

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, classification_report


# Step 1: Data Collection

# Assuming you have a CSV file named 'hr_data.csv' in the same directory

data = pd.read_csv('hr_data.csv')


# Step 2: Data Preprocessing

# Perform any necessary data cleaning and transformation here

# Handle missing values, outliers, data type conversions, etc.


# Step 3: Exploratory Data Analysis (EDA)

# Example code for EDA


# Display the first few rows of the dataset

print(data.head())


# Check the dimensions of the dataset (rows, columns)

print(data.shape)


# Summary statistics of the dataset

print(data.describe())


# Correlation matrix

correlation_matrix = data.corr()

print(correlation_matrix)
```

```
# Example visualization
```

```
# Distribution of employee satisfaction levels
```

```
plt.figure(figsize=(8, 6))  
sns.histplot(data['satisfaction_level'], kde=True)  
plt.xlabel('Satisfaction Level')  
plt.ylabel('Frequency')  
plt.title('Distribution of Employee Satisfaction Levels')  
plt.show()
```

```
# Boxplot of employee satisfaction levels by churn status
```

```
plt.figure(figsize=(6, 8))  
sns.boxplot(data['churn'], data['satisfaction_level'])  
plt.xticks([0, 1], ['No Churn', 'Churn'])  
plt.xlabel('Churn')  
plt.ylabel('Satisfaction Level')  
plt.title('Employee Satisfaction Levels by Churn Status')  
plt.show()
```

```
# Step 4: Predictive Modeling
```

```
# Split the dataset into features (X) and target variable (y)
```

```
X = data.drop('churn', axis=1)  
y = data['churn']
```

```
# 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_state=42)
```

```
# Build and train a logistic regression model
```

```
model = LogisticRegression()  
model.fit(X_train, y_train)
```

```
# Predict on the test set
```

```
y_pred = model.predict(X_test)
```

```
# Evaluate the model
```

```
accuracy = accuracy_score(y_test, y_pred)
```

```
classification_report = classification_report(y_test, y_pred)
```

```
print('Accuracy:', accuracy)
```

```
print('Classification Report:\n', classification_report)
```

```
# Step 10: Conclusion
```

```
# Summarize your findings and insights.
```