSatisfaction | -1.7594 | 0.0787 | Not statistically significant |
| StockOptionLevel | -5.3048 | 1.3010e-07 | Statistically significant |
| TotalWorkingYears | -6.6523 | 4.0619e-11 | Statistically significant |
| TrainingTimesLastYear | -2.2829 | 0.0226 | Statistically significant |
| WorkLifeBalance | -2.4548 | 0.0142 | Statistically significant |
| YearsAtCompany | -5.1963 | 2.3189e-07 | Statistically significant |
| YearsInCurrentRole | -6.2320 | 6.0032e-10 | Statistically significant |
| YearsSinceLastPromotion | -1.2658 | 0.2058 | Not statistically significant |
| YearsWithCurrManager | -6.0591 | 1.7370e-09 | Statistically significant |

A "Not statistically significant" means that there isn't strong statistical evidence to conclude that a particular relationship or difference exists between the variables. However, it doesn't necessarily mean the information is not important or that the feature should be deleted.

2. Data visualization:

2.1. Correlation Matrix:

```
correlation_matrix = df.corr()
plt.figure(figsize=(30, 12))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.show()
```



Most features are not strongly correlated to attrition, **Overtime** is having the most correlation with 0.25.

- Total working years, age, and job level exhibit a strong positive correlation (correlation coefficient > 0.55).

- Notably, there is an exceptionally high correlation (0.95) between job level and monthly income, which is logically expected given their relationships.

Correlation values

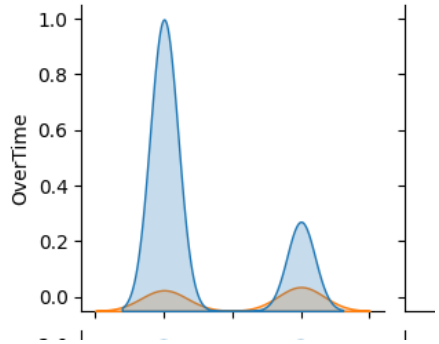
```
attrition_correlation = df.corr()["Attrition"].sort_values(ascending=False)
print(attrition_correlation)
```

```
Attrition                   1.000000
OverTime                    0.246118
BusinessTravel              0.127006
DistanceFromHome            0.077924
EducationField              0.051567
NumCompaniesWorked          0.043494
MonthlyRate                 0.015170
PerformanceRating           0.002889
JobRole                     0.002715
HourlyRate                 -0.006846
EmployeeNumber             -0.010577
PercentSalaryHike          -0.013478
Gender                     -0.029453
Education                  -0.031373
YearsSinceLastPromotion    -0.033019
RelationshipSatisfaction   -0.045872
DailyRate                  -0.056652
TrainingTimesLastYear      -0.059478
WorkLifeBalance            -0.063939
```

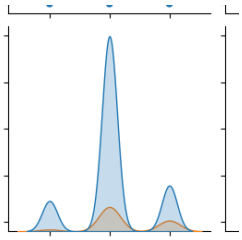
We can select features with the best correlation value with attrition and using seaborn library to creates a pair plot visualization:

```
df1 = df[['Attrition','OverTime', 'BusinessTravel', 'DistanceFromHome', 'EducationField', 'NumCompaniesWorked']]
sns.pairplot(df1, hue="Attrition", diag_kind="kde")
plt.show()
```


- Analysing plots separately:



The attrition rate is higher when overtime is involved compared to the attrition rate when no overtime is required.



Individuals who engage in frequent business travel are more prone to attrition compared to those who do not travel for work. The question arises: Why are employees with business travel commitments choosing to leave their positions? The specific reasons may require HR's insights and understanding.

