Advanced Certification Programme in AI and MLOps

A programme by IISc and TalentSprint

Mini Project Notebook: Employee Attrition Prediction

Problem Statement

To predict employee attrition using CatBoost and XgBoost

Learning Objectives

At the end of the experiment, you will be able to

- explore the employee attrition dataset
- apply CatBoost and XgBoost on the dataset
- tune the model hyperparameters to improve accuracy
- evaluate the model using suitable metrics

Introduction

Employee attrition is the gradual reduction in employee numbers. Employee attrition happens when the size of your workforce diminishes over time. This means that employees are leaving faster than they are hired. Employee attrition happens when employees retire, resign, or simply aren't replaced. Although employee attrition can be company-wide, it may also be confined to specific parts of a business.

Employee attrition can happen for several reasons. These include unhappiness about employee benefits or the pay structure, a lack of employee development opportunities, and even poor conditions in the workplace.

To know more about the factors that lead to employee attrition, refer here.

Gradient Boosted Decision Trees

- Gradient boosted decision trees (GBDTs) are one of the most important machine learning models.
- GBDTs originate from AdaBoost, an algorithm that ensembles weak learners and uses the majority vote, weighted by their individual accuracy, to solve binary classification problems. The weak learners in this case are decision trees with a single split, called decision stumps.
- Some of the widely used gradient boosted decision trees are XgBoost, CatBoost and LightGBM.

Dataset

The dataset used for this mini-project is HR Employee Attrition dataset. This dataset is synthetically created by IBM data scientists. There are 35 features and 1470 records.

There are numerical features such as:

- Age
- DistanceFromHome
- EmployeeNumber
- PerformanceRating

There are several categorical features such as:

- JobRole
- EducationField
- Department
- BusinessTravel

Dependent or target feature is 'attrition' which has values as Yes/No.

Download the data

```
#@title Download the data
!wget -qq https://cdn.iisc.talentsprint.com/CDS/Datasets/wa_fn_usec_hr_employee_attrition_tsv.csv
```

```
print("Data Downloaded Successfuly!!")
```

```
Data Downloaded Successfuly!!
```

Grading = 10 Points

Install CatBoost

```
!pip -qq install catboost
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.7/98.7 MB 7.7 MB/s eta 0:00:00
```

Import Required Packages

```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
from sklearn.metrics import roc_auc_score, accuracy_score, confusion_matrix, f1_score
from sklearn.model_selection import train_test_split
from lightgbm import LGBMClassifier
from xgboost import XGBClassifier
from catboost import CatBoostClassifier, metrics
import warnings
warnings.filterwarnings("ignore")
plt.style.use('fivethirtyeight')
pd.set_option('display.max_columns', 100)
%matplotlib inline
```

Please refer to the ReadMe before proceeding further.

Part-A

Load the Dataset

Exercise 1: Read the dataset [0.5 Mark]

Hint: pd.read_csv()

```
# read the dataset
# YOUR CODE HERE
import pandas as pd

# The file is named 'wa_fn_usec_hr_employee_attrition_tsv.csv' and is in the same directory
#file_name = 'wa_fn_usec_hr_employee_attrition_tsv.csv'
#dataset = pd.read_csv(file_name, delimiter='\t')  # TSV format uses tab as delimiter

# Display the first few rows to ensure it's loaded correctly
#print(dataset.head(10))
# Reading the dataset as CSV instead of TSV
dataset = pd.read_csv(file_name, delimiter=',')  # Using comma as delimiter

# Check the first few rows of the dataset to ensure it's loaded properly:
print("Dataset head:")
print(dataset.head())
print("\n-------------------------\n")


