Module 2 Assignment — Case Study

Utkarsh Kothari

Master of Professional Studies in Analytics, Northeastern University

ALY 6110: Data Management and Big Data

Prof. Mohammad Shafiqul Islam

June 09, 2024

Introduction

This report's goal is to use PySpark, an effective open-source data processing engine that prioritizes speed, usability, and advanced analytics, to analyze the Boston Housing dataset. The Boston Housing dataset, which includes data gathered by the US Census Service regarding housing in the Boston, Massachusetts area, is a well-known dataset in the fields of machine learning and data science. Its variety of continuous and categorical variables, combined with its relatively small size, make it an excellent foundation for building machine learning models.

The dataset contains a number of explanatory variables, including the average number of rooms per dwelling, the rate of property taxes, and the crime rate. The median value of owner-occupied homes is the target variable. This study examines the relationship between owner-occupied home median value (MEDV) and average number of rooms per dwelling (RM). Bins are created from the "RM" attribute, and statistics are computed for each bin.

The format of the report is as follows: The dataset is loaded and any missing values are checked in order to prepare the data. The coding procedure is then explained, along with how bins for the "RM" attribute are made and statistics are computed for each bin. The findings are presented in the results and analysis section, supported by a number of visual aids including box plots, bar plots, scatter plots, and a heatmap of the correlation matrix. A review of the conclusions drawn from the analysis comes at the end of the report.

Objectives

This report's main goal is to use PySpark to conduct a thorough analysis of the Boston Housing dataset. This involves understanding the features and structure of the data, preprocessing it as needed, and drawing conclusions that are relevant to the data.

The relationship between the median value of owner-occupied homes ("MEDV") and the average number of rooms per dwelling ("RM") is a major area of focus for the analysis. One of the goals is to group the housing records into bins according to the "RM" attribute in order to make this easier. A more detailed examination of the relationship between the number of rooms and the median value of homes is made possible by this binning process.

Finally, the report aims to present the findings in a clear and understandable manner. This involves calculating various statistics for each “RM” bin, such as the average, minimum, and maximum “MEDV”. It also involves creating several visualizations, including bar plots, scatter plots, box plots, and a heatmap of the correlation matrix. These visualizations serve to highlight the key trends and patterns in the data, thereby aiding in the interpretation of the results.

A brief description of each column in the Boston Housing dataset:

CRIM: This is the per capita crime rate by town.

ZN: This is the proportion of residential land zoned for lots over 25,000 sq.ft.

INDUS: This is the proportion of non-retail business acres per town.

CHAS: This is a Charles River dummy variable (equals 1 if tract bounds river; 0 otherwise).

NOX: This is the nitric oxides concentration (parts per 10 million).

RM: This is the average number of rooms per dwelling.

AGE: This is the proportion of owner-occupied units built prior to 1940.

DIS: This is the weighted distances to five Boston employment centers.

RAD: This is the index of accessibility to radial highways.

TAX: This is the full-value property-tax rate per $10,000.

PTRATIO: This is the pupil-teacher ratio by town.

B: This is calculated as 1000(Bk - 0.63)^2, where Bk is the proportion of people of African American descent by town.

LSTAT: This is the percentage lower status of the population.

MEDV: This is the median value of owner-occupied homes in $1000s.

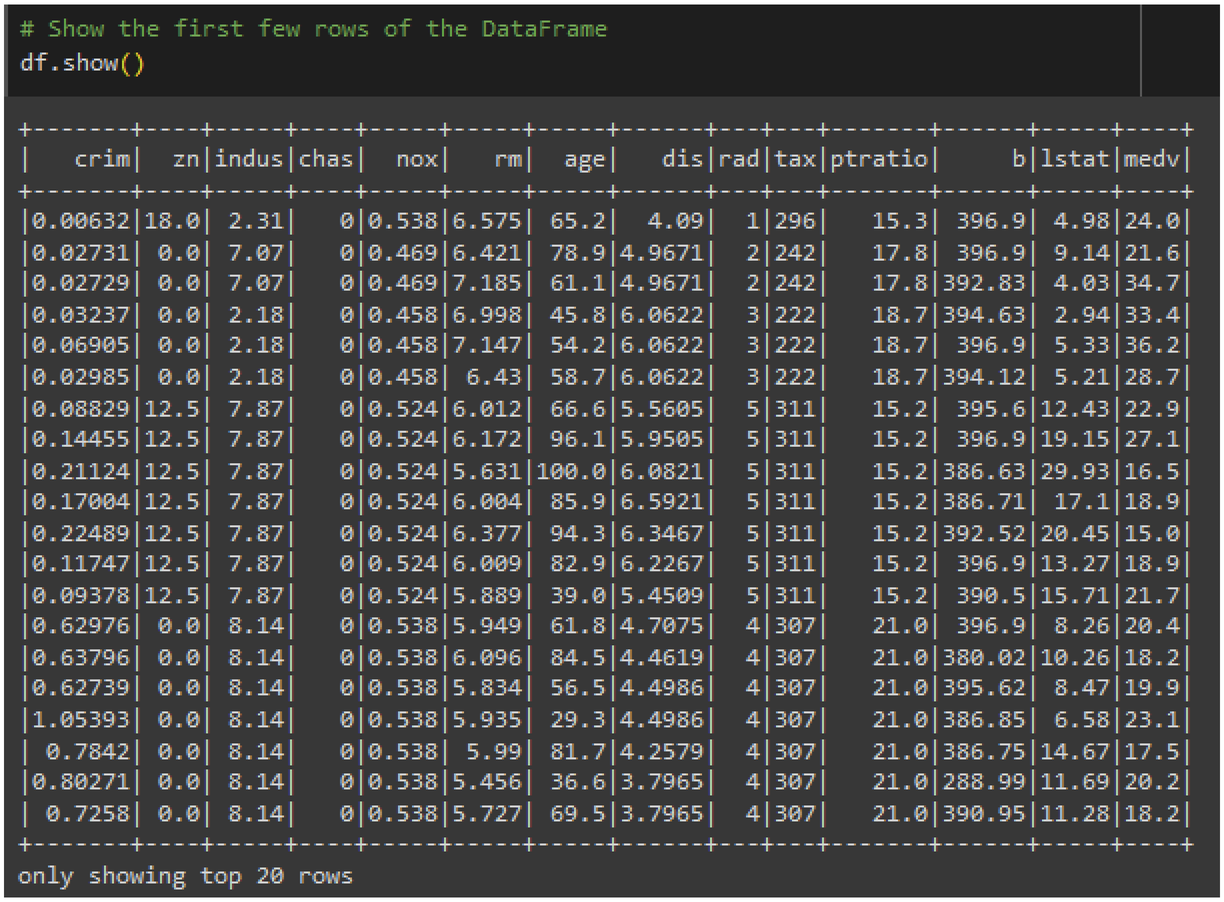
(Harrison & Rubinfeld, 1996)

