ibm-atr-analysis

July 2, 2025

0.0.1 IBM HR Attrition Analysis

Business Objective IBM wants to identify key factors that lead to employee attrition (voluntary resignation), so that interventions can be made to reduce turnover, improve retention, and lower HR costs.

0.0.2 1. Import Libraries

[84]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

from sklearn.preprocessing import LabelEncoder
```

0.0.3 2. Load & Understand the Dataset

[86]:
```
df = pd.read_csv("HR_Employee_Attrition.csv")
```

[87]:
```
df.head()
```

[87]:
```
   Age Attrition     BusinessTravel  DailyRate              Department  \
0   41       Yes      Travel_Rarely       1102                   Sales
1   49        No  Travel_Frequently        279  Research & Development
2   37       Yes      Travel_Rarely       1373  Research & Development
3   33        No  Travel_Frequently       1392  Research & Development
4   27        No      Travel_Rarely        591  Research & Development

   DistanceFromHome  Education EducationField  EmployeeCount  EmployeeNumber  \
0                 1          2  Life Sciences              1               1
1                 8          1  Life Sciences              1               2
2                 2          2          Other              1               4
3                 3          4  Life Sciences              1               5
4                 2          1        Medical              1               7

   …  RelationshipSatisfaction StandardHours  StockOptionLevel  \
```

```
0  …                          1            80                  0
1  …                          4            80                  1
2  …                          2            80                  0
3  …                          3            80                  0
4  …                          4            80                  1

   TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                  8                      0               1               6
1                 10                      3               3              10
2                  7                      3               3               0
3                  8                      3               3               8
4                  6                      3               3               2

  YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
0                  4                        0                     5
1                  7                        1                     7
2                  0                        0                     0
3                  7                        3                     0
4                  2                        2                     2

[5 rows x 35 columns]
```

[88]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 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
dtypes: int64(26), object(9)
memory usage: 402.1+ KB
```

[89]:
```
df.describe()
```

[89]:

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

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

	JobLevel	…	RelationshipSatisfaction	StandardHours	\
count	1470.000000	…	1470.000000	1470.0	
mean	2.063946	…	2.712245	80.0	
std	1.106940	…	1.081209	0.0	
min	1.000000	…	1.000000	80.0	

```
25%       1.000000  …                 2.000000           80.0
50%       2.000000  …                 3.000000           80.0
75%       3.000000  …                 4.000000           80.0
max       5.000000  …                 4.000000           80.0

       StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
count       1470.000000        1470.000000            1470.000000
mean           0.793878          11.279592               2.799320
std            0.852077           7.780782               1.289271
min            0.000000           0.000000               0.000000
25%            0.000000           6.000000               2.000000
50%            1.000000          10.000000               3.000000
75%            1.000000          15.000000               3.000000
max            3.000000          40.000000               6.000000

       WorkLifeBalance  YearsAtCompany  YearsInCurrentRole  \
count      1470.000000     1470.000000         1470.000000
mean          2.761224        7.008163            4.229252
std           0.706476        6.126525            3.623137
min           1.000000        0.000000            0.000000
25%           2.000000        3.000000            2.000000
50%           3.000000        5.000000            3.000000
75%           3.000000        9.000000            7.000000
max           4.000000       40.000000           18.000000

       YearsSinceLastPromotion  YearsWithCurrManager
count              1470.000000           1470.000000
mean                  2.187755              4.123129
std                   3.222430              3.568136
min                   0.000000              0.000000
25%                   0.000000              2.000000
50%                   1.000000              3.000000
75%                   3.000000              7.000000
max                  15.000000             17.000000

[8 rows x 26 columns]
```

0.0.4 3. Data Cleaning & Preprocessing

[91]:
```
df.nunique()
```

[91]:
```
Age                         43
Attrition                    2
BusinessTravel               3
DailyRate                  886
Department                   3
DistanceFromHome            29
```

```
Education                     5
EducationField                6
EmployeeCount                 1
EmployeeNumber             1470
EnvironmentSatisfaction       4
Gender                        2
HourlyRate                   71
JobInvolvement                4
JobLevel                      5
JobRole                       9
JobSatisfaction               4
MaritalStatus                 3
MonthlyIncome              1349
MonthlyRate                1427
NumCompaniesWorked           10
Over18                        1
OverTime                      2
PercentSalaryHike            15
PerformanceRating             2
RelationshipSatisfaction      4
StandardHours                 1
StockOptionLevel              4
TotalWorkingYears            40
TrainingTimesLastYear         7
WorkLifeBalance               4
YearsAtCompany               37
YearsInCurrentRole           19
YearsSinceLastPromotion      16
YearsWithCurrManager         18
dtype: int64
```

[92]:
```
df.drop(['EmployeeCount' , 'StandardHours' , 'Over18' , 'EmployeeNumber'],␣
 ↪axis=1, inplace=True)  ---   # only 1 value present
```

