BUAN 6356.004 Business Analytics with R

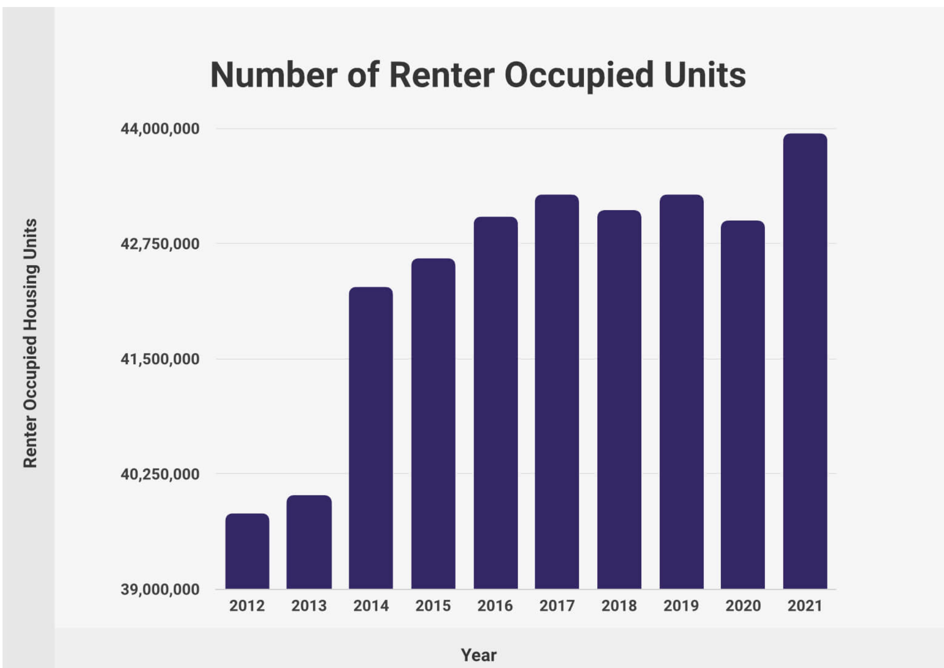
GROUP 6

Business Analytics with R - Project Report

Team 6:
Neeti Mishra (NXM230008)
Sneha Bhowmick (SXB220050)
Tulit Pal (TXP220006)
Srikruthi Shilpika Reddy Gangapalli (SXG230007)

Executive Summary

About 34% of the US population are renters, and this figure is expected to grow over the next years[1]. The graph below demonstrates the number of renters occupying housing units over the years:



It is evident from this chart that there has been a gradual increase in the number of renter-occupied housing units over the past years. This increase in renters and rental properties brings with it the perplexing question of an appropriate rent amount.

This amount can be determined by analyzing past data and utilizing different parameters. Additionally, it is helpful to cluster the database based on characteristics like the number of bedrooms, square footage, tolerance towards pets, etc., to extract more granular information from the dataset. Furthermore, the dataset will be divided for training and testing the algorithm. Thus, with the help of this dataset and employing business intelligence techniques like data preprocessing and clustering, we can predict the rent amount of a rental property with precision.

1 Mariotti, Tony. "Renting Statistics: Trends & Demographics (2022)." RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

BUAN 6356.004 Business Analytics with R

GROUP 6

Data Description

The dataset consists of 22 attributes: 12 numeric, and 10 categorical Title and body contain the details and specifications of the apartment; address, city, state, latitude and longitude represent the location of the apartment. Amenities contain the various facilities provided in the apartment and have pets shows if and what pets are allowed. No. of Bathrooms and Bedrooms contains the room specifics of the apartment price and price display show the cost of the apartment and fee represents any additional fees to be paid square feet represents the total area of the apartment source shows the source of the apartment listing (website).

Data Pre-processing & Cleaning

Data cleaning is an important step in the analytics process, where a dataset is analyzed to minimize formatting errors, unit mismatches, missing data, and miscellaneous inaccuracies. This step is essential in ensuring data accuracy, consistency, and reliability by improving the overall quality of the data and leading to better analysis, more accurate modeling, and informed decision-making.

In our dataset, we approached data pre-processing by isolating the columns that contained missing values. Then, we processed these columns using different strategies depending on the data in the columns. For the *bedrooms* column, we replaced the missing data values with the average value of the column. Our reason for this approach was that the row contained important data (*price*, *cityname*, *state*, *square_feet*, etc.) that we could use even when the *bedrooms* column was not contributing as much. On the other hand, the records with missing *latitude* and *longitude* values were removed from the dataset. This was done since the number of records with this inconsistency was substantially low (10/10,000 ≈ 0.1%) and because missing *latitude* and *longitude* values may impact clustering and association later.