Analysis & Interpretation

The analysis began with the loading of the Boston Housing dataset into a PySpark DataFrame and then continues as described below.

Figure 1

Viewing a few rows of the dataset



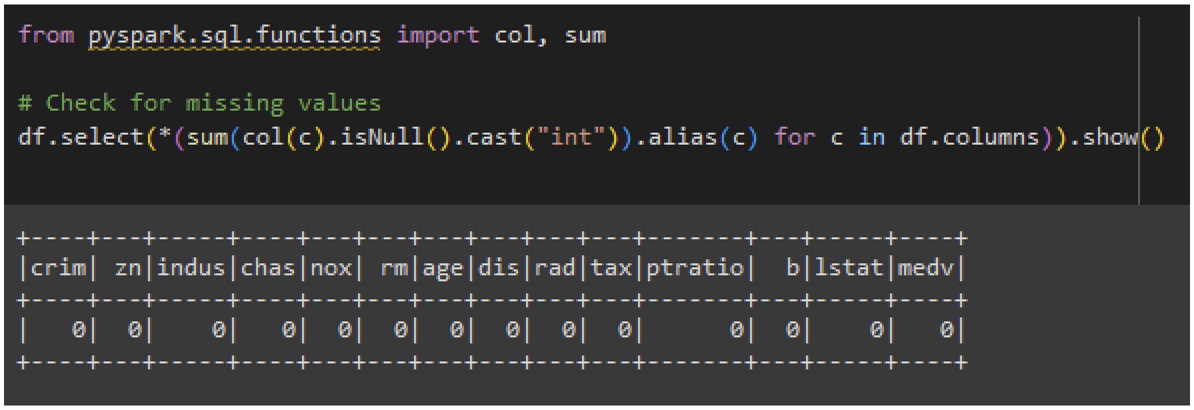
```
# Show the first few rows of the DataFrame
df.show()
```

```
+-------+----+-----+----+-----+-----+-----+------+---+---+-------+------+-----+----+
|   crim|  zn|indus|chas|  nox|   rm|  age|   dis|rad|tax|ptratio|     b|lstat|medv|
+-------+----+-----+----+-----+-----+-----+------+---+---+-------+------+-----+----+
|0.00632|18.0| 2.31|   0|0.538|6.575| 65.2|  4.09|  1|296|   15.3| 396.9| 4.98|24.0|
|0.02731| 0.0| 7.07|   0|0.469|6.421| 78.9|4.9671|  2|242|   17.8| 396.9| 9.14|21.6|
|0.02729| 0.0| 7.07|   0|0.469|7.185| 61.1|4.9671|  2|242|   17.8|392.83| 4.03|34.7|
|0.03237| 0.0| 2.18|   0|0.458|6.998| 45.8|6.0622|  3|222|   18.7|394.63| 2.94|33.4|
|0.06905| 0.0| 2.18|   0|0.458|7.147| 54.2|6.0622|  3|222|   18.7| 396.9| 5.33|36.2|
|0.02985| 0.0| 2.18|   0|0.458| 6.43| 58.7|6.0622|  3|222|   18.7|394.12| 5.21|28.7|
|0.08829|12.5| 7.87|   0|0.524|6.012| 66.6|5.5605|  5|311|   15.2| 395.6|12.43|22.9|
|0.14455|12.5| 7.87|   0|0.524|6.172| 96.1|5.9505|  5|311|   15.2| 396.9|19.15|27.1|
|0.21124|12.5| 7.87|   0|0.524|5.631|100.0|6.0821|  5|311|   15.2|386.63|29.93|16.5|
|0.17004|12.5| 7.87|   0|0.524|6.004| 85.9|6.5921|  5|311|   15.2|386.71| 17.1|18.9|
|0.22489|12.5| 7.87|   0|0.524|6.377| 94.3|6.3467|  5|311|   15.2|392.52|20.45|15.0|
|0.11747|12.5| 7.87|   0|0.524|6.009| 82.9|6.2267|  5|311|   15.2| 396.9|13.27|18.9|
|0.09378|12.5| 7.87|   0|0.524|5.889| 39.0|5.4509|  5|311|   15.2| 390.5|15.71|21.7|
|0.62976| 0.0| 8.14|   0|0.538|5.949| 61.8|4.7075|  4|307|   21.0| 396.9| 8.26|20.4|
|0.63796| 0.0| 8.14|   0|0.538|6.096| 84.5|4.4619|  4|307|   21.0|380.02|10.26|18.2|
|0.62739| 0.0| 8.14|   0|0.538|5.834| 56.5|4.4986|  4|307|   21.0|395.62| 8.47|19.9|
|1.05393| 0.0| 8.14|   0|0.538|5.935| 29.3|4.4986|  4|307|   21.0|386.85| 6.58|23.1|
| 0.7842| 0.0| 8.14|   0|0.538| 5.99| 81.7|4.2579|  4|307|   21.0|386.75|14.67|17.5|
|0.80271| 0.0| 8.14|   0|0.538|5.456| 36.6|3.7965|  4|307|   21.0|288.99|11.69|20.2|
| 0.7258| 0.0| 8.14|   0|0.538|5.727| 69.5|3.7965|  4|307|   21.0|390.95|11.28|18.2|
+-------+----+-----+----+-----+-----+-----+------+---+---+-------+------+-----+----+
only showing top 20 rows
```