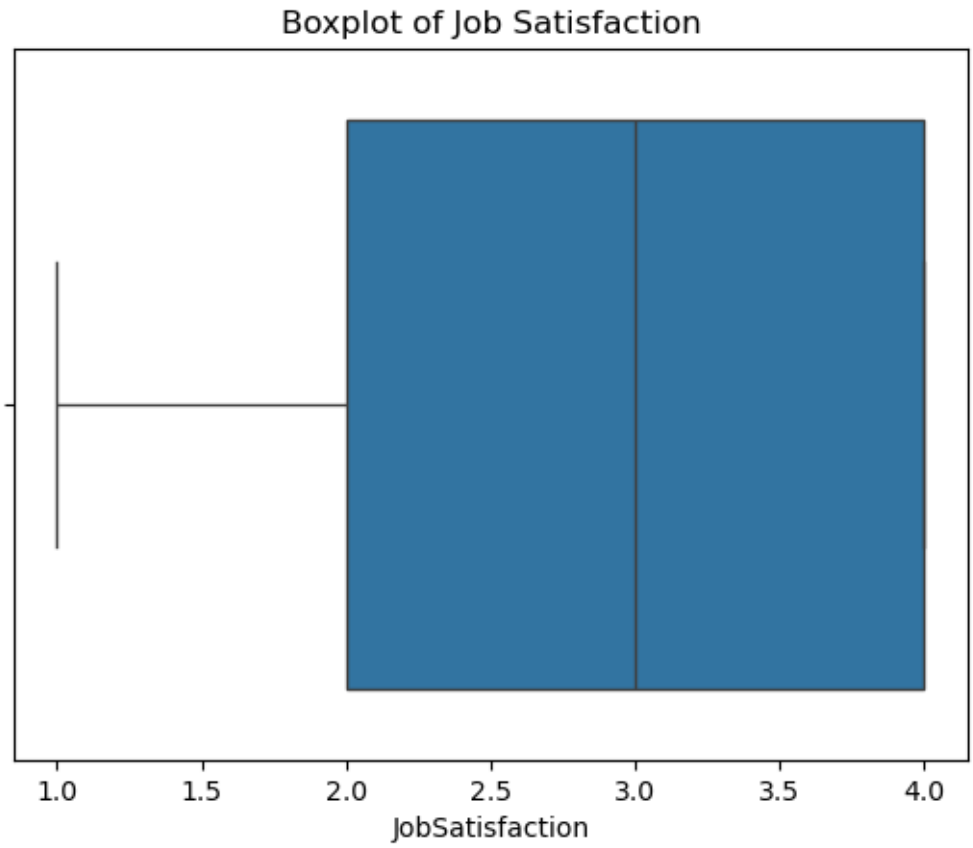
[93]:
```
df.head()
```

[93]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfaction
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	2
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	3
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	4
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	4
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1

```
3                3          4  Life Sciences                        4
4                2          1        Medical                        1

   Gender  …  PerformanceRating  RelationshipSatisfaction  StockOptionLevel  \
0  Female  …                  3                         1                 0
1    Male  …                  4                         4                 1
2    Male  …                  3                         2                 0
3  Female  …                  3                         3                 0
4    Male  …                  3                         4                 1

  TotalWorkingYears  TrainingTimesLastYear WorkLifeBalance  YearsAtCompany  \
0                 8                      0               1               6
1                10                      3               3              10
2                 7                      3               3               0
3                 8                      3               3               8
4                 6                      3               3               2

   YearsInCurrentRole  YearsSinceLastPromotion YearsWithCurrManager
0                   4                        0                    5
1                   7                        1                    7
2                   0                        0                    0
3                   7                        3                    0
4                   2                        2                    2

[5 rows x 31 columns]
```

[94]:
```
sns.boxplot(x=df['JobSatisfaction'])
plt.title('Boxplot of Job Satisfaction')
plt.show()
```



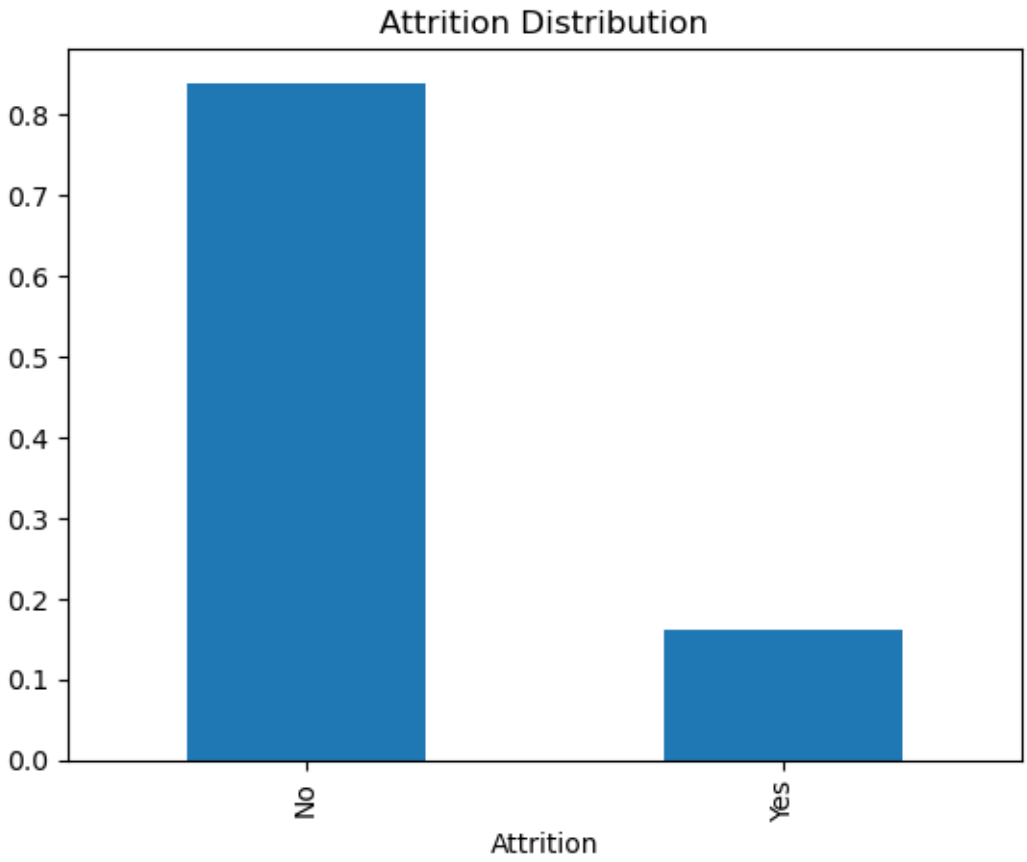
0.0.5 4. Univariate Analysis

[96]:
```
df['Attrition'].value_counts(normalize=True)*100  --- #  Attrition by percentage
```

[96]:
```
Attrition
No     83.877551
Yes    16.122449
Name: proportion, dtype: float64
```