```

```
Dataset head:
   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

   environmentsatisfaction  gender  hourlyrate  jobinvolvement  joblevel  \
0                        2  Female          94               3         2
1                        3    Male          61               2         2
2                        4    Male          92               2         1
3                        4  Female          56               3         1
4                        1    Male          40               3         1

                jobrole  jobsatisfaction maritalstatus  monthlyincome  \
0        Sales Executive                4        Single           5993
1     Research Scientist                2       Married           5130
2  Laboratory Technician                3        Single           2090
3     Research Scientist                3       Married           2909
4  Laboratory Technician                2       Married           3468

   monthlyrate  numcompaniesworked over18 overtime  percentsalaryhike  \
0        19479                   8      Y      Yes                 11
1        24907                   1      Y       No                 23
2         2396                   6      Y      Yes                 15
3        23159                   1      Y      Yes                 11
4        16632                   9      Y       No                 12

   performancerating  relationshipsatisfaction  standardhours  \
0                  3                         1             80
1                  4                         4             80
2                  3                         2             80
3                  3                         3             80
4                  3                         4             80

   stockoptionlevel  totalworkingyears  trainingtimeslastyear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   worklifebalance  yearsatcompany  yearsincurrentrole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   yearssincelastpromotion  yearswithcurrmanager
```

```
# Check the shape of dataframe.
# YOUR CODE HERE
#print(dataset.shape)
dataset.shape

```

```
(1470, 35)
```

There can be more than one file to read as this is introduced as a competition, dataset has one file for training the model. Their can be other files as one containing the test features and the other can be the true labels.

Data Exploration

- Check for missing values
- Check for consistent data type across a feature
- Check for outliers or inconsistencies in data columns
- Check for correlated features
- Do we have a target label imbalance
- How our independent variables are distributed relative to our target label
- Are there features that have strong linear or monotonic relationships? Making correlation heatmaps makes it easy to identify possible collinearity

Exercise 2: Create a `List` of numerical and categorical columns. Display a statistical description of the dataset. Remove missing values (if any) [0.5 Mark]

Hint: Use `for` to iterate through each column.

```
# YOUR CODE HERE
import pandas as pd

# Assuming you've already read the dataset:
```

```
# dataset = pd.read_csv(file_name, delimiter='\t')

# Create empty lists for numerical and categorical columns
numerical_columns = []
categorical_columns = []

# Iterate through columns to classify them as numerical or categorical using a for loop
for column in dataset.columns:
    if dataset[column].dtype in ['int64', 'float64']:
        numerical_columns.append(column)
    else:
        categorical_columns.append(column)

print("Numerical Columns:", numerical_columns)
print("Categorical Columns:", categorical_columns)

# Display a statistical description of the dataset
print("\nStatistical Description of the Dataset:\n")
print(dataset.describe(include='all'))

# Remove rows with missing values
missing_before = dataset.isnull().sum().sum()
dataset.dropna(inplace=True)
missing_after = dataset.isnull().sum().sum()

print(f"\nNumber of missing values before removal: {missing_before}")
print(f"Number of missing values after removal: {missing_after}")
```

```
Numerical Columns: ['age', 'dailyrate', 'distancefromhome', 'education', 'employeecount', 'employeenumber', 'environments
Categorical Columns: ['attrition', 'businesstravel', 'department', 'educationfield', 'gender', 'jobrole', 'maritalstatus'

Statistical Description of the Dataset:

                age attrition businesstravel    dailyrate  \
count   1470.000000      1470           1470  1470.000000
unique          NaN         2              3          NaN
top             NaN        No  Travel_Rarely          NaN
freq            NaN      1233           1043          NaN
mean      36.923810       NaN            NaN   802.485714
std        9.135373       NaN            NaN   403.509100
min       18.000000       NaN            NaN   102.000000
25%       30.000000       NaN            NaN   465.000000
50%       36.000000       NaN            NaN   802.000000
75%       43.000000       NaN            NaN  1157.000000
max       60.000000       NaN            NaN  1499.000000

                    department  distancefromhome    education educationfield  \
count                     1470       1470.000000  1470.000000           1470
unique                       3               NaN          NaN              6
top     Research & Development               NaN          NaN  Life Sciences
freq                       961               NaN          NaN            606
mean                       NaN          9.192517     2.912925            NaN
std                        NaN          8.106864     1.024165            NaN
min                        NaN          1.000000     1.000000            NaN
25%                        NaN          2.000000     2.000000            NaN
50%                        NaN          7.000000     3.000000            NaN
75%                        NaN         14.000000     4.000000            NaN
max                        NaN         29.000000     5.000000            NaN

        employeecount  employeenumber  environmentsatisfaction gender  \
count          1470.0     1470.000000               1470.000000   1470
unique            NaN             NaN                       NaN      2
top               NaN             NaN                       NaN   Male
freq              NaN             NaN                       NaN    882
mean              1.0     1024.865306                  2.721769    NaN
std               0.0      602.024335                  1.093082    NaN
min               1.0        1.000000                  1.000000    NaN
25%               1.0      491.250000                  2.000000    NaN
50%               1.0     1020.500000                  3.000000    NaN
75%               1.0     1555.750000                  4.000000    NaN
max               1.0     2068.000000                  4.000000    NaN

         hourlyrate  jobinvolvement     joblevel          jobrole  \
count   1470.000000     1470.000000  1470.000000             1470
unique          NaN             NaN          NaN                9
top             NaN             NaN          NaN  Sales Executive
freq            NaN             NaN          NaN              326
mean      65.891156        2.729932     2.063946              NaN
std       20.329428        0.711561     1.106940              NaN
min       30.000000        1.000000     1.000000              NaN
25%       48.000000        2.000000     1.000000              NaN
```

```
50%       66.000000        3.000000     2.000000              NaN
75%       83.750000        3.000000     3.000000              NaN
max      100.000000        4.000000     5.000000              NaN

        jobsatisfaction maritalstatus  monthlyincome   monthlyrate  \
```

