FUNDAMENTALS OF MACHINE LEARNING

LAB ASSIGNMENT - 4

NAME - Kaparotu Venkata Surya Tharani
USN - 22BTRAD018
BRANCH - AI & DE

Questions -

1. Load a dataset with outliers values (Boston Housing Dataset).

CODE :

```
# importing modules
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

# loading the data
f1=pd.read_csv("HousingData.csv")
print(f1.head())
```

Output :

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO       B  LSTAT  MEDV
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296     15.3  396.90   4.98  24.0
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242     17.8  396.90   9.14  21.6
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242     17.8  392.83   4.03  34.7
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222     18.7  394.63   2.94  33.4
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222     18.7  396.90    NaN  36.2
PS C:\Users\kvsth\Desktop\Term 7\Fundamentals of ML\Module 2>
```

2. Implement one-hot encoding.

- One hot encoding is a technique that we use to represent categorical variables as numerical values in a machine learning model.
- Each column contains "0" or "1" corresponding to which column it has been placed.
- A one-hot encoding is a representation of categorical variables as binary vectors.
- Many machine learning algorithms cannot work with categorical data directly. The categories must be converted into numbers. This is required for both input and output variables that are categorical.
- A one hot encoding allows the representation of categorical data to be more expressive.

Code :

```
one_hot_encoded_data = pd.get_dummies(f1, columns = ['CHAS'])
print(one_hot_encoded_data.head())
```

Output :

```
      CRIM    ZN  INDUS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO       B  LSTAT  MEDV  CHAS_0.0  CHAS_1.0
0  0.00632  18.0   2.31  0.538  6.575  65.2  4.0900    1  296     15.3  396.90   4.98  24.0         1         0
1  0.02731   0.0   7.07  0.469  6.421  78.9  4.9671    2  242     17.8  396.90   9.14  21.6         1         0
2  0.02729   0.0   7.07  0.469  7.185  61.1  4.9671    2  242     17.8  392.83   4.03  34.7         1         0
3  0.03237   0.0   2.18  0.458  6.998  45.8  6.0622    3  222     18.7  394.63   2.94  33.4         1         0
4  0.06905   0.0   2.18  0.458  7.147  54.2  6.0622    3  222     18.7  396.90    NaN  36.2         1         0
PS C:\Users\kvsth\Desktop\Term 7\Fundamentals of ML\Module 2>
```

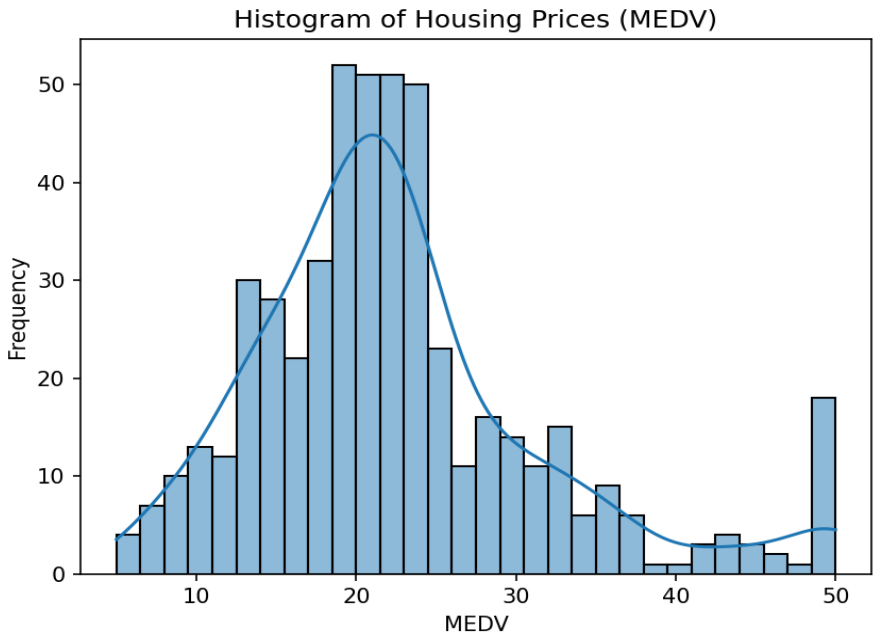
3. Create visualizations for different aspects of a dataset using Matplotlib or Seaborn.

Code :

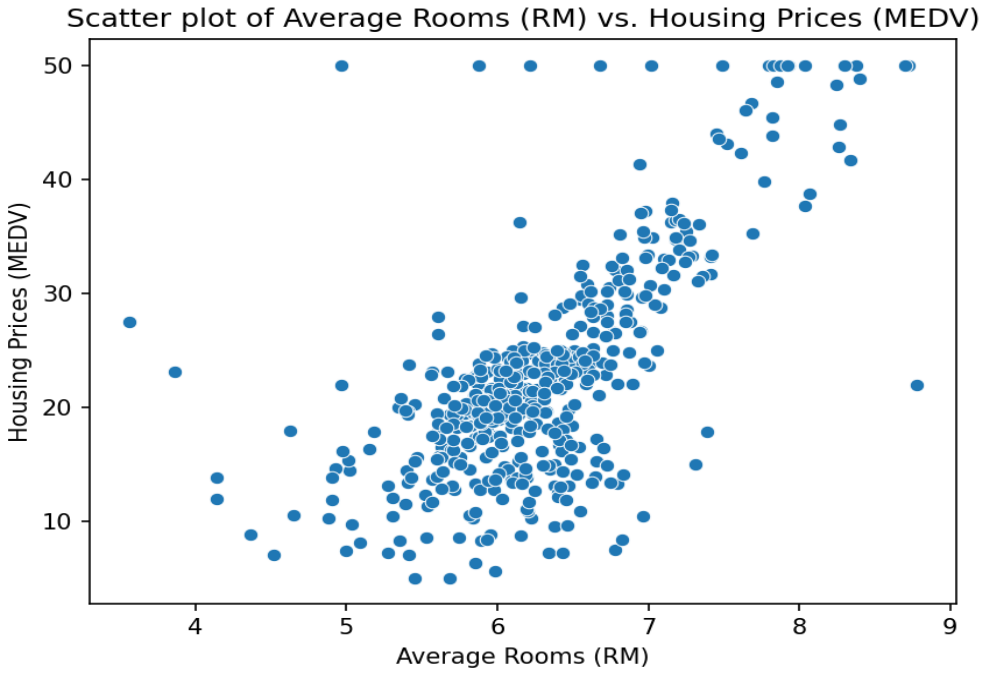
```
# Histogram of target variable (housing prices)
sns.histplot(f1['MEDV'], bins=30, kde=True)
plt.title('Histogram of Housing Prices (MEDV)')
plt.xlabel('MEDV')
plt.ylabel('Frequency')
plt.show()