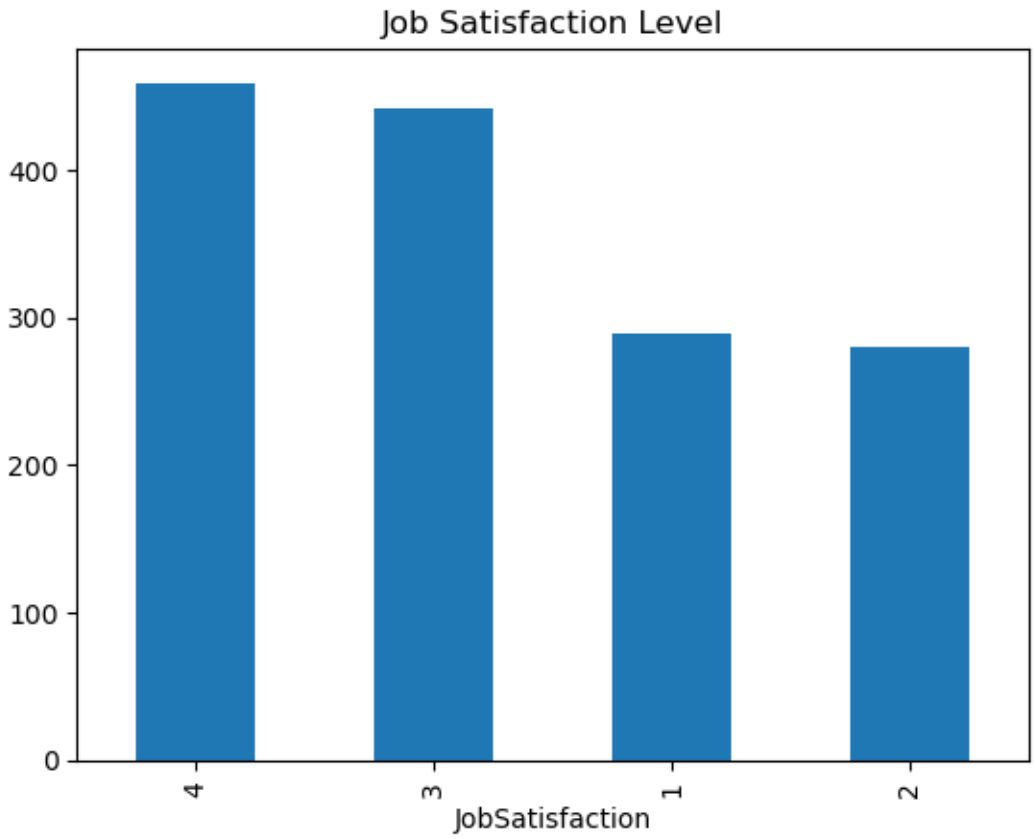
[97]:
```
df['Attrition'].value_counts(normalize=True).plot(kind='bar' , title='Attrition␣
↪Distribution')
```

[97]:
```
<Axes: title={'center': 'Attrition Distribution'}, xlabel='Attrition'>
```



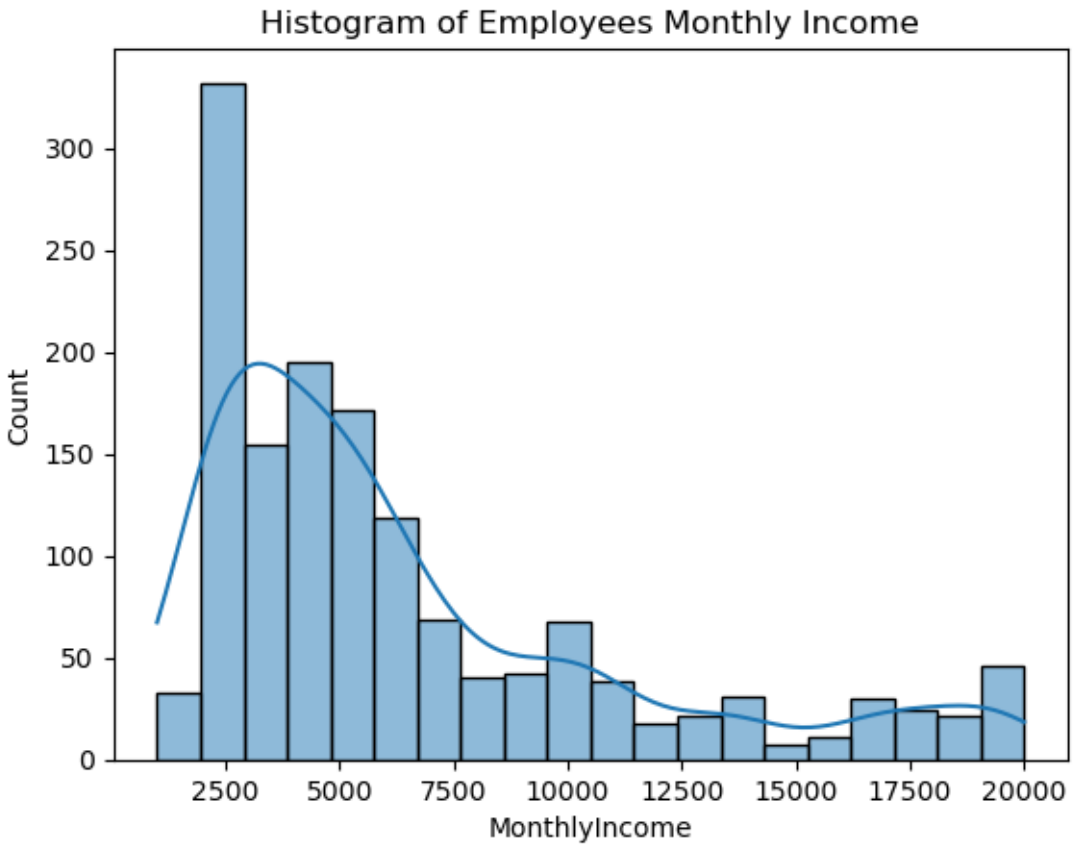
[155]:
```
df['JobSatisfaction'].value_counts().plot(kind='bar',title='Job Satisfaction␣
↪Level')
```

[155]: <Axes: title={'center': 'Job Satisfaction Level'}, xlabel='JobSatisfaction'>