### 2.2. Calculation of Attrition rate for some specific categories

Let's calculate attrition rate for categorical features:

```
# Define the column mappings
column_mappings = {
    'Attrition': {'No': 0, 'Yes': 1},
    'BusinessTravel': {'Travel_Frequently': 2, 'Travel_Rarely': 1, 'Non-Travel': 0},
    'Department': {'Research & Development': 3, 'Sales': 2, 'Human Resources': 1},
    'EducationField': {
        'Life Sciences': 1, 'Other': 6, 'Medical': 2, 'Marketing': 3,
        'Technical Degree': 4, 'Human Resources': 5
    },
    'Gender': {'Male': 0, 'Female': 1},
    'JobRole': {
        'Research Scientist': 1, 'Laboratory Technician': 2, 'Manufacturing Director': 3,
        'Healthcare Representative': 4, 'Manager': 5, 'Sales Representative': 6,
        'Research Director': 7, 'Sales Executive': 8, 'Human Resources': 9
    },
    'MaritalStatus': {'Married': 2, 'Single': 1, 'Divorced': 3},
    'OverTime': {'No': 0, 'Yes': 1}
}

# Loop through the scenarios and calculate attrition rates for all features
for feature, mapping in column_mappings.items():
    for value, encoded_value in mapping.items():
        attrition_condition = (df['Attrition'] == 1) & (df[feature] == encoded_value)
        total_attrition = df[attrition_condition]['Attrition'].count()
        average_employees = df[df[feature] == encoded_value]['Attrition'].count()
        attrition_rate = total_attrition / average_employees
        print(f"Attrition Rate for {feature}: {value}: {attrition_rate:.8%}")
```

| | | |
|---|---|---|
| Attrition Rate for BusinessTravel | Travel_Frequently | 24.90974729% |
| Attrition Rate for BusinessTravel | Travel_Rarely | 14.95685523% |
| Attrition Rate for BusinessTravel | Non-Travel | 8.00000000% |
| Attrition Rate for Department | Research & Development | 13.83975026% |
| Attrition Rate for Department | Sales | 20.62780269% |
| Attrition Rate for Department | Human Resources | 19.04761905% |
| Attrition Rate for EducationField | Life Sciences | 14.68646865% |
| Attrition Rate for EducationField | Other | 13.41463415% |
| Attrition Rate for EducationField | Medical | 13.57758621% |
| Attrition Rate for EducationField | Marketing | 22.01257862% |
| Attrition Rate for EducationField | Technical Degree | 24.24242424% |
| Attrition Rate for EducationField | Human Resources | 25.92592593% |
| Attrition Rate for Gender | Male | 17.00680272% |
| Attrition Rate for Gender | Female | 14.79591837% |

| | | |
|---|---|---|
| Attrition Rate for JobRole | Research Scientist | 16.09589041% |
| Attrition Rate for JobRole | Laboratory Technician | 23.93822394% |
| Attrition Rate for JobRole | Manufacturing Director | 6.89655172% |
| Attrition Rate for JobRole | Healthcare Representative | 6.87022901% |
| Attrition Rate for JobRole | Manager | 4.90196078% |
| Attrition Rate for JobRole | Sales Representative | 39.75903614% |
| Attrition Rate for JobRole | Research Director | 2.50000000% |
| Attrition Rate for JobRole | Sales Executive | 17.48466258% |
| Attrition Rate for JobRole | Human Resources | 23.07692308% |
| Attrition Rate for MaritalStatus | Married | 12.48142645% |
| Attrition Rate for MaritalStatus | Single | 25.53191489% |
| Attrition Rate for MaritalStatus | Divorced | 10.09174312% |
| Attrition Rate for OverTime | No | 10.43643264% |
| Attrition Rate for OverTime | Yes | 30.52884615% |
| Attrition Rate for PerformanceRating | 3 | 16.07717042% |
| Attrition Rate for PerformanceRating | 4 | 16.37168142% |

Calculate other attrition rate for numerical values based on defined categories.

```
# Define age categories
age_bins = [0, 25, 40, 100]
age_labels = ['Under 25', '25-40', 'Over 40']

# Add 'AgeCategory' column
df['AgeCategory'] = pd.cut(df['Age'], bins=age_bins, labels=age_labels, right=False)

# Calculate attrition rates for different age categories
scenarios = {
    'AgeCategory: Under 25': {'Attrition': 1, 'AgeCategory': 'Under 25'},
    'AgeCategory: 25-40': {'Attrition': 1, 'AgeCategory': '25-40'},
    'AgeCategory: Over 40': {'Attrition': 1, 'AgeCategory': 'Over 40'},
}

for scenario, conditions in scenarios.items():
    attrition_condition = (df['Attrition'] == conditions['Attrition']) & (df['AgeCategory'] == conditions['AgeCategory'])
    total_attrition = df[attrition_condition]['Attrition'].count()
    average_employees = df[df['AgeCategory'] == conditions['AgeCategory']]['Attrition'].count()
    attrition_rate = total_attrition / average_employees
    print(f"Attrition Rate for {scenario}: {attrition_rate:.8%}")
```