Next, some columns contained “null” as a text for some unavailable values. These values were replaced by 0 for numeric columns and “None” or “Unavailable” for other columns to enforce consistency.

The *bathrooms* column was originally encoded as a character column despite containing only integer values. This column was reformatted as a numeric column.

Additionally, some of the *price* values were provided as weekly values, while others were presented as monthly values. This column was transformed to reflect the monthly *price* for all the records.

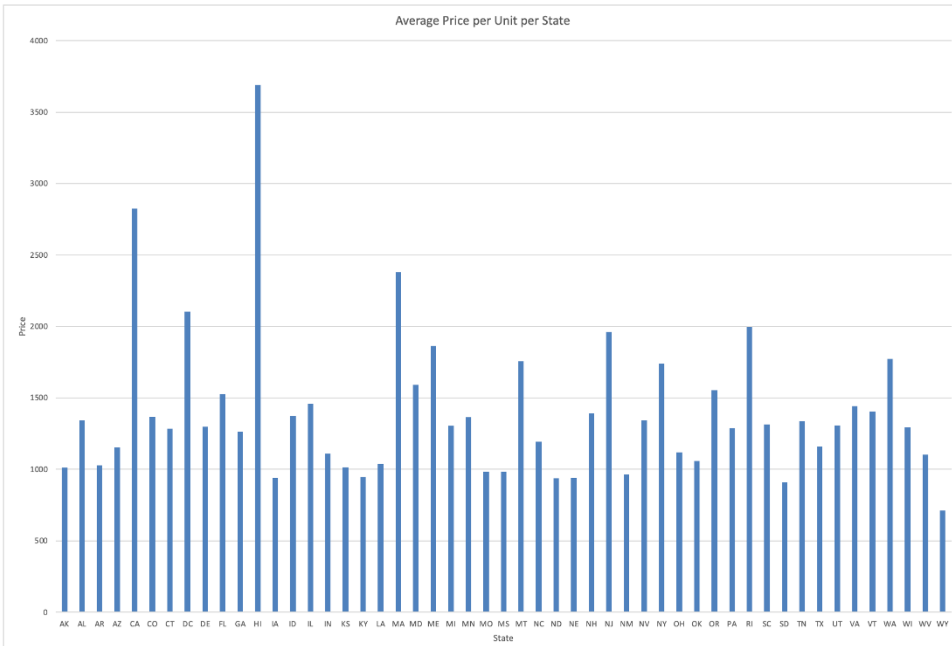
Finally, the outliers were removed using Cooks’D to prevent it from skewing the forecast results.

Initial Analysis

Our initial analysis was instrumental in gaining a deeper understanding of the dataset and extracting concise summaries from various columns. After thorough data cleaning and preprocessing, we utilized visualization techniques to uncover valuable insights that will play a pivotal role in our subsequent analysis. Our primary focus was on visualizing the data to discern patterns related to the average price per housing unit concerning different parameters, such as state, number of bedrooms, and bathrooms.

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

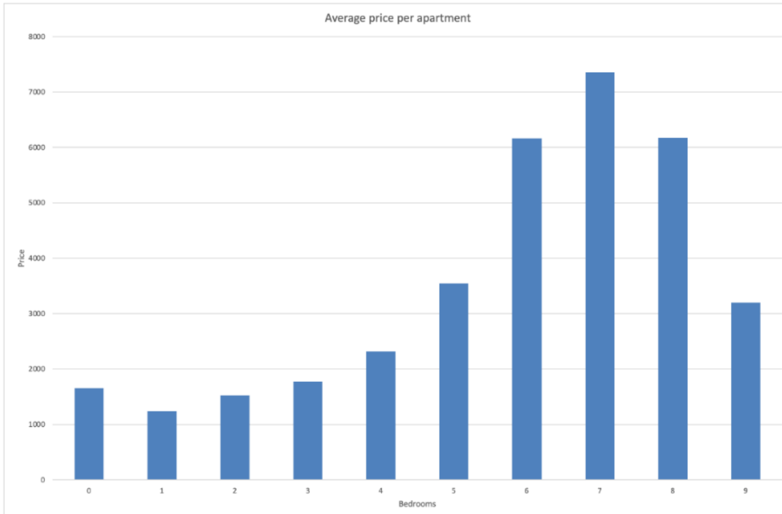
The charts provide insights on the variations in housing prices among different states, providing insights into areas that offer more affordable housing options as well as those at the upper end of the price range. We also conducted a thorough analysis of how housing unit prices are influenced by the number of bedrooms and bathrooms, a vital resource for renters looking to make informed decisions in the housing market. By comparing their specific preferences to our analyzed data, renters can make more informed decisions. Additionally, we examined the sources of these listings, a crucial aspect for both renters in their housing search and businesses aiming to evaluate their online presence.