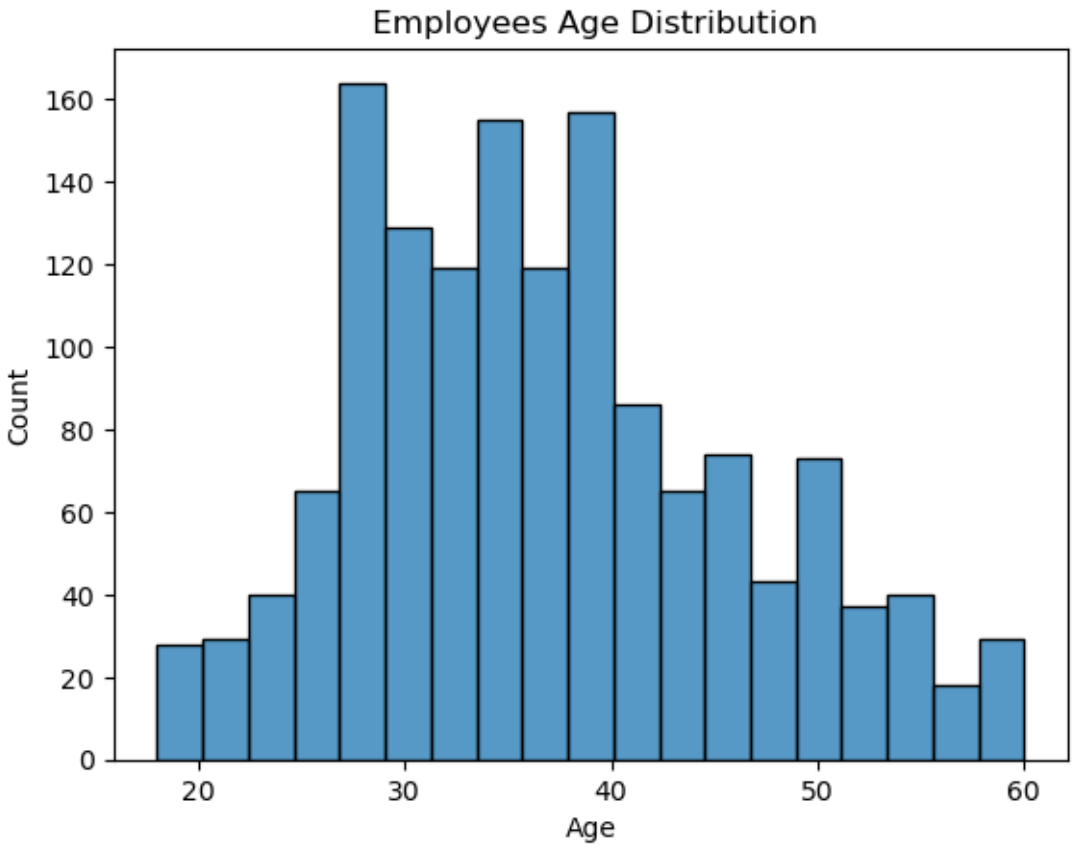
[159]:
```
sns.histplot(df['MonthlyIncome'], kde=True)
plt.title('Histogram of Employees Monthly Income')
```

[159]: Text(0.5, 1.0, 'Histogram of Employees Monthly Income')

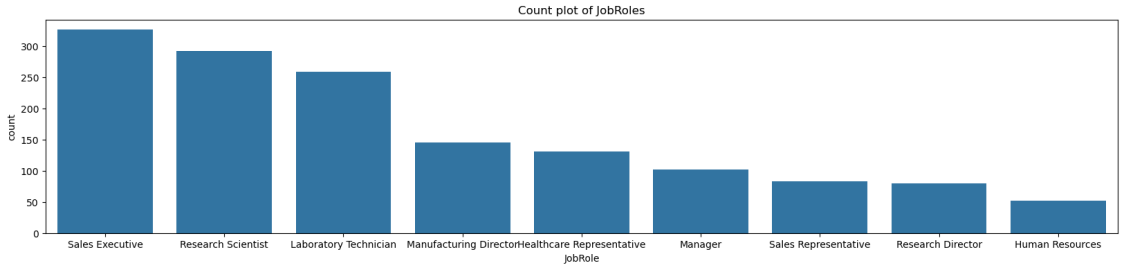


[163]:
```
sns.histplot(df['Age'])
plt.title(' Employees Age Distribution')
```

[163]:
```
Text(0.5, 1.0, ' Employees Age Distribution')
```



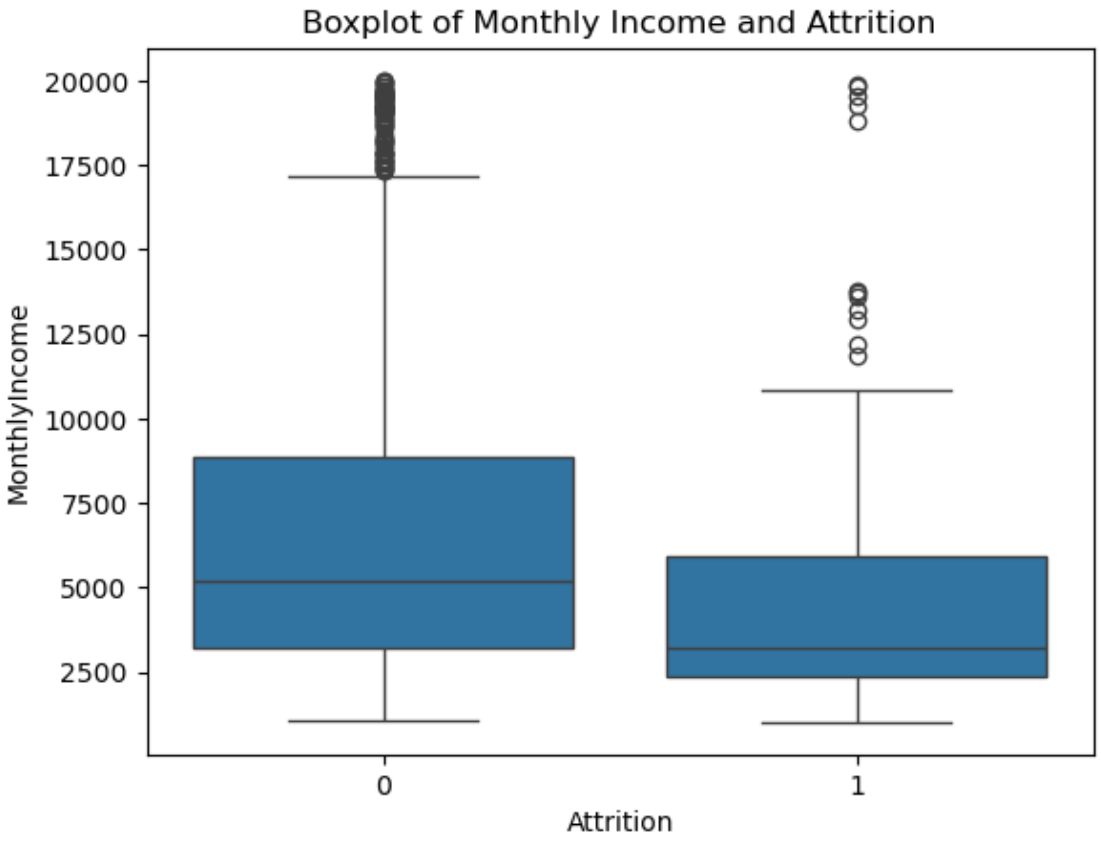
[101]:
```
plt.figure(figsize=(20,4))
sns.countplot(x='JobRole',data=df)
plt.title("Count plot of JobRoles")
plt.show()
```



