

Module 2 Assignment — Case Study

Jatin Satija

Master of Professional Studies in Analytics, Northeastern University

ALY 6110: Data Management and Big Data

Prof. Mohammad Shafiqul Islam

Jun 09, 2024

Introduction

The objective of this report is to examine the Boston Housing dataset using PySpark, an efficient open-source data processing engine that puts an emphasis on speed, usability, and sophisticated analytics. We have worked on the Boston Housing dataset—which comprises information obtained by the US Census Service about housing in the Boston, Massachusetts area. Because of its small size and range of continuous and categorical data, it's a great starting point for machine learning models.

Several explanatory variables are included in the dataset, such as the average number of rooms per dwelling, the rate of property taxes, and the rate of crime. The target variable is owner-occupied home median value.

This report examines at the relationship between the average number of rooms per dwelling (RM) and the median owner-occupied home value (MEDV). The "RM" attribute is used to form bins, and statistics are calculated for each bin.

Objective

This report aims to conduct a comprehensive analysis of the Boston Housing dataset using PySpark. The primary focus is to examine the relationship between the median value of owner-occupied homes ("MEDV") and the average number of rooms per dwelling ("RM"). The analysis involves binning the housing records based on the "RM" attribute, enabling a detailed exploration of this relationship. Various statistics for each "RM" bin, such as average, minimum, and maximum "MEDV," are calculated. The findings are presented clearly through several visualizations, including bar plots, scatter plots and a heatmap of the correlation matrix. These visualizations help highlight key trends and patterns in the data, facilitating the interpretation of the results.

Features of 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 $1000(B_k - 0.63)^2$, where B_k 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)

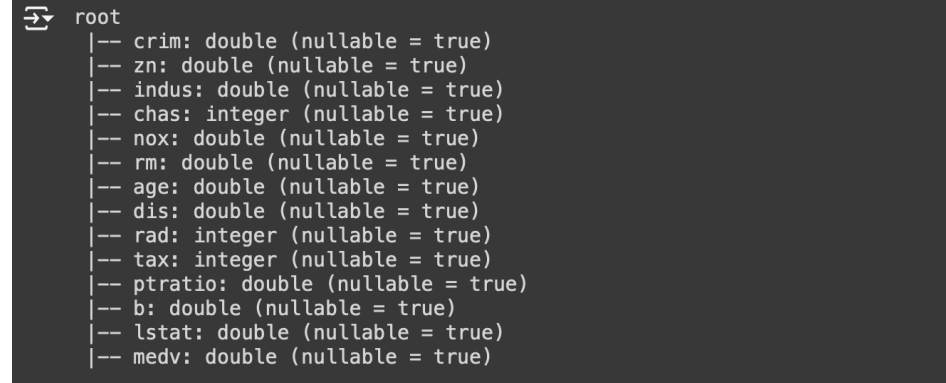
Analysis

Figure 1

Loading and Inspecting the Boston Housing Dataset with PySpark

```
[ ] df = spark.read.csv("BostonHousing.csv", header=True, inferSchema=True)

[ ] df.printSchema() ⚠
```



```
root
|-- crim: double (nullable = true)
|-- zn: double (nullable = true)
|-- indus: double (nullable = true)
|-- chas: integer (nullable = true)
|-- nox: double (nullable = true)
|-- rm: double (nullable = true)
|-- age: double (nullable = true)
|-- dis: double (nullable = true)
|-- rad: integer (nullable = true)
|-- tax: integer (nullable = true)
|-- ptratio: double (nullable = true)
|-- b: double (nullable = true)
|-- lstat: double (nullable = true)
|-- medv: double (nullable = true)
```

Note. The code uses PySpark's `read.csv` method to read the Boston Housing dataset from a CSV file. The Data Frame's schema, which includes each column's structure and data type, is then displayed using the `printSchema` method.

Figure 2

Displaying the First 5 Rows of the Boston Housing Dataset

```
[ ] df.show(5) 🔦
```

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

only showing top 5 rows

Note. The first five rows of the dataset are shown to provide a basic overview which helps in understanding its structure.

