ARTIFICIAL INTELLITENCE AND DATA SCIENCE

# ASSESSEMENT BY TEMITOPE AKADIRI

Contents

[QUESTION 2: FINDINGS FROM DATASET 4](#_Toc139361146)

[INTRODUCTION 4](#_Toc139361147)

[METHODOLOGY 4](#_Toc139361148)

[FINDINGS / ANALYSIS 5](#_Toc139361149)

[Figure 1: Basic statistics on trained dataset 5](#_Toc139361150)

[Figure 2: A Correlation analysis between all the variables 6](#_Toc139361151)

[Figure 3: Multiple Linear Regression coefficients using Price (target) against the features 7](#_Toc139361152)

[Figure 4: A scatter plot of the predicted price versus actual price 8](#_Toc139361153)

[Random Forest 9](#_Toc139361154)

[Figure 6: A scatter plot of the predicted price versus actual price (Random Forest) 9](#_Toc139361155)

[Knn Regression 10](#_Toc139361156)

[Figure 8: A truth graph of the predicted price versus actual price (Knn Regression) 11](#_Toc139361157)

[Figure 9: A comparison between actual values of the price, and the predicted prices generated in the models 12](#_Toc139361158)

[Feature Importance 12](#_Toc139361159)

[Figure 7: Out-of-bag Feature Importance for all the inputs 13](#_Toc139361160)

[CONCLUSION: 14](#_Toc139361161)

[APPENDIX 1: PROJECTED PRICES AND CODES 15](#_Toc139361162)

[Price Projection 15](#_Toc139361163)

[Code 39](#_Toc139361164)

[IMPORTING TRAINED DATASET 39](#_Toc139361165)

[IMPORTING TEST DATASET 40](#_Toc139361166)

[CHECKING INFORMATION OF TEST AND TRAINED DATA 41](#_Toc139361167)

[BASIC STATISTICS ON TRAINED DATA TO PROVIDE MORE UNDERSTANDING OF THE DATA 43](#_Toc139361168)

[DATA VISUALIZATION WITH PAIR PLOTS 44](#_Toc139361169)

[CORRELATION ANALYSIS BETWEEN ALL THE FEATURES 45](#_Toc139361170)

[MULTIPLE LINEAR REGRESSION MODEL 45](#_Toc139361171)

[APPLICATION OF RANDOM FOREST 52](#_Toc139361172)

[Feature Importance 53](#_Toc139361173)

[OUT-OF-BAG (OOB) Feature Importance 53](#_Toc139361174)

[Using KNeighborsRegressor 55](#_Toc139361175)

[PRICE PREDICTIONS WITH MULTIPLE LINEAR REGRESSION VS RANDOM FOREST 56](#_Toc139361176)

# QUESTION 2: FINDINGS FROM DATASET

# INTRODUCTION

Artificial Intelligence supports organizations prices projections experience by analysing vast volumes of data. For instance, machine learning algorithms can gather data on buyer habits and pricing to create complex algorithms that reliably project changes in price. Given how quite simple it is to use and how much time it saves, many

organisations stand to benefit from this.

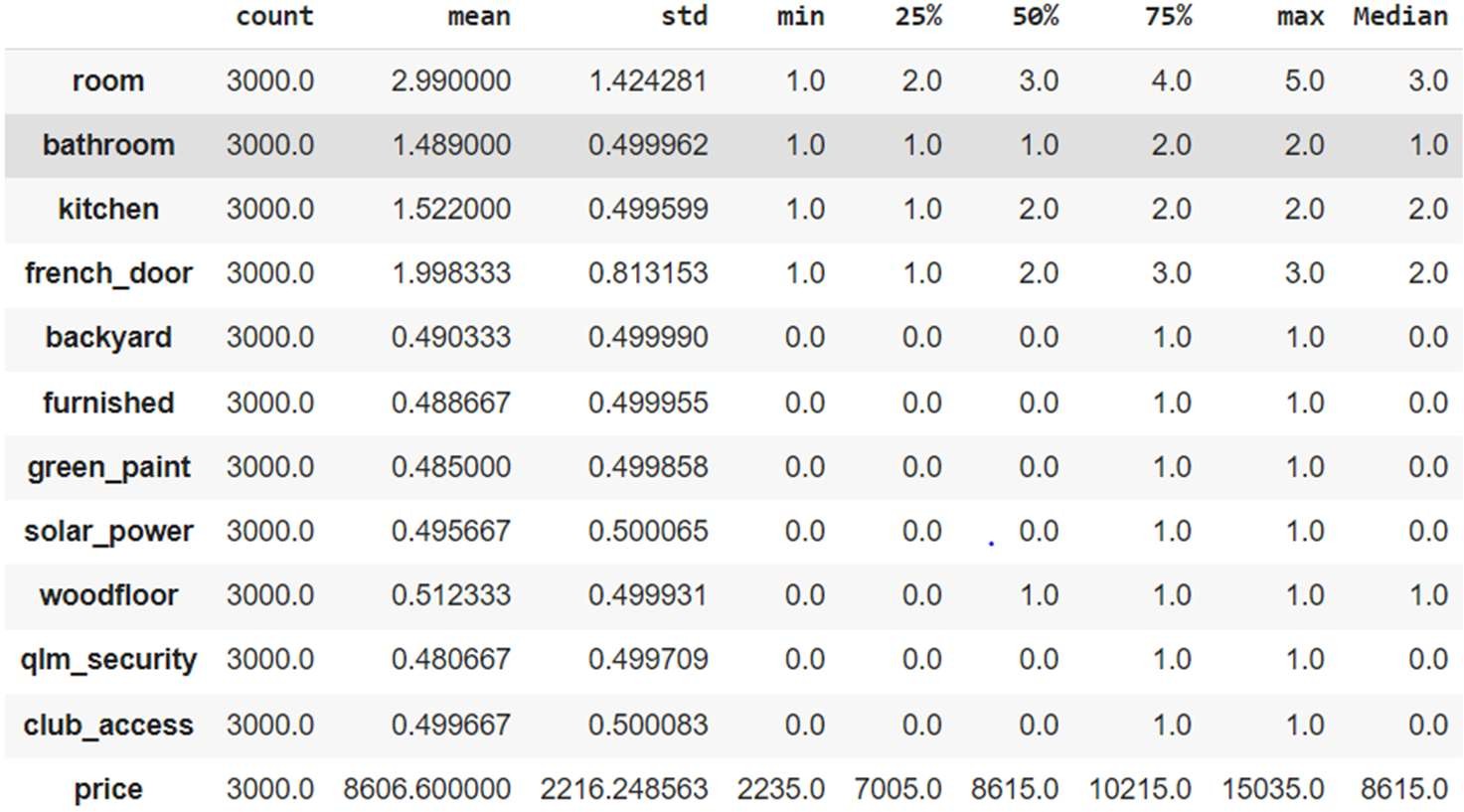
This report compares various regression models that were applied to a housing dataset to forecast price changes.

# METHODOLOGY

For this report, python programming Language was used on Google Collab to analyse and create different regression models to predict the house prices in the provided dataset. This trained dataset has 3000 rows and 12 columns while test dataset has 999 rows and 12 columns. The features in both housing dataset included room, bathroom, kitchen, French door, backyard, furnished, green paint, solar power, wood floor, qlm security, and club access. Using a multiple linear regression, Random forest, and Knn Regression, the price (target) with assumed unit pounds will be predicted.

# FINDINGS / ANALYSIS

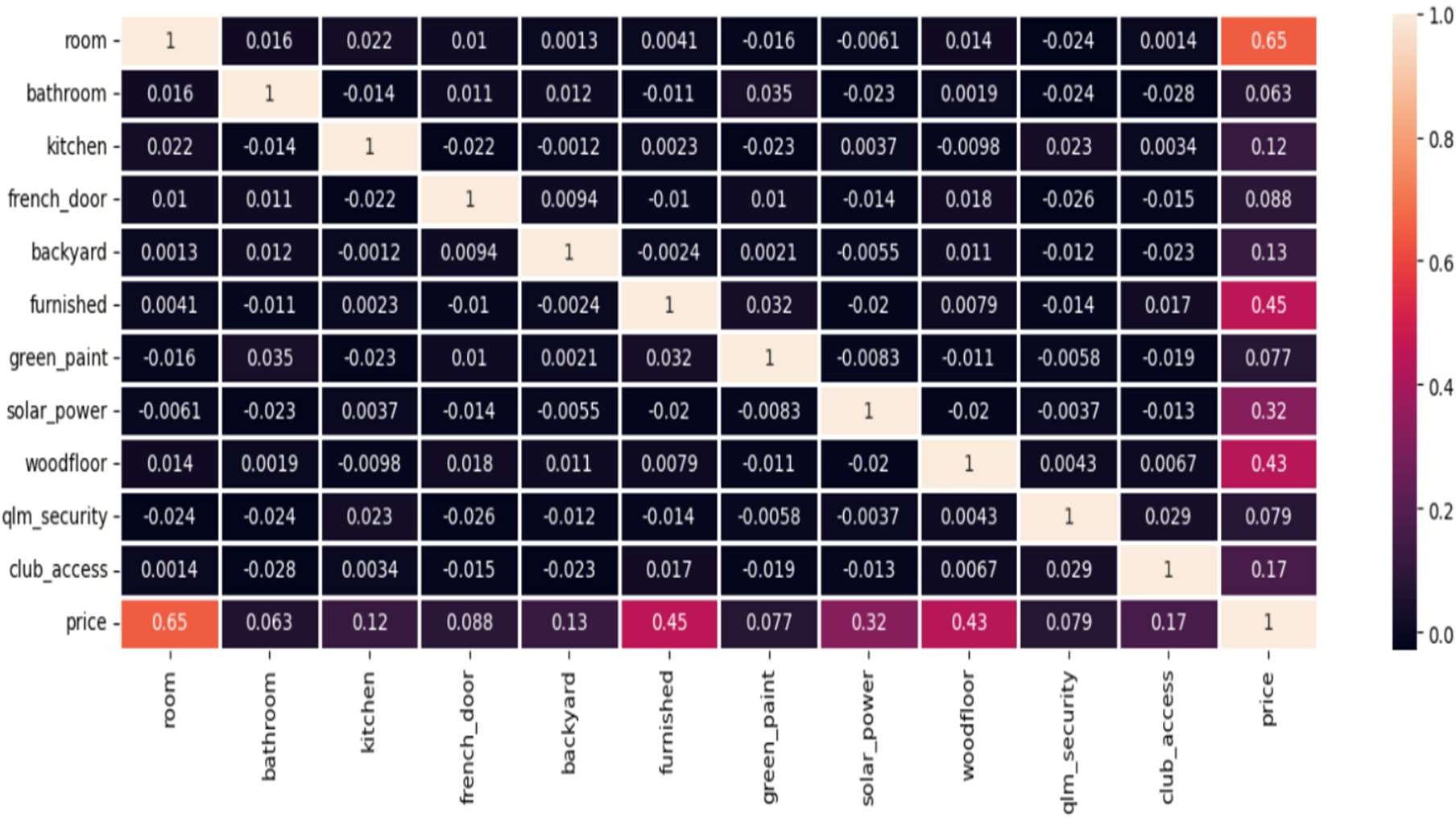
To understand this dataset, Basic statistics was performed on the trained dataset which explores information on the mean, median, standard deviation, minimum & maximum value, and percentile of the data.