0.0.6 5. Bivariate Analysis

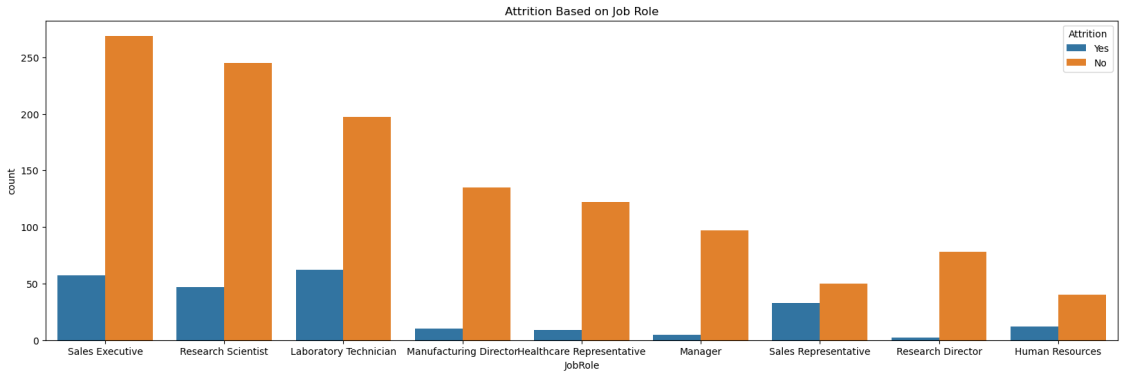
[165]:
```
sns.boxplot(x='Attrition', y='MonthlyIncome', data=df)
plt.title('Boxplot of Monthly Income and Attrition')
```

[165]: Text(0.5, 1.0, 'Boxplot of Monthly Income and Attrition')



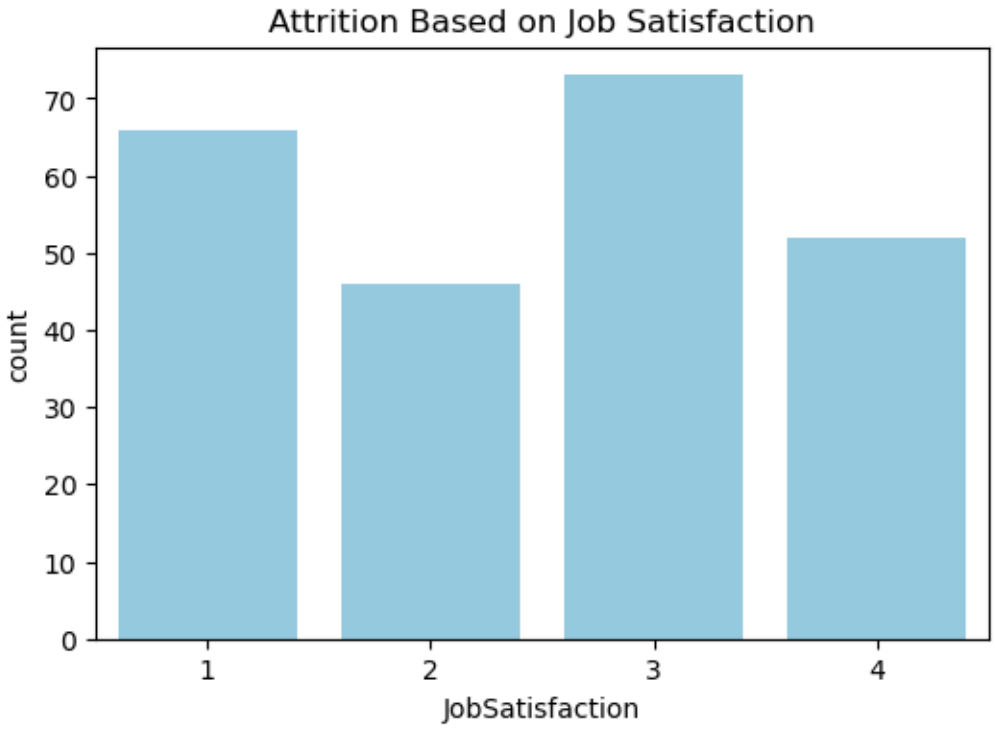
[104]:
```
plt.figure(figsize=(20,6))
sns.countplot(x='JobRole', hue='Attrition', data=df)
plt.title('Attrition Based on Job Role')
```

[104]: Text(0.5, 1.0, 'Attrition Based on Job Role')