Figure 3

Checking for Null Values in the Boston Housing Dataset

```
[ ] from pyspark.sql.functions import col, sum, when
null_counts = df.select([col(column).isNull().alias(column) for column in df.columns]).\
.....select([count(when(col(column), 1)).alias(column) for column in df.columns])
null_counts.show()
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv	RM_Bin
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note. Performing a null value check to make sure the dataset was complete and to ensure the dataset's integrity and identify any missing data.

Figure 4*Rounding Numerical Columns to Two Decimal Places*

```
[ ] numerical_columns = [col_name for col_name, dtype in df.dtypes if dtype in ('double', 'int')]
for col_name in numerical_columns:
    df = df.withColumn(col_name, round(col(col_name), 2))

df.show(5)
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0.01	18.0	2.31	0	0.54	6.58	65.2	4.09	1	296	15.3	396.9	4.98	24.0
0.03	0.0	7.07	0	0.47	6.42	78.9	4.97	2	242	17.8	396.9	9.14	21.6
0.03	0.0	7.07	0	0.47	7.19	61.1	4.97	2	242	17.8	392.83	4.03	34.7
0.03	0.0	2.18	0	0.46	7.0	45.8	6.06	3	222	18.7	394.63	2.94	33.4
0.07	0.0	2.18	0	0.46	7.15	54.2	6.06	3	222	18.7	396.9	5.33	36.2

only showing top 5 rows

Note. All numerical columns in the Boston Housing dataset were rounded to two decimal places in order to improve their readability and consistency.

Figure 5*Exploring the RM column stats*

```
from pyspark.sql.functions import min, max, avg
df.select(min("RM"), max("RM"), avg("RM")).show()
```

min(RM)	max(RM)	avg(RM)
3.56	8.78	6.285217391304348

Note. In order to determine the bin ranges, we printed the min, max, and average values in order to comprehend the distribution of the "RM" column. It reveals that the average, maximum, and minimum number of rooms in a dwelling are roughly 3.56, 8.78, and 6.28, respectively.

Figure 6

Binning and Statistical Analysis of the Boston Housing Dataset

```

from pyspark.sql.functions import mean, stddev, min, max, count

def categorize_rm(rm):
    if rm < 5:
        return "< 5"
    elif 5 <= rm < 6:
        return "5 - 6"
    elif 6 <= rm < 7:
        return "6 - 7"
    elif 7 <= rm < 8:
        return "7 - 8"
    else:
        return ">= 8"

# Register the UDF
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

categorize_rm_udf = udf(categorize_rm, StringType())

# Apply the binning function to create a new column "RM_Bin"
df = df.withColumn("RM_Bin", categorize_rm_udf(col("rm")))

# Group by "RM_Bin" and calculate statistics
stats_df = df.groupBy("RM_Bin").agg(
    count("*").alias("count"),
    mean("medv").alias("mean_medv"),
    stddev("medv").alias("stddev_medv"),
    min("medv").alias("min_medv"),
    max("medv").alias("max_medv")
).orderBy("RM_Bin")

# Display the statistics for each bin
stats_df.show()

```

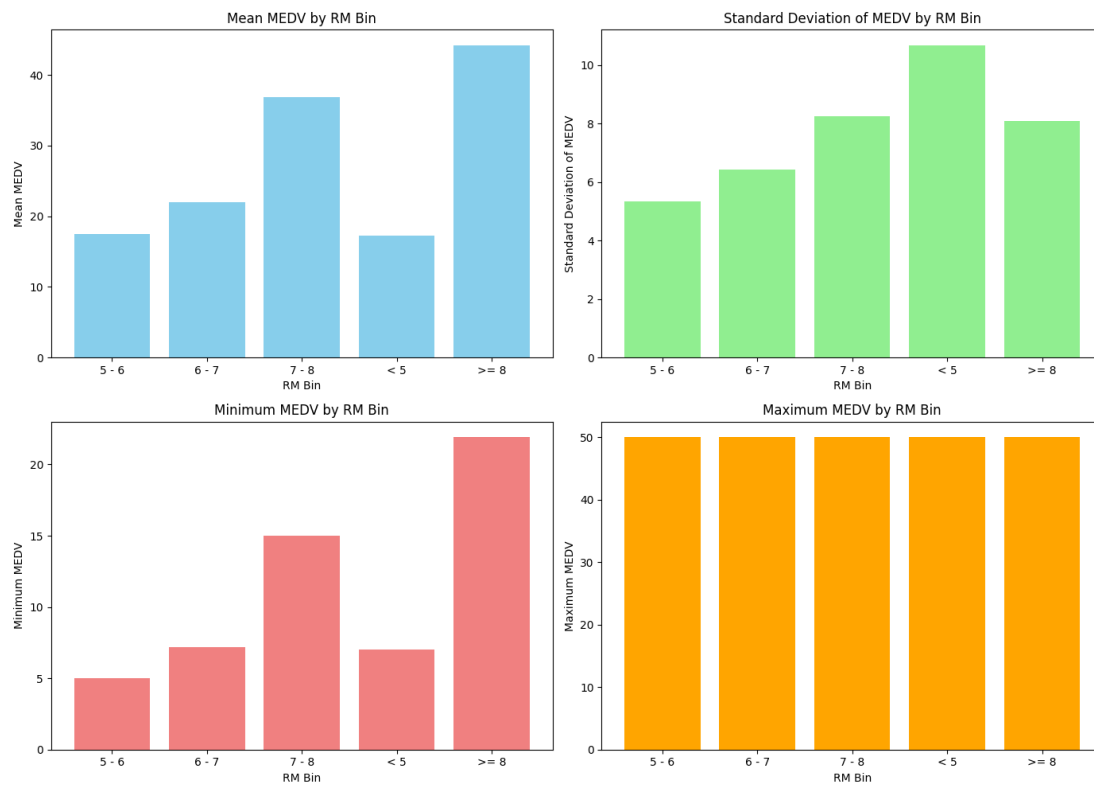
RM_Bin	count	mean_medv	stddev_medv	min_medv	max_medv
5 - 6	157	17.44968152866241	5.348755237457921	5.0	50.0
6 - 7	269	21.978810408921916	6.430677762883202	7.2	50.0
7 - 8	52	36.849999999999994	8.248862072443451	15.0	50.0
< 5	15	17.26	10.677399362338058	7.0	50.0
>= 8	13	44.2	8.092383250110357	21.9	50.0