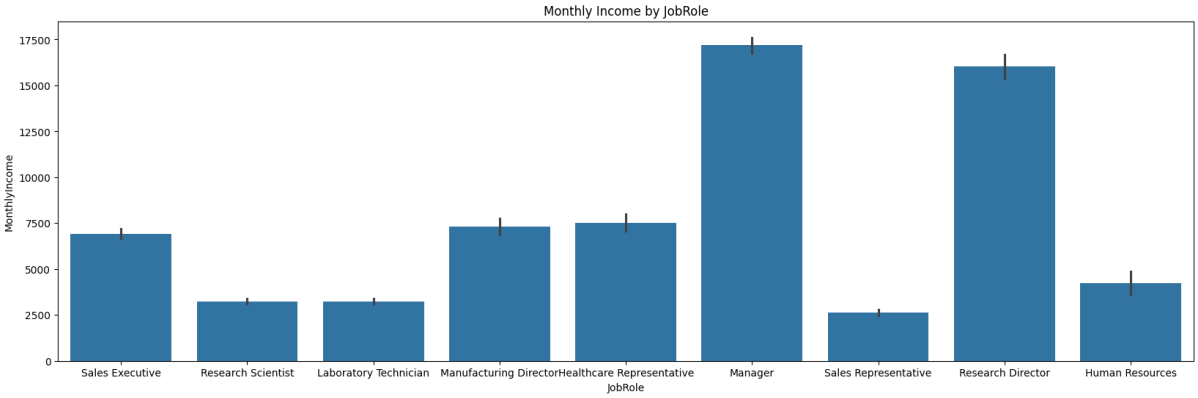
| | | |
|---|---|---|
| Attrition Rate for AgeCategory | Under 25 | 39.17525773% |
| Attrition Rate for AgeCategory | 25-40 | 16.68625147% |
| Attrition Rate for AgeCategory | Over 40 | 10.91954023% |
| Attrition Rate for MonthlyRateCategories | Under 10000 | 14.78439425% |
| Attrition Rate for MonthlyRateCategories | 1000-20000 | 16.58119658% |
| Attrition Rate for MonthlyRateCategories | Over 20000 | 17.08542714% |
| Attrition Rate for DistanceFromHomeCategories | Under 10 | 14.14893617% |
| Attrition Rate for DistanceFromHomeCategories | 10-20 | 18.27242525% |
| Attrition Rate for DistanceFromHomeCategories | Over 20 | 21.39737991% |

-Employees who are single may be more inclined to explore alternative career options (Attrition Rate 25.5) , while those who are married or divorced might prioritize stability and job security.

-Attrition rates vary by age, with employees under 25 experiencing an attrition rate of 39.17%, suggesting that younger employees may seek better opportunities.

