MACHINE LEARNING

ASSIGNMENT 1

NAME: DURGAM NAGA DEERAJ REDDY

USN: 22BTRAD013

Q: Load a dataset with missing values (Boston Housing Dataset).

```
[ ]  # DURGAM NAGA DEERAJ REDDY(22BTRAD013)
     import pandas as pd

     # Load the Boston Housing dataset
     data = pd.read_csv('/content/HousingData.csv')
```

CODE:

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
import pandas as pd


# Load the Boston Housing dataset
data = pd.read_csv('/content/HousingData.csv')
```

Q. Explore the description of the dataset

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
# Get the description of the dataset
print(data.describe())
```

```
             CRIM          ZN       INDUS        CHAS         NOX          RM  \
count  486.000000  486.000000  486.000000  486.000000  506.000000  506.000000
mean     3.611874   11.211934   11.083992    0.069959    0.554695    6.284634
std      8.720192   23.388876    6.835896    0.255340    0.115878    0.702617
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%      0.081900    0.000000    5.190000    0.000000    0.449000    5.885500
50%      0.253715    0.000000    9.690000    0.000000    0.538000    6.208500
75%      3.560263   12.500000   18.100000    0.000000    0.624000    6.623500
max     88.976200  100.000000   27.740000    1.000000    0.871000    8.780000

              AGE         DIS         RAD         TAX     PTRATIO           B  \
count  486.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean    68.518519    3.795043    9.549407  408.237154   18.455534  356.674032
std     27.999513    2.105710    8.707259  168.537116    2.164946   91.294864
min      2.900000    1.129600    1.000000  187.000000   12.600000    0.320000
25%     45.175000    2.100175    4.000000  279.000000   17.400000  375.377500
50%     76.800000    3.207450    5.000000  330.000000   19.050000  391.440000
75%     93.975000    5.188425   24.000000  666.000000   20.200000  396.225000
max    100.000000   12.126500   24.000000  711.000000   22.000000  396.900000

            LSTAT        MEDV
count  486.000000  506.000000
mean    12.715432   22.532806
std      7.155871    9.197104
min      1.730000    5.000000
25%      7.125000   17.025000
50%     11.430000   21.200000
75%     16.955000   25.000000
max     37.970000   50.000000
```

CODE:

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
# Get the description of the dataset
print(data.describe())
```

Q. Identify the number of missing values corresponding to each feature

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
# Check for missing values
print(data.isnull().sum())
```

```
CRIM       20
ZN         20
INDUS      20
CHAS       20
NOX         0
RM          0
AGE        20
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT      20
MEDV        0
dtype: int64
```

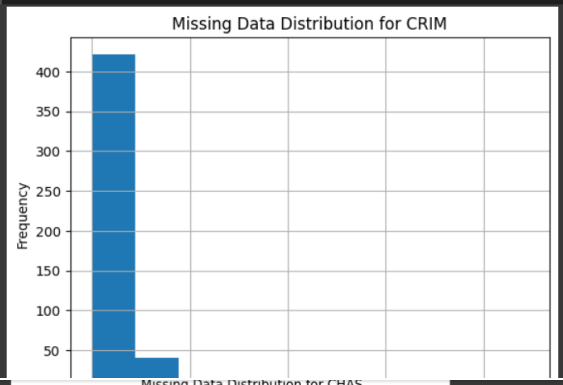
CODE:

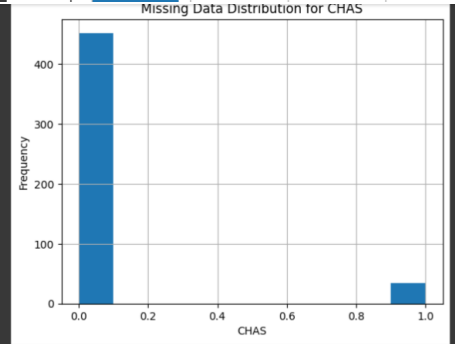
```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
# Check for missing values
print(data.isnull().sum())
```

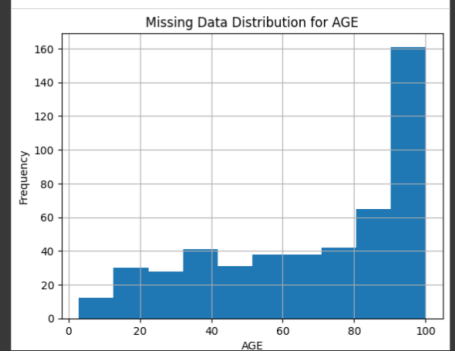
Q. Explore and visualize the missing data patterns.