Note. We created the `categorize_rm` Python function, which divides the "RM" values into pre-defined categories. In order to generate a new column called "RM_Bin" that holds the bin categories, we then applied the UDF to the "RM" column. Finally, we computed the count, mean, standard deviation, minimum, and maximum of the "MEDV" column for each bin after grouping the DataFrame by "RM_Bin".

The mean, standard deviation, minimum and maximum "MEDV" values for each "RM" bin are displayed in the table.

Figure 7

Visualization of MEDV Statistics by RM Bin



Note. We used bar charts to illustrate several statistics in order to better understand the relationship between the median value of owner-occupied dwellings ("MEDV") and the average number of rooms per dwelling ("RM").

Here, we focus on interpreting the mean MEDV for each RM bin:

1. RM Bin: <5

The mean median value of a home with fewer than five rooms is roughly \$17,260. The fact that this value is comparable to the mean MEDV for the bin of 5 to 6 rooms suggests that smaller dwellings are often worth less.

2. RM Bin: $5 - 6$

The mean median value of homes with 5 to 6 rooms is roughly \$17,450. This implies that the market value of homes in this range is comparatively lower than that of homes with more rooms.

3. RM Bin: $6 - 7$

The mean median value of homes with 6 to 7 rooms is roughly \$21,980. As the average number of rooms rises from the previous bin, this suggests that the worth of the home is increasing.

4. RM Bin: $7 - 8$

The mean median value of a home with 7 or 8 rooms is much greater, at roughly \$36,850. This significant rise implies that the homes in this bin are significantly more costly, indicating that larger homes have a higher market worth.

5. RM Bin: ≥ 8

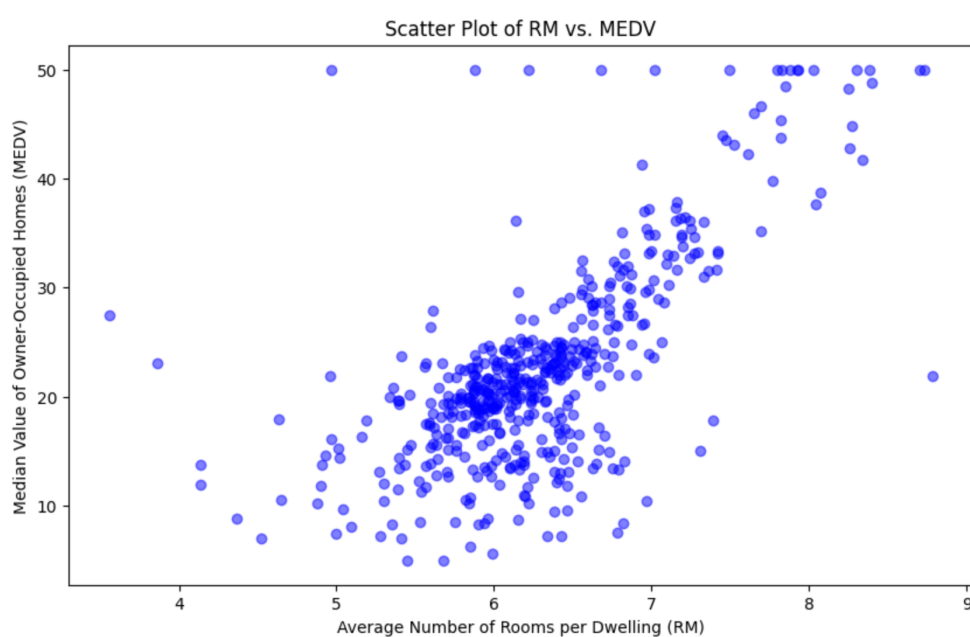
At almost \$44,200, homes with 8 or more rooms have the highest mean median value. According to this, the largest homes have the highest average value.