First, we want to get a sense of our data:

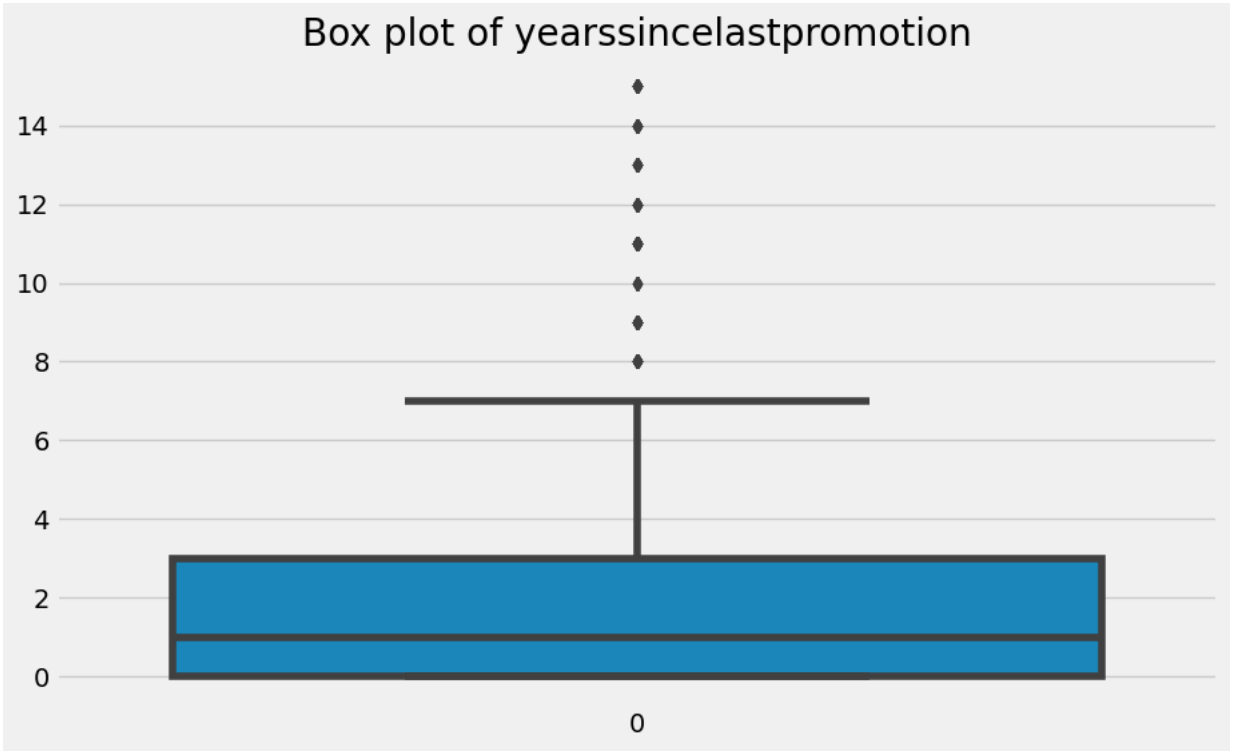
- What features have the most divergent distributions based on target class
- Do we have a target label imbalance
- How our independent variables are distributed relative to our target label
- Are there features that have strong linear or monotonic relationships, making correlation heatmaps makes it easy to identify possible colinearity