Note. We can understand the variables and values with these first 20 lines of the code.

Figure 2

Checking for missing values



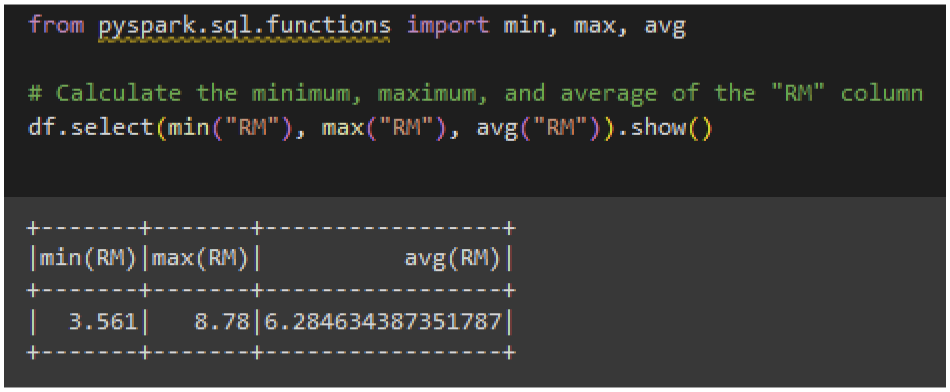
```
from pyspark.sql.functions import col, sum

# Check for missing values
df.select(*(sum(col(c).isNull().cast("int")).alias(c) for c in df.columns)).show()
```

```
+----+---+-----+----+---+---+---+---+---+---+-------+---+-----+----+
|crim| zn|indus|chas|nox| rm|age|dis|rad|tax|ptratio|  b|lstat|medv|
+----+---+-----+----+---+---+---+---+---+---+-------+---+-----+----+
|   0|  0|    0|   0|  0|  0|  0|  0|  0|  0|      0|  0|    0|   0|
+----+---+-----+----+---+---+---+---+---+---+-------+---+-----+----+
```

Note. The dataset was checked for missing values, and it was found that there were no missing values in the dataset, which is a good starting point for any data analysis task.

Figure 3

Calculating stats of RM column for further operations



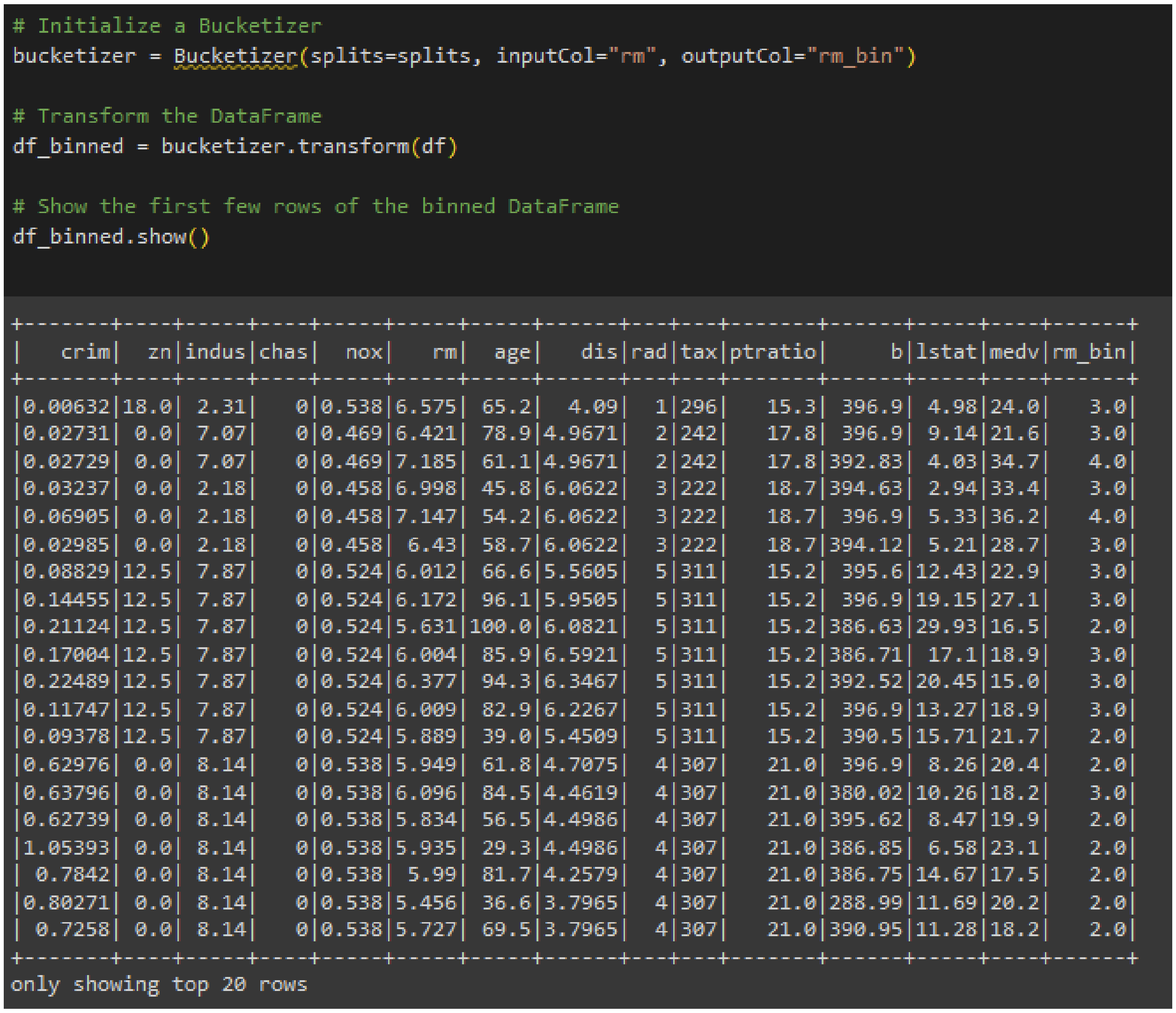
```
from pyspark.sql.functions import min, max, avg

# Calculate the minimum, maximum, and average of the "RM" column
df.select(min("RM"), max("RM"), avg("RM")).show()
```

```
+-------+-------+-----------------+
|min(RM)|max(RM)|          avg(RM)|
+-------+-------+-----------------+
|  3.561|   8.78|6.284634387351787|
+-------+-------+-----------------+
```