# Correlation matrix heatmap
correlation_matrix = f1.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)
plt.title('Correlation Matrix')
plt.show()

# Scatter plot for a pair of features and the target variable
sns.scatterplot(data=f1, x='RM', y='MEDV')
plt.title('Scatter plot of Average Rooms (RM) vs. Housing Prices (MEDV)')
plt.xlabel('Average Rooms (RM)')
plt.ylabel('Housing Prices (MEDV)')
plt.show()

# box plot for RM feature
sns.boxplot(x=f1['RM'])
plt.title('Boxplot of Average Rooms (RM)')
plt.xlabel('Average Rooms (RM)')
plt.show()
```

Output :

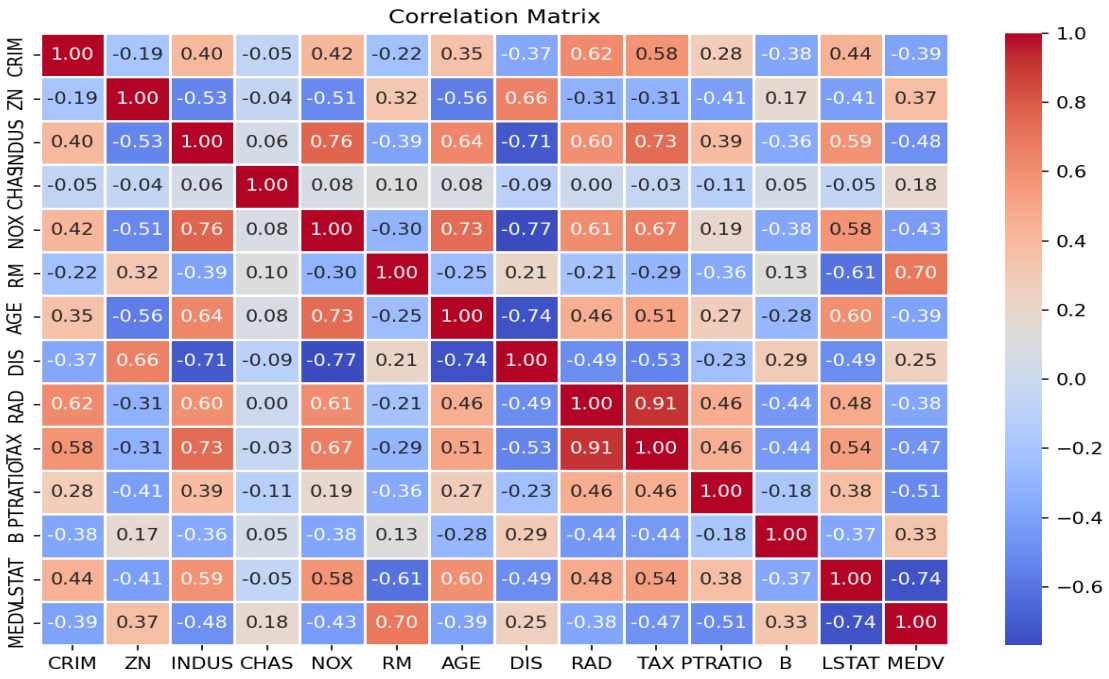
- Creating a histogram of the 'MEDV' (median value of owner-occupied homes) feature. This will help us to visualize the distribution of the data and identify any outliers.



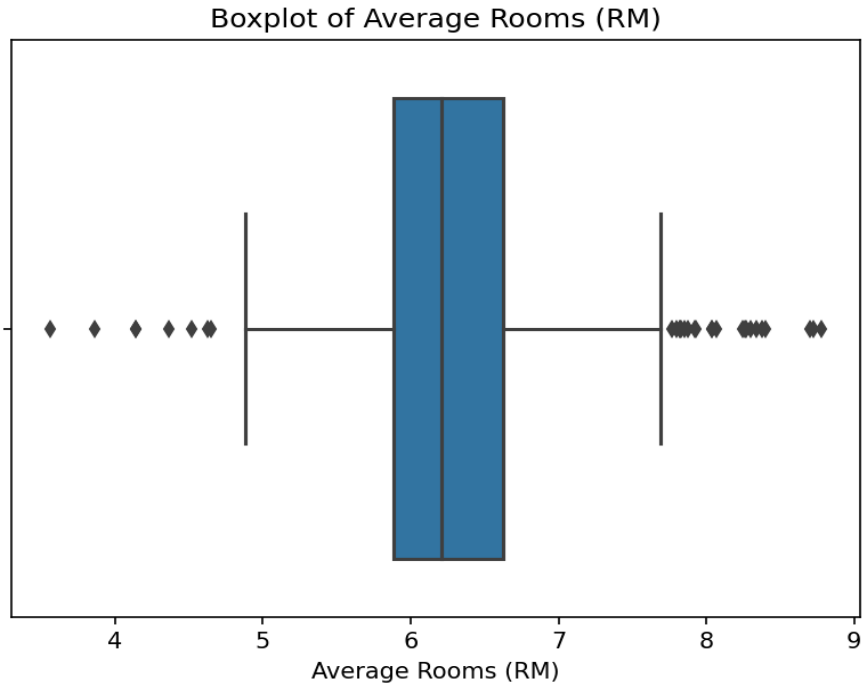
- Creating a scatter plot of 'MEDV' vs. 'CRIM' (per capita crime rate) to visualize the relationship between the two variables, to determine if there is a correlation between crime rates and property values.



❖ Creating a heatmap of the correlation matrix to visualize the relationships between all of the numerical features in the dataset. This will help us to identify any potential collinearity issues.



❖ Creating a box plot of 'RM' (average number of rooms per dwelling) feature to see how the distribution of this variable compares to the distribution of other numerical features indataset.



4. Interpret the visualizations to gain insights into the dataset.

Histogram of Housing Prices(MEDV) -
Interpretation : The distribution of home prices is revealed by the histogram. A distribution that is skewed to the right indicates that many houses are priced lower, whereas a distribution that is skewed to the left indicates prices that are higher. A smoothed curve is superimposed on the histogram by the kernel density estimate (kde), giving a more continuous depiction of the data distribution.

Correlation Matrix -