# Figure 1: Basic statistics on trained dataset

The figure 1 shows that on an average, customers buy houses that cost about £8615 which has just 3 rooms, 1 bathroom, 2 kitchens, 2 French doors with a wood floor but without the other features.

To further understand the degree of relationship between all the variables, a correlation matrix was generated.



# Figure 2: A Correlation analysis between all the variables

The figure 2 shows that there is a positive correlation between price and all the features with the room number having the strongest correlation at 0.65. Following this assumption then a price prediction can be performed with these features using machine learning algorithms.

Using the **Multiple Linear Regression model,** the equation below was generated (Figure 3).

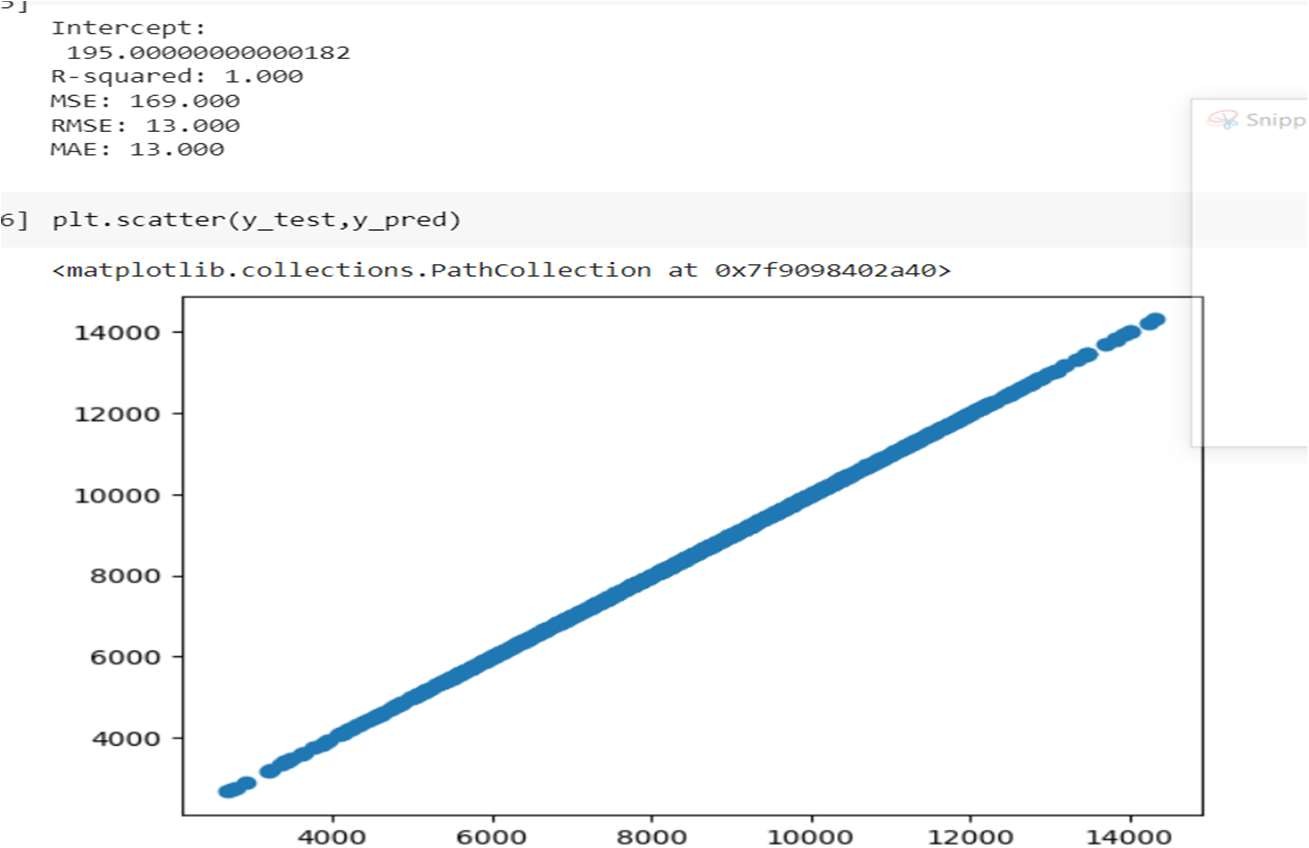
Price= 195 + 1000\*room + 300\*bathroom + 500\*kitchen + 240\*French door + 560\*backyard + 2000\*furnished + 370\*green paint + 1530\*solar power + 1890\*wood floor + 440\*qlm-security + 730\*club access.



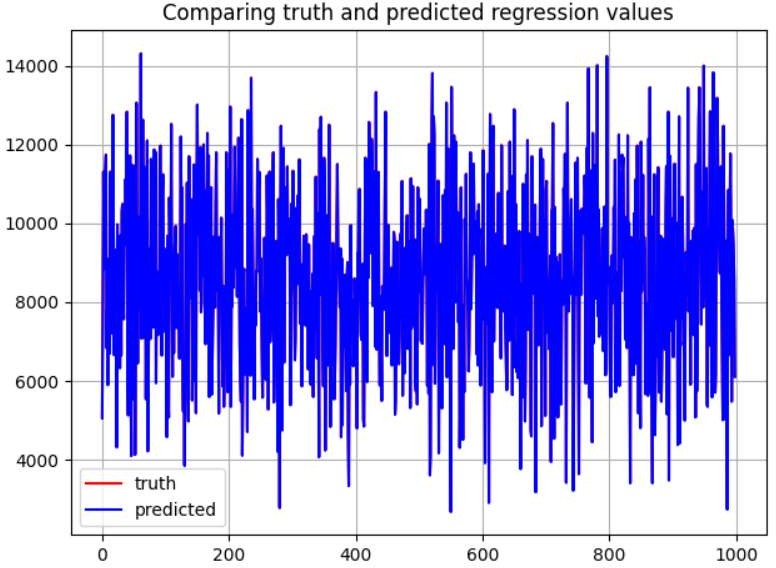
# Figure 3: Multiple Linear Regression coefficients using Price (target) against the features

Also, the R-squared value was equal to 1 which implies that 100% of the variation in the price can be explained by the regression model. The mean square error is also relatively low (169). This implies that this is an exceptionally reliable model.

To visualise the difference between the predicted price against the actual price, a scatter plot was generated. As seen in figure 4 and 5, there were no outliers which implies that the difference in the predicted price and the actual price is insignificant.



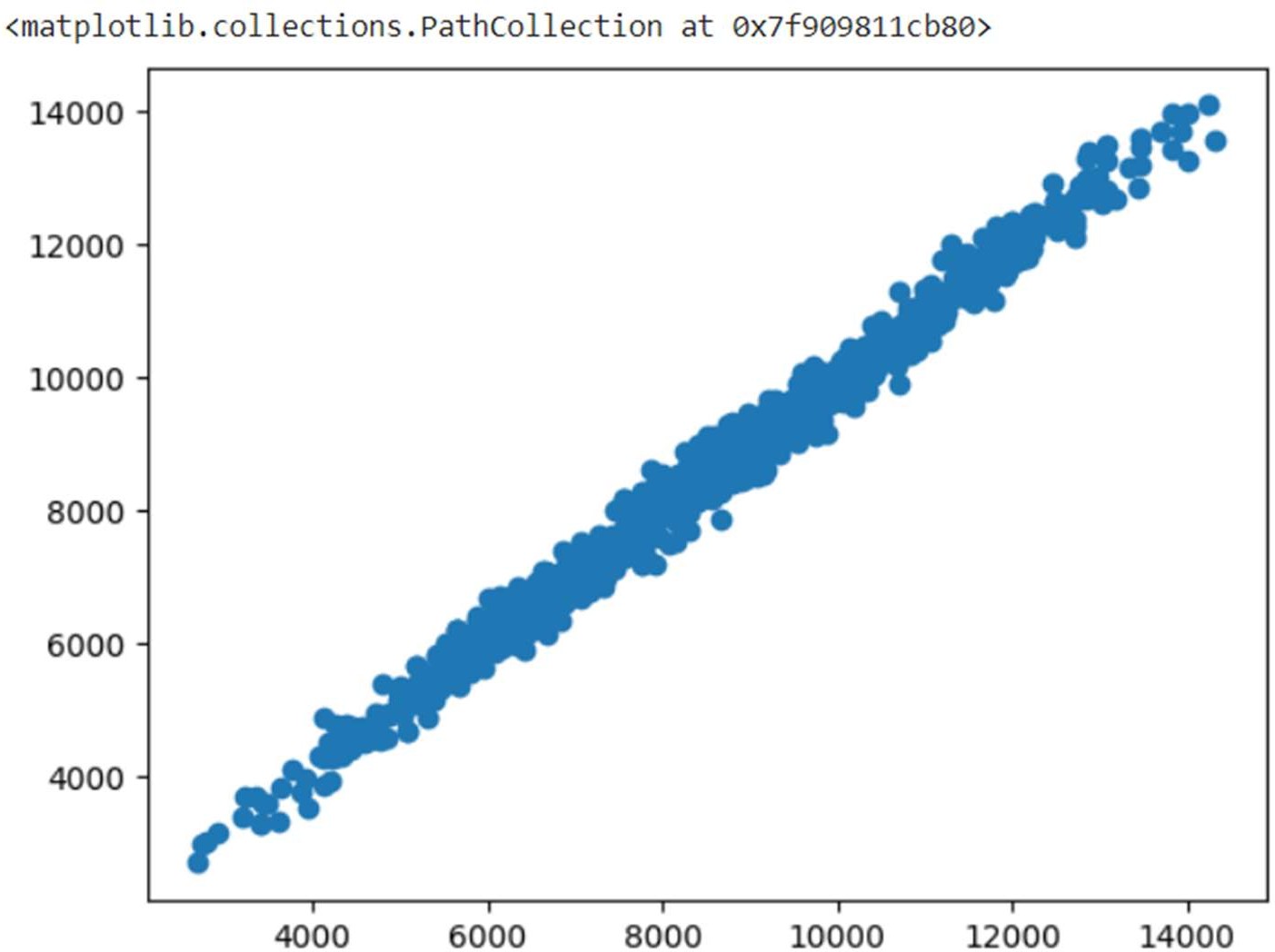
# Figure 4: A scatter plot of the predicted price versus actual price



