**Outlier Treatments**

Instructions:

Please share your answers filled inline in the word document. Submit Python code and R code files wherever applicable as Python File (.py) and R file as .r extension files.

Please ensure you update all the details:

**Name: ANANDAKRISHNAN K V**

**Batch Id: 190421**

**Topic: Preliminaries for Data Analysis**

**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 the techniques for treating such values. Explore on various other techniques to treat these values, you can go through this link:

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

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

**

**Ans:-**

**Python code:**

############## Outliers Treatment #################

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats.mstats import winsorize

boston = pd.DataFrame(boston\_datacsv)

boston.shape

bstn = boston.copy()

b1 = boston.copy()

bstn.isna().sum()

#getting boston collumns names

bstn.columns

# Boxplots

for i in bstn.describe().columns:

sns.boxplot(bstn[i].dropna())

plt.show()

#Columns which doesn't have outliers

sns.boxplot(bstn['indus']);plt.title('box plot for boston.indus')

sns.boxplot(bstn['nox']);plt.title('box plot for boston.nox')

sns.boxplot(bstn['age']);plt.title('box plot for boston.age')

sns.boxplot(bstn['rad']);plt.title('box plot for boston.rad')

sns.boxplot(bstn['tax']);plt.title('box plot for boston.tax')

#Outlier Treatment for boston.crim

sns.boxplot(bstn.crim);plt.title('box plot for boston.crim')

q1 = bstn['crim'].quantile(0.25)

q1

q3 = bstn['crim'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['crim']<lower\_limit, True, np.where(bstn['crim']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['crim'][outliers]

outliers.sum()

outlier\_values

#------------------------------------------------------------------------------

#Trimming

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['crim']);plt.title('box plot for boston.crim after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

boston = pd.DataFrame(boston\_datacsv)

bstn = boston.copy()

bstn['crim'] = np.where(bstn['crim']<q1, lower\_limit, np.where(bstn['crim']>q3, upper\_limit, bstn['crim']))

sns.boxplot(bstn['crim']);plt.title('box plot for boston.crim after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['crim']);plt.title('box plot for boston.crim before winsorization')

b1['crim'] = winsorize(bstn['crim'], limits = [0,0.12])

sns.boxplot(b1['crim']);plt.title('box plot for boston.crim after winsorization')

#from feature\_engine.outliers import Winsorizer

#winsorizer = Winsorizer(capping\_method='iqr',tail='both',fold=1.5,variables=['crim'])

#df\_t = winsorizer.fit\_transform(bstn[['crim']])

#------------------------------------------------------------------------------

#outlier treatment for boston.zn

bstn=boston.copy()

q1 = bstn['zn'].quantile(0.25)

q1

q3 = bstn['zn'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['zn']<lower\_limit, True, np.where(bstn['zn']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['zn'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['zn']);plt.title('box plot for boston.zn')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outlier\_values), ]

sns.boxplot(bstn['zn']);plt.title('box plot for boston.zn after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['zn'] = np.where(bstn['zn']<q1, lower\_limit, np.where(bstn['zn']>q3, upper\_limit, bstn['zn']))

sns.boxplot(bstn['zn']);plt.title('box plot for boston.zn after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['zn']);plt.title('box plot for boston.zn before winsorization')

b1['zn'] = winsorize(bstn['zn'], limits = [0,0.13])

sns.boxplot(b1['zn']);plt.title('box plot for boston.zn after winsorization')

#------------------------------------------------------------------------------

#outlier treatment for boston.rm

bstn=boston.copy()

q1 = bstn['rm'].quantile(0.25)

q1

q3 = bstn['rm'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['rm']<lower\_limit, True, np.where(bstn['rm']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['rm'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['rm']);plt.title('box plot for boston.rm')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['rm']);plt.title('box plot for boston.rm after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['rm'] = np.where(bstn['rm']<q1, lower\_limit, np.where(bstn['rm']>q3, upper\_limit, bstn['rm']))

sns.boxplot(bstn['rm']);plt.title('box plot for boston.rm after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['rm']);plt.title('box plot for boston.rm before winsorization')

b1['rm'] = winsorize(bstn['rm'], limits = [0.04,0.04])

sns.boxplot(b1['rm']);plt.title('box plot for boston.rm after winsorization')

#------------------------------------------------------------------------------

#outlier treatment for boston.dis

bstn=boston.copy()

q1 = bstn['dis'].quantile(0.25)

q1

q3 = bstn['dis'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['dis']<lower\_limit, True, np.where(bstn['dis']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['dis'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['dis']);plt.title('box plot for boston.dis')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['dis']);plt.title('box plot for boston.dis after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['dis'] = np.where(bstn['dis']<q1, lower\_limit, np.where(bstn['dis']>q3, upper\_limit, bstn['dis']))

sns.boxplot(bstn['dis']);plt.title('box plot for boston.dis after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['dis']);plt.title('box plot for boston.dis before winsorization')

b1['dis'] = winsorize(bstn['dis'], limits = [0,0.02])

sns.boxplot(b1['dis']);plt.title('box plot for boston.dis after winsorization')