Average Price per State Bar Chart

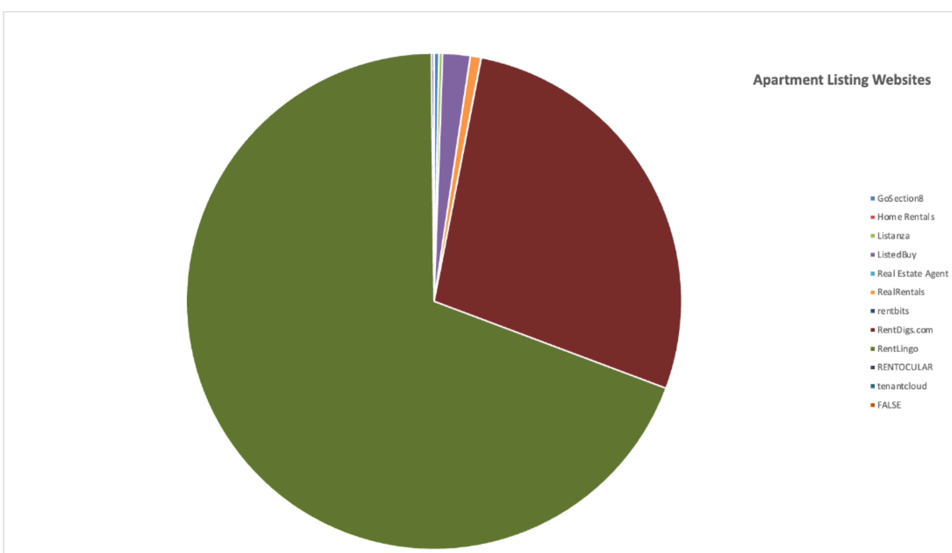
The Average Price per State Bar Chart provides a clear visual representation of the cost of housing across different states. Each bar represents a specific state, and its height corresponds to the average price of housing units in that state. This chart allows us to easily compare and contrast the affordability of housing in different states across the country. We can also use this to identify the states with the highest and lowest average rental prices. This information is valuable for individuals looking to move or invest in real estate.

1 Mariotti, Tony. "Renting Statistics: Trends & Demographics (2022)." RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.



Average Price per No. of Bedrooms Bar Chart

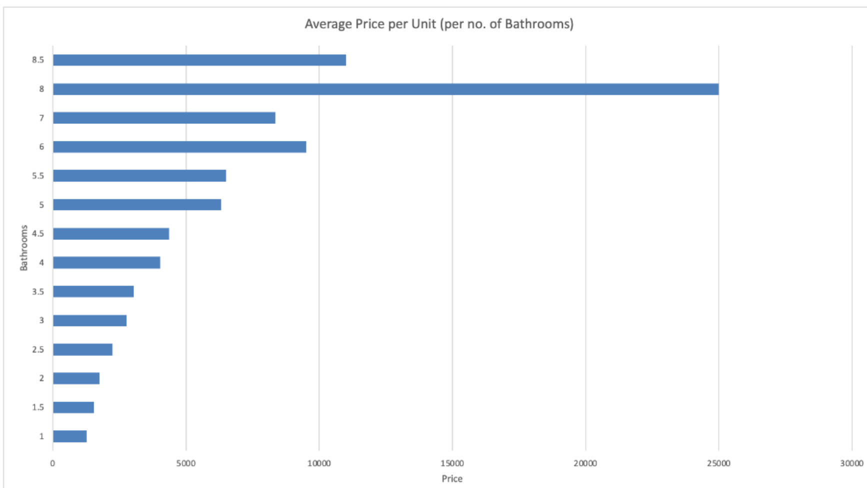
The average price per bedrooms bar chart illustrates the relationship between the average price of housing units and the number of bedrooms in those houses. Each bar in the chart corresponds to a specific number of bedrooms (e.g., 1-bedroom, 2-bedroom, 3-bedroom, etc.), and the height of each bar represents the average price of housing units with that specific bedroom count. This visualization enables viewers to discern how housing prices vary based on the number of bedrooms, helping potential buyers, sellers, and investors make informed decisions about housing options, budgets, and market trends.



Apartment Listing Websites Pie-chart

1 Mariotti, Tony. "Renting Statistics: Trends & Demographics (2022)." RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

The Apartment Listing Websites pie chart is a visual representation that displays the proportion of listings found on different websites. Each segment of the pie chart represents a specific website, and the size of each segment corresponds to the percentage of total listings available on that website. This chart provides a quick and easy way to understand which websites dominate the online listing market, making it useful for consumers exploring housing options and businesses evaluating their online presence.