Note. We printed the min, max and avg values to understand the distribution of the "RM" column to decide on the bin ranges. It shows that the minimum, maximum, and average number of rooms per dwelling are approximately 3.56, 8.78, and 6.28 respectively

Figure 4

Binning of the "RM" attribute



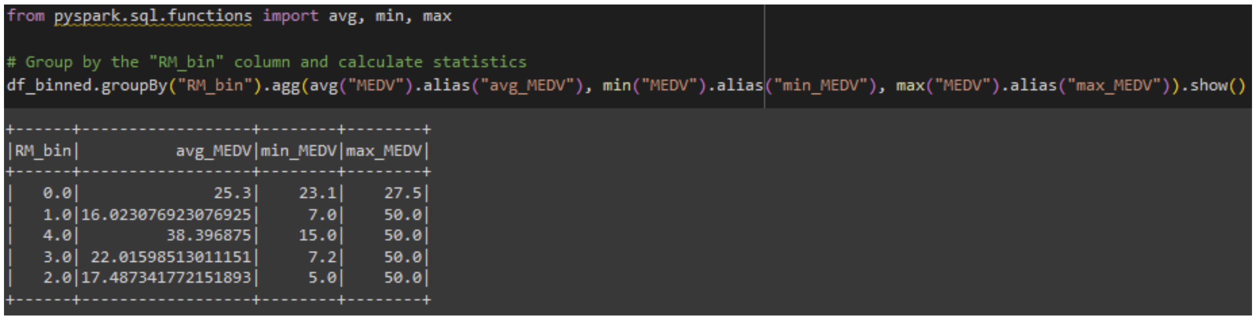
```
# Initialize a Bucketizer
bucketizer = Bucketizer(splits=splits, inputCol="rm", outputCol="rm_bin")

# Transform the DataFrame
df_binned = bucketizer.transform(df)

# Show the first few rows of the binned DataFrame
df_binned.show()
```

```
+-------+----+-----+----+-----+-----+-----+------+---+---+-------+------+-----+----+------+
|   crim|  zn|indus|chas|  nox|   rm|  age|   dis|rad|tax|ptratio|     b|lstat|medv|rm_bin|
+-------+----+-----+----+-----+-----+-----+------+---+---+-------+------+-----+----+------+
|0.00632|18.0| 2.31|   0|0.538|6.575| 65.2|  4.09|  1|296|   15.3| 396.9| 4.98|24.0|   3.0|
|0.02731| 0.0| 7.07|   0|0.469|6.421| 78.9|4.9671|  2|242|   17.8| 396.9| 9.14|21.6|   3.0|
|0.02729| 0.0| 7.07|   0|0.469|7.185| 61.1|4.9671|  2|242|   17.8|392.83| 4.03|34.7|   4.0|
|0.03237| 0.0| 2.18|   0|0.458|6.998| 45.8|6.0622|  3|222|   18.7|394.63| 2.94|33.4|   3.0|
|0.06905| 0.0| 2.18|   0|0.458|7.147| 54.2|6.0622|  3|222|   18.7| 396.9| 5.33|36.2|   4.0|
|0.02985| 0.0| 2.18|   0|0.458| 6.43| 58.7|6.0622|  3|222|   18.7|394.12| 5.21|28.7|   3.0|
|0.08829|12.5| 7.87|   0|0.524|6.012| 66.6|5.5605|  5|311|   15.2| 395.6|12.43|22.9|   3.0|
|0.14455|12.5| 7.87|   0|0.524|6.172| 96.1|5.9505|  5|311|   15.2| 396.9|19.15|27.1|   3.0|
|0.21124|12.5| 7.87|   0|0.524|5.631|100.0|6.0821|  5|311|   15.2|386.63|29.93|16.5|   2.0|
|0.17004|12.5| 7.87|   0|0.524|6.004| 85.9|6.5921|  5|311|   15.2|386.71| 17.1|18.9|   3.0|
|0.22489|12.5| 7.87|   0|0.524|6.377| 94.3|6.3467|  5|311|   15.2|392.52|20.45|15.0|   3.0|
|0.11747|12.5| 7.87|   0|0.524|6.009| 82.9|6.2267|  5|311|   15.2| 396.9|13.27|18.9|   3.0|
|0.09378|12.5| 7.87|   0|0.524|5.889| 39.0|5.4509|  5|311|   15.2| 390.5|15.71|21.7|   2.0|
|0.62976| 0.0| 8.14|   0|0.538|5.949| 61.8|4.7075|  4|307|   21.0| 396.9| 8.26|20.4|   2.0|
|0.63796| 0.0| 8.14|   0|0.538|6.096| 84.5|4.4619|  4|307|   21.0|380.02|10.26|18.2|   3.0|
|0.62739| 0.0| 8.14|   0|0.538|5.834| 56.5|4.4986|  4|307|   21.0|395.62| 8.47|19.9|   2.0|
|1.05393| 0.0| 8.14|   0|0.538|5.935| 29.3|4.4986|  4|307|   21.0|386.85| 6.58|23.1|   2.0|
| 0.7842| 0.0| 8.14|   0|0.538| 5.99| 81.7|4.2579|  4|307|   21.0|386.75|14.67|17.5|   2.0|
|0.80271| 0.0| 8.14|   0|0.538|5.456| 36.6|3.7965|  4|307|   21.0|288.99|11.69|20.2|   2.0|
| 0.7258| 0.0| 8.14|   0|0.538|5.727| 69.5|3.7965|  4|307|   21.0|390.95|11.28|18.2|   2.0|
+-------+----+-----+----+-----+-----+-----+------+---+---+-------+------+-----+----+------+
only showing top 20 rows
```

