emp-2

September 28, 2023

#

Employee attrition (2512)

0.0.1 Imports

[ ]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import warnings
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
from sklearn.preprocessing import StandardScaler
from scipy import stats
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

0.0.2 Data Loading

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

0.0.3 Null checking

[ ]:
```
# df.shape
df.isnull().sum()
```

[ ]:
```
Age                         0
Attrition                   0
BusinessTravel              0
DailyRate                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
```

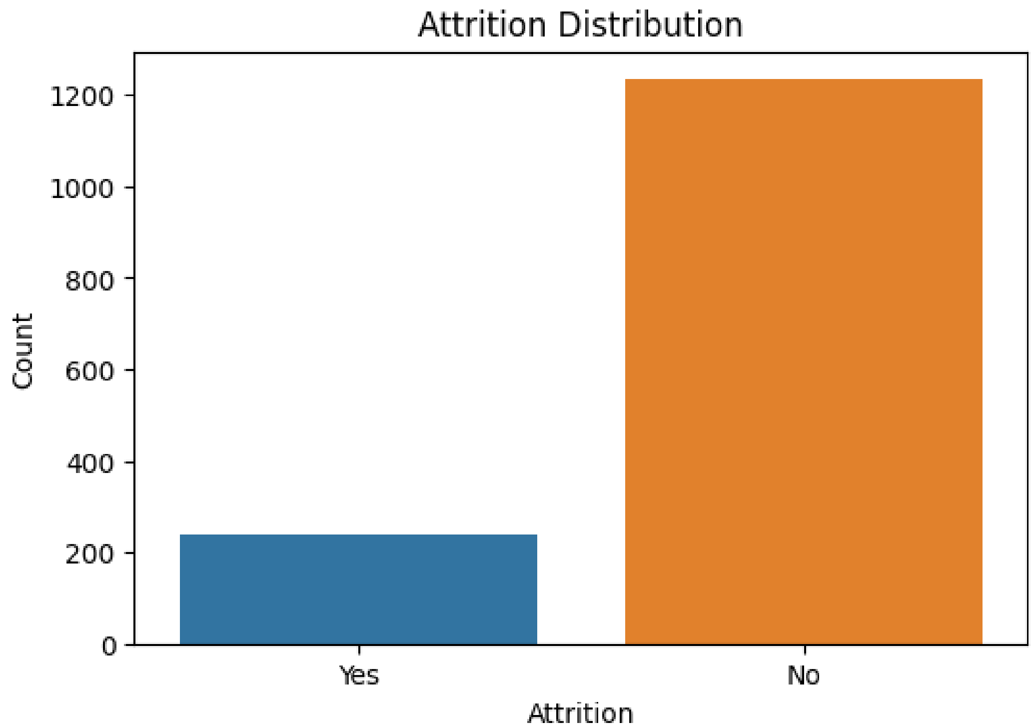
```
EmployeeCount               0
EmployeeNumber              0
EnvironmentSatisfaction     0
Gender                      0
HourlyRate                  0
JobInvolvement              0
JobLevel                    0
JobRole                     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
MonthlyRate                 0
NumCompaniesWorked          0
Over18                      0
OverTime                    0
PercentSalaryHike           0
PerformanceRating           0
RelationshipSatisfaction    0
StandardHours               0
StockOptionLevel            0
TotalWorkingYears           0
TrainingTimesLastYear       0
WorkLifeBalance             0
YearsAtCompany              0
YearsInCurrentRole          0
YearsSinceLastPromotion     0
YearsWithCurrManager        0
dtype: int64
```