Average price per unit (with respect to Bathrooms)

The average price per bathrooms bar chart illustrates the relationship between the average price of housing units and the number of bathrooms in those houses. Each bar in the chart represents a specific bathroom count (like 1 bathroom, 2 bathrooms, 3 bathrooms, and so on), and the height of each bar tells us the average price of houses with that particular bathroom count. This chart gives us a straightforward way to see how housing prices are influenced by the number of bathrooms.

Prediction and Regression

Regression is a tool that enables analytics and business professionals to make predictions with a certain degree of veracity and can be extremely crucial to the success of a business. In our case, we employ regression to predict the price of a house depending on various parameters. These parameters are listed below:

square_feet	square_feet^3	bathrooms	bedrooms
square_feet × bathrooms	square_feet^3 × bedrooms	region	

Here, region is a derived parameter. It is obtained from the *states* column in the dataset. It is based on the geographic division of all the US states:

- **Northeast:** Maine, New Hampshire, Vermont, Massachusetts, Rhode Island, Connecticut, New York, New Jersey, Pennsylvania

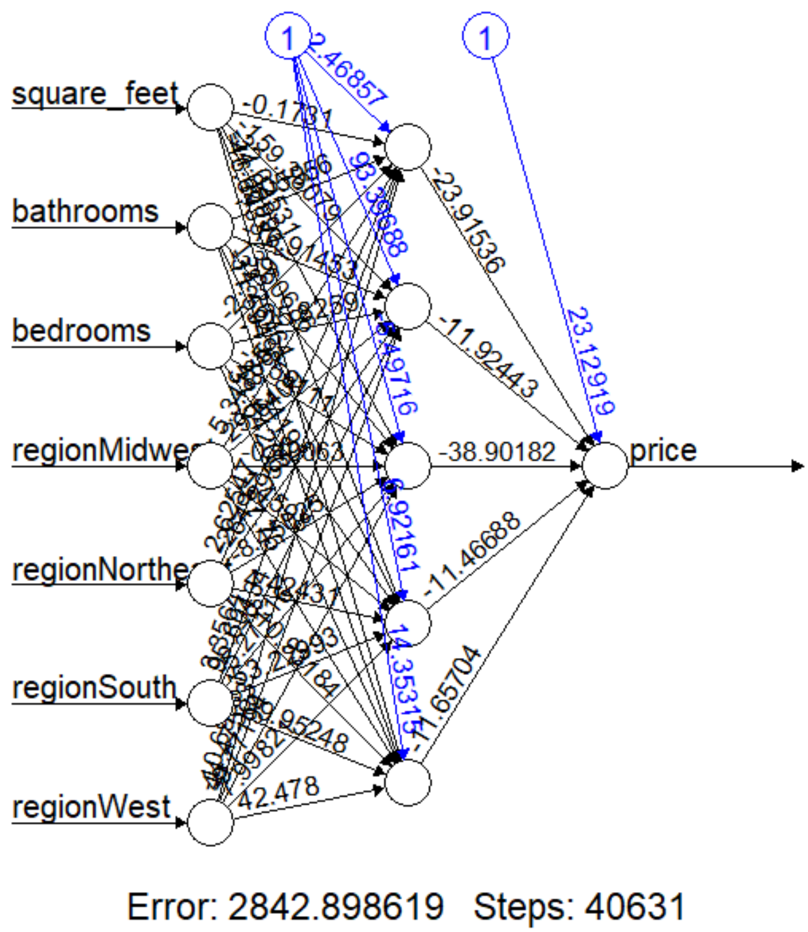
1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

- **South:** Delaware, Maryland, Virginia, West Virginia, North Carolina, South Carolina, Georgia, Florida, Alabama, Mississippi, Tennessee, Kentucky
- **Midwest:** Ohio, Indiana, Illinois, Michigan, Wisconsin, Minnesota, Iowa, Missouri
- **West:** North Dakota, South Dakota, Nebraska, Kansas, Oklahoma, Texas, Montana, Wyoming, Colorado, New Mexico, Arizona, Utah, Nevada, Idaho, Oregon, Washington, California, Alaska, Hawaii

We noticed that the relationship represents a quadratic curve and not a linear one and added the *square_feet*^{3} term to compensate for this effect. Additionally, we also account for the interaction between *square_feet* and *bathrooms* and between *square_feet*^{3} and *bedrooms*. This helps us achieve an adjusted R^2 value of almost 28.5%.

Neural Networks

Additionally, we employed neural networks to gauge the optimality of the two models. In this case, the dependent variable, price, was forecasted based on *square_feet*, *bathrooms*, *bedrooms*, and *region*. The data was standardized and converted into z-scores for a better fit, and the model was prepared with one hidden layer with five neurons. The visual representation of the model is given below:



1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

Since region is a categorical variable, it was encoded into dummy variables representing each of the regions.