Note. The binning process categorizes the "RM" attribute into five bins. Each bin represents a range of "RM" values. The bin index in the "rm_bin" column corresponds to these ranges.

Figure 5

Statistics of the "MEDV" attribute for each "RM" bin



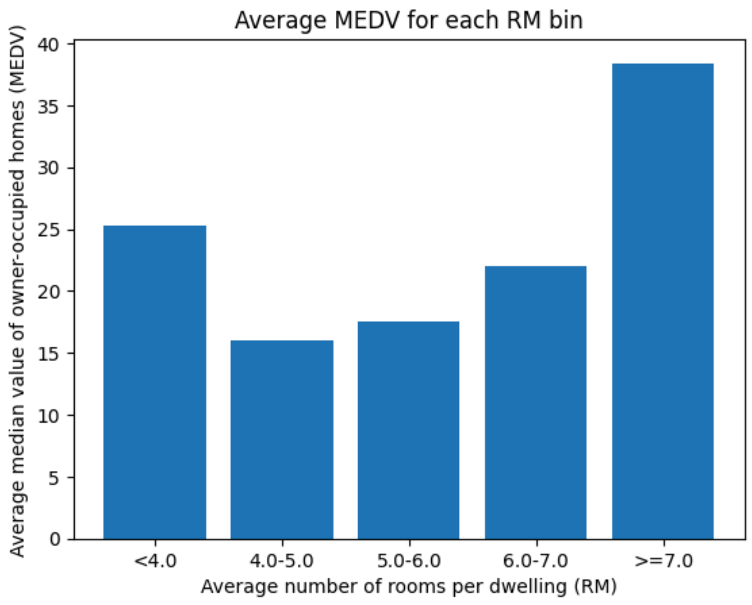
```
from pyspark.sql.functions import avg, min, max

# Group by the "RM_bin" column and calculate statistics
df_binned.groupBy("RM_bin").agg(avg("MEDV").alias("avg_MEDV"), min("MEDV").alias("min_MEDV"), max("MEDV").alias("max_MEDV")).show()
```

```
+------+------------------+--------+--------+
|RM_bin|          avg_MEDV|min_MEDV|max_MEDV|
+------+------------------+--------+--------+
|   0.0|              25.3|    23.1|    27.5|
|   1.0|16.023076923076925|     7.0|    50.0|
|   4.0|         38.396875|    15.0|    50.0|
|   3.0| 22.01598513011151|     7.2|    50.0|
|   2.0|17.487341772151893|     5.0|    50.0|
+------+------------------+--------+--------+
```

Note. The table shows the average, minimum, and maximum "MEDV" for each "RM" bin. These statistics provide insights into the distribution of "MEDV" values within each bin.

Figure 6

Bar plot of the average MEDV for each RM bin



Note. Bin 0 (RM < 4.0): The average median value of owner-occupied homes (MEDV) is approximately $25,300. This suggests that homes with fewer than 4 rooms are relatively expensive, which could be due to factors not included in this dataset.

Bin 1 (4.0 <= RM < 5.0): The average MEDV is approximately $16,023. This suggests that homes with 4 to 5 rooms are less expensive than those with fewer rooms.

Bin 2 (5.0 <= RM < 6.0): The average MEDV is approximately $17,487. This suggests that homes with 5 to 6 rooms are slightly more expensive than those with 4 to 5 rooms.