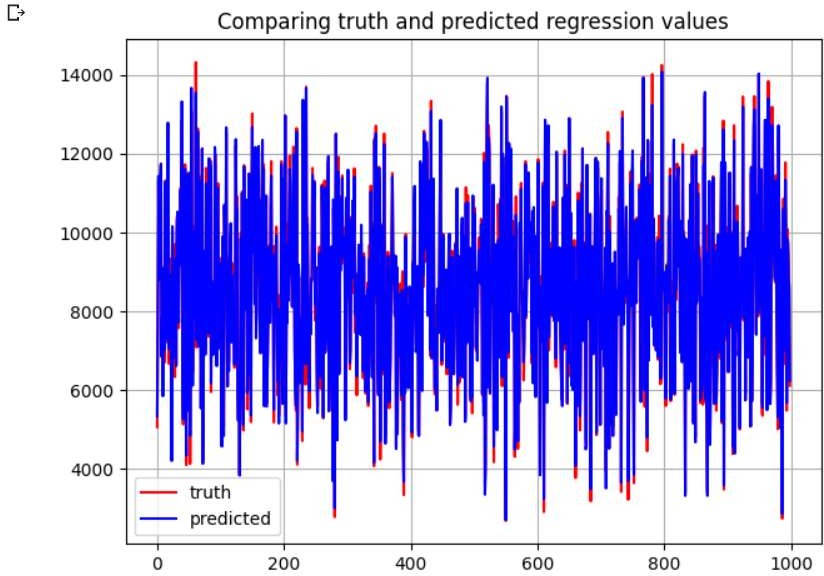
**Figure 5: A truth graph of the predicted price versus actual price**

# Random Forest

To check the reliability of the multiple linear regression model, a Random Forest model was also used as a method to predict house prices. The R-squared value was 0.9894 which implies that 98.94% of the variation in the price can be explained by this model. The mean square error of 51904. The figure 6 is scatter plot to visualize the predicted price against the actual price. This has more outliers than the multiple linear regression model which implies that the multiple linear regression is more suitable for the price prediction.



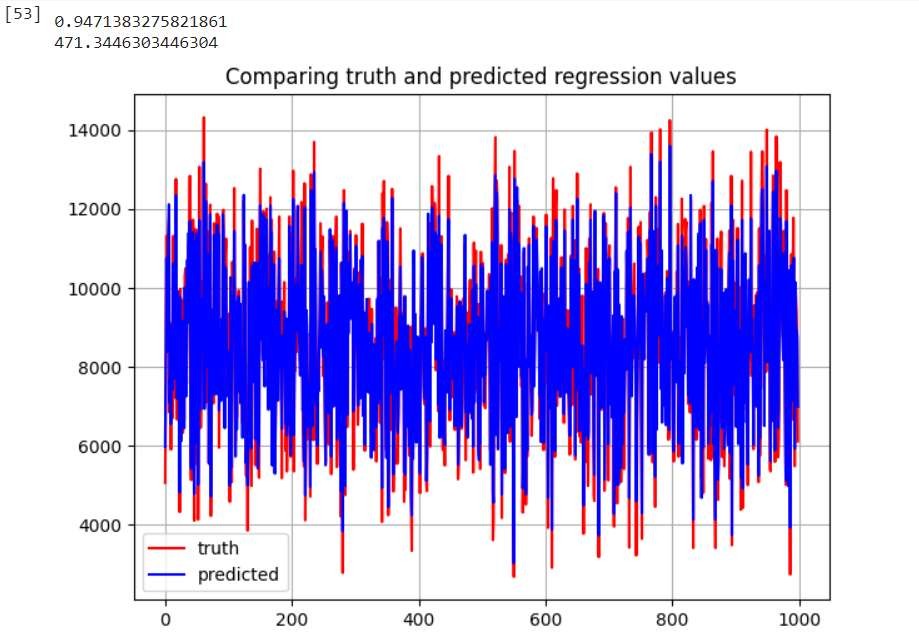
# Figure 6: A scatter plot of the predicted price versus actual price (Random Forest)



**Figure 7: A truth graph of the predicted price versus actual price (Random Forest)**

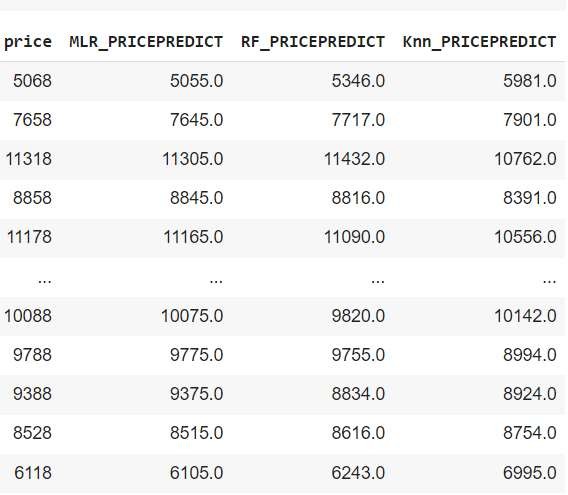
# Knn Regression

Finally, Knn regression was used to predict the price. R-squared value was 0.9471 which implies that 94.71% of the variation in the price can be explained by this model. The mean square error of 471. The figure 8 is a truth graph to visualize the predicted price against the actual price. This has more outliers than the Random Forest model which implies that it is less suitable for the price prediction.



# Figure 8: A truth graph of the predicted price versus actual price (Knn Regression)

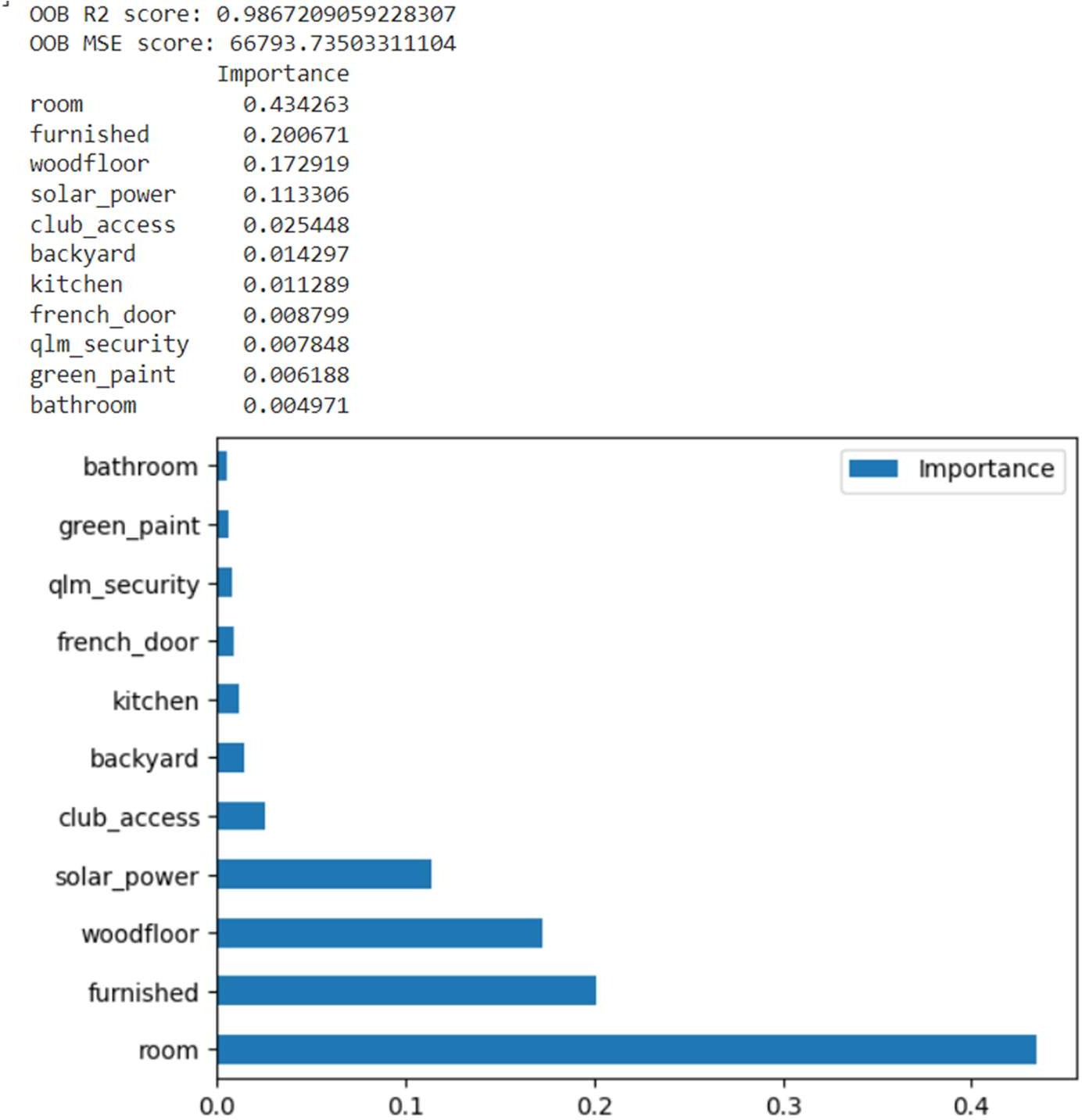
The figure 9 below shows the actual value comparison among the actual price, predicted price from the multiple linear regression (MLR\_PRICEPREDICT), Random Forest (RF\_pricepredict), and Knn Regression . This shows that multiple linear regression model is more reliable as the difference between the actual price is insignificant.



# Figure 9: A comparison between actual values of the price, and the predicted prices generated in the models

# Feature Importance

By applying the out-of-box feature importance on the random forest model, values were assigned to the features to show the relative relevance of each feature when predicting the price.



# Figure 7: Out-of-bag Feature Importance for all the inputs

The Figure 7 shows the order of feature importance with room having 0.4342, furnished having 0.2007, wood floor having 0.1729, solar power having 0.1133, club access having 0.0254, backyard having 0.0143, kitchen having 0.0113, French door having 0.0088, qlm-security having 0.007848, green paint having 0.0062, and bathroom having 0.004971. This implies that room is most valuable feature in this model then furnished, wood floor, solar power, club access, backyard, kitchen, French door, qlm-security, green paint, and bathroom. This also implies that the removal of bathroom feature will cause an insignificant change to the regression model adopted earlier.