Comparing the Models

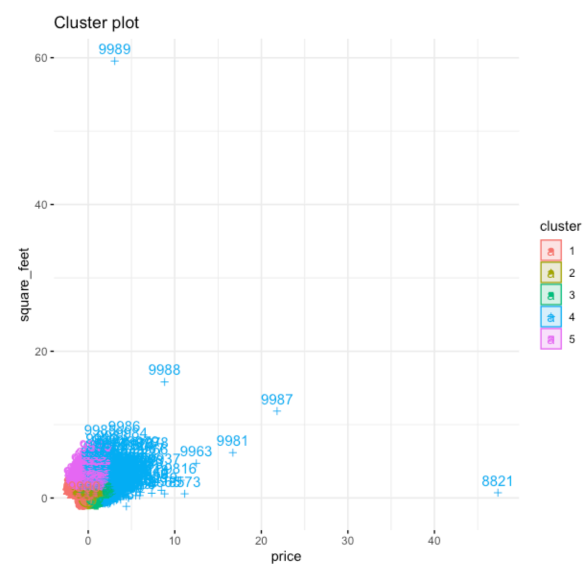
Between the regression model and the neural network model, the regression model appears to perform better for this dataset. These details are summarized in the table below:

	Regression Model	Neural Network Model
R-Squared Value	0.2842	0.187343
Relative Root Mean Square Error	9.089306	73.55552

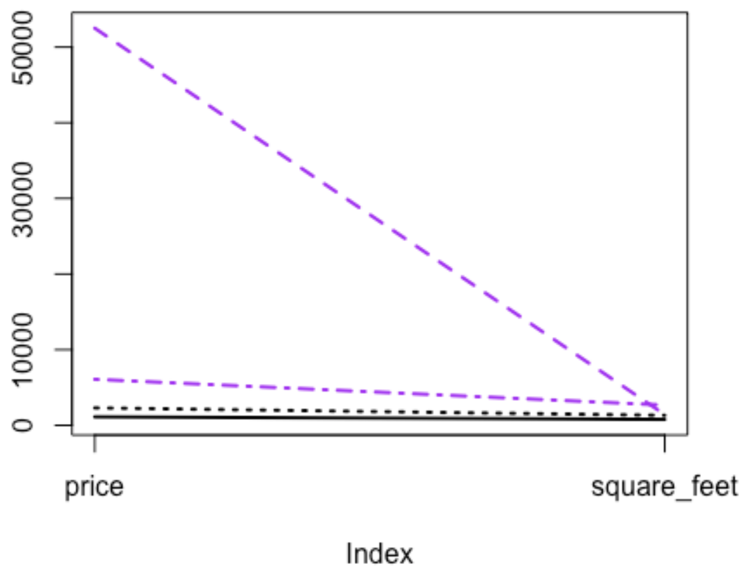
Here, the relative RMSE value of the regression analysis is under 10, while that of the neural network is almost 74. The regression analysis performs much better in terms of the R-squared metric as well. Thus, the regression model is better suited for this dataset.

Clustering Analysis

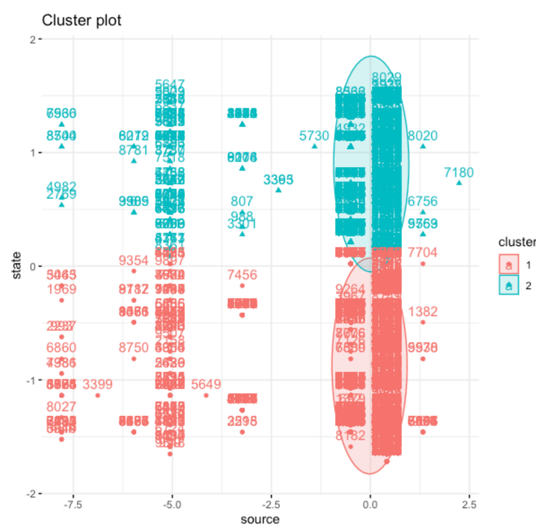
Cluster Analysis is a widespread tool in Business Analytics that uses data mining techniques to segment various smaller groups containing similar characteristics and features. The method works through many datasets and analyses features with the most common aspects, curating them together in smaller groups for easier access. In our case, we would employ clustering for price and square feet. We have also employed clustering for source and state variables.