Clearly, we dont have any null values to be processsed in the dataset

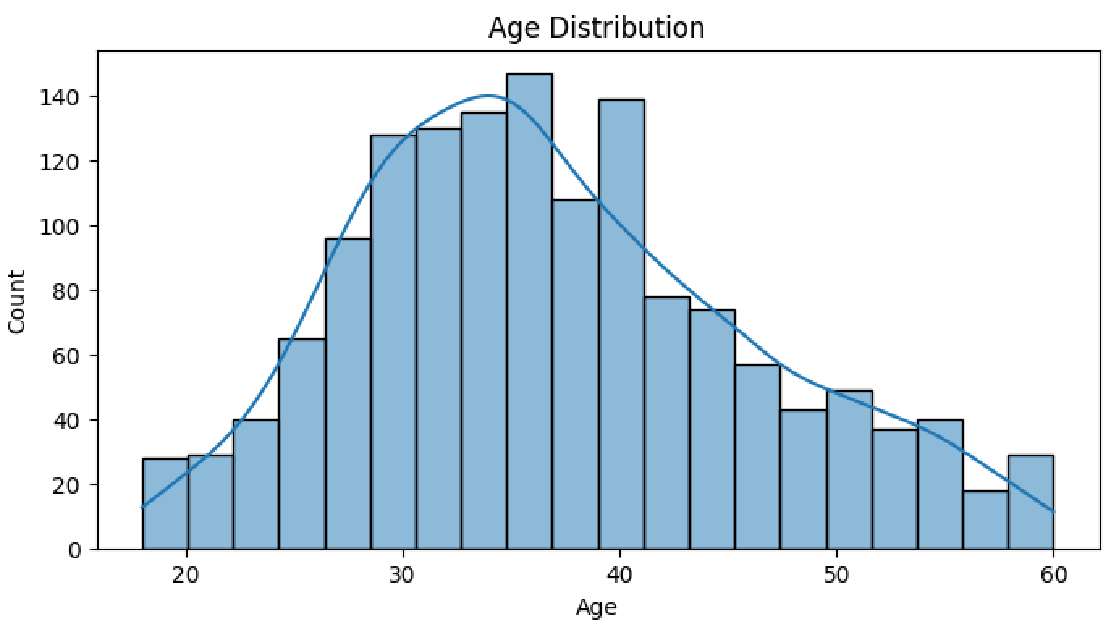
Data Visualizations

[ ]:
```python
# Visualization 1: Bar Chart for Attrition

plt.figure(figsize=(6, 4))
sns.countplot(x='Attrition', data=df)
plt.title('Attrition Distribution')
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.show()

# Visualization 2: Histogram for Age

plt.figure(figsize=(8, 4))
sns.histplot(df['Age'], bins=20, kde=True)
plt.title('Age Distribution')
```

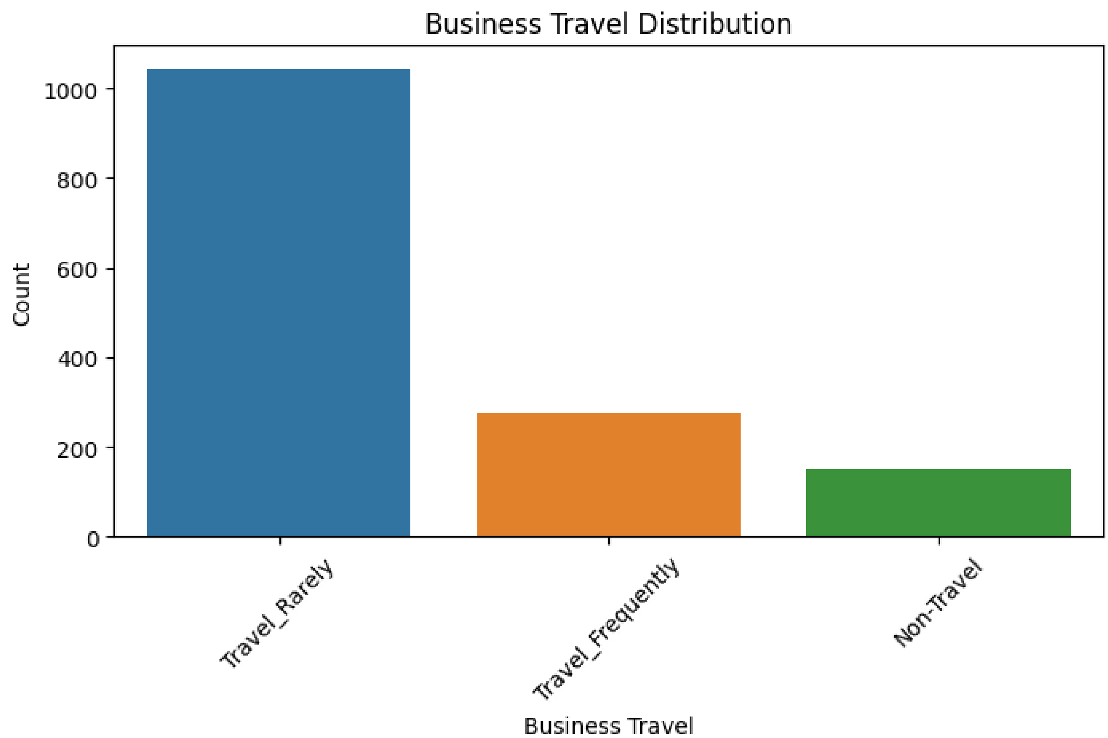
```
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

# Visualization 3: Count Plot for Business Travel

plt.figure(figsize=(8, 4))
sns.countplot(x='BusinessTravel', data=df)
plt.title('Business Travel Distribution')
plt.xlabel('Business Travel')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()

# Visualization 4: Correlation Heatmap for Numerical Features

numerical_features = df.select_dtypes(include=['int64', 'float64']).columns
correlation_matrix = df[numerical_features].corr()
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap for Numerical Features')
plt.show()
# Visualization 5: Attrition by Department

plt.figure(figsize=(8, 4))
sns.countplot(x='Department', hue='Attrition', data=df)
plt.title('Attrition by Department')
plt.xlabel('Department')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Attrition', loc='upper right')
plt.show()
```