# CONCLUSION:

In conclusion, these findings show that there is a reasonable relationship between the dependent variable (price)(target) and the independent variables (room, furnished, wood floor, solar power, club access, backyard, kitchen, French door, qlm-security, green paint, and bathroom) through the correlation matrix and regression models applied (multiple linear regression, random forest and Knn Regression).

It also implies that the three regression models are reliable tools to predict the price since the R-Squared values are (Multiple linear regression: 1, Random Forest: 0.9872,

And Knn Regression: 0.9471).

Finally, the feature importance identified the room variable as the most importance feature in the Random forest model.

# APPENDIX 1: PROJECTED PRICES AND CODES

# Price Projection

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual price** | **MLR**  **PRICE** | **RF\_PRICE** | **Knn\_PRICE** |
| 5068 | 5055 | 5346 | 5981 |
| 7658 | 7645 | 7717 | 7901 |
| 11318 | 11305 | 11432 | 10762 |
| 8858 | 8845 | 8816 | 8391 |
| 11178 | 11165 | 11090 | 10556 |
| 11388 | 11375 | 11452 | 11162 |
| 11748 | 11735 | 11728 | 12116 |
| 6848 | 6835 | 6870 | 7096 |
| 7828 | 7815 | 7869 | 7706 |
| 5908 | 5895 | 5854 | 6556 |
| 9108 | 9095 | 9101 | 9046 |
| 8308 | 8295 | 8528 | 7755 |
| 6708 | 6695 | 7033 | 7438 |
| 11318 | 11305 | 11284 | 10614 |
| 7218 | 7205 | 7303 | 7708 |
| 9178 | 9165 | 9499 | 8995 |
| 8788 | 8775 | 8864 | 8658 |
| 12758 | 12745 | 12784 | 12346 |
| 9758 | 9745 | 9843 | 9692 |
| 6668 | 6655 | 7049 | 7194 |
| 9338 | 9325 | 8922 | 9918 |
| 7798 | 7785 | 7455 | 8105 |
| 7088 | 7075 | 7134 | 7414 |
| 4328 | 4315 | 4213 | 4831 |
| 9978 | 9965 | 10159 | 9769 |
| 6868 | 6855 | 7347 | 6119 |
| 7348 | 7335 | 7404 | 8194 |
| 8908 | 8895 | 9105 | 7464 |
| 6338 | 6325 | 6815 | 6461 |
| 9708 | 9695 | 9322 | 9169 |
| 6658 | 6645 | 6617 | 6454 |
| 10318 | 10305 | 10038 | 10012 |
| 10498 | 10485 | 10528 | 9478 |
| 8208 | 8195 | 8248 | 7258 |
| 7578 | 7565 | 7721 | 8441 |
| 10368 | 10355 | 10223 | 10712 |
| 9488 | 9475 | 9173 | 9192 |

|  |  |  |  |
| --- | --- | --- | --- |
| 11128 | 11115 | 11079 | 11394 |
| 9778 | 9765 | 9406 | 9128 |
| 12838 | 12825 | 13326 | 11698 |
| 6308 | 6295 | 6722 | 7166 |
| 5138 | 5125 | 5271 | 5425 |
| 9838 | 9825 | 9969 | 9531 |
| 8908 | 8895 | 8954 | 8448 |
| 11728 | 11715 | 11614 | 11412 |
| 6218 | 6205 | 6123 | 5682 |
| 4108 | 4095 | 4345 | 4789 |
| 11488 | 11475 | 11423 | 11156 |
| 9578 | 9565 | 9566 | 9608 |
| 5548 | 5535 | 5488 | 6101 |
| 11468 | 11455 | 11515 | 11376 |
| 10408 | 10395 | 10586 | 10066 |
| 4138 | 4125 | 4857 | 5019 |
| 7748 | 7735 | 7884 | 7588 |
| 13068 | 13055 | 13670 | 11718 |
| 9608 | 9595 | 9680 | 9175 |
| 6458 | 6445 | 6120 | 6928 |
| 8378 | 8365 | 8392 | 8365 |
| 10168 | 10155 | 10024 | 9715 |
| 7848 | 7835 | 7809 | 8519 |
| 8288 | 8275 | 7859 | 8695 |
| 14318 | 14305 | 13544 | 13184 |
| 7088 | 7075 | 7284 | 6941 |
| 8448 | 8435 | 8486 | 8995 |
| 12638 | 12625 | 12553 | 12204 |
| 7108 | 7095 | 7124 | 8016 |
| 11448 | 11435 | 11315 | 10872 |
| 10568 | 10555 | 10577 | 10129 |
| 10108 | 10095 | 9945 | 9535 |
| 5548 | 5535 | 5549 | 5575 |
| 8248 | 8235 | 8130 | 7886 |
| 12108 | 12095 | 12132 | 11862 |
| 4228 | 4215 | 4138 | 4721 |
| 5878 | 5865 | 6230 | 6556 |
| 9368 | 9355 | 9522 | 9442 |
| 9258 | 9245 | 8661 | 8722 |
| 7078 | 7065 | 6909 | 7295 |
| 11638 | 11625 | 11243 | 11344 |
| 10868 | 10855 | 10971 | 10489 |
| 10838 | 10825 | 10576 | 9842 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8798 | 8785 | 8822 | 8634 |
| 7368 | 7355 | 7449 | 7314 |
| 11878 | 11865 | 11877 | 11604 |
| 8328 | 8315 | 8210 | 8985 |
| 11818 | 11805 | 11714 | 11374 |
| 5958 | 5945 | 6176 | 6902 |
| 8788 | 8775 | 8733 | 8868 |
| 8288 | 8275 | 8273 | 8206 |
| 7178 | 7165 | 6898 | 7142 |
| 8258 | 8245 | 8256 | 9202 |
| 7588 | 7575 | 7646 | 8025 |
| 11938 | 11925 | 12167 | 11865 |
| 11978 | 11965 | 11735 | 11441 |
| 7988 | 7975 | 7917 | 7965 |
| 6658 | 6645 | 6617 | 6454 |
| 10808 | 10795 | 11062 | 10529 |
| 11248 | 11235 | 11132 | 10459 |
| 8208 | 8195 | 8274 | 8234 |
| 9368 | 9355 | 9239 | 8395 |
| 7458 | 7445 | 7399 | 8021 |
| 7248 | 7235 | 7040 | 7245 |
| 8948 | 8935 | 8723 | 8398 |
| 4588 | 4575 | 4578 | 4999 |
| 9688 | 9675 | 9901 | 10272 |
| 5748 | 5735 | 5836 | 5462 |
| 5088 | 5075 | 4845 | 5452 |
| 10658 | 10645 | 10705 | 9788 |
| 9718 | 9705 | 9764 | 9592 |
| 8618 | 8605 | 8598 | 9034 |
| 12528 | 12515 | 12666 | 11442 |
| 8178 | 8165 | 7810 | 9241 |
| 6118 | 6105 | 6101 | 5734 |
| 9188 | 9175 | 9215 | 8985 |
| 6718 | 6705 | 6489 | 6416 |
| 9278 | 9265 | 9188 | 9656 |
| 9618 | 9605 | 9640 | 9011 |
| 9938 | 9925 | 10234 | 9788 |
| 7358 | 7345 | 7219 | 7385 |
| 9228 | 9215 | 8924 | 9606 |
| 7798 | 7785 | 7498 | 7084 |
| 8408 | 8395 | 7862 | 8201 |
| 7548 | 7535 | 7517 | 7894 |
| 6578 | 6565 | 6659 | 6212 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8678 | 8665 | 8584 | 8394 |
| 12208 | 12195 | 12365 | 12355 |
| 10208 | 10195 | 10234 | 9715 |
| 10918 | 10905 | 10623 | 11099 |
| 5798 | 5785 | 5762 | 5574 |
| 5238 | 5225 | 5234 | 5756 |
| 6108 | 6095 | 6110 | 5955 |
| 3858 | 3845 | 3840 | 5004 |
| 6248 | 6235 | 6197 | 6375 |
| 9988 | 9975 | 9849 | 10148 |
| 6478 | 6465 | 6033 | 6854 |
| 11038 | 11025 | 10998 | 10066 |
| 6578 | 6565 | 6551 | 6308 |
| 4988 | 4975 | 5125 | 5939 |
| 11708 | 11695 | 11666 | 10549 |
| 8118 | 8105 | 8051 | 7626 |
| 9148 | 9135 | 8762 | 9099 |
| 10618 | 10605 | 10836 | 10644 |
| 10338 | 10325 | 9922 | 9694 |
| 5518 | 5505 | 5938 | 6178 |
| 7448 | 7435 | 7516 | 7339 |
| 8908 | 8895 | 8702 | 8305 |
| 11498 | 11485 | 11489 | 10641 |
| 6008 | 5995 | 6391 | 5438 |
| 7278 | 7265 | 7335 | 6238 |
| 5198 | 5185 | 5370 | 6059 |
| 10708 | 10695 | 10698 | 9975 |
| 13018 | 13005 | 12678 | 12092 |
| 9748 | 9735 | 9517 | 8699 |
| 10438 | 10425 | 10348 | 9965 |
| 11198 | 11185 | 11240 | 9625 |
| 8308 | 8295 | 8185 | 8171 |
| 11938 | 11925 | 12167 | 11865 |
| 7758 | 7745 | 7322 | 8309 |
| 8018 | 8005 | 7812 | 7509 |
| 10368 | 10355 | 10296 | 9556 |
| 10048 | 10035 | 10241 | 9785 |
| 12008 | 11995 | 12192 | 11729 |
| 8678 | 8665 | 8959 | 8228 |
| 10838 | 10825 | 10774 | 10022 |
| 6948 | 6935 | 6944 | 6919 |
| 9438 | 9425 | 9499 | 9315 |
| 9708 | 9695 | 9350 | 8374 |