The results indicate that there are various segments in the housing market: larger homes (7-8 rooms and ≥ 8 rooms) attract premium prices, while smaller homes (< 5 rooms and 5-6 rooms) are cheaper. Mid-sized homes (6-7 rooms) have moderate values.

The bins corresponding to 6-7 and 7-8 rooms, as well as the bin for households with 8 or more rooms, show the largest increases in mean MEDV. This implies that upsizing a mid-sized house to a larger one adds significantly to its worth.

Figure 8

Scatterplot to Analyse the Relationship Between Room Count and Median Home Value

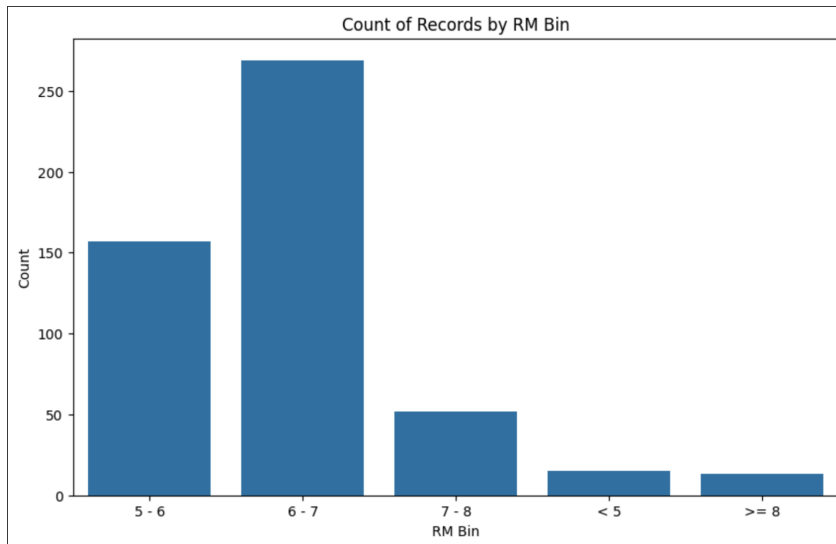


Note. The scatter plot demonstrates a positive correlation between the "RM" (average number of rooms per dwelling) and "MEDV" (median value of owner-occupied homes) attributes. As the

number of rooms increases, the median value of the homes tends to rise, indicating that larger homes are generally more valuable.

Figure 8

Frequency Distribution of Homes by Number of Rooms (RM)



Note. Understanding the distribution of the dataset based on the average number of rooms per dwelling is possible with the help of the count statistics for each RM_Bin. With 157 entries, the "5 - 6" bin has the second most records, indicating that a significant number of the dwellings in the dataset have between 5 and 6 rooms. With 269 records, "6 - 7" is the largest category, suggesting that houses with six to seven rooms are also rather popular. On the other hand, the "7 - 8" bin contains only 52 records, indicating a lower frequency of residences with 7 to 8 rooms. There are only 15 items in the "< 5" bucket, which comprises dwellings with fewer than five rooms, indicating that smaller homes are not as common in this dataset. Homes with eight or more rooms are also the least frequent, as evidenced by the "≥ 8" bin, which has the fewest

records, just thirteen. These counts aid in our comprehension of the dataset's distribution of dwelling sizes, demonstrating the clear majority of houses with five to seven rooms.

Conclusion

In conclusion, we used PySpark to perform a thorough study of the Boston Housing dataset.

After preparing the dataset to manage missing values and round numerical columns, we started by examining its features and structure. The average number of rooms per dwelling (RM) was then used to group housing records into bins, allowing for a thorough analysis of the correlation between RM and the median value of owner-occupied dwellings (MEDV). We determined several statistics for every RM bin using PySpark functions and user-defined functions (UDFs), providing insight into the distribution of house values for varying room counts. The investigation that followed showed some interesting patterns, like a positive relationship between the number of rooms and the median house value.

In addition, we employed bar charts to illustrate our results, presenting the distribution of MEDV within RM categories and offering an insight into the differences in home values according to the number of rooms. The visuals improved our understanding of the dataset and simplified the interpretation of the statistical findings. We have gained important insights into the factors impacting home prices through careful data preparation, analysis, and visualization.

References

(1996, October 10). Boston Dataset. Retrieved June 10, 2024, from
<https://www.cs.toronto.edu/%7Edelve/data/boston/bostonDetail.html>

Boston Housing. (n.d.). GitHub. Retrieved June 10, 2024, from
https://github.com/jatinsatija/Boston_Housing

Quick Start - Spark 3.5.1 Documentation. (n.d.). Apache Spark. Retrieved June 10, 2024, from
<https://spark.apache.org/docs/latest/quick-start.html>