#------------------------------------------------------------------------------

#outlier treatment for boston.ptratio

bstn=boston.copy()

q1 = bstn['ptratio'].quantile(0.25)

q1

q3 = bstn['ptratio'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['ptratio']<lower\_limit, True, np.where(bstn['ptratio']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['ptratio'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['ptratio']);plt.title('box plot for boston.ptratio')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['ptratio']);plt.title('box plot for boston.ptratio after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['ptratio'] = np.where(bstn['ptratio']<q1, lower\_limit, np.where(bstn['ptratio']>q3, upper\_limit, bstn['ptratio']))

sns.boxplot(bstn['ptratio']);plt.title('box plot for boston.ptratio after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['ptratio']);plt.title('box plot for boston.ptratio before winsorization')

b1['ptratio'] = winsorize(bstn['ptratio'], limits = [0.03,0.1])

sns.boxplot(b1['ptratio']);plt.title('box plot for boston.ptratio after winsorization')

#------------------------------------------------------------------------------

#outlier treatment for boston.black

bstn=boston.copy()

q1 = bstn['black'].quantile(0.25)

q1

q3 = bstn['black'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['black']<lower\_limit, True, np.where(bstn['black']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['black'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['black']);plt.title('box plot for boston.black')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['black']);plt.title('box plot for boston.black after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['black'] = np.where(bstn['black']<q1, lower\_limit, np.where(bstn['black']>q3, upper\_limit, bstn['black']))

sns.boxplot(bstn['black']);plt.title('box plot for boston.black after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['black']);plt.title('box plot for boston.black before winsorization')

b1['black'] = winsorize(bstn['black'], limits = [0.16,0])

sns.boxplot(b1['black']);plt.title('box plot for boston.black after winsorization')

#-----------------------------------------------------------------------------

#outlier treatment for boston.lstat

bstn=boston.copy()

q1 = bstn['lstat'].quantile(0.25)

q1

q3 = bstn['lstat'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['lstat']<lower\_limit, True, np.where(bstn['lstat']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['lstat'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['lstat']);plt.title('box plot for boston.lstat')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['lstat']);plt.title('box plot for boston.lstat after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['lstat'] = np.where(bstn['lstat']<q1, lower\_limit, np.where(bstn['lstat']>q3, upper\_limit, bstn['lstat']))

sns.boxplot(bstn['lstat']);plt.title('box plot for boston.lstat after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['lstat']);plt.title('box plot for boston.lstat before winsorization')

b1['lstat'] = winsorize(bstn['lstat'], limits = [0,0.01])

sns.boxplot(b1['lstat']);plt.title('box plot for boston.lstat after winsorization')

#------------------------------------------------------------------------------

#outlier treatment for boston.medv

bstn=boston.copy()

q1 = bstn['medv'].quantile(0.25)

q1

q3 = bstn['medv'].quantile(0.75)

q3

IQR = q3 - q1

IQR

lower\_limit = q1-(1.5\*IQR)

lower\_limit

upper\_limit = q3+(1.5\*IQR)

upper\_limit

#finding outliers indexes

outliers = np.where(bstn['medv']<lower\_limit, True, np.where(bstn['medv']>upper\_limit, True, False))

outliers

#finding outlier values

outlier\_values = bstn['medv'][outliers]

outliers.sum()

outlier\_values

sns.boxplot(bstn['medv']);plt.title('box plot for boston.medv')

#------------------------------------------------------------------------------

#Trimming

bstn = boston.copy()

bstn = bstn.loc[~(outliers), ]

sns.boxplot(bstn['medv']);plt.title('box plot for boston.medv after trimming')

bstn.shape #getting dimensions or shape of bstn

boston.shape #getting dimensions or shape of boston

#------------------------------------------------------------------------------

#Replacing

bstn = boston.copy()

bstn['medv'] = np.where(bstn['medv']<q1, lower\_limit, np.where(bstn['medv']>q3, upper\_limit, bstn['medv']))

sns.boxplot(bstn['medv']);plt.title('box plot for boston.medv after replacing')

#------------------------------------------------------------------------------

#Winsorization

bstn=boston.copy()

sns.boxplot(bstn['medv']);plt.title('box plot for boston.medv before winsorization')

b1['medv'] = winsorize(bstn['medv'], limits = [0.01,0.07])

sns.boxplot(b1['medv']);plt.title('box plot for boston.medv after winsorization')

#comparision for both boxplots after Winsorization

bstn.boxplot()

b1.boxplot()

**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 Boston.csv file.
3. Research and perform all possible steps for obtaining solution
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