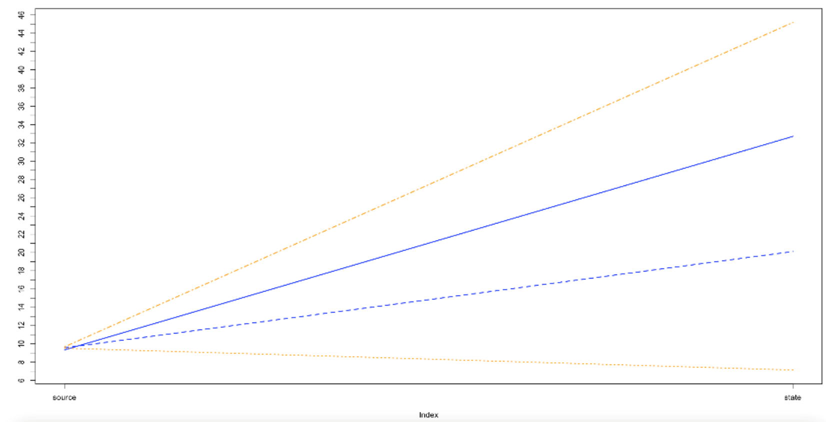
1 Mariotti, Tony. "Renting Statistics: Trends & Demographics (2022)." RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.



Price and Square feet: Cluster 1's price remains near 1000 and square feet is below 1000 while Cluster 2 shows maximum variation with price being above 52000 while square feet is around 1400. For Cluster 3, the price is around 2300 for square feet of around 1300, while for Cluster 4 the price is around 6000 for square feet of around 2600.



1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.



State and source: For all states the source is Rent lingo (10).

Conclusion

In summary, this project utilized Business Intelligence techniques, including clustering, regression, and neural networks, to analyze rental apartment data and address the primary goal of predicting price optimization and market segmentation. Additionally, it provided valuable insights for more effective business strategies. Through clustering, distinct market segments were identified, while regression analysis illuminated factors influencing rental prices. The incorporation of neural networks showcased a commitment to innovative predictive modeling. The project outcomes contribute essential information for strategic decision-making in the dynamic rental housing market.

References

The dataset was obtained from https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified.