|  |  |  |  |
| --- | --- | --- | --- |
| 12298 | 12285 | 12347 | 12082 |
| 7418 | 7405 | 7448 | 8106 |
| 11728 | 11715 | 11614 | 11412 |
| 5608 | 5595 | 5601 | 5504 |
| 9318 | 9305 | 9493 | 10112 |
| 8158 | 8145 | 7979 | 7614 |
| 10918 | 10905 | 10939 | 10595 |
| 5668 | 5655 | 5588 | 5302 |
| 6658 | 6645 | 6543 | 6305 |
| 7828 | 7815 | 7344 | 7769 |
| 8208 | 8195 | 8337 | 7968 |
| 9678 | 9665 | 9612 | 9454 |
| 7878 | 7865 | 7716 | 7151 |
| 10458 | 10445 | 10652 | 9871 |
| 11808 | 11795 | 11446 | 10746 |
| 8318 | 8305 | 8271 | 8579 |
| 8898 | 8885 | 8953 | 8634 |
| 8338 | 8325 | 7992 | 7518 |
| 5188 | 5175 | 5509 | 6431 |
| 9658 | 9645 | 9439 | 8792 |
| 8908 | 8895 | 8461 | 8218 |
| 8658 | 8645 | 7964 | 8031 |
| 10148 | 10135 | 9936 | 9398 |
| 7718 | 7705 | 7548 | 7852 |
| 5878 | 5865 | 6230 | 6556 |
| 8418 | 8405 | 8050 | 9225 |
| 5358 | 5345 | 5160 | 6278 |
| 6288 | 6275 | 6538 | 7424 |
| 7288 | 7275 | 7287 | 6546 |
| 7358 | 7345 | 7219 | 7385 |
| 11808 | 11795 | 11726 | 11564 |
| 9918 | 9905 | 9592 | 9365 |
| 5728 | 5715 | 5656 | 5772 |
| 11358 | 11345 | 11364 | 10858 |
| 9548 | 9535 | 8991 | 8229 |
| 7538 | 7525 | 7909 | 7659 |
| 12968 | 12955 | 12963 | 12251 |
| 5358 | 5345 | 5160 | 6278 |
| 8008 | 7995 | 8089 | 8058 |
| 7898 | 7885 | 7702 | 8441 |
| 10238 | 10225 | 9683 | 10082 |
| 8378 | 8365 | 8277 | 8918 |
| 10178 | 10165 | 10389 | 9822 |

|  |  |  |  |
| --- | --- | --- | --- |
| 11948 | 11935 | 11607 | 12076 |
| 10698 | 10685 | 11212 | 9664 |
| 8908 | 8895 | 9023 | 8761 |
| 8988 | 8975 | 8912 | 8476 |
| 8148 | 8135 | 8114 | 7785 |
| 9338 | 9325 | 9112 | 9866 |
| 12178 | 12165 | 11984 | 10585 |
| 9618 | 9605 | 9576 | 9461 |
| 6978 | 6965 | 6965 | 6895 |
| 6088 | 6075 | 6052 | 7176 |
| 5488 | 5475 | 5616 | 5614 |
| 12648 | 12635 | 12554 | 12048 |
| 4118 | 4105 | 4220 | 4754 |
| 5758 | 5745 | 6292 | 7478 |
| 8758 | 8745 | 9183 | 9436 |
| 8848 | 8835 | 8890 | 8882 |
| 6178 | 6165 | 6206 | 6488 |
| 7858 | 7845 | 7804 | 7578 |
| 6558 | 6545 | 6902 | 7232 |
| 7798 | 7785 | 7498 | 7084 |
| 4718 | 4705 | 4965 | 4966 |
| 12878 | 12865 | 13366 | 12474 |
| 8008 | 7995 | 7998 | 8012 |
| 7448 | 7435 | 7410 | 6508 |
| 9208 | 9195 | 9074 | 9396 |
| 6148 | 6135 | 6664 | 7056 |
| 13698 | 13685 | 13654 | 12935 |
| 9058 | 9045 | 8690 | 9156 |
| 10368 | 10355 | 10202 | 10979 |
| 8848 | 8835 | 8482 | 8108 |
| 6848 | 6835 | 6719 | 6952 |
| 5548 | 5535 | 5977 | 6005 |
| 8018 | 8005 | 7967 | 7635 |
| 7418 | 7405 | 7422 | 7384 |
| 8798 | 8785 | 8822 | 8634 |
| 8778 | 8765 | 8812 | 9532 |
| 11638 | 11625 | 11456 | 10912 |
| 9148 | 9135 | 9220 | 8494 |
| 10888 | 10875 | 10767 | 10461 |
| 11308 | 11295 | 11276 | 10044 |
| 7778 | 7765 | 8185 | 8279 |
| 9118 | 9105 | 9126 | 9794 |
| 8328 | 8315 | 8210 | 8985 |

|  |  |  |  |
| --- | --- | --- | --- |
| 6688 | 6675 | 6124 | 7242 |
| 5588 | 5575 | 5427 | 5512 |
| 7408 | 7395 | 7370 | 7212 |
| 6818 | 6805 | 6929 | 6911 |
| 9638 | 9625 | 9471 | 9092 |
| 8068 | 8055 | 7581 | 7762 |
| 6048 | 6035 | 6339 | 6838 |
| 6678 | 6665 | 6902 | 7445 |
| 11218 | 11205 | 11174 | 11486 |
| 9958 | 9945 | 9849 | 10271 |
| 5288 | 5275 | 5323 | 5729 |
| 8978 | 8965 | 8879 | 9299 |
| 11318 | 11305 | 11124 | 11225 |
| 6998 | 6985 | 6955 | 7599 |
| 7318 | 7305 | 7318 | 7009 |
| 10718 | 10705 | 10331 | 10512 |
| 7758 | 7745 | 7310 | 8958 |
| 11158 | 11145 | 10982 | 10488 |
| 11808 | 11795 | 11939 | 11016 |
| 5468 | 5455 | 5481 | 6169 |
| 6428 | 6415 | 6234 | 6539 |
| 9528 | 9515 | 9477 | 9572 |
| 8448 | 8435 | 8379 | 8155 |
| 9558 | 9545 | 9648 | 9816 |
| 8818 | 8805 | 8992 | 9134 |
| 4218 | 4205 | 3702 | 4858 |
| 10618 | 10605 | 10836 | 10644 |
| 9018 | 9005 | 8847 | 8901 |
| 2788 | 2775 | 3013 | 3845 |
| 9718 | 9705 | 9560 | 9638 |
| 12478 | 12465 | 12507 | 12154 |
| 10458 | 10445 | 10747 | 9579 |
| 4758 | 4745 | 4734 | 4994 |
| 11068 | 11055 | 10785 | 10242 |
| 11918 | 11905 | 11536 | 11958 |
| 6478 | 6465 | 6412 | 6781 |
| 10068 | 10055 | 10064 | 9806 |
| 9388 | 9375 | 9031 | 8848 |
| 10918 | 10905 | 10858 | 10631 |
| 9208 | 9195 | 9357 | 9361 |
| 11448 | 11435 | 11378 | 11351 |
| 9488 | 9475 | 9498 | 9255 |
| 9308 | 9295 | 9427 | 9225 |

|  |  |  |  |
| --- | --- | --- | --- |
| 10118 | 10105 | 9863 | 10136 |
| 9098 | 9085 | 8999 | 9984 |
| 5398 | 5385 | 5245 | 6879 |
| 8958 | 8945 | 8511 | 8525 |
| 10508 | 10495 | 10893 | 10439 |
| 8868 | 8855 | 8597 | 8665 |
| 11338 | 11325 | 11592 | 11061 |
| 8558 | 8545 | 8586 | 8846 |
| 8108 | 8095 | 8192 | 8269 |
| 6538 | 6525 | 6685 | 7022 |
| 7078 | 7065 | 7026 | 7041 |
| 9178 | 9165 | 9339 | 9009 |
| 10408 | 10395 | 10345 | 10091 |
| 8468 | 8455 | 8841 | 7212 |
| 10718 | 10705 | 10821 | 10706 |
| 9958 | 9945 | 10032 | 10332 |
| 11628 | 11615 | 11411 | 10806 |
| 7468 | 7455 | 7922 | 6765 |
| 8978 | 8965 | 8912 | 9389 |
| 8378 | 8365 | 8307 | 7932 |
| 5878 | 5865 | 6356 | 6031 |
| 8418 | 8405 | 8050 | 9225 |
| 8458 | 8445 | 8407 | 8599 |
| 9688 | 9675 | 9683 | 9719 |
| 7718 | 7705 | 8117 | 7396 |
| 8068 | 8055 | 7896 | 8875 |
| 7418 | 7405 | 7406 | 7352 |
| 9728 | 9715 | 10193 | 8619 |
| 7568 | 7555 | 7593 | 7736 |
| 9468 | 9455 | 9250 | 8852 |
| 6308 | 6295 | 6481 | 5705 |
| 9468 | 9455 | 9382 | 9095 |
| 7488 | 7475 | 7334 | 7401 |
| 8278 | 8265 | 8379 | 8566 |
| 5918 | 5905 | 5976 | 5896 |
| 7748 | 7735 | 7674 | 8051 |
| 7768 | 7755 | 8059 | 7736 |
| 8728 | 8715 | 8565 | 8568 |
| 5608 | 5595 | 5703 | 6481 |
| 8298 | 8285 | 7851 | 7626 |
| 9688 | 9675 | 9338 | 9735 |
| 7358 | 7345 | 6960 | 7084 |
| 11188 | 11175 | 11044 | 11189 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7888 | 7875 | 8012 | 8665 |
| 6588 | 6575 | 6334 | 7172 |
| 10258 | 10245 | 10140 | 9804 |
| 7798 | 7785 | 7455 | 8105 |
| 4078 | 4065 | 4167 | 4941 |
| 12398 | 12385 | 12334 | 11524 |
| 8218 | 8205 | 8399 | 8824 |
| 12708 | 12695 | 12514 | 11764 |
| 9038 | 9025 | 9162 | 9579 |
| 10158 | 10145 | 9693 | 10608 |
| 6888 | 6875 | 6586 | 6942 |
| 5878 | 5865 | 5890 | 5699 |
| 11668 | 11655 | 11630 | 11179 |
| 11048 | 11035 | 10965 | 10354 |
| 4248 | 4235 | 4702 | 4455 |
| 7188 | 7175 | 7386 | 7948 |
| 10208 | 10195 | 10158 | 9911 |
| 6398 | 6385 | 6510 | 7338 |
| 7238 | 7225 | 7206 | 8101 |
| 9498 | 9485 | 9408 | 7971 |
| 12508 | 12495 | 12243 | 12276 |
| 8618 | 8605 | 8598 | 9034 |
| 4848 | 4835 | 4649 | 5286 |
| 6278 | 6265 | 5931 | 6879 |
| 6558 | 6545 | 6610 | 7578 |
| 8178 | 8165 | 8225 | 7575 |
| 9508 | 9495 | 9401 | 8252 |
| 8968 | 8955 | 8754 | 8641 |
| 5548 | 5535 | 5705 | 6925 |
| 5998 | 5985 | 6088 | 6335 |
| 6908 | 6895 | 6792 | 6885 |
| 8428 | 8415 | 8481 | 9062 |
| 10188 | 10175 | 9746 | 9355 |
| 11508 | 11495 | 11415 | 10541 |
| 9438 | 9425 | 9600 | 7429 |
| 9228 | 9215 | 8930 | 9286 |
| 8438 | 8425 | 8559 | 8244 |
| 7878 | 7865 | 7917 | 7871 |
| 9368 | 9355 | 9414 | 9248 |
| 4588 | 4575 | 4514 | 5344 |
| 9118 | 9105 | 9126 | 9794 |
| 4888 | 4875 | 4896 | 5274 |
| 8348 | 8335 | 8282 | 8598 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7868 | 7855 | 7775 | 7726 |
| 6868 | 6855 | 6904 | 6502 |
| 8698 | 8685 | 8434 | 9048 |
| 7078 | 7065 | 7505 | 8128 |
| 5758 | 5745 | 5897 | 5999 |
| 8558 | 8545 | 8540 | 8428 |
| 9018 | 9005 | 8505 | 8265 |
| 4578 | 4565 | 4878 | 4705 |
| 3348 | 3335 | 3675 | 4252 |
| 7708 | 7695 | 7462 | 7954 |
| 5928 | 5915 | 5945 | 5676 |
| 9958 | 9945 | 9891 | 10951 |
| 7378 | 7365 | 7172 | 7829 |
| 8068 | 8055 | 8132 | 9349 |
| 7978 | 7965 | 7652 | 7994 |
| 6388 | 6375 | 6053 | 6085 |
| 6748 | 6735 | 6853 | 7474 |
| 8608 | 8595 | 8633 | 8769 |
| 7948 | 7935 | 7866 | 7684 |
| 8008 | 7995 | 8048 | 8345 |
| 7278 | 7265 | 7673 | 7464 |
| 4808 | 4795 | 4930 | 5305 |
| 9248 | 9235 | 9201 | 8804 |
| 9228 | 9215 | 8924 | 9606 |
| 10678 | 10665 | 10377 | 10776 |
| 10878 | 10865 | 11181 | 10574 |
| 8158 | 8145 | 7836 | 8684 |
| 7188 | 7175 | 6974 | 7472 |
| 7058 | 7045 | 7174 | 7211 |
| 8598 | 8585 | 8532 | 8305 |
| 8978 | 8965 | 9151 | 8761 |
| 5028 | 5015 | 4866 | 5118 |
| 4858 | 4845 | 4849 | 5802 |
| 8448 | 8435 | 8538 | 7841 |
| 11668 | 11655 | 11807 | 11879 |
| 7128 | 7115 | 7385 | 6988 |
| 10358 | 10345 | 10326 | 9998 |
| 5988 | 5975 | 5999 | 6628 |
| 11638 | 11625 | 11729 | 10894 |
| 8628 | 8615 | 8920 | 8628 |
| 12578 | 12565 | 12518 | 12049 |
| 11568 | 11555 | 11652 | 11446 |
| 9258 | 9245 | 9584 | 9092 |