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
import matplotlib.pyplot as plt

# Check for missing values and visualize them
for col in data.columns:
    if data[col].isnull().sum() > 0:
        data[col].hist()
        plt.title(f'Missing Data Distribution for {col}')
        plt.xlabel(col)
        plt.ylabel('Frequency')
        plt.show()
```







CODE:

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
import matplotlib.pyplot as plt

# Check for missing values and visualize them
for col in data.columns:
    if data[col].isnull().sum() > 0:
        data[col].hist()
        plt.title(f'Missing Data Distribution for {col}')
        plt.xlabel(col)
        plt.ylabel('Frequency')
        plt.show()
```

Q. Handle missing values using imputation method for a specific feature

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
from sklearn.impute import SimpleImputer

# Load the Boston Housing dataset
data = pd.read_csv('HousingData.csv')

# Check for missing values
print(data.isnull().sum())

# Impute missing values using mean imputation for 'CHAS' feature
imputer = SimpleImputer(strategy='mean')
imputer.fit(data[['CHAS']])
data['CHAS'] = imputer.transform(data[['CHAS']])

# Check for missing values after imputation
print(data.isnull().sum())
```

```
CRIM      20
ZN        20
INDUS     20
CHAS      20
NOX        0
RM         0
AGE       20
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT     20
MEDV       0
dtype: int64
CRIM      20
ZN        20
INDUS     20
CHAS       0
NOX        0
RM         0
AGE       20
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT     20
MEDV       0
dtype: int64
```

CODE:

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
from sklearn.impute import SimpleImputer

# Load the Boston Housing dataset
data = pd.read_csv('HousingData.csv')

# Check for missing values
```

```
print(data.isnull().sum())

# Impute missing values using mean imputation for 'CHAS' feature
imputer = SimpleImputer(strategy='mean')
imputer.fit(data[['CHAS']])
data['CHAS'] = imputer.transform(data[['CHAS']])

# Check for missing values after imputation
print(data.isnull().sum())
```

Q. Handle missing values using tuple removal method.

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
import pandas as pd

# Load the Boston Housing dataset
data = pd.read_csv('HousingData.csv')

# Drop rows with missing values
data_no_missing = data.dropna()

# Show the output
print(data_no_missing)
```

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  \
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296
1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242
2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242
3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222
5    0.02985   0.0   2.18   0.0  0.458  6.430  58.7  6.0622    3  222
..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...
499  0.17783   0.0   9.69   0.0  0.585  5.569  73.5  2.3999    6  391
500  0.22438   0.0   9.69   0.0  0.585  6.027  79.7  2.4982    6  391
502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273
503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273

     PTRATIO       B  LSTAT  MEDV
0       15.3  396.90   4.98  24.0
1       17.8  396.90   9.14  21.6
2       17.8  392.83   4.03  34.7
3       18.7  394.63   2.94  33.4
5       18.7  394.12   5.21  28.7
..       ...     ...    ...   ...
499     19.2  395.77  15.10  17.5
500     19.2  396.90  14.33  16.8
502     21.0  396.90   9.08  20.6
503     21.0  396.90   5.64  23.9
504     21.0  393.45   6.48  22.0

[394 rows x 14 columns]
```

CODE:

```
# DURGAM NAGA DEERAJ REDDY(22BTRAD013)
import pandas as pd

# Load the Boston Housing dataset
data = pd.read_csv('HousingData.csv')

# Drop rows with missing values
data_no_missing = data.dropna()

# Show the output
print(data_no_missing)
```

GITHUB:

https://github.com/DeeruReddy/Machine_learning