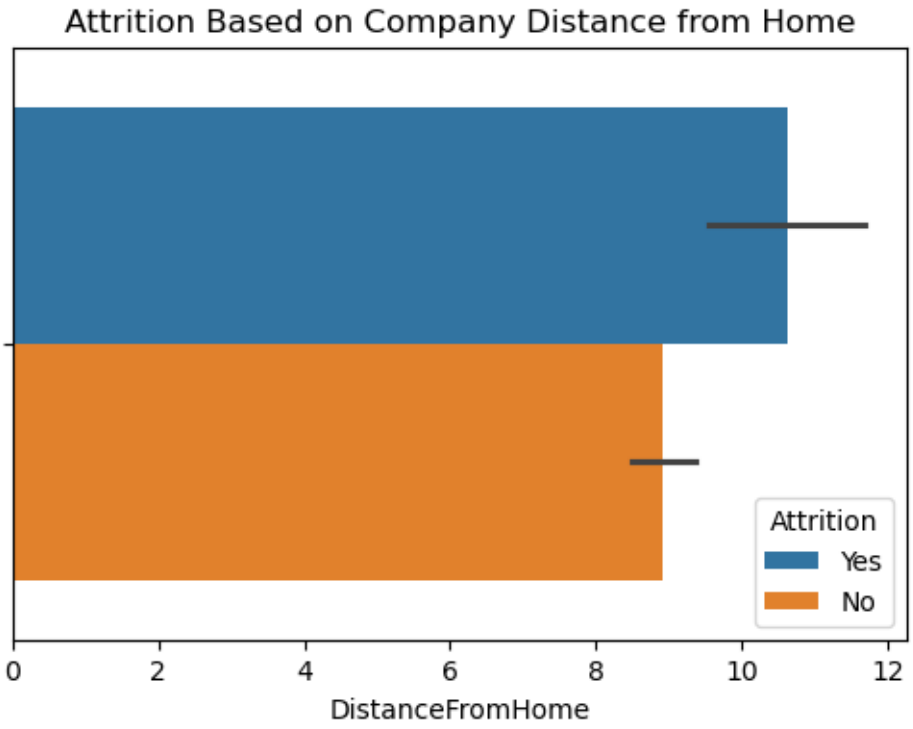
[105]:
```
plt.figure(figsize=(6,4))
sns.countplot(x='JobSatisfaction', data=df[df['Attrition'] == 'Yes'],␣
↪color='skyblue')
plt.title('Attrition Based on Job Satisfaction')
```

[105]: Text(0.5, 1.0, 'Attrition Based on Job Satisfaction')



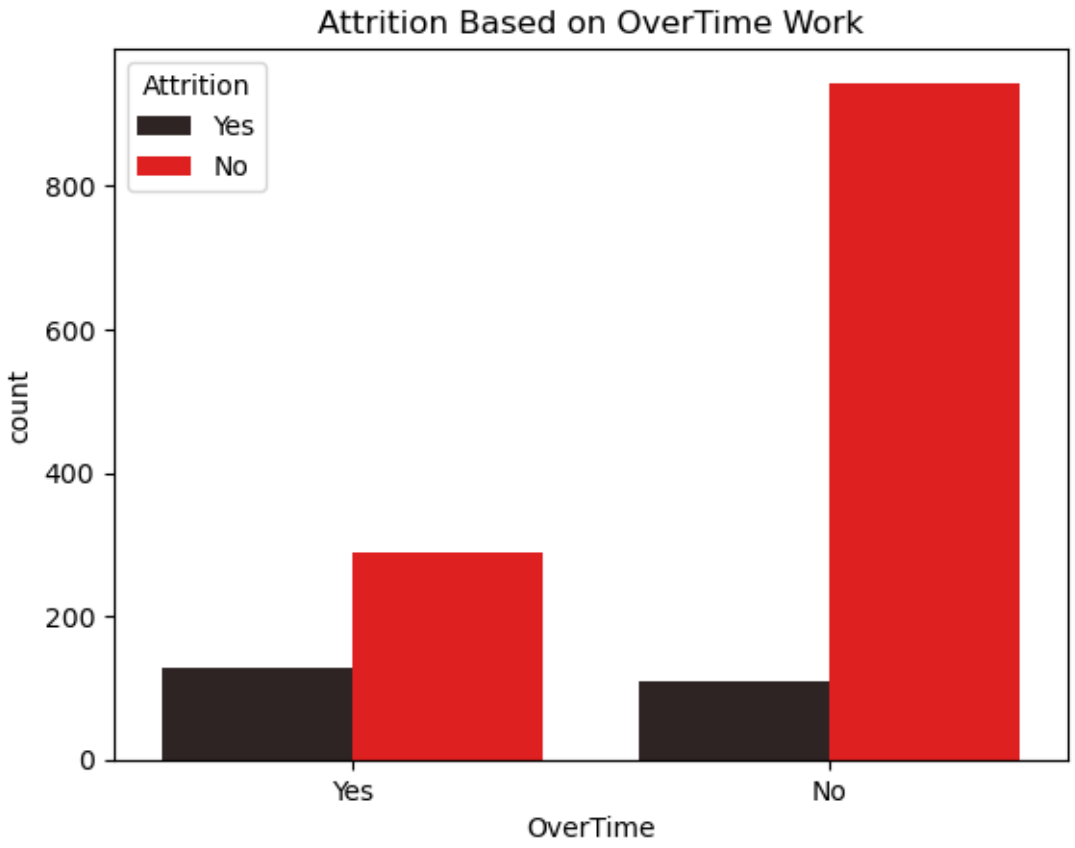
[106]:
```
plt.figure(figsize=(6,4))
sns.barplot(x='DistanceFromHome',hue='Attrition',data = df)
plt.title('Attrition Based on Company Distance from Home')
```

[106]: Text(0.5, 1.0, 'Attrition Based on Company Distance from Home')



