11.1 Project 3. Draft. Milestone 2 - Jennifer Barrera Conde

November 6, 2024

[3]:
```
# Importing necessary libraries
import pandas as pd
import numpy as np
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.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.impute import SimpleImputer

# Load data
df = pd.read_csv('Impact_of_Remote_Work_on_Mental_Health.csv')

# Initial inspection
print("Data Shape:", df.shape)
print("Column Names:", df.columns)
print("Data Types:\n", df.dtypes)
print("Missing Values:\n", df.isnull().sum())
print("Basic Statistics:\n", df.describe())
```

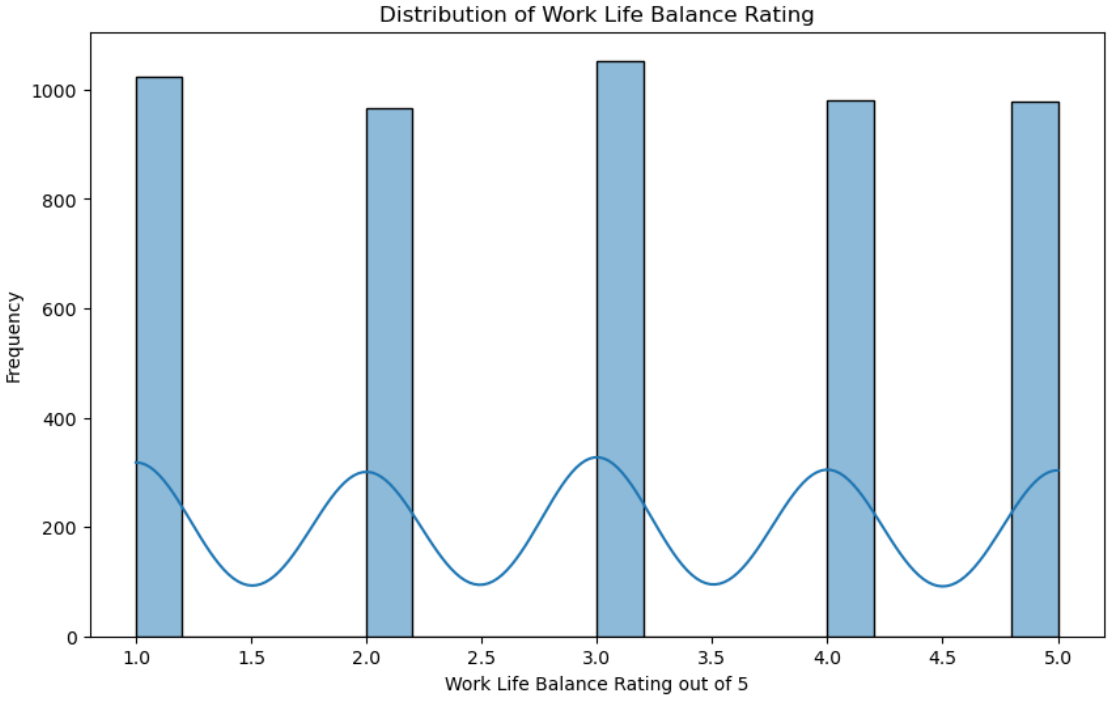
```
Data Shape: (5000, 20)
Column Names: Index(['Employee_ID', 'Age', 'Gender', 'Job_Role', 'Industry',
       'Years_of_Experience', 'Work_Location', 'Hours_Worked_Per_Week',
       'Number_of_Virtual_Meetings', 'Work_Life_Balance_Rating',
       'Stress_Level', 'Mental_Health_Condition',
       'Access_to_Mental_Health_Resources', 'Productivity_Change',
       'Social_Isolation_Rating', 'Satisfaction_with_Remote_Work',
       'Company_Support_for_Remote_Work', 'Physical_Activity', 'Sleep_Quality',
       'Region'],
      dtype='object')
Data Types:
 Employee_ID                          object
Age                                   int64
Gender                               object
Job_Role                             object
Industry                             object
Years_of_Experience                   int64
```

```
Work_Location                         object
Hours_Worked_Per_Week                  int64
Number_of_Virtual_Meetings             int64
Work_Life_Balance_Rating               int64
Stress_Level                          object
Mental_Health_Condition               object
Access_to_Mental_Health_Resources     object
Productivity_Change                   object
Social_Isolation_Rating                int64
Satisfaction_with_Remote_Work         object
Company_Support_for_Remote_Work        int64
Physical_Activity                     object
Sleep_Quality                         object
Region                                object
dtype: object
Missing Values:
 Employee_ID                             0
Age                                      0
Gender                                   0
Job_Role                                 0
Industry                                 0
Years_of_Experience                      0
Work_Location                            0
Hours_Worked_Per_Week                    0
Number_of_Virtual_Meetings               0
Work_Life_Balance_Rating                 0
Stress_Level                             0
Mental_Health_Condition               1196
Access_to_Mental_Health_Resources        0
Productivity_Change                      0
Social_Isolation_Rating                  0
Satisfaction_with_Remote_Work            0
Company_Support_for_Remote_Work          0
Physical_Activity                     1629
Sleep_Quality                            0
Region                                   0
dtype: int64
Basic Statistics:
                Age  Years_of_Experience  Hours_Worked_Per_Week  \
count  5000.000000          5000.000000            5000.000000
mean     40.995000            17.810200              39.614600
std      11.296021            10.020412              11.860194
min      22.000000             1.000000              20.000000
25%      31.000000             9.000000              29.000000
50%      41.000000            18.000000              40.000000
75%      51.000000            26.000000              50.000000
max      60.000000            35.000000              60.000000
```

```
       Number_of_Virtual_Meetings  Work_Life_Balance_Rating  \
count                 5000.000000               5000.000000
mean                     7.559000                  2.984200
std                      4.636121                  1.410513
min                      0.000000                  1.000000
25%                      4.000000                  2.000000
50%                      8.000000                  3.000000
75%                     12.000000                  4.000000
max                     15.000000                  5.000000

       Social_Isolation_Rating  Company_Support_for_Remote_Work
count              5000.000000                      5000.000000
mean                  2.993800                         3.007800
std                   1.394615                         1.399046
min                   1.000000                         1.000000
25%                   2.000000                         2.000000
50%                   3.000000                         3.000000
75%                   4.000000                         4.000000
max                   5.000000                         5.000000
```

