MACHINE LEARNING

LAB WORK 2

Name: Arjun Unnikrishnan USN: 22BTRAD004

Boston Housing Dataset

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

Code:

```
import pandas as pd

# Load the CSV file into a pandas DataFrame

boston_df_with_outliers = pd.read_csv('HousingData.csv')

# Display the DataFrame with outliers

print(boston_df_with_outliers.head())
```

Output:

Name: Arjun Unnikrishnan

USN: 22BTRAD004

Lab 2

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

In [1]:
```
import pandas as pd
# Load the CSV file into a pandas DataFrame
boston_df_with_outliers = pd.read_csv('HousingData.csv')
# Display the DataFrame with outliers
print(boston_df_with_outliers.head())
```

```
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222     18.7

        B  LSTAT  MEDV
0  396.90   4.98  24.0
1  396.90   9.14  21.6
2  392.83   4.03  34.7
3  394.63   2.94  33.4
4  396.90    NaN  36.2
```

Question 2. Use visualization or statistical methods to detect outliers.

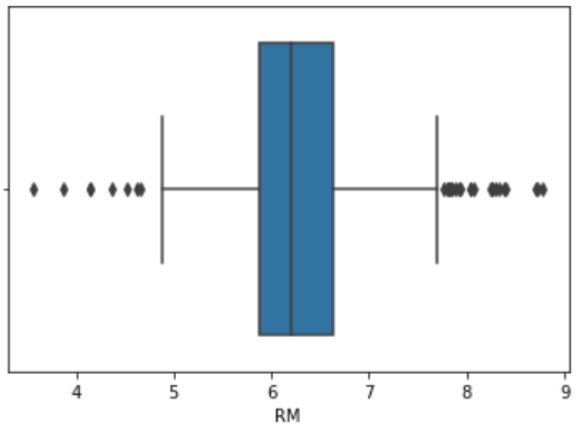
Code:

```
#Box plots

import seaborn as sns

import matplotlib.pyplot as plt

# Example using Seaborn box plot

sns.boxplot(x=boston_df_with_outliers['RM'])
```

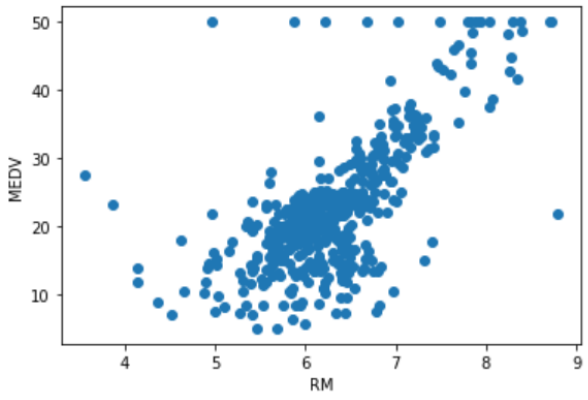
```
#Scatter plots
plt.show()
plt.scatter(x=boston_df_with_outliers['RM'], y=boston_df_with_outliers['MEDV'])
plt.xlabel('RM')
plt.ylabel('MEDV')
plt.show()

#InterQuartileRange
# Calculate quartiles and IQR
Q1 = boston_df_with_outliers['RM'].quantile(0.25)
Q3 = boston_df_with_outliers['RM'].quantile(0.75)
IQR = Q3 - Q1
# Define the lower and upper bounds for outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
# Identify outliers
outliers = boston_df_with_outliers[(boston_df_with_outliers['RM'] < lower_bound) |
(boston_df_with_outliers['RM'] > upper_bound)]
# Display outliers
print("Outliers based on IQR:")
print(outliers[['RM', 'MEDV']])
# Visualize the data with outliers
plt.figure(figsize=(10, 6))
sns.scatterplot(x='RM', y='MEDV', data=boston_df_with_outliers, color='blue', label='Data')
sns.scatterplot(x='RM', y='MEDV', data=outliers, color='red', label='Outliers')
plt.xlabel('RM')
plt.ylabel('MEDV')
plt.legend()
plt.show()
```

Output:

In [3]:
```
#Box plots
import seaborn as sns
import matplotlib.pyplot as plt
# Example using Seaborn box plot
sns.boxplot(x=boston_df_with_outliers['RM'])
plt.show()
```