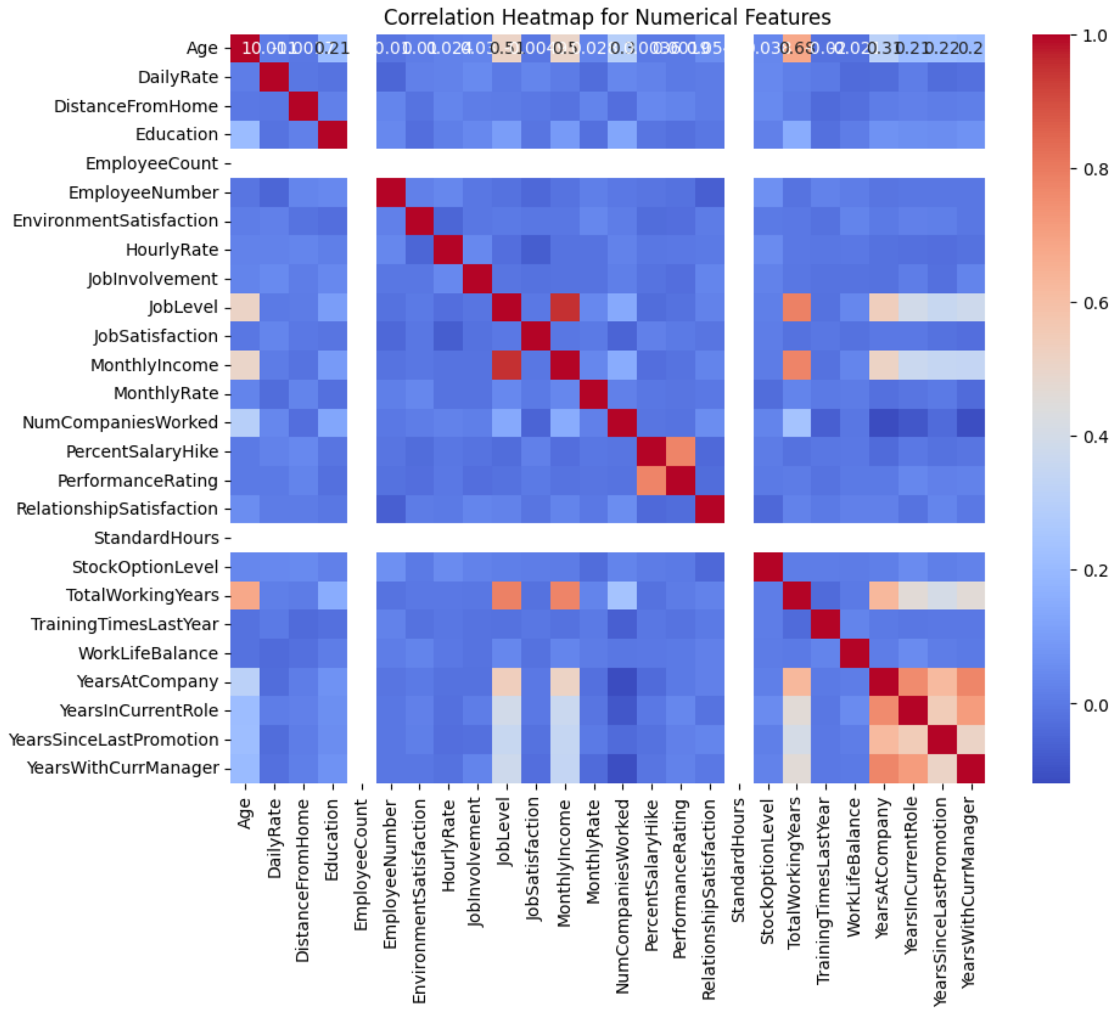


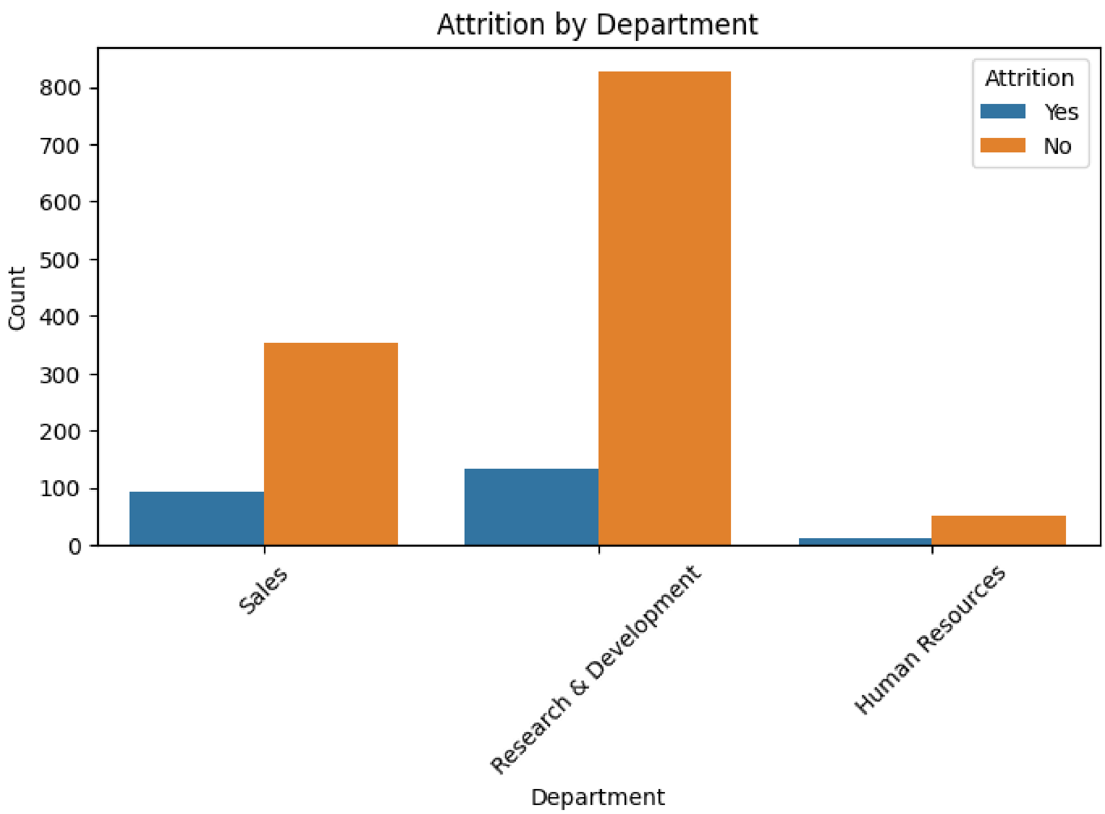
Attrition Distribution
1200
1000
800
600
400
200
0
Count
Yes
No
Attrition