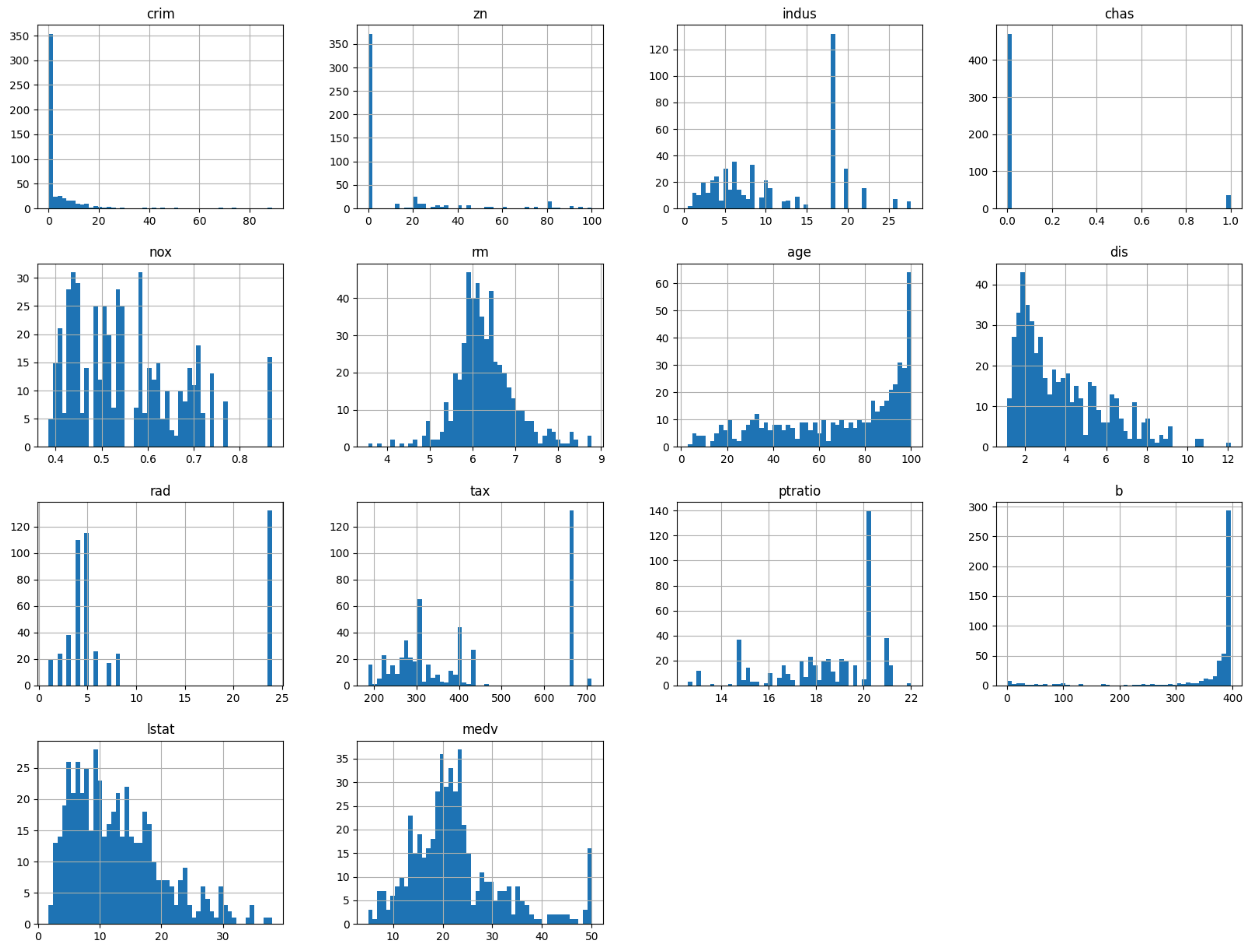
Bin 3 (6.0 <= RM < 7.0): The average MEDV is approximately $22,016. This suggests that homes with 6 to 7 rooms are more expensive than those with fewer rooms.

Bin 4 (RM >= 7.0): The average MEDV is approximately $38,397. This suggests that homes with 7 or more rooms are the most expensive among the categories.

From these results, we can observe that there seems to be a positive correlation between the number of rooms and the median value of the homes. As the number of rooms increases, the median value of the homes also increases. This is consistent with the common understanding that larger homes (with more rooms) tend to be more expensive.