|  |  |  |  |
| --- | --- | --- | --- |
| 10988 | 10975 | 11331 | 10454 |
| 12148 | 12135 | 12300 | 10759 |
| 11638 | 11625 | 11581 | 11408 |
| 7918 | 7905 | 7897 | 8194 |
| 10208 | 10195 | 10191 | 10004 |
| 11318 | 11305 | 11289 | 11299 |
| 8828 | 8815 | 9223 | 9362 |
| 6878 | 6865 | 7023 | 7042 |
| 13338 | 13325 | 13080 | 11828 |
| 6808 | 6795 | 6824 | 6209 |
| 10028 | 10015 | 10267 | 9204 |
| 7258 | 7245 | 7053 | 6792 |
| 11308 | 11295 | 11365 | 11101 |
| 7008 | 6995 | 7194 | 7685 |
| 5898 | 5885 | 6029 | 6216 |
| 8148 | 8135 | 7562 | 7919 |
| 6918 | 6905 | 6763 | 6365 |
| 5508 | 5495 | 5489 | 6025 |
| 8408 | 8395 | 8576 | 8926 |
| 7918 | 7905 | 7898 | 8755 |
| 7818 | 7805 | 7459 | 7354 |
| 8868 | 8855 | 9110 | 8875 |
| 10978 | 10965 | 10907 | 10468 |
| 12838 | 12825 | 12855 | 11736 |
| 8258 | 8245 | 8728 | 7909 |
| 6648 | 6635 | 6750 | 6914 |
| 7648 | 7635 | 7901 | 8528 |
| 9968 | 9955 | 9928 | 9738 |
| 7558 | 7545 | 7843 | 8706 |
| 8678 | 8665 | 8277 | 8212 |
| 8158 | 8145 | 8272 | 8618 |
| 7218 | 7205 | 7315 | 7681 |
| 8878 | 8865 | 9060 | 8782 |
| 6408 | 6395 | 6577 | 6506 |
| 8148 | 8135 | 8456 | 7654 |
| 8198 | 8185 | 8022 | 7588 |
| 9788 | 9775 | 9591 | 9336 |
| 9638 | 9625 | 9967 | 9535 |
| 5158 | 5145 | 5122 | 5531 |
| 5508 | 5495 | 5600 | 6338 |
| 6858 | 6845 | 7071 | 7075 |
| 9538 | 9525 | 9373 | 9662 |
| 7928 | 7915 | 8368 | 8508 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7098 | 7085 | 7073 | 6581 |
| 6438 | 6425 | 6407 | 6496 |
| 7068 | 7055 | 7483 | 7838 |
| 8308 | 8295 | 8402 | 9176 |
| 9748 | 9735 | 9700 | 9552 |
| 8628 | 8615 | 8504 | 8421 |
| 11078 | 11065 | 11120 | 11344 |
| 5638 | 5625 | 6139 | 6858 |
| 8278 | 8265 | 8473 | 7948 |
| 9378 | 9365 | 9386 | 8561 |
| 6208 | 6195 | 6362 | 6229 |
| 7548 | 7535 | 7679 | 7276 |
| 9268 | 9255 | 9366 | 9074 |
| 9498 | 9485 | 9732 | 9076 |
| 10208 | 10195 | 10373 | 9022 |
| 6828 | 6815 | 6329 | 6869 |
| 8098 | 8085 | 7907 | 8785 |
| 7568 | 7555 | 7369 | 6694 |
| 9538 | 9525 | 9915 | 9946 |
| 5328 | 5315 | 5204 | 5985 |
| 11148 | 11135 | 10746 | 10564 |
| 7338 | 7325 | 7180 | 7105 |
| 8568 | 8555 | 8594 | 8402 |
| 10948 | 10935 | 10760 | 10098 |
| 8568 | 8555 | 8594 | 8402 |
| 7428 | 7415 | 7236 | 8419 |
| 9598 | 9585 | 9618 | 9255 |
| 5968 | 5955 | 5903 | 6848 |
| 5748 | 5735 | 5719 | 5641 |
| 9128 | 9115 | 9142 | 8871 |
| 5418 | 5405 | 5818 | 6069 |
| 10918 | 10905 | 10939 | 10595 |
| 8278 | 8265 | 8128 | 7759 |
| 9858 | 9845 | 9804 | 10219 |
| 10628 | 10615 | 10477 | 10506 |
| 8698 | 8685 | 8434 | 9048 |
| 7018 | 7005 | 7087 | 7105 |
| 8288 | 8275 | 8273 | 8206 |
| 6958 | 6945 | 6755 | 7812 |
| 8398 | 8385 | 8998 | 8852 |
| 9188 | 9175 | 8902 | 8678 |
| 9878 | 9865 | 9551 | 9934 |
| 9248 | 9235 | 9020 | 9886 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8728 | 8715 | 8723 | 8884 |
| 10348 | 10335 | 10199 | 9835 |
| 8708 | 8695 | 8739 | 8041 |
| 5888 | 5875 | 5909 | 5525 |
| 10148 | 10135 | 10306 | 10268 |
| 5688 | 5675 | 5366 | 5914 |
| 12018 | 12005 | 11788 | 11158 |
| 3618 | 3605 | 3356 | 4806 |
| 4118 | 4105 | 3963 | 4568 |
| 9708 | 9695 | 9185 | 9398 |
| 12608 | 12595 | 12573 | 11849 |
| 13818 | 13805 | 13926 | 12858 |
| 6418 | 6405 | 6071 | 6278 |
| 12718 | 12705 | 12389 | 12412 |
| 12248 | 12235 | 12009 | 11439 |
| 5948 | 5935 | 5918 | 5918 |
| 8448 | 8435 | 8471 | 8366 |
| 9068 | 9055 | 9151 | 8185 |
| 8278 | 8265 | 8194 | 8581 |
| 9688 | 9675 | 9451 | 9018 |
| 11088 | 11075 | 11214 | 10622 |
| 4178 | 4165 | 4579 | 4775 |
| 7448 | 7435 | 7338 | 7765 |
| 9418 | 9405 | 9235 | 9144 |
| 9478 | 9465 | 9700 | 9788 |
| 7918 | 7905 | 7171 | 7482 |
| 5318 | 5305 | 4991 | 5426 |
| 9068 | 9055 | 9151 | 8185 |
| 10718 | 10705 | 9807 | 10604 |
| 9888 | 9875 | 9706 | 9552 |
| 6948 | 6935 | 6837 | 6329 |
| 10848 | 10835 | 10726 | 10166 |
| 7338 | 7325 | 7362 | 7511 |
| 13068 | 13055 | 13196 | 11994 |
| 6108 | 6095 | 6337 | 6489 |
| 11838 | 11825 | 11992 | 11391 |
| 11908 | 11895 | 11667 | 11164 |
| 7558 | 7545 | 7843 | 8706 |
| 7548 | 7535 | 7517 | 7894 |
| 5378 | 5365 | 5648 | 5551 |
| 2688 | 2675 | 2717 | 3035 |
| 13468 | 13455 | 13439 | 12765 |
| 10228 | 10215 | 10240 | 10384 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7998 | 7985 | 8044 | 7705 |
| 6828 | 6815 | 6967 | 6734 |
| 12238 | 12225 | 12270 | 12549 |
| 8578 | 8565 | 8764 | 8519 |
| 8008 | 7995 | 7998 | 8012 |
| 12078 | 12065 | 12126 | 11115 |
| 9918 | 9905 | 10136 | 9441 |
| 9298 | 9285 | 9648 | 8861 |
| 10288 | 10275 | 9898 | 9875 |
| 7498 | 7485 | 7493 | 7742 |
| 9338 | 9325 | 9448 | 9304 |
| 4318 | 4305 | 4505 | 4929 |
| 5598 | 5585 | 5436 | 5696 |
| 10918 | 10905 | 10447 | 11008 |
| 6018 | 6005 | 5991 | 6366 |
| 7568 | 7555 | 7546 | 7569 |
| 4518 | 4505 | 4713 | 4772 |
| 7918 | 7905 | 7897 | 8194 |
| 5728 | 5715 | 5685 | 5878 |
| 10118 | 10105 | 10248 | 10535 |
| 8808 | 8795 | 8487 | 8019 |
| 11258 | 11245 | 11110 | 11086 |
| 8618 | 8605 | 8547 | 8845 |
| 8488 | 8475 | 8413 | 8779 |
| 7658 | 7645 | 7591 | 7415 |
| 6258 | 6245 | 6714 | 6795 |
| 6928 | 6915 | 7250 | 6894 |
| 8668 | 8655 | 9026 | 9264 |
| 11808 | 11795 | 11726 | 11564 |
| 7838 | 7825 | 7902 | 7511 |
| 8828 | 8815 | 8820 | 8512 |
| 9948 | 9935 | 9469 | 10884 |
| 9218 | 9205 | 8842 | 9651 |
| 11518 | 11505 | 11516 | 11221 |
| 9688 | 9675 | 9676 | 10352 |
| 9488 | 9475 | 9511 | 8921 |
| 10048 | 10035 | 10241 | 9785 |
| 9338 | 9325 | 9378 | 7954 |
| 6688 | 6675 | 6931 | 7265 |
| 10308 | 10295 | 10540 | 10412 |
| 6408 | 6395 | 6182 | 6102 |
| 10488 | 10475 | 10641 | 10495 |
| 6878 | 6865 | 7080 | 7504 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8388 | 8375 | 8156 | 9192 |
| 5898 | 5885 | 5832 | 6088 |
| 7758 | 7745 | 7310 | 8958 |
| 9868 | 9855 | 9930 | 8936 |
| 9608 | 9595 | 9422 | 9839 |
| 11858 | 11845 | 11772 | 11556 |
| 7578 | 7565 | 7660 | 8431 |
| 8378 | 8365 | 8392 | 8365 |
| 3928 | 3915 | 3852 | 4696 |
| 8888 | 8875 | 8427 | 9551 |
| 8508 | 8495 | 8648 | 8599 |
| 7468 | 7455 | 7265 | 6975 |
| 9728 | 9715 | 9590 | 9685 |
| 11268 | 11255 | 11183 | 10802 |
| 2918 | 2905 | 3248 | 3886 |
| 9418 | 9405 | 9235 | 9144 |
| 12778 | 12765 | 12856 | 11191 |
| 9918 | 9905 | 10197 | 10278 |
| 6708 | 6695 | 7033 | 7438 |
| 5728 | 5715 | 5685 | 5878 |
| 12118 | 12105 | 12133 | 11724 |
| 12478 | 12465 | 12717 | 11691 |
| 7018 | 7005 | 7349 | 7424 |
| 8698 | 8685 | 8703 | 8184 |
| 9228 | 9215 | 9540 | 8218 |
| 8988 | 8975 | 9533 | 9724 |
| 9748 | 9735 | 9592 | 8796 |
| 7878 | 7865 | 7924 | 7244 |
| 9308 | 9295 | 9332 | 9215 |
| 10778 | 10765 | 10656 | 10516 |
| 7858 | 7845 | 7765 | 7665 |
| 8378 | 8365 | 8182 | 8655 |
| 8718 | 8705 | 8469 | 8674 |
| 6588 | 6575 | 6802 | 6809 |
| 11988 | 11975 | 12052 | 11456 |
| 8878 | 8865 | 8849 | 8179 |
| 8178 | 8165 | 8068 | 7965 |
| 7778 | 7765 | 7845 | 8074 |
| 7848 | 7835 | 7764 | 7349 |
| 9438 | 9425 | 9623 | 8975 |
| 7398 | 7385 | 7471 | 7914 |
| 6608 | 6595 | 6597 | 7359 |
| 5778 | 5765 | 6057 | 6271 |