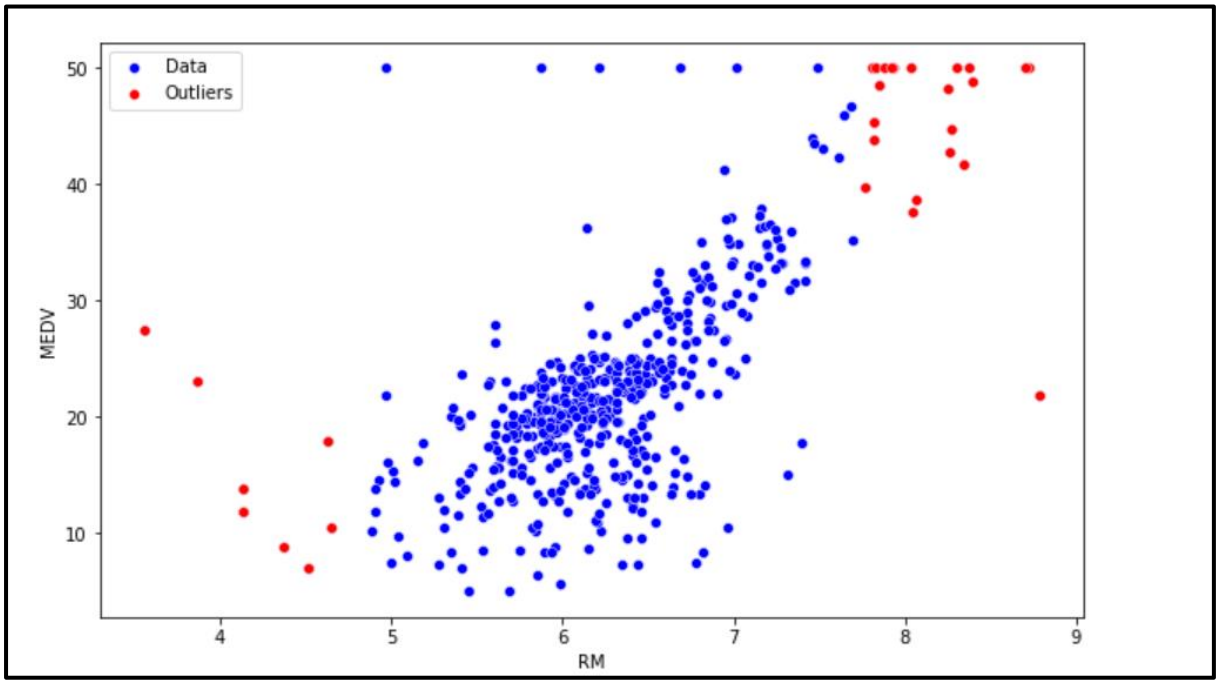


In [4]:
```
#Scatter Plots
plt.scatter(x=boston_df_with_outliers['RM'], y=boston_df_with_outliers['MEDV'])
plt.xlabel('RM')
plt.ylabel('MEDV')
plt.show()
```



Outliers based on IQR:

	RM	MEDV
97	8.069	38.7
98	7.820	43.8
162	7.802	50.0
163	8.375	50.0
166	7.929	50.0
180	7.765	39.8
186	7.831	50.0
195	7.875	50.0
203	7.853	48.5
204	8.034	50.0
224	8.266	44.8
225	8.725	50.0
226	8.040	37.6
232	8.337	41.7
233	8.247	48.3
253	8.259	42.8
257	8.704	50.0
262	8.398	48.8
267	8.297	50.0
280	7.820	45.4
283	7.923	50.0
364	8.780	21.9
365	3.561	27.5
367	3.863	23.1
374	4.138	13.8
384	4.368	8.8
386	4.652	10.5
406	4.138	11.9
412	4.628	17.9
414	4.519	7.0



Question 3. Implement a strategy to handle outliers (e.g., removal and transformation).

Code:

#Outlier Removal

Identify and remove outliers based on IQR

Q1 = boston_df_with_outliers['RM'].quantile(0.25)

Q3 = boston_df_with_outliers['RM'].quantile(0.75)

IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR

upper_bound = Q3 + 1.5 * IQR

Remove outliers

boston_df_no_outliers = boston_df_with_outliers[(boston_df_with_outliers['RM'] >= lower_bound) & (boston_df_with_outliers['RM'] <= upper_bound)]

#Transformation

Log transformation

boston_df_transformed = pd.DataFrame()

boston_df_transformed['RM'] = np.log1p(boston_df_with_outliers['RM'])

boston_df_transformed['MEDV'] = np.log1p(boston_df_with_outliers['MEDV'])

Output:

```
In [15]: #Outlier Removal
         # Identify and remove outliers based on IQR
         Q1 = boston_df_with_outliers['RM'].quantile(0.25)
         Q3 = boston_df_with_outliers['RM'].quantile(0.75)
         IQR = Q3 - Q1
         lower_bound = Q1 - 1.5 * IQR
         upper_bound = Q3 + 1.5 * IQR
         # Remove outliers
         boston_df_no_outliers = boston_df_with_outliers[(boston_df_with_outliers['RM'] >= lower_bound) & (boston_df_with_outliers['RM']

In [16]: #Transformation
         # Log transformation
         boston_df_transformed = pd.DataFrame()
         boston_df_transformed['RM'] = np.log1p(boston_df_with_outliers['RM'])
         boston_df_transformed['MEDV'] = np.log1p(boston_df_with_outliers['MEDV'])
```

GitHub Link: https://github.com/arj1-1n/ML