Figure 7

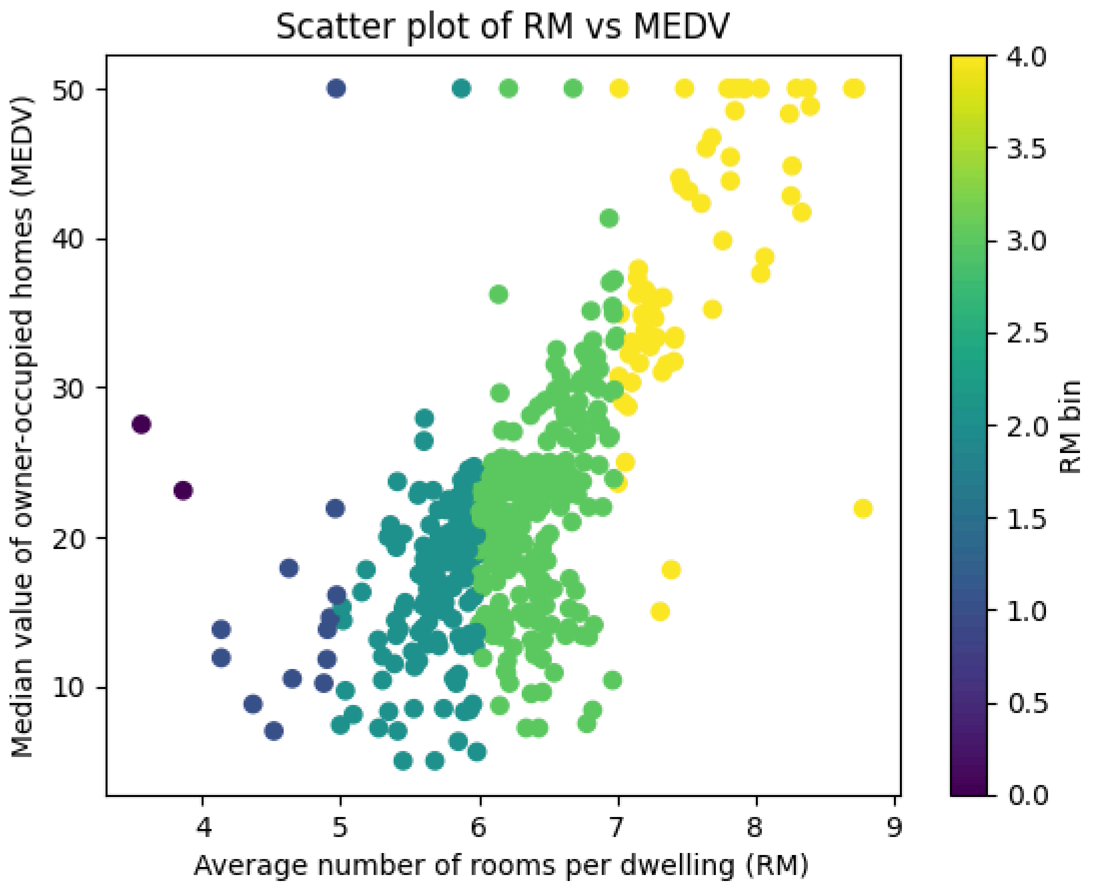
Histograms of each attribute



Note: The histograms provide an overview of the distribution of values in each attribute. They can help identify patterns and outliers in the data.

Figure 8

Scatter plot of "RM" vs "MEDV"

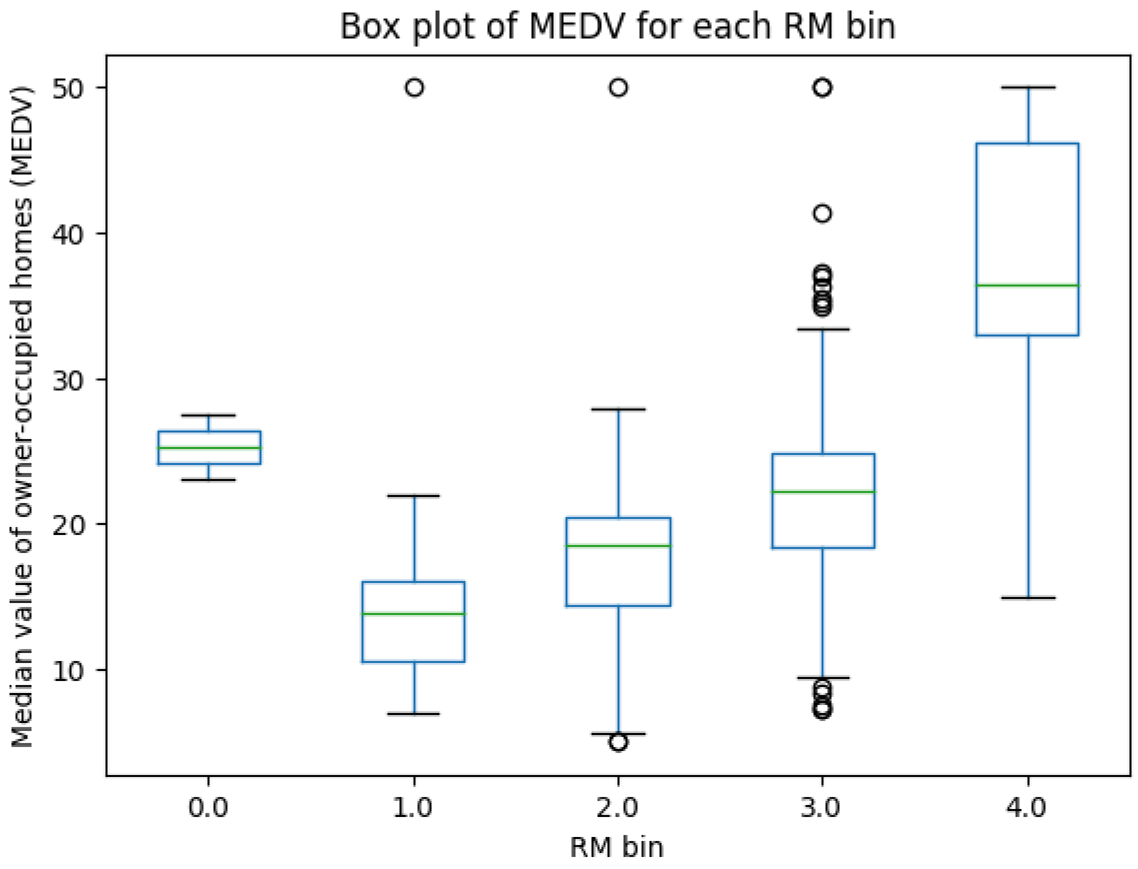


Note: The scatter plot visualizes the relationship between the "RM" and "MEDV" attributes. Each point represents a record in the dataset. The color of the point indicates the "RM" bin.

We can see a positive correlation between the number of rooms and the median value of the homes.