Age Distribution
140
120
100
80
60
40
20
0
Count
20
30
40
50
60
Age


Business Travel Distribution
Count
1000
800
600
400
200
0
Travel_Rarely
Travel_Frequently
Non-Travel
Business Travel


Correlation Heatmap for Numerical Features
Age
DailyRate
DistanceFromHome
Education
EmployeeCount
EmployeeNumber
EnvironmentSatisfaction
HourlyRate
JobInvolvement
JobLevel
JobSatisfaction
MonthlyIncome
MonthlyRate
NumCompaniesWorked
PercentSalaryHike
PerformanceRating
RelationshipSatisfaction
StandardHours
StockOptionLevel
TotalWorkingYears
TrainingTimesLastYear
WorkLifeBalance
YearsAtCompany
YearsInCurrentRole
YearsSinceLastPromotion
YearsWithCurrManager
1.0
0.8
0.6
0.4
0.2
0.0



0.0.4 Outlier Detection

[ ]:
```
numerical_features = ["Age", "DailyRate", "DistanceFromHome", "HourlyRate",␣
↪"MonthlyIncome", "MonthlyRate", "NumCompaniesWorked", "PercentSalaryHike",␣
↪"TotalWorkingYears", "YearsAtCompany", "YearsInCurrentRole",␣
↪"YearsSinceLastPromotion", "YearsWithCurrManager"]

z_scores = np.abs(stats.zscore(df[numerical_features]))

#Z-score calc,
threshold = 5

# Find and print the indices of outliers
outlier_indices = (z_scores > threshold).any(axis=1)
outliers = df[outlier_indices]

print("Outliers:")
print(outliers)
```

```
Outliers:
     Age Attrition BusinessTravel  DailyRate              Department  \
```

```
126   58       Yes  Travel_Rarely        147  Research & Development

     DistanceFromHome  Education EducationField  EmployeeCount  \
126                23          4        Medical              1

     EmployeeNumber  …  RelationshipSatisfaction StandardHours  \
126             165  …                         4            80

     StockOptionLevel  TotalWorkingYears  TrainingTimesLastYear  \
126                 1                 40                      3

    WorkLifeBalance  YearsAtCompany YearsInCurrentRole  \
126               2              40                 10

     YearsSinceLastPromotion  YearsWithCurrManager
126                       15                     6

[1 rows x 35 columns]
```

Hence for the given threshold, we can see that there is only one outlier here, so we shall not worry about it.