Check for outliers

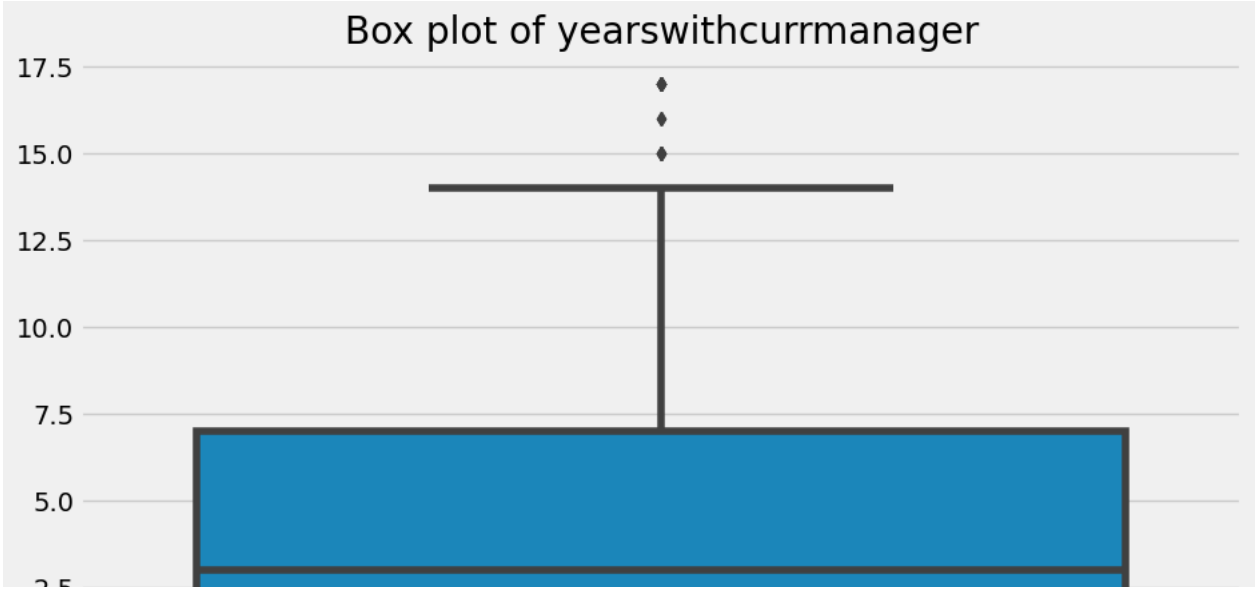
Exercise 3: Create a box plot to check for outliers [0.5 Mark]

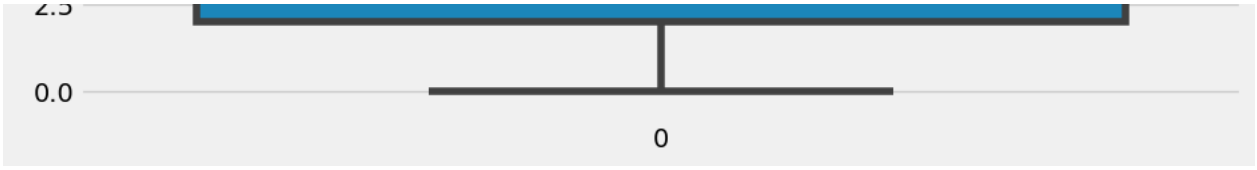
```
# Check for outliers
# YOUR CODE HERE
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming you've already read your dataset into a DataFrame named 'dataset'

# Extract numerical columns if you haven't already
numerical_columns = [column for column in dataset.columns if dataset[column].dtype in ['int64', 'float64']]

# Create box plots for each numerical column and print the outliers
for column in numerical_columns:
    plt.figure(figsize=(10, 6))
    sns.boxplot(dataset[column])
    plt.title(f"Box plot of {column}")
    plt.show()

    # Calculate IQR for the column
    Q1 = dataset[column].quantile(0.25)
    Q3 = dataset[column].quantile(0.75)
    IQR = Q3 - Q1

    # Determine bounds for outliers
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    # Extract outliers
    outliers = dataset[(dataset[column] < lower_bound) | (dataset[column] > upper_bound)][column]

    # Print outliers if they exist
    if outliers.empty:
        print(f"No outliers found for {column}.\n")
    else:
        print(f"Outliers for {column}:")
        print(outliers, "\n")


```

```
190     18
231     17
281     16
417     15
466     16
595     15
716     16
746     16
861     15
976     16
1024    17
1150    15
1156    15
1221    15
1327    17
1351    17
1430    16
Name: yearsincurrentrole, dtype: int64
```




```
Outliers for yearssincelastpromotion:
15       8
45      15
46       8
55       8
61       9
        ..
1414    12
1425     8
1444     9
1447    11
1462     9
Name: yearssincelastpromotion, Length: 107, dtype: int64
```






```
Outliers for yearswithcurrmanager:
28      17
123     15
153     15
187     15
231     15
386     17
561     16
616     17
635     15
686     17
875     17
926     17
1078    17
1348    16
Name: yearswithcurrmanager, dtype: int64
```

Handling outliers

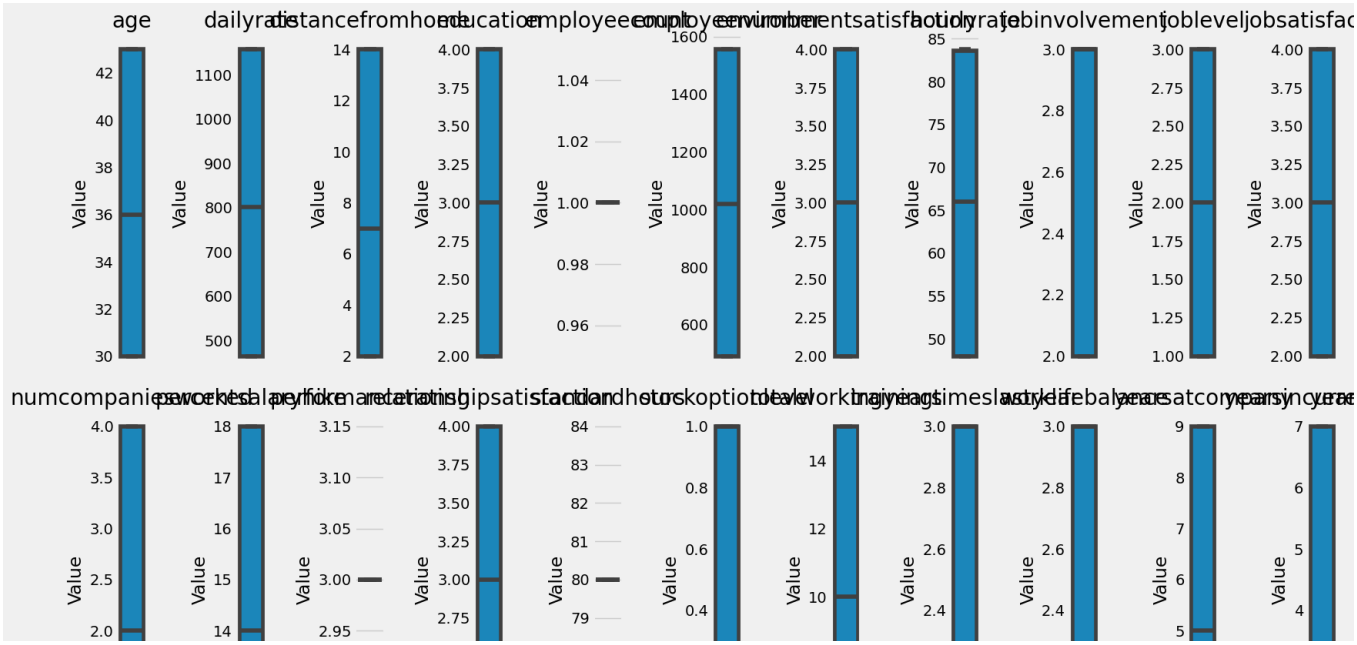
Exercise 4: Use lower bound as 25% and upper bound as 75% to handle the outliers [0.5 Mark]

```
# YOUR CODE HERE
import pandas as pd

# Assuming you've already read your dataset into a DataFrame named 'dataset'

# Check the first few rows of the dataset to ensure it's loaded properly:
print("Dataset head:")
print(dataset.head())
print("\n------------------------\n")

# Extract numerical columns if you haven't already
numerical_columns = [column for column in dataset.columns if dataset[column].dtype in ['int64', 'float64']]

# Display the numerical columns to verify:
print("Numerical columns:")
print(numerical_columns)
print("\n------------------------\n")

# For each numerical column, cap the data with the 25% and 75% quantiles
for column in numerical_columns:
    Q1 = dataset[column].quantile(0.25)
    Q3 = dataset[column].quantile(0.75)

    # Replacing values below Q1 with Q1 and values above Q3 with Q3
    dataset[column] = dataset[column].apply(lambda x: Q1 if x < Q1 else (Q3 if x > Q3 else x))

# Check the revised data
print("Revised Data Description:")
```

```
print(dataset[numerical_columns].describe())

```

```
Dataset head:
   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

   environmentsatisfaction  gender  hourlyrate  jobinvolvement  joblevel  \
0                        2  Female          94               3         2
1                        3    Male          61               2         2
2                        4    Male          92               2         1
3                        4  Female          56               3         1
4                        1    Male          40               3         1

                jobrole  jobsatisfaction maritalstatus  monthlyincome  \
0        Sales Executive                4        Single           5993
1     Research Scientist                2       Married           5130
2  Laboratory Technician                3        Single           2090
3     Research Scientist                3       Married           2909
4  Laboratory Technician                2       Married           3468

   monthlyrate  numcompaniesworked over18 overtime  percentsalaryhike  \
0        19479                   8      Y      Yes                 11
1        24907                   1      Y       No                 23
2         2396                   6      Y      Yes                 15
3        23159                   1      Y      Yes                 11
4        16632                   9      Y       No                 12

   performancerating  relationshipsatisfaction  standardhours  \
0                  3                         1             80
1                  4                         4             80
2                  3                         2             80
3                  3                         3             80
4                  3                         4             80

   stockoptionlevel  totalworkingyears  trainingtimeslastyear  \
0                 0                  8                      0
1                 1                 10                      3
2                 0                  7                      3
3                 0                  8                      3
4                 1                  6                      3

   worklifebalance  yearsatcompany  yearsincurrentrole  \
0                1               6                   4
1                3              10                   7
2                3               0                   0
3                3               8                   7
4                3               2                   2

   yearssincelastpromotion  yearswithcurrmanager
```

```
# Recheck for outliers
# YOUR CODE HERE
import matplotlib.pyplot as plt
import seaborn as sns

# Setting the size for the plots
plt.figure(figsize=(20, 10))

# Plotting boxplots for all numerical columns
for i, column in enumerate(numerical_columns, 1):
    plt.subplot(2, len(numerical_columns)//2, i)
    sns.boxplot(y=dataset[column])
    plt.title(column)
    plt.ylabel('Value')
    plt.tight_layout()

plt.show()

```



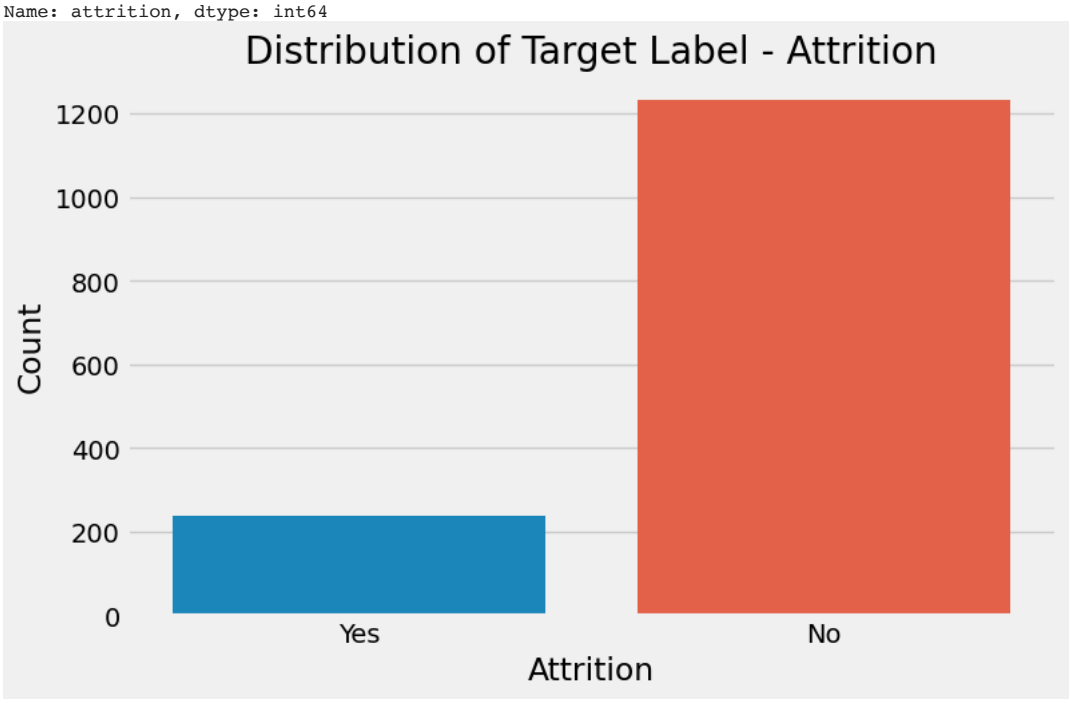
Target label imbalance

Exercise 5: Check if there is an imbalance in target label [0.5 Mark]

Hint: Use value_counts()

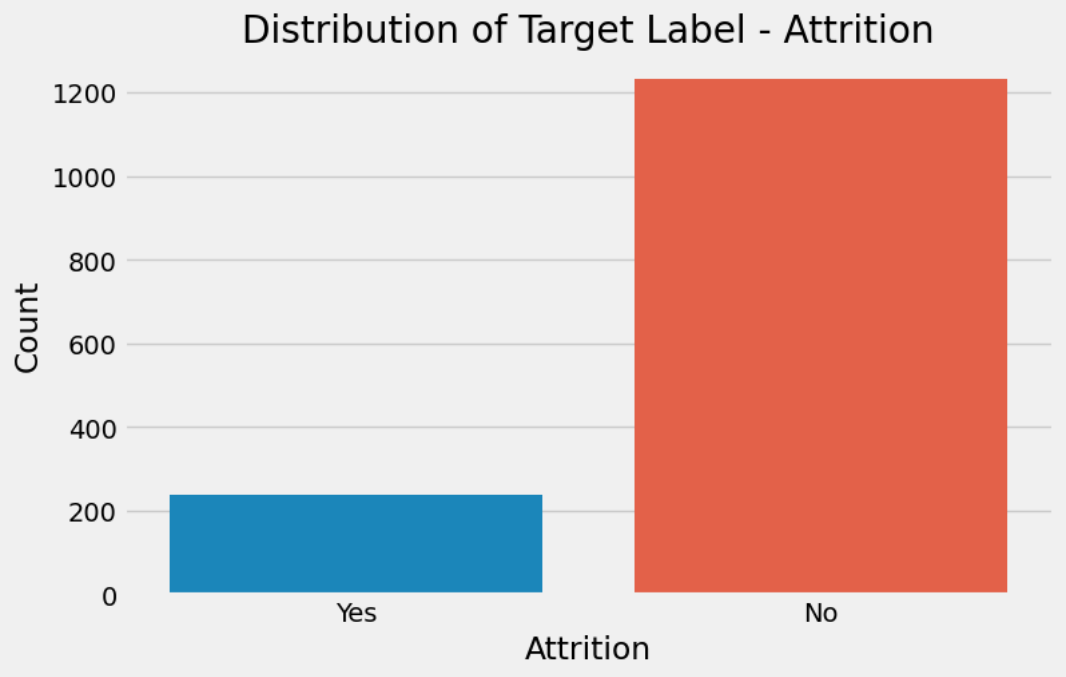
```
# Count of unique values in Attrition column
# YOUR CODE HERE
# Checking the distribution of the target label
print(dataset['attrition'].value_counts())

# Plotting the distribution
plt.figure(figsize=(8, 5))
sns.countplot(data=dataset, x='attrition')
plt.title("Distribution of Target Label - Attrition")
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.show()

```

```
No     1233
Yes     237
Name: attrition, dtype: int64
```




```
# Plot barplot to visualize balance/imbalance
# YOUR CODE HERE
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting the distribution
plt.figure(figsize=(8, 5))
sns.countplot(data=dataset, x='attrition')
plt.title("Distribution of Target Label - Attrition")
plt.xlabel('Attrition')
plt.ylabel('Count')
plt.show()

```



If there is any imbalance in the dataset then a few techniques can be utilised (optional):

1. SMOTE
2. Cross Validation
3. Regularizing the model's parameters

Plot pairplot

Exercise 6: Visualize the relationships between the predictor variables and the target variable using a pairplot [0.5 Mark]

Hint: Use sns.pairplot

```
# Visualize a pairplot with relevant features
# YOUR CODE HERE
import seaborn as sns

# Selecting a few features for demonstration purposes. Adjust this list as needed.
selected_features = ['age', 'dailyrate', 'distancefromhome', 'monthlyincome']

# Adding the target label to visualize the relationships based on the target classes
selected_features.append('attrition')

# Creating the pairplot
sns.pairplot(dataset[selected_features], hue='attrition', plot_kws={'alpha':0.5})
plt.show()

```

Explore Correlation

- Plotting the Heatmap

Exercise 7: Visualize the correlation among IBM employee attrition numerical features using a heatmap [0.5 Mark]

```
# Visualize heatmap
# YOUR CODE HERE
import matplotlib.pyplot as plt
import seaborn as sns

# Compute the correlation matrix
corr_matrix = dataset[numerical_columns].corr()

# Create a heatmap to visualize the correlations
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title("Correlation Heatmap of Numerical Features")
plt.show()

```