|  |  |  |  |
| --- | --- | --- | --- |
| 5818 | 5805 | 5602 | 6264 |
| 10818 | 10805 | 10881 | 10728 |
| 10348 | 10335 | 9921 | 10428 |
| 7038 | 7025 | 7173 | 7992 |
| 12078 | 12065 | 11977 | 11726 |
| 8818 | 8805 | 8712 | 8984 |
| 8068 | 8055 | 8498 | 8896 |
| 11208 | 11195 | 11134 | 10694 |
| 5658 | 5645 | 5693 | 5759 |
| 9488 | 9475 | 9097 | 8642 |
| 6358 | 6345 | 6692 | 7411 |
| 12898 | 12885 | 12893 | 12256 |
| 7688 | 7675 | 8325 | 7591 |
| 9118 | 9105 | 9185 | 9318 |
| 10488 | 10475 | 10485 | 10346 |
| 6238 | 6225 | 6116 | 6375 |
| 6988 | 6975 | 6908 | 7425 |
| 6088 | 6075 | 6012 | 6522 |
| 8118 | 8105 | 7939 | 8411 |
| 9808 | 9795 | 9640 | 8548 |
| 7088 | 7075 | 7110 | 7254 |
| 3778 | 3765 | 4091 | 4496 |
| 8218 | 8205 | 8367 | 7318 |
| 8478 | 8465 | 8744 | 7954 |
| 5598 | 5585 | 5632 | 6266 |
| 11008 | 10995 | 10810 | 10104 |
| 9068 | 9055 | 9170 | 9905 |
| 7378 | 7365 | 7452 | 7722 |
| 4988 | 4975 | 5349 | 6174 |
| 6468 | 6455 | 6456 | 6356 |
| 11248 | 11235 | 11002 | 10646 |
| 8278 | 8265 | 8471 | 8631 |
| 12118 | 12105 | 12133 | 11724 |
| 7308 | 7295 | 7254 | 7845 |
| 9748 | 9735 | 9715 | 9536 |
| 9228 | 9215 | 9540 | 8218 |
| 9648 | 9635 | 9513 | 9905 |
| 4768 | 4755 | 4506 | 6062 |
| 4608 | 4595 | 4599 | 5271 |
| 11708 | 11695 | 11715 | 11951 |
| 9158 | 9145 | 8908 | 9291 |
| 8308 | 8295 | 7854 | 8281 |
| 10258 | 10245 | 10054 | 10192 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8218 | 8205 | 7799 | 7756 |
| 10478 | 10465 | 10450 | 9996 |
| 3188 | 3175 | 3496 | 3739 |
| 7648 | 7635 | 7901 | 8528 |
| 9658 | 9645 | 9645 | 10141 |
| 7428 | 7415 | 7515 | 7398 |
| 6918 | 6905 | 6933 | 6976 |
| 8758 | 8745 | 8510 | 9605 |
| 9108 | 9095 | 9383 | 8635 |
| 11038 | 11025 | 10998 | 10066 |
| 11898 | 11885 | 11883 | 11839 |
| 4868 | 4855 | 4927 | 5148 |
| 7368 | 7355 | 7323 | 6969 |
| 11638 | 11625 | 11800 | 11269 |
| 8078 | 8065 | 8145 | 8165 |
| 8978 | 8965 | 8823 | 8605 |
| 10108 | 10095 | 9661 | 10689 |
| 5128 | 5115 | 5130 | 6049 |
| 10828 | 10815 | 10447 | 10094 |
| 11078 | 11065 | 11376 | 10758 |
| 7238 | 7225 | 7051 | 7054 |
| 8568 | 8555 | 8305 | 8456 |
| 8108 | 8095 | 8072 | 7572 |
| 6828 | 6815 | 6608 | 6719 |
| 8488 | 8475 | 8413 | 8779 |
| 9188 | 9175 | 9621 | 9308 |
| 3958 | 3945 | 3514 | 4285 |
| 10308 | 10295 | 10540 | 10412 |
| 10408 | 10395 | 10422 | 9999 |
| 12548 | 12535 | 12259 | 12402 |
| 8748 | 8735 | 9034 | 9132 |
| 4548 | 4535 | 4520 | 5022 |
| 10418 | 10405 | 10015 | 10469 |
| 5148 | 5135 | 5126 | 5018 |
| 9148 | 9135 | 9060 | 9584 |
| 6818 | 6805 | 6865 | 8245 |
| 9008 | 8995 | 9218 | 9042 |
| 5088 | 5075 | 4845 | 5452 |
| 8988 | 8975 | 8694 | 9285 |
| 6118 | 6105 | 6369 | 6041 |
| 9218 | 9205 | 9513 | 8685 |
| 7448 | 7435 | 7516 | 7339 |
| 5418 | 5405 | 5378 | 6836 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7978 | 7965 | 8003 | 7649 |
| 8548 | 8535 | 8509 | 8694 |
| 7948 | 7935 | 7921 | 7934 |
| 10898 | 10885 | 10603 | 10941 |
| 10168 | 10155 | 10294 | 9872 |
| 11768 | 11755 | 12066 | 11418 |
| 10228 | 10215 | 9979 | 9462 |
| 3428 | 3415 | 3658 | 4686 |
| 9668 | 9655 | 9800 | 8916 |
| 13068 | 13055 | 12909 | 12035 |
| 8778 | 8765 | 9129 | 8225 |
| 7778 | 7765 | 7670 | 7259 |
| 9378 | 9365 | 9278 | 8318 |
| 8218 | 8205 | 8399 | 8824 |
| 9988 | 9975 | 9769 | 9816 |
| 10618 | 10605 | 10654 | 9614 |
| 8738 | 8725 | 8946 | 9149 |
| 8538 | 8525 | 8591 | 8509 |
| 3228 | 3215 | 3704 | 4604 |
| 5188 | 5175 | 5509 | 6431 |
| 7778 | 7765 | 8011 | 7519 |
| 9068 | 9055 | 8992 | 8724 |
| 11548 | 11535 | 11249 | 11296 |
| 8218 | 8205 | 8150 | 8596 |
| 11208 | 11195 | 11497 | 9784 |
| 11668 | 11655 | 12080 | 11122 |
| 6848 | 6835 | 6849 | 6744 |
| 3648 | 3635 | 3860 | 4299 |
| 9688 | 9675 | 9676 | 10352 |
| 10318 | 10305 | 10271 | 9372 |
| 9338 | 9325 | 9448 | 9304 |
| 8208 | 8195 | 8236 | 8198 |
| 10378 | 10365 | 10326 | 9786 |
| 9748 | 9735 | 9353 | 9402 |
| 10028 | 10015 | 9934 | 10055 |
| 10228 | 10215 | 10240 | 10384 |
| 11218 | 11205 | 11099 | 10744 |
| 10748 | 10735 | 10639 | 10609 |
| 5948 | 5935 | 5709 | 5775 |
| 10088 | 10075 | 9841 | 9659 |
| 11928 | 11915 | 11826 | 11466 |
| 8688 | 8675 | 8417 | 9338 |
| 13938 | 13925 | 13919 | 13389 |