[4]:
```
# Exploratory Data Analysis (EDA)
plt.figure(figsize=(10, 6))
sns.histplot(df['Work_Life_Balance_Rating'], bins=20, kde=True)
plt.title('Distribution of Work Life Balance Rating')
plt.xlabel('Work Life Balance Rating out of 5')
plt.ylabel('Frequency')
plt.show()
```

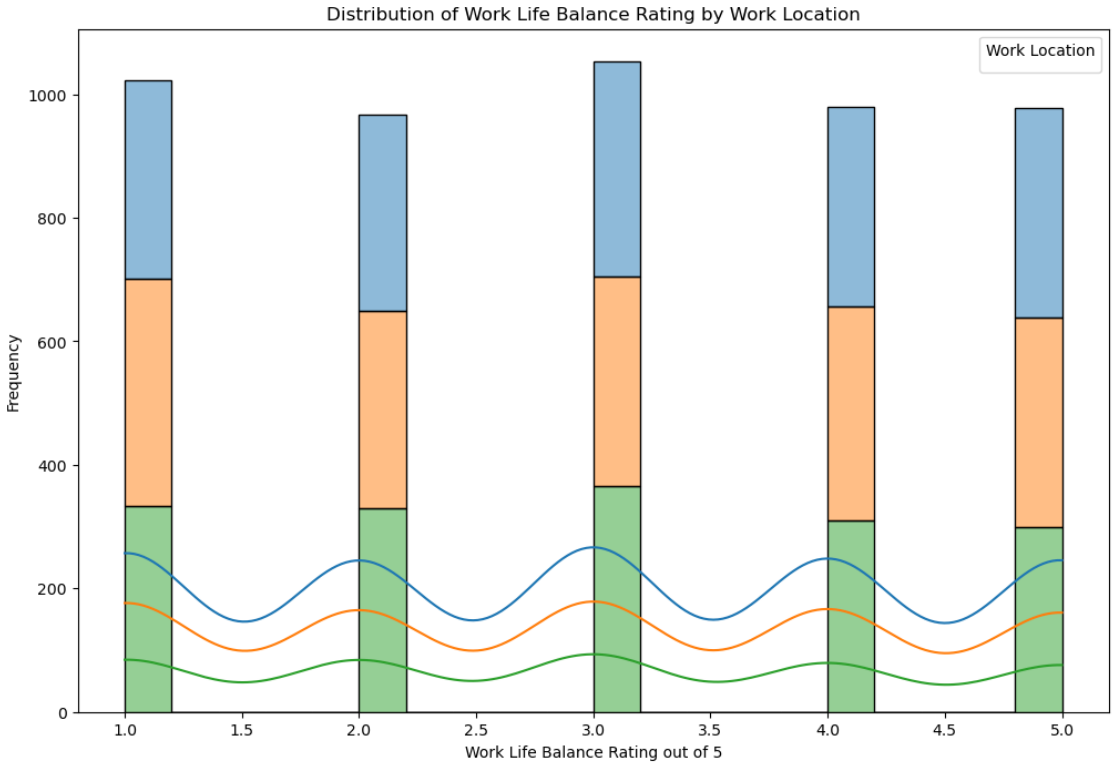


[8]:
```
import matplotlib.pyplot as plt

# Define specific colors for each work location
palette = {
    'Remote': 'blue',
    'Hybrid': 'green',
    'Onsite': 'orange'
}

# Plot the distribution of Work Life Balance Rating by Work Location
plt.figure(figsize=(12, 8))
sns.histplot(data=df, x='Work_Life_Balance_Rating', hue='Work_Location',␣
 ↪bins=20, kde=True, multiple='stack')
plt.title('Distribution of Work Life Balance Rating by Work Location')
plt.xlabel('Work Life Balance Rating out of 5')
plt.ylabel('Frequency')
plt.legend(title='Work Location')
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label
start with an underscore are ignored when legend() is called with no argument.
```



[10]:
```
# Drop non-numeric columns, such as employee IDs
df_numeric = df.select_dtypes(include=[float, int])

# Check for any categorical columns to encode if they are meaningful
# For example, using one-hot encoding for 'Work_Location'
if 'Work_Location' in df.columns:
    df_encoded = pd.get_dummies(df[['Work_Location']], drop_first=True)
    df_numeric = pd.concat([df_numeric, df_encoded], axis=1)

# Plot the correlation heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(df_numeric.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap of Variables')
plt.show()
```