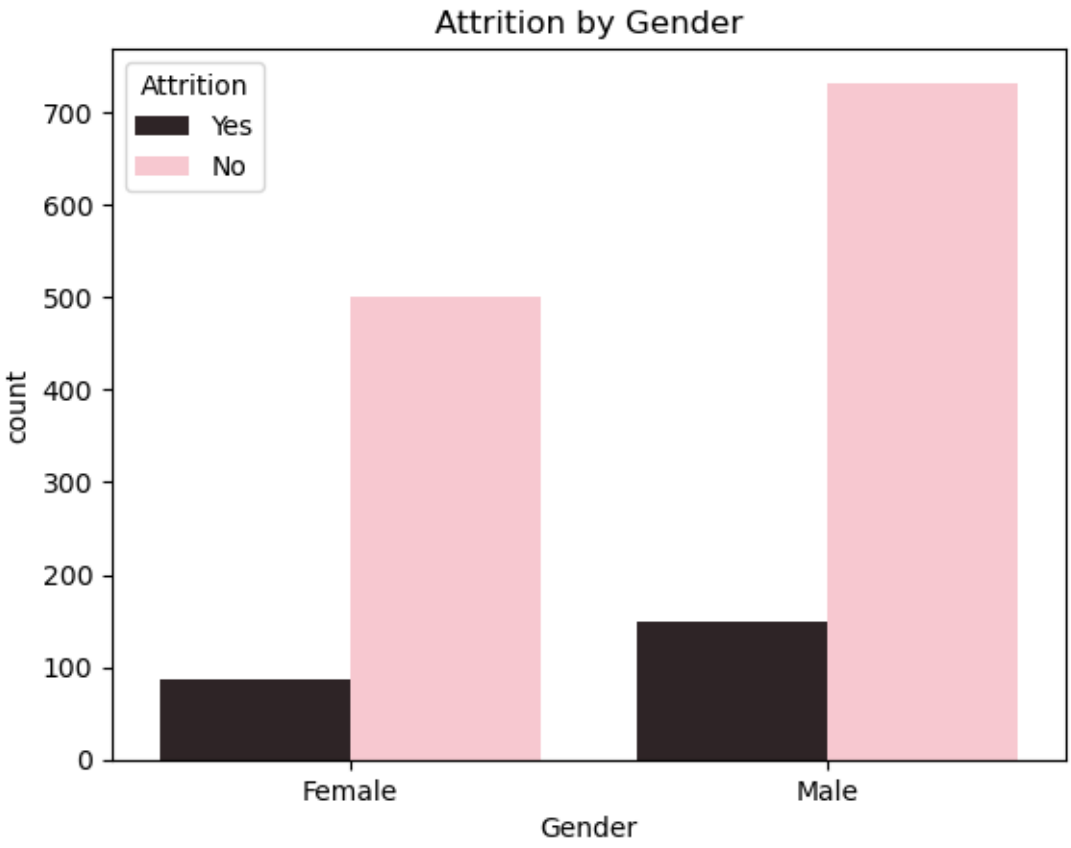
[107]:
```
sns.countplot(x='OverTime', hue='Attrition', data=df, color='red')
plt.title('Attrition Based on OverTime Work')
```

[107]: Text(0.5, 1.0, 'Attrition Based on OverTime Work')



[108]:
```
sns.countplot(x='Gender', hue='Attrition', data=df, color='pink')
plt.title('Attrition by Gender')
```

[108]: Text(0.5, 1.0, 'Attrition by Gender')



[109]:
```
df.groupby('Attrition')['JobSatisfaction'].mean().round(2)
```

[109]:
```
Attrition
No     2.78
Yes    2.47
Name: JobSatisfaction, dtype: float64
```

[110]:
```
df.groupby('Attrition')['YearsAtCompany'].mean().round(2)
```

[110]:
```
Attrition
No     7.37
Yes    5.13
Name: YearsAtCompany, dtype: float64
```

[111]:
```
df.groupby('Attrition')['WorkLifeBalance'].mean().round(2)
```

[111]:
```
Attrition
No     2.78
Yes    2.66
Name: WorkLifeBalance, dtype: float64
```