0.0.5 Splitting into Dependent and Independent variables

[ ]:
```python
X = df.drop('Attrition', axis=1)
y = df['Attrition']
X
```

[ ]:
```
       Age     BusinessTravel  DailyRate              Department  \
0       41      Travel_Rarely       1102                   Sales
1       49  Travel_Frequently        279  Research & Development
2       37      Travel_Rarely       1373  Research & Development
3       33  Travel_Frequently       1392  Research & Development
4       27      Travel_Rarely        591  Research & Development
…     …                …        …                      …
1465   36  Travel_Frequently        884  Research & Development
1466   39      Travel_Rarely        613  Research & Development
1467   27      Travel_Rarely        155  Research & Development
1468   49  Travel_Frequently       1023                   Sales
1469   34      Travel_Rarely        628  Research & Development

      DistanceFromHome  Education EducationField  EmployeeCount  \
0                    1          2  Life Sciences              1
1                    8          1  Life Sciences              1
2                    2          2          Other              1
3                    3          4  Life Sciences              1
4                    2          1        Medical              1
```

```
…                 …        …            …            …
1465             23        2      Medical            1
1466              6        1      Medical            1
1467              4        3  Life Sciences          1
1468              2        3      Medical            1
1469              8        3      Medical            1

      EmployeeNumber  EnvironmentSatisfaction  … RelationshipSatisfaction  \
0                  1                        2  …                        1
1                  2                        3  …                        4
2                  4                        4  …                        2
3                  5                        4  …                        3
4                  7                        1  …                        4
…                …                        …  …                      …
1465            2061                        3  …                        3
1466            2062                        4  …                        1
1467            2064                        2  …                        2
1468            2065                        4  …                        4
1469            2068                        2  …                        1

      StandardHours  StockOptionLevel  TotalWorkingYears  \
0                80                 0                  8
1                80                 1                 10
2                80                 0                  7
3                80                 0                  8
4                80                 1                  6
…               …                 …                 …
1465             80                 1                 17
1466             80                 1                  9
1467             80                 1                  6
1468             80                 0                 17
1469             80                 0                  6

     TrainingTimesLastYear  WorkLifeBalance YearsAtCompany  \
0                        0                1              6
1                        3                3             10
2                        3                3              0
3                        3                3              8
4                        3                3              2
…                      …               …             …
1465                     3                3              5
1466                     5                3              7
1467                     0                3              6
1468                     3                2              9
1469                     3                4              4

      YearsInCurrentRole  YearsSinceLastPromotion  YearsWithCurrManager
```

```
0                      4                         0                      5
1                      7                         1                      7
2                      0                         0                      0
3                      7                         3                      0
4                      2                         2                      2
…                   …                       …                    …
1465                   2                         0                      3
1466                   7                         1                      7
1467                   2                         0                      3
1468                   6                         0                      8
1469                   3                         1                      2

[1470 rows x 34 columns]
```

0.0.6 Encoding categorical variables

[ ]:
```
categorical_columns = ["BusinessTravel", "Department", "EducationField",␣
↪"Gender", "JobRole", "MaritalStatus", "OverTime","Over18"]

X_encoded = pd.get_dummies(X, columns=categorical_columns)
```

Feature Scaling is not required here, since only Decision trees and Logistic regression are being done neither of which are scale sensitive

0.0.7 Splitting into testing and training

[ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.
↪2, random_state=42)
y_train = y_train.map({'Yes': 1, 'No': 0})
```

[ ]:
```
X_train
# y_train.shape
# X_test.shape
# y_test.shape
# y_train
```

[ ]:
```
      Age  DailyRate  DistanceFromHome  Education  EmployeeCount  \
1097   24        350                21          2              1
727    18        287                 5          2              1
254    29       1247                20          2              1
1175   39        492                12          3              1
1341   31        311                20          3              1
…     …        …               …        …           …
1130   35        750                28          3              1
1294   41        447                 5          3              1
```

```
860   22        1256                  3          4              1
1459  29        1378                 13          2              1
1126  50         264                  9          3              1

      EmployeeNumber  EnvironmentSatisfaction  HourlyRate  JobInvolvement  \
1097            1551                        3          57               2
727             1012                        2          73               3
254              349                        4          45               3
1175            1654                        4          66               3
1341            1881                        2          89               3
…                  …                        …           …               …
1130            1596                        2          46               4
1294            1814                        2          85               4
860             1203                        3          48               2
1459            2053                        4          46               2
1126            1591                        3          59               3

      JobLevel  …  JobRole_Research Director  JobRole_Research Scientist  \
1097         1  …                      False                       False
727          1  …                      False                        True
254          2  …                      False                       False
1175         2  …                      False                       False
1341         2  …                      False                       False
…          …  …                          …                           …
1130         2  …                      False                       False
1294         2  …                      False                       False
860          1  …                      False                        True
1459         2  …                      False                       False
1126         5  …                      False                       False

      JobRole_Sales Executive  JobRole_Sales Representative  \
1097                    False                         False
727                     False                         False
254                      True                         False
1175                    False                         False
1341                    False                         False
…                           …                             …
1130                    False                         False
1294                    False                         False
860                     False                         False
1459                    False                         False
1126                    False                         False

