```
# -*- coding: utf-8 -*-
"""Alaiba__Nawaz_Day2.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/18tcRHYfqiW5ggcWlYA3Uo_yuLVydgXGl
"""

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
from sklearn.metrics import mean_squared_error, r2_score

#loading data in dataframes
data = pd.read_csv("/content/train.csv")

#getting information of data
data.info()

"""Visualizing all features"""

#visualising ID by plotting histogram and box plot
ID = data['ID']
plt.title("ID histogram")
plt.hist(ID,ec='red')
plt.show()
print("\n")
plt.title("ID boxplot")
plt.boxplot(ID)
plt.show()

#visualising crim    by plotting histogram and box plot
```

```
crim = data['crim']
plt.title("crim histogram")
plt.hist(crim,ec='red')
plt.show()
print("\n")
plt.title("crim boxplot")
plt.boxplot(crim)
plt.show()

#visualising zn by plotting histogram and box plot
zn = data['zn']
plt.title("zn histogram")
plt.hist(zn,ec='red')
plt.show()
print("\n")
plt.title("zn boxplot")
plt.boxplot(zn)
plt.show()

#visualising indus by plotting histogram and box plot
indus = data['indus']
plt.title("indus histogram")
plt.hist(indus,ec='red')
plt.show()
print("\n")
plt.title("indus boxplot")
plt.boxplot(indus)
plt.show()

#visualising chas    by plotting histogram and box plot
chas = data['chas']
plt.title("chas histogram")
plt.hist(chas,ec='red')
plt.show()
print("\n")
plt.title("chas boxplot")
plt.boxplot(chas)
plt.show()

#visualising nox by plotting histogram and box plot
```

```
nox = data['nox']
plt.title("nox histogram")
plt.hist(nox,ec='red')
plt.show()
print("\n")
plt.title("nox boxplot")
plt.boxplot(nox)
plt.show()

#visualising rm by plotting histogram and box plot
rm = data['rm']
plt.title("rm histogram")
plt.hist(rm,ec='red')
plt.show()
print("\n")
plt.title("rm boxplot")
plt.boxplot(rm)
plt.show()

#visualising age by plotting histogram and box plot
age = data['age']
plt.title("age histogram")
plt.hist(age,ec='red')
plt.show()
print("\n")
plt.title("age boxplot")
plt.boxplot(age)
plt.show()

#visualising dis by plotting histogram and box plot
dis = data['dis']
plt.title("dis histogram")
plt.hist(dis,ec='red')
plt.show()
print("\n")
plt.title("dis boxplot")
plt.boxplot(dis)
plt.show()

#visualising rad  by plotting histogram and box plot
```

```
rad = data['rad']
plt.title("rad histogram")
plt.hist(rad,ec='red')
plt.show()
print("\n")
plt.title("rad boxplot")
plt.boxplot(rad)
plt.show()

#visualising tax by plotting histogram and box plot
tax = data['tax']
plt.title("tax histogram")
plt.hist(tax,ec='red')
plt.show()
print("\n")
plt.title("tax boxplot")
plt.boxplot(tax)
plt.show()

#visualising ptratio by plotting histogram and box plot
ptratio = data['ptratio']
plt.title("ptratio histogram")
plt.hist(ptratio,ec='red')
plt.show()
print("\n")
plt.title("ptratio boxplot")
plt.boxplot(ptratio)
plt.show()

#visualising black by plotting histogram and box plot
black = data['black']
plt.title("black histogram")
plt.hist(black,ec='red')
plt.show()
print("\n")
plt.title("black boxplot")
plt.boxplot(black)
plt.show()

#visualising lstat by plotting histogram and box plot
```

```
lstat = data['lstat']
plt.title("lstat histogram")
plt.hist(lstat,ec='red')
plt.show()
print("\n")
plt.title("lstat boxplot")
plt.boxplot(lstat)
plt.show()

#visualising medv by plotting histogram and box plot
medv = data['medv']
plt.title("medv histogram")
plt.hist(medv,ec='red')
plt.show()
print("\n")
plt.title("medv boxplot")
plt.boxplot(medv)
plt.show()

#plotting heatmap to see which variable is most coorelated to the target
variable
corelation = data.corr()
plt.figure(figsize=(20,20))
sns.heatmap(corelation , annot=True)
plt.show()
#from heatmap we can see that the rm is most coorelated with medv having
value 0.69 so we take it as x

#splitting data in train and test sets
x = data['rm']
y = data['medv']
#20% testing and 80% training
X_train , X_test , y_train , y_test = train_test_split(x,y,test_size=0.2)

#preprocessing

#handeling missing values by removing them
imputer = IterativeImputer(random_state=0)
X_train = imputer.fit_transform(np.array(X_train).reshape(-1, 1))
X_test = imputer.fit_transform(np.array(X_test).reshape(-1, 1))
```

```
#feature scaling by minmax
scaler = MinMaxScaler()
X_train = scaler.fit_transform(np.array(X_train).reshape(-1, 1))
X_test = scaler.transform(np.array(X_test).reshape(-1, 1))

#applying linear regression
lr = LinearRegression()
lr.fit(X_train, y_train)
pred = lr.predict(X_test)

#Evaluate the model's performance on the testing data using metrics such
as mean squared error (MSE) or R-squared.
mse = mean_squared_error(y_test, pred)
r_square = r2_score(y_test, pred)
print('Mean Squared Error: ', mse)
print('R-square: ', r_square)
#0.63 r square shows that 63% of the variability in the dependent variable
is explained by the independent variable.
```