Figure 9

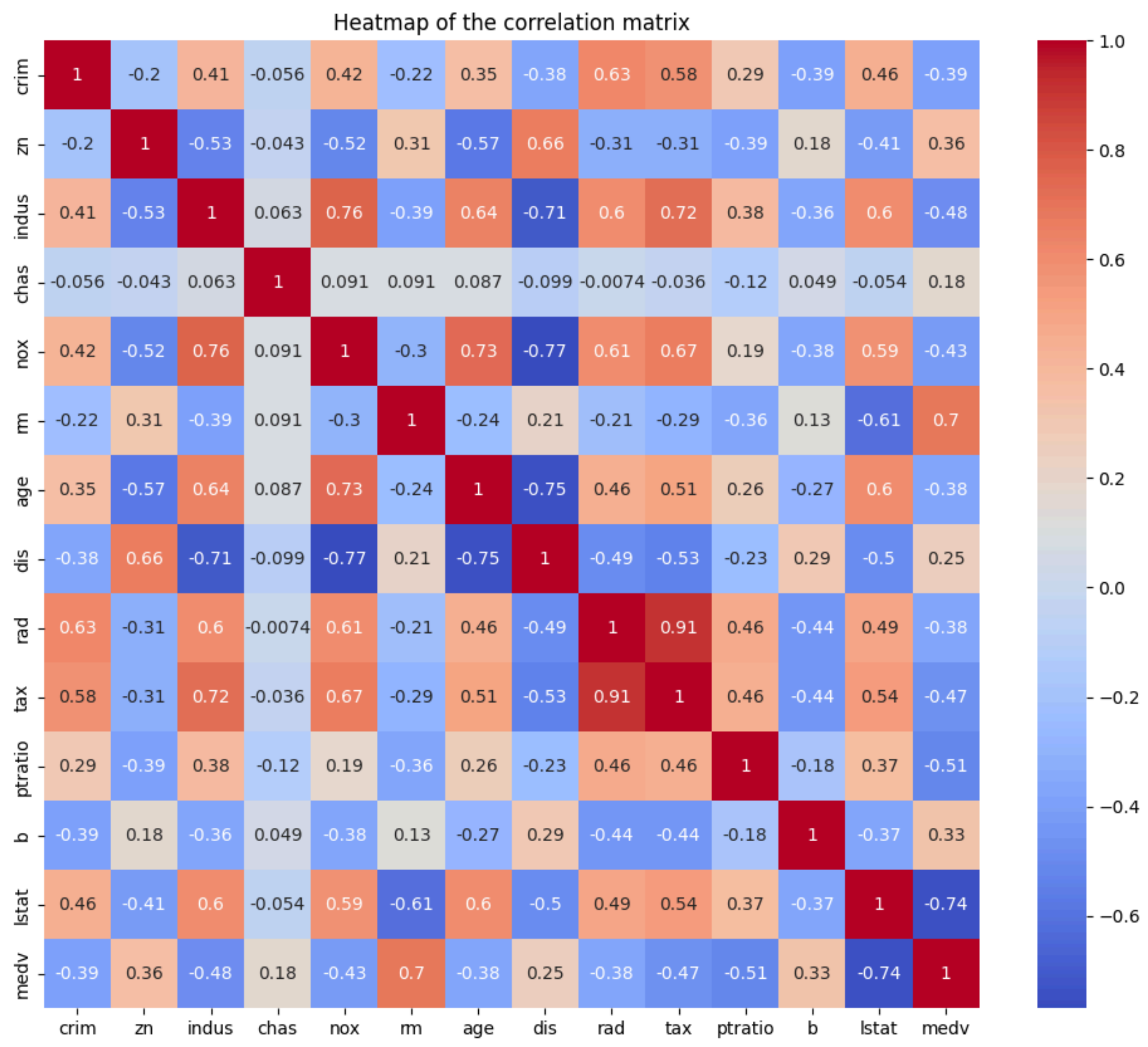
Box plots of "MEDV" for each "RM" bin



Note: The box plots show the distribution of "MEDV" values within each "RM" bin. They provide insights into the spread and outliers of the "MEDV" values.

Figure 10

Heatmap of the correlation matrix



Note: The heatmap visualizes the correlation between different pairs of attributes. A high positive or negative correlation indicates a strong relationship between two attributes.

Conclusion

Understanding the Boston housing market has been greatly enhanced by the PySpark-assisted analysis of the Boston Housing dataset. A thorough analysis was conducted on the dataset, which includes several attributes like the median value of owner-occupied homes, the average number of rooms per dwelling, and the crime rate. The primary discovery of the search is the positive connection that exists between the median property value and the average number of rooms in each residence. The median value of the houses rises in proportion to the number of rooms installed. The general perception that larger (more spacious) homes are typically more expensive is supported by this.

This analysis's use of PySpark demonstrated the tool's ability to manage sizable datasets and carry out challenging data processing operations. The DataFrame API and built-in functions of PySpark were utilized to efficiently perform the binning of the "RM" attribute and the calculation of statistics for every bin. Furthermore, PySpark's compatibility with well-known Python libraries like matplotlib and seaborn was demonstrated through the creation of a variety of visualizations, including bar plots, scatter plots, box plots, and a heatmap of the correlation matrix.

References

Apache Spark. (2024, February 24). *PySpark Overview — PySpark master documentation*. Apache Spark. Retrieved June 7, 2024, from https://spark.apache.org/docs/latest/api/python/index.html

Harrison, D., & Rubinfeld, D. L. (1996, October 10). *Boston Housing Dataset*. Boston Dataset. Retrieved June 7, 2024, from https://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html

GodlikeAnalyst. (2024). *Boston-Housing-PySpark-Analysis* [Software]. GitHub.
https://github.com/GodlikeAnalyst/Boston-Housing-PySpark-Analysis