      MaritalStatus_Divorced  MaritalStatus_Married  MaritalStatus_Single  \
1097                    True                  False                 False
727                    False                  False                  True
254                     True                  False                 False
```

```
1175                     False                    True                 False
1341                      True                   False                 False
…                        …                    …                   …
1130                     False                    True                 False
1294                     False                   False                  True
860                      False                    True                 False
1459                     False                    True                 False
1126                     False                    True                 False

      OverTime_No  OverTime_Yes  Over18_Y
1097         True         False      True
727          True         False      True
254          True         False      True
1175         True         False      True
1341         True         False      True
…             …            …        …
1130         True         False      True
1294         True         False      True
860         False          True      True
1459        False          True      True
1126        False          True      True

[1176 rows x 55 columns]
```

0.0.8 Linear regression

[ ]:
```python
logistic_regression_model = LogisticRegression(random_state=42)

# Fit the model on the training data
logistic_regression_model.fit(X_train, y_train)



# Predict on the test data
y_pred = logistic_regression_model.predict(X_test)
y_pred = np.where(y_pred == 0, 'No', 'Yes')

# Convert y_test to string labels if it's not already
y_test = y_test.astype(str)

# Now both y_test and y_pred have string labels

# Evaluate the model's performance using y_test
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
```

```python
# Print the results
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

```
Accuracy: 0.8673469387755102
Confusion Matrix:
 [[255   0]
 [ 39   0]]
Classification Report:
               precision    recall  f1-score   support

          No       0.87      1.00      0.93       255
         Yes       0.00      0.00      0.00        39

    accuracy                           0.87       294
   macro avg       0.43      0.50      0.46       294
weighted avg       0.75      0.87      0.81       294
```

0.0.9 Decision Tree Classifier

[ ]:
```python
decision_tree_model = DecisionTreeClassifier(random_state=42)
decision_tree_model.fit(X_train, y_train)

# Predict on the test data
y_true = y_test.map({'No': 0, 'Yes': 1})
y_pred = decision_tree_model.predict(X_test)
accuracy = accuracy_score(y_true, y_pred)
conf_matrix = confusion_matrix(y_true, y_pred)
class_report = classification_report(y_true, y_pred)



# Print the results
print("Accuracy:", accuracy)
print("Confusion Matrix:\n", conf_matrix)
print("Classification Report:\n", class_report)
```

```
Accuracy: 0.7653061224489796
Confusion Matrix:
 [[217  38]
 [ 31   8]]
Classification Report:
               precision    recall  f1-score   support

           0       0.88      0.85      0.86       255
           1       0.17      0.21      0.19        39
```

```
    accuracy                           0.77       294
   macro avg       0.52      0.53      0.53       294
weighted avg       0.78      0.77      0.77       294
```

0.0.10 Random Forest Method

```
[ ]: categorical_columns = ["BusinessTravel", "Department", "EducationField",␣
      ↪"Gender", "JobRole", "MaritalStatus", "OverTime", "Over18"]
     X_encoded = pd.get_dummies(X, columns=categorical_columns)

     # Splitting into testing and training
     X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.
      ↪2, random_state=42)

     # Create and fit the Random Forest model
     random_forest_model = RandomForestClassifier(random_state=42)
     random_forest_model.fit(X_train, y_train)

     # Predict on the test data
     y_pred = random_forest_model.predict(X_test)

     # Evaluate the model's performance
     accuracy = accuracy_score(y_test, y_pred)
     conf_matrix = confusion_matrix(y_test, y_pred)
     class_report = classification_report(y_test, y_pred)

     # Print the results
     print("Accuracy:", accuracy)
     print("Confusion Matrix:\n", conf_matrix)
     print("Classification Report:\n", class_report)
```

```
Accuracy: 0.8741496598639455
Confusion Matrix:
 [[253   2]
 [ 35   4]]
Classification Report:
               precision    recall  f1-score   support

          No       0.88      0.99      0.93       255
         Yes       0.67      0.10      0.18        39

    accuracy                           0.87       294
   macro avg       0.77      0.55      0.55       294
weighted avg       0.85      0.87      0.83       294
```