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

BUAN 6356.004 Business Analytics with R

GROUP 6

Code

```
# Importing the dataset

library(readxl)

Apartments_Data_Initial <- read_excel("Apartments Data.xlsx", sheet = "Apartments Data (10K)")


###

# Data cleaning and Pre-processing

###


# Making a copy of the Dataset

Apartments_Data <- Apartments_Data_Initial


# Getting the names of columns with missing values

print(names(Apartments_Data)[colSums(is.na(Apartments_Data)) > 0])


# The above function returned: "bedrooms"  "latitude"  "longitude"

# Replacing the N.A. values in the 'bedrooms' column with the average value of the column

avg_bedrooms <- mean(Apartments_Data$bedrooms, na.rm = TRUE)

Apartments_Data$bedrooms <- ifelse(is.na(Apartments_Data$bedrooms), avg_bedrooms,
Apartments_Data$bedrooms)


# Removing the N.A. values in the dataset but only 'latitude' and 'longitude' columns contain N.A. values

Apartments_Data <- na.omit(Apartments_Data)


# Verifying that no columns have missing values anymore

print(names(Apartments_Data)[colSums(is.na(Apartments_Data)) > 0])


# The above function returned: 0
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
# Removing columns with "null" values

# Replacing amenities = "null" with amenities = "None"

Apartments_Data$amenities <- ifelse(Apartments_Data$amenities == "null", "None",
Apartments_Data$amenities)


# Replacing bathrooms = "null" with bathrooms = 0

Apartments_Data$bathrooms <- ifelse(Apartments_Data$bathrooms == "null", 0,
Apartments_Data$bathrooms)


# Replacing pets_allowed = "null" with pets_allowed = "None"

Apartments_Data$pets_allowed <- ifelse(Apartments_Data$pets_allowed == "null", "None",
Apartments_Data$pets_allowed)


# Replacing address = "null" with address = "Unavailable"

Apartments_Data$address <- ifelse(Apartments_Data$address == "null", "Unavailable",
Apartments_Data$address)


# Replacing cityname = "null" with cityname = "Unavailable"

Apartments_Data$cityname <- ifelse(Apartments_Data$cityname == "null", "Unavailable",
Apartments_Data$cityname)


# Replacing state = "null" with state = "Unavailable"

Apartments_Data$state <- ifelse(Apartments_Data$state == "null", "Unavailable",
Apartments_Data$state)


# Formatting the 'bathrooms' column as numeric

Apartments_Data$bathrooms <- as.numeric(Apartments_Data$bathrooms)
```

```
# Checking if all the prices are for the same amount of time

print(unique(Apartments_Data$price_type))


# The above function returned: "Monthly" "Weekly" "Monthly|Weekly"


# Updating 'Weekly' and 'Monthly|Weekly' prices to 'Monthly' prices to maintain consistency

Apartments_Data$price[Apartments_Data$price_type == "Weekly" | Apartments_Data$price_type == "Monthly|Weekly"] <- Apartments_Data$price[Apartments_Data$price_type == "Weekly" | Apartments_Data$price_type == "Monthly|Weekly"] * 4

Apartments_Data$price_type[Apartments_Data$price_type == "Weekly" | Apartments_Data$price_type == "Monthly|Weekly"] <- "Monthly"

#Clustering

library(cluster)

library(ggplot2)

library(factoextra)

# Select relevant variables for clustering

selected_vars <- c( "price", "square_feet")

cluster_data <- Apartments_Data[selected_vars]

#run pam algorithm, metric =“euclidean”

set.seed(2)

km <- kmeans(cluster_data, 4)

km$cluster

# centroids

km$centers

km$withinss

km$size

min(km$centers)

max(km$centers)
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
fviz_cluster(km, cluster_data, ellipse.type ="euclid",  ggtheme = theme_minimal())

# plot an empty scatter plot

plot(c(0), xaxt = 'n', ylab = "", type = "l",

    ylim = c(min(km$centers), max(km$centers)), xlim = c(1, 2))

# label x-axes

axis(1, at = c(1:2), labels = names(cluster_data))


# plot centroids

for (i in c(1:4))

  lines(km$centers[i,], lty = i, lwd = 2, col = ifelse(i %in% c(1, 2),

                                                       "black", "purple"))

# name clusters

text(x =0.5, y = km$centers[, 1], labels = paste("Cluster", c(1:2)))

Apartments_Data$source <- as.numeric(Apartments_Data$source)

Apartments_Data$state <- as.numeric(Apartments_Data$state)

# Select relevant variables for clustering

selected_vars <- c( "source", "state")

cluster_data1 <- Apartments_Data[selected_vars]

#run pam algorithm, metric =“euclidean”

set.seed(2)

km <- kmeans(cluster_data1, 4)

km$cluster

# centroids

km$centers

km$withinss

km$size

fviz_cluster(km, cluster_data1, ellipse.type ="euclid",  ggtheme = theme_minimal())
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
# plot an empty scatter plot
plot(c(0), xaxt = 'n', ylab = "", type = "l",
    ylim = c(min(km$centers), max(km$centers)), xlim = c(1, 2))
# label x-axes
axis(1, at = c(1:2), labels = names(cluster_data1))
axis(2, at = axTicks(2, axp = c(0, 55, 55)))
# plot centroids
for (i in c(1:4))
  lines(km$centers[i,], lty = i, lwd = 2, col = ifelse(i %in% c(1, 2),
                                       "blue", "orange"))
# name clusters
text(x =0.5, y = km$centers[, 1], labels = paste("Cluster", c(1:2)))


# Adding a new region column
region <- ""
Apartments_Data <- cbind(Apartments_Data, region)


Apartments_Data$region[Apartments_Data$state == "NY" | Apartments_Data$state == "MA" |
Apartments_Data$state == "NJ" | Apartments_Data$state == "PA" | Apartments_Data$state == "CT" |
Apartments_Data$state == "RI" | Apartments_Data$state == "NH" | Apartments_Data$state == "VT" |
Apartments_Data$state == "ME"] <- "Northeast"

Apartments_Data$region[Apartments_Data$state == "VA" | Apartments_Data$state == "NC" |
Apartments_Data$state == "GA" | Apartments_Data$state == "FL" | Apartments_Data$state == "AL" |
Apartments_Data$state == "MD" | Apartments_Data$state == "TN" | Apartments_Data$state == "DE" |
Apartments_Data$state == "SC" | Apartments_Data$state == "KY" | Apartments_Data$state == "LA" |
Apartments_Data$state == "AR" | Apartments_Data$state == "WV" | Apartments_Data$state == "MS"]
<- "South"

Apartments_Data$region[Apartments_Data$state == "IN" | Apartments_Data$state == "IL" |
Apartments_Data$state == "IA" | Apartments_Data$state == "MN" | Apartments_Data$state == "MI" |
Apartments_Data$state == "WI" | Apartments_Data$state == "OH" | Apartments_Data$state == "MO"]
<- "Midwest"
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
Apartments_Data$region[Apartments_Data$state == "DC" | Apartments_Data$state == "WA" |
Apartments_Data$state == "CA" | Apartments_Data$state == "AZ" | Apartments_Data$state == "TX" |
Apartments_Data$state == "CO" | Apartments_Data$state == "NM" | Apartments_Data$state == "AK" |
Apartments_Data$state == "OR" | Apartments_Data$state == "NV" | Apartments_Data$state == "UT" |
Apartments_Data$state == "OK" | Apartments_Data$state == "NE" | Apartments_Data$state == "ND" |
Apartments_Data$state == "KS" | Apartments_Data$state == "ID" | Apartments_Data$state == "HI" |
Apartments_Data$state == "MT" | Apartments_Data$state == "SD" | Apartments_Data$state == "WY"]
<- "West"

Apartments_Data$region[Apartments_Data$state == "Unavailable"] <- "Unavailable"

###

# Forming the Regression

###


# Running the regression

square_feet3 <- Apartments_Data$square_feet * Apartments_Data$square_feet  *
Apartments_Data$square_feet

reg_model <- lm(price ~ square_feet + square_feet3 + bathrooms + bedrooms + square_feet*bathrooms
+ square_feet3*bedrooms + state + cityname, data = Apartments_Data)


# Regression Summary

summary(reg_model)

###

# Removing outliers

###


# Calculating Cook's distance

cooksd <- cooks.distance(reg_model)


# Identifying influential observations (outliers)

influential_obs <- which(cooksd > 4 / length(cooksd))
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
# Printing influential observations

cat("Influential Observations (Outliers):", influential_obs, "\n")


# Removing influential observations from the dataset

Apartments_Data_no_outliers <- Apartments_Data[-influential_obs, ]


# Fitting a new model without outliers

square_feet3 <- Apartments_Data_no_outliers$square_feet *
Apartments_Data_no_outliers$square_feet * Apartments_Data_no_outliers$square_feet

reg_model_no_outliers <- lm(price ~ square_feet3 + bathrooms + bedrooms + square_feet3*bathrooms
+ square_feet*bedrooms + region, data = Apartments_Data_no_outliers)


summary(reg_model_no_outliers)


# Calculating errors for the regression

predicted_values <- predict(reg_model_no_outliers, Apartments_Data_no_outliers, type = "response")

actual_values <- Apartments_Data_no_outliers$price


mse <- mean((actual_values - predicted_values)^2, na.rm = TRUE)

rmse <- sqrt(mse)


# Calculating relative RMSE

cv_rmse <- (rmse / (max(predicted_values) - min(predicted_values))) * 100

print(cv_rmse)

###

# Apply Neural Network

###
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
library(neuralnet)

# Function to standardize values in terms of z-scores
z_score_standardize <- function(x) {
  if (is.numeric(x)) {
    return((x - mean(x)) / sd(x))
  } else {
    return(x)
  }
}

Apartments_Data_standardized_initial <- as.data.frame(lapply(Apartments_Data, z_score_standardize))

# Encoding region into numeric variables
encoded_data <- model.matrix(~ region - 1, data = Apartments_Data_standardized_initial)
Apartments_Data_standardized <- cbind(Apartments_Data_standardized_initial[, c("price",
"square_feet", "bathrooms", "bedrooms")], encoded_data)

# Identifying the odd and even rows
odd_rows <- seq(1, nrow(Apartments_Data_standardized), by = 2)
even_rows <- seq(2, nrow(Apartments_Data_standardized), by = 2)

# Creating the training and validation datasets
training_data <- Apartments_Data_standardized[odd_rows, ]
validation_data <- Apartments_Data_standardized[even_rows, ]

# Defining the neural network model
```