Interpretation : The correlation matrix for a number of features, including the target variable ('MEDV,') is shown in the heatmap. Stronger correlations are represented by darker colors. For example, a positive correlation between 'MEDV' and 'RM', or average number of rooms, indicates that homes with more rooms typically cost more. In a similar vein, an inverse relationship between "LSTAT" (the population's percentage with a lower status) and "MEDV" suggests that areas with higher proportions of people with lower statuses may also have cheaper housing.

Scatter Plot of Average Rooms (RM) vs. Housing Prices (MEDV) -
Interpretation : The relationship between average room count ('RM') and median home price ('MEDV') is depicted in the scatter plot. The scatter plot shows a positive trend, indicating that houses with more rooms typically have higher prices. This corroborates the correlation matrix heatmap's observation of a positive correlation.

Boxplot of Average Rooms (RM) -
Interpretation : The boxplot for the 'RM' feature shows the distribution of the average number of rooms per dwelling. You can observe the median (middle line in the box), the interquartile range (box), and potential outliers (points outside the whiskers). A wider interquartile range may indicate variations in the size of dwellings.

5. Perform Univariate and multivariate analysis for the dataset.

Univariate Analysis - Univariate Analysis is a type of data visualization where we visualize only a single variable at a time. Univariate Analysis helps us to analyze the distribution of the variable present in the data so that we can perform further analysis.

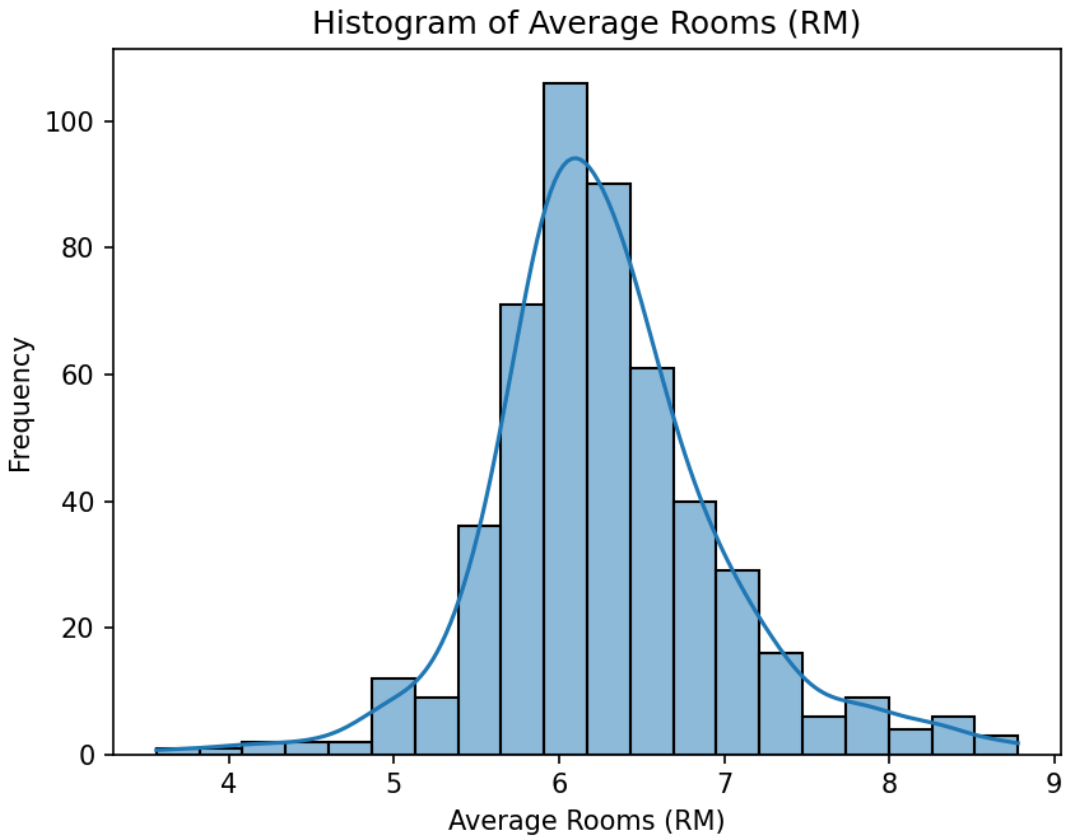
Multivariate Analysis - It is an extension of bivariate analysis which means it involves multiple variables at the same time to find correlation between them. Multivariate Analysis is a set of statistical model that examine patterns in multidimensional data by considering at once, several data variable.

Code :

```
# Univariate analysis using histograms or KDE plots
sns.histplot(f1['RM'], bins=20 )#, kde=True)
plt.title('Histogram of Average Rooms (RM)')
plt.xlabel('Average Rooms (RM)')
plt.ylabel('Frequency')
plt.show()

# Multivariate analysis using scatter plots
sns.pairplot(f1[['RM', 'LSTAT', 'PTRATIO', 'MEDV']])
plt.suptitle('Pairplot of selected features and Housing Prices (MEDV)', y=1.02)
plt.show()
```

Output :

Univariate analysis of the dataset -



Multivariate Analysis of the dataset -