|  |  |  |  |
| --- | --- | --- | --- |
| 9788 | 9775 | 9755 | 8994 |
| 5588 | 5575 | 6008 | 5814 |
| 11348 | 11335 | 11397 | 11436 |
| 11078 | 11065 | 11120 | 11344 |
| 9748 | 9735 | 9209 | 9284 |
| 4458 | 4445 | 4804 | 5222 |
| 12298 | 12285 | 12347 | 12082 |
| 6968 | 6955 | 6754 | 6412 |
| 10468 | 10455 | 10343 | 10301 |
| 11488 | 11475 | 11204 | 10552 |
| 11118 | 11105 | 11025 | 10358 |
| 9438 | 9425 | 9319 | 9192 |
| 8098 | 8085 | 7907 | 8785 |
| 14018 | 14005 | 13234 | 13196 |
| 7658 | 7645 | 7604 | 7446 |
| 10358 | 10345 | 10333 | 10955 |
| 7138 | 7125 | 7163 | 6595 |
| 8178 | 8165 | 8174 | 8482 |
| 7308 | 7295 | 7236 | 7026 |
| 7218 | 7205 | 7010 | 7465 |
| 9078 | 9065 | 8526 | 8964 |
| 9888 | 9875 | 9762 | 9809 |
| 6778 | 6765 | 6851 | 6528 |
| 7748 | 7735 | 7335 | 7858 |
| 10428 | 10415 | 9831 | 10246 |
| 9568 | 9555 | 9438 | 8978 |
| 6158 | 6145 | 6297 | 7626 |
| 6328 | 6315 | 6396 | 6411 |
| 14248 | 14235 | 14075 | 13596 |
| 9158 | 9145 | 8573 | 10092 |
| 8498 | 8485 | 8737 | 8199 |
| 8898 | 8885 | 8953 | 8634 |
| 8398 | 8385 | 8998 | 8852 |
| 6408 | 6395 | 6492 | 6102 |
| 5608 | 5595 | 5793 | 5866 |
| 5638 | 5625 | 6139 | 6858 |
| 9188 | 9175 | 9146 | 9779 |
| 7988 | 7975 | 8484 | 7701 |
| 8368 | 8355 | 8226 | 7612 |
| 10418 | 10405 | 10312 | 9834 |
| 9008 | 8995 | 9218 | 9042 |
| 11618 | 11605 | 11435 | 11075 |
| 5728 | 5715 | 5773 | 6054 |

|  |  |  |  |
| --- | --- | --- | --- |
| 7128 | 7115 | 7385 | 6988 |
| 8748 | 8735 | 8400 | 8089 |
| 7018 | 7005 | 6997 | 6884 |
| 6358 | 6345 | 6029 | 6212 |
| 12258 | 12245 | 12353 | 11324 |
| 5888 | 5875 | 5922 | 5559 |
| 11268 | 11255 | 11282 | 10502 |
| 10478 | 10465 | 10385 | 10045 |
| 8298 | 8285 | 8560 | 7844 |
| 11748 | 11735 | 11426 | 10912 |
| 11428 | 11415 | 11420 | 11031 |
| 11678 | 11665 | 11631 | 10598 |
| 9148 | 9135 | 9153 | 9025 |
| 10178 | 10165 | 10306 | 10042 |
| 7068 | 7055 | 6957 | 7286 |
| 8538 | 8525 | 8630 | 8316 |
| 7858 | 7845 | 7796 | 8104 |
| 6848 | 6835 | 6714 | 7075 |
| 12238 | 12225 | 11990 | 12349 |
| 7398 | 7385 | 7402 | 7554 |
| 10018 | 10005 | 9905 | 9629 |
| 7358 | 7345 | 7390 | 8376 |
| 3418 | 3405 | 3327 | 4138 |
| 9658 | 9645 | 9842 | 10221 |
| 7758 | 7745 | 7848 | 8802 |
| 8278 | 8265 | 8585 | 8528 |
| 9508 | 9495 | 9486 | 9858 |
| 6158 | 6145 | 6038 | 6472 |
| 10258 | 10245 | 10310 | 10518 |
| 6818 | 6805 | 7010 | 7211 |
| 10978 | 10965 | 11224 | 11141 |
| 11128 | 11115 | 11079 | 11394 |
| 6448 | 6435 | 6222 | 6404 |
| 11978 | 11965 | 11945 | 11632 |
| 10058 | 10045 | 9731 | 9654 |
| 5188 | 5175 | 5156 | 4689 |
| 9548 | 9535 | 9501 | 9124 |
| 7338 | 7325 | 7251 | 7655 |
| 4818 | 4805 | 4683 | 5955 |
| 12078 | 12065 | 11977 | 11726 |
| 8278 | 8265 | 8212 | 8099 |
| 8888 | 8875 | 8761 | 9054 |
| 11238 | 11225 | 10995 | 10962 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8548 | 8535 | 8806 | 8484 |
| 9658 | 9645 | 9353 | 9602 |
| 9288 | 9275 | 9174 | 9541 |
| 5678 | 5665 | 5752 | 6062 |
| 7328 | 7315 | 7264 | 7508 |
| 6028 | 6015 | 5968 | 6501 |
| 7978 | 7965 | 8362 | 8786 |
| 5638 | 5625 | 6139 | 6858 |
| 12158 | 12145 | 11703 | 11098 |
| 9048 | 9035 | 9154 | 9795 |
| 13458 | 13445 | 13561 | 12705 |
| 5728 | 5715 | 5773 | 6054 |
| 6018 | 6005 | 6034 | 6286 |
| 5378 | 5365 | 5362 | 5432 |
| 3418 | 3405 | 3327 | 4138 |
| 10188 | 10175 | 9500 | 10965 |
| 8578 | 8565 | 8424 | 8342 |
| 11248 | 11235 | 11305 | 10519 |
| 4638 | 4625 | 4621 | 5149 |
| 6038 | 6025 | 6040 | 5548 |
| 7378 | 7365 | 7162 | 7406 |
| 6638 | 6625 | 7085 | 7408 |
| 9678 | 9665 | 9592 | 9339 |
| 11428 | 11415 | 11394 | 10478 |
| 7868 | 7855 | 8350 | 7882 |
| 5718 | 5705 | 5655 | 6459 |
| 7568 | 7555 | 7534 | 7485 |
| 9688 | 9675 | 9530 | 9875 |
| 5488 | 5475 | 5680 | 5928 |
| 6468 | 6455 | 6456 | 6356 |
| 8098 | 8085 | 7856 | 8141 |
| 9608 | 9595 | 9946 | 9966 |
| 8408 | 8395 | 8455 | 8851 |
| 6988 | 6975 | 6984 | 7018 |
| 6808 | 6795 | 6965 | 6022 |
| 10298 | 10285 | 10111 | 10196 |
| 11458 | 11445 | 11786 | 10896 |
| 5168 | 5155 | 5232 | 5831 |
| 8688 | 8675 | 8645 | 8529 |
| 12838 | 12825 | 12610 | 12086 |
| 3488 | 3475 | 3599 | 3744 |
| 10698 | 10685 | 10825 | 9998 |
| 11718 | 11705 | 11776 | 10692 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8308 | 8295 | 8784 | 8312 |
| 7288 | 7275 | 7418 | 7562 |
| 9448 | 9435 | 9422 | 9321 |
| 10138 | 10125 | 9966 | 9439 |
| 6618 | 6605 | 6459 | 6484 |
| 9818 | 9805 | 9333 | 8518 |
| 11518 | 11505 | 11516 | 11221 |
| 7588 | 7575 | 7722 | 6052 |
| 7938 | 7925 | 8036 | 8718 |
| 10018 | 10005 | 9953 | 8788 |
| 9238 | 9225 | 8982 | 8929 |
| 4388 | 4375 | 4581 | 5204 |
| 8118 | 8105 | 8182 | 7186 |
| 12718 | 12705 | 12340 | 11726 |
| 4428 | 4415 | 4399 | 4934 |
| 7588 | 7575 | 7665 | 8425 |
| 10248 | 10235 | 10070 | 10175 |
| 7618 | 7605 | 7512 | 7101 |
| 10678 | 10665 | 10271 | 10875 |
| 9568 | 9555 | 9923 | 9324 |
| 6918 | 6905 | 6879 | 6595 |
| 7058 | 7045 | 7174 | 7211 |
| 5008 | 4995 | 5050 | 5528 |
| 8088 | 8075 | 8098 | 8829 |
| 9278 | 9265 | 9379 | 10214 |
| 8768 | 8755 | 9000 | 8658 |
| 8178 | 8165 | 8257 | 8376 |
| 13448 | 13435 | 13189 | 11756 |
| 7778 | 7765 | 7842 | 8488 |
| 9148 | 9135 | 9168 | 9215 |
| 8808 | 8795 | 9254 | 8334 |
| 5918 | 5905 | 5747 | 6026 |
| 9558 | 9545 | 9595 | 8288 |
| 8568 | 8555 | 8554 | 8006 |
| 8938 | 8925 | 8949 | 8222 |
| 9818 | 9805 | 9876 | 9571 |
| 7178 | 7165 | 6832 | 6995 |
| 9888 | 9875 | 9762 | 9801 |
| 9778 | 9765 | 10034 | 9746 |
| 5088 | 5075 | 5087 | 5246 |
| 11508 | 11495 | 11663 | 10579 |
| 5748 | 5735 | 5717 | 6198 |
| 6908 | 6895 | 7174 | 7739 |

|  |  |  |  |
| --- | --- | --- | --- |
| 11228 | 11215 | 10820 | 10851 |
| 12708 | 12695 | 12686 | 12368 |
| 13458 | 13445 | 13123 | 12496 |
| 12468 | 12455 | 12613 | 11849 |
| 8778 | 8765 | 8812 | 9532 |
| 7448 | 7435 | 7410 | 6508 |
| 9818 | 9805 | 9608 | 9308 |
| 10808 | 10795 | 10935 | 10126 |
| 7878 | 7865 | 7956 | 8361 |
| 14008 | 13995 | 14029 | 13082 |
| 8128 | 8115 | 8151 | 8124 |
| 8608 | 8595 | 8633 | 8769 |
| 10818 | 10805 | 10910 | 10941 |
| 11408 | 11395 | 11606 | 11448 |
| 5428 | 5415 | 5706 | 6518 |
| 5358 | 5345 | 5560 | 5609 |
| 9868 | 9855 | 9930 | 8936 |
| 