1 Mariotti, Tony. "Renting Statistics: Trends & Demographics (2022)." RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
nn <- neuralnet(price ~ square_feet + bathrooms + bedrooms + regionMidwest + regionNortheast + regionSouth + regionWest,
            data = training_data,
            linear.output = F,
            hidden = 5,          # Number of hidden layers and neurons
            learningrate = 1.5)   # Adjust the learning rate as needed

plot(nn, rep = "best")

predictions <- predict(nn, validation_data, type = "response")
actual_values <- validation_data$price

# Calculate Mean Absolute Error (MAE)
mae <- mean(abs(predictions - actual_values))

# Calculate Mean Squared Error (MSE)
mse <- mean((predictions - actual_values)^2)

# Calculate Root Mean Squared Error (RMSE)
rmse <- sqrt(mse)

# Calculating relative RMSE
cv_rmse <- (rmse / (max(predictions) - min(predictions))) * 100

# Calculating R-squared
rsquared <- 1 - sum((actual_values - predictions)^2) / sum((actual_values - mean(actual_values))^2)

cat("Mean Absolute Error (MAE):", mae, "\n")
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.

```
cat("Mean Squared Error (MSE):", mse, "\n")

cat("Root Mean Squared Error (RMSE):", rmse, "\n")

cat("Relative Root Mean Squared Error (RMSE):", cv_rmse, "\n")

cat("R-Squared:", rsquared, "\n")
```

1 Mariotti, Tony. “Renting Statistics: Trends & Demographics (2022).” RubyHome.com, 6 Aug. 2022, www.rubyhome.com/blog/renting-stats/. Accessed 28 Oct. 2023.