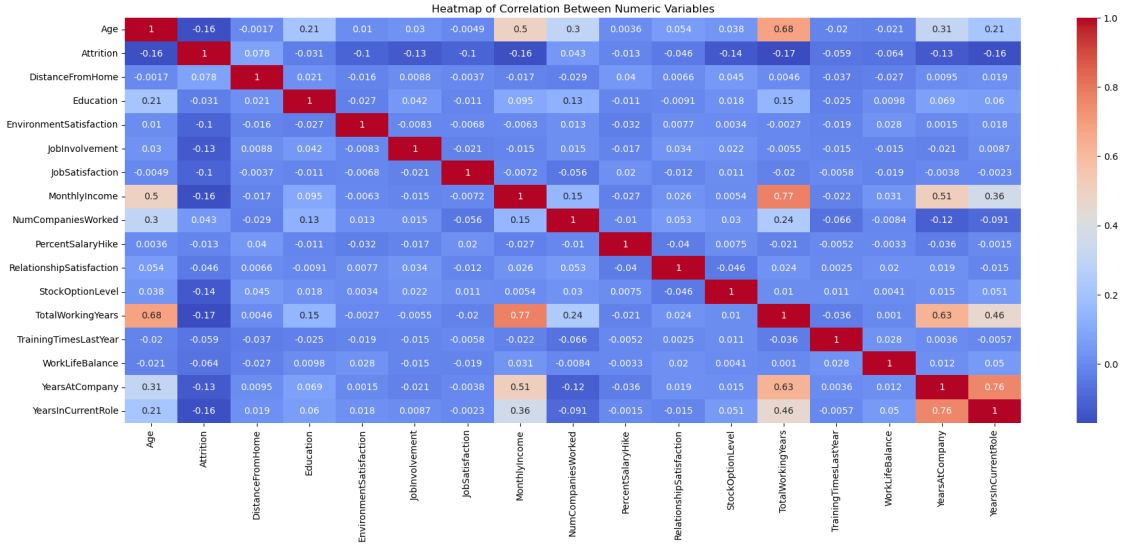
0.0.7 6. Feature Scaling

[112]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 31 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   EnvironmentSatisfaction   1470 non-null   int64
 9   Gender                    1470 non-null   object
 10  HourlyRate                1470 non-null   int64
 11  JobInvolvement            1470 non-null   int64
 12  JobLevel                  1470 non-null   int64
 13  JobRole                   1470 non-null   object
 14  JobSatisfaction           1470 non-null   int64
 15  MaritalStatus             1470 non-null   object
 16  MonthlyIncome             1470 non-null   int64
 17  MonthlyRate               1470 non-null   int64
 18  NumCompaniesWorked        1470 non-null   int64
 19  OverTime                  1470 non-null   object
 20  PercentSalaryHike         1470 non-null   int64
 21  PerformanceRating         1470 non-null   int64
 22  RelationshipSatisfaction  1470 non-null   int64
 23  StockOptionLevel          1470 non-null   int64
 24  TotalWorkingYears         1470 non-null   int64
 25  TrainingTimesLastYear     1470 non-null   int64
 26  WorkLifeBalance           1470 non-null   int64
 27  YearsAtCompany            1470 non-null   int64
 28  YearsInCurrentRole        1470 non-null   int64
 29  YearsSinceLastPromotion   1470 non-null   int64
 30  YearsWithCurrManager      1470 non-null   int64
dtypes: int64(23), object(8)
memory usage: 356.1+ KB
```

[169]:
```
corr=df.select_dtypes(include='number').corr()
plt.figure(figsize=(22,8))
sns.heatmap(corr,annot=True,cmap='coolwarm')
plt.title('Heatmap of Correlation Between Numeric Variables')
```

[169]: Text(0.5, 1.0, 'Heatmap of Correlation Between Numeric Variables')



[114]:
```
df.drop(columns=[
    'JobLevel',␣
  ↪'YearsWithCurrManager','YearsSinceLastPromotion','PerformanceRating','HourlyRate','DailyRat
], inplace=True)
```

[115]:
```
le = LabelEncoder()
df['Attrition']=le.fit_transform(df['Attrition'])
df['Attrition'].head()
```

[115]:
```
0    1
1    0
2    1
3    0
4    0
Name: Attrition, dtype: int32
```

[117]:
```
selected_features = ['OverTime', 'JobSatisfaction', 'MonthlyIncome',
                     'EnvironmentSatisfaction', 'WorkLifeBalance',
                     'DistanceFromHome', 'YearsAtCompany']

X = df[selected_features]
y = df['Attrition']
```

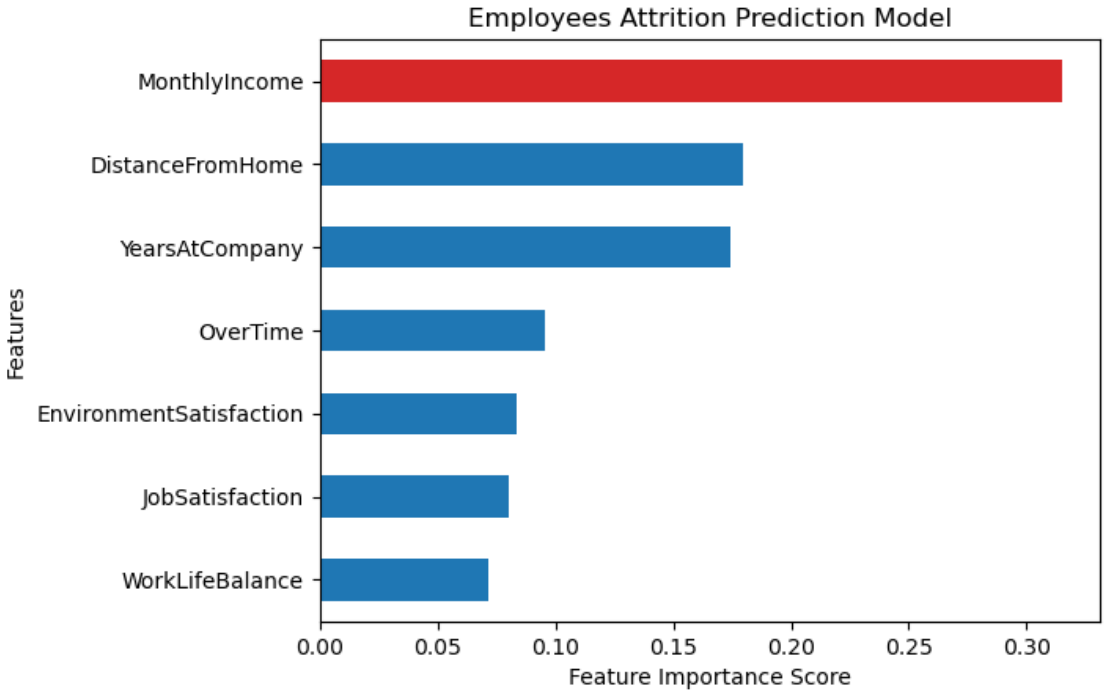
[118]:
```
le = LabelEncoder()
X['OverTime'] = le.fit_transform(X['OverTime'])  # Yes = 1, No = 0
```

0.0.8 7. Model Building (Model - Random Forest Classifier)

[119]:
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

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

model = RandomForestClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8469387755102041
```

[175]:
```
feat_imp = pd.Series(model.feature_importances_, index=X.columns)
feat_imp.sort_values().plot(kind='barh', title='Feature Importance')
colors = ['#1f77b4' if val < 0.2 else '#d62728' for val in feat_imp.
 ↪sort_values()]
feat_imp.sort_values().plot(kind='barh', color=colors)
plt.title('Employees Attrition Prediction Model')
plt.xlabel("Feature Importance Score")
plt.ylabel("Features")
plt.show()
```



[ ]:
```
#  - MonthlyIncome has the highest impact on attrition.
#  - Employees with shorter tenure are more likely to leave.
#  - Distance from home and overtime contribute significantly.
```

0.0.9 8. Final Insights Section (Markdown)

[ ]:
```
## Top Findings:
- Employees with low income, long commute, and overtime are more likely to␣
 ↪leave.
- Job satisfaction and environment also play a role.

## Recommendations:
- Improve salary structure for low-income bands.
- Offer flexible/remote work for those with long commutes.
- Monitor job satisfaction via regular surveys.
```