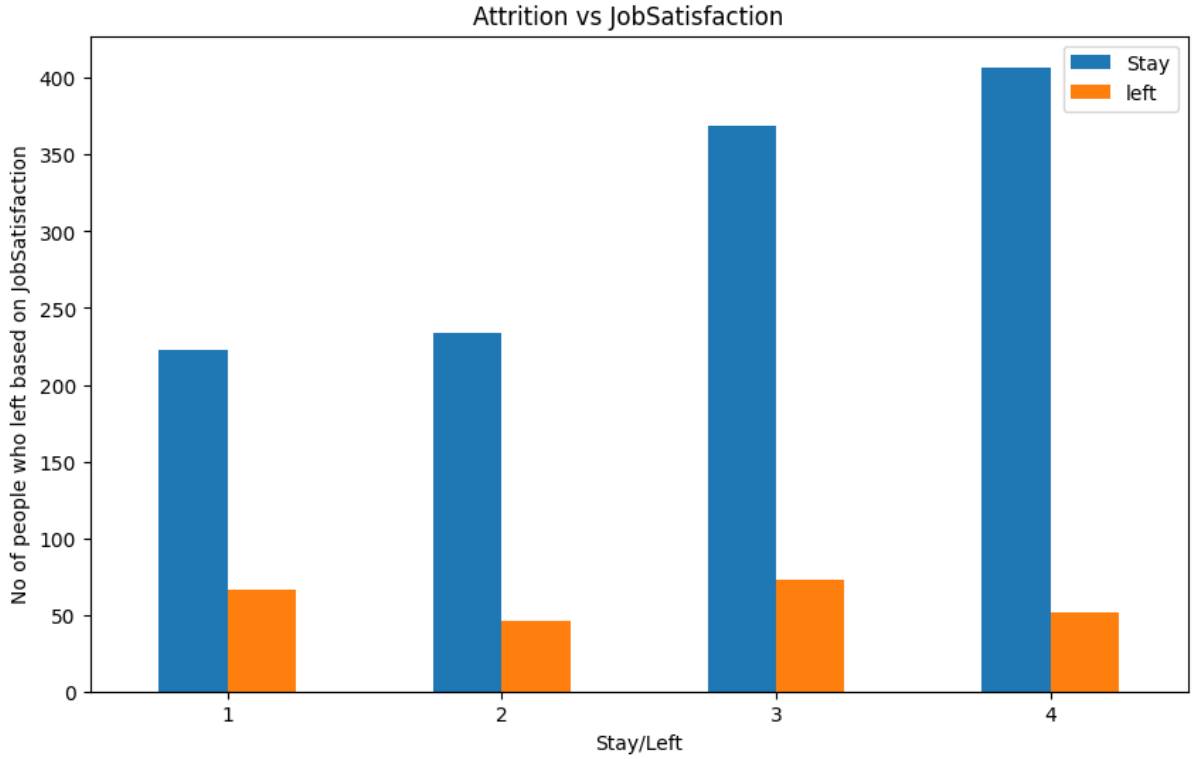
- ⇨ When examining attrition based on age and marital status, those with fewer responsibilities are more likely to leave, possibly indicating that the job alone isn't satisfying.

-Sales Representatives exhibit a notably high attrition rate of 39.75%. To understand why Sales Representatives are leaving their jobs, let's investigate their monthly income.




⇨ Sales Representative are having the less monthly income comparing to other job roles.

-Everytime the distance from home increase, attrition rate increase. It's important for the human resource responsible to resolve any transportation problems.




| | | |
|---|---|---|
| Attrition Rate for JobSatisfaction | 1 | 22.83737024% |
| Attrition Rate for JobSatisfaction | 2 | 16.42857143% |
| Attrition Rate for JobSatisfaction | 3 | 16.51583710% |
| Attrition Rate for JobSatisfaction | 4 | 11.32897603% |

The less job satisfaction implies high attrition rate.

**Conclusions**:
Employee attrition among single and young individuals with low engagement is primarily driven by factors such as overtime, business travel, extended distances from home, and monthly income. To maintain a low attrition rate, it is essential to ensure that employees experience high levels of job satisfaction.

**How to select features for model training:**
While certain features like age, overtime, job satisfaction, distance from home, monthly income, business travel, and job role may have a more significant and direct impact on attrition, it's important not to disregard or ignore features that have a low attrition correlation or are not statistically significant. Some features may not have a strong individual impact but can contribute to interactions with other features. These interactions might be significant in predicting attrition. Including more features can make the model more robust and less prone to overfitting. We may assign different weights to features based on their impact after training the model.

List of features:

- Age
- BusinessTravel
- DailyRate
- Department
- DistanceFromHome
- Education
- EducationField
- EmployeeNumber
- EnvironmentSatisfaction
- Gender
- HourlyRate
- JobInvolvement
- JobLevel
- JobRole
- JobSatisfaction
- MaritalStatus
- MonthlyIncome
- MonthlyRate
- NumCompaniesWorked
- OverTime
- PercentSalaryHike
- PerformanceRating
- RelationshipSatisfaction
- StockOptionLevel
- TotalWorkingYears
- TrainingTimes
- LastYearWork
- LifeBalance
- YearsAtCompany
- YearsInCurrentRole
- YearsSinceLastPromotion
- YearsWithCurrManager