**Outlier Treatments**

**Instructions**:

Please share your answers filled inline in the word document. Submit code files wherever applicable.

Please ensure you update all the details:

**Name: \_\_Seema Iyengar\_\_\_\_**

**Batch Id: \_\_\_EDA\_06042023\_10AM\_\_\_**

**Topic: Data Pre-Processing**

**Problem Statement:**

Most of the datasets have extreme values or exceptions in their observations. These values affect the predictions (Accuracy) of the model in one way or the other, removing these values is not a very good option. For these types of scenarios, we have various techniques to treat such values.

Refer: <https://360digitmg.com/mindmap-data-science>

1. Prepare the dataset by performing the preprocessing techniques, to treat the outliers.

A picture containing shape, arrow

Description automatically generated**

**Hints:**

For each assignment, the solution should be submitted in the below format

1. Work on each feature to create a data dictionary as displayed in the image displayed below:
2. Hint: Boston dataset is publicly available. Refer to the Boston.csv file.
3. Research and perform all possible steps for obtaining the solutions.
4. All the codes (executable programs) should execute without errors.
5. Code modularization should be followed.
6. Each line of code should have comments explaining the logic and why you are using that function
7. Detailed explanation of your approach is mandatory.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

boston = pd.read\_csv(r"C:\Users\seema\OneDrive\Desktop\COURSE\360DigiTMG Course\EDA\_06042023\_10AM\Module02 - C.Data Preprocessing\InClass\_DataPreprocessing\_datasets\Boston.csv")

boston.info()

# to check the outliers, use boxplot for each and every column

# crim

sns.boxplot(boston.crim)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for crim based on IQR)

IQR = boston['crim'].quantile(0.75) - boston['crim'].quantile(0.25)

lower\_limit = boston['crim'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['crim'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_crim'] = pd.DataFrame(np.where(boston['crim'] > upper\_limit, upper\_limit, np.where(boston['crim'] < lower\_limit, lower\_limit, boston['crim'])))

sns.boxplot(boston.boston\_crim)

# as per the above boxplot now there are no outliers in 'crim' column

# zn column

sns.boxplot(boston.zn)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for zn based on IQR)

IQR = boston['zn'].quantile(0.75) - boston['zn'].quantile(0.25)

lower\_limit = boston['zn'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['zn'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_zn'] = pd.DataFrame(np.where(boston['zn'] > upper\_limit, upper\_limit, np.where(boston['zn'] < lower\_limit, lower\_limit, boston['zn'])))

sns.boxplot(boston.boston\_zn)

# as per the above boxplot now there are no outliers in 'zn' column

# indus column

sns.boxplot(boston.indus)

# as per the box plot, outliers do not exist hence no need to treat them

#chas

sns.boxplot(boston.chas)

# as per the box plot, outliers do not exist hence no need to treat them

#nox

sns.boxplot(boston.nox)

# as per the box plot, outliers do not exist hence no need to treat them

#rm

sns.boxplot(boston.rm)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for rm based on IQR)

IQR = boston['rm'].quantile(0.75) - boston['rm'].quantile(0.25)

lower\_limit = boston['rm'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['rm'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_rm'] = pd.DataFrame(np.where(boston['rm'] > upper\_limit, upper\_limit, np.where(boston['rm'] < lower\_limit, lower\_limit, boston['rm'])))

sns.boxplot(boston.boston\_rm)

# as per the above boxplot now there are no outliers in 'rm' column

#age

sns.boxplot(boston.age)

# as per the box plot, outliers do not exist hence no need to treat them

#dis

sns.boxplot(boston.dis)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for dis based on IQR)

IQR = boston['dis'].quantile(0.75) - boston['dis'].quantile(0.25)

lower\_limit = boston['dis'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['dis'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_dis'] = pd.DataFrame(np.where(boston['dis'] > upper\_limit, upper\_limit, np.where(boston['dis'] < lower\_limit, lower\_limit, boston['dis'])))

sns.boxplot(boston.boston\_dis)

# as per the above boxplot now there are no outliers in 'dis' column

#rad

sns.boxplot(boston.rad)

# as per the box plot, outliers do not exist hence no need to treat them

#tax

sns.boxplot(boston.tax)

# as per the box plot, outliers do not exist hence no need to treat them

#ptratio

sns.boxplot(boston.ptratio)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for ptratio based on IQR)

IQR = boston['ptratio'].quantile(0.75) - boston['ptratio'].quantile(0.25)

lower\_limit = boston['ptratio'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['ptratio'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_ptratio'] = pd.DataFrame(np.where(boston['ptratio'] > upper\_limit, upper\_limit, np.where(boston['ptratio'] < lower\_limit, lower\_limit, boston['ptratio'])))

sns.boxplot(boston.boston\_ptratio)

# as per the above boxplot now there are no outliers in 'ptratio' column

#black

sns.boxplot(boston.black)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for black based on IQR)

IQR = boston['black'].quantile(0.75) - boston['black'].quantile(0.25)

lower\_limit = boston['black'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['black'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_black'] = pd.DataFrame(np.where(boston['black'] > upper\_limit, upper\_limit, np.where(boston['black'] < lower\_limit, lower\_limit, boston['black'])))

sns.boxplot(boston.boston\_black)

# as per the above boxplot now there are no outliers in 'black' column

#lstat

sns.boxplot(boston.lstat)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for lstat based on IQR)

IQR = boston['lstat'].quantile(0.75) - boston['lstat'].quantile(0.25)

lower\_limit = boston['lstat'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['lstat'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_lstat'] = pd.DataFrame(np.where(boston['lstat'] > upper\_limit, upper\_limit, np.where(boston['lstat'] < lower\_limit, lower\_limit, boston['lstat'])))

sns.boxplot(boston.boston\_lstat)

# as per the above boxplot now there are no outliers in 'lstat' column

#medv

sns.boxplot(boston.medv)

# as per the box plot lot of outliers exist, hence need to remove them using the below techniques

# Detection of outliers (find limits for medv based on IQR)

IQR = boston['medv'].quantile(0.75) - boston['medv'].quantile(0.25)

lower\_limit = boston['medv'].quantile(0.25) - (IQR \* 1.5)

upper\_limit = boston['medv'].quantile(0.75) + (IQR \* 1.5)

############### 2. Replace ###############

# Replace the outliers by the maximum and minimum limit

boston['boston\_medv'] = pd.DataFrame(np.where(boston['medv'] > upper\_limit, upper\_limit, np.where(boston['medv'] < lower\_limit, lower\_limit, boston['medv'])))

sns.boxplot(boston.boston\_medv)

# as per the above boxplot now there are no outliers in 'medv' column

# as per the analysis, all outliers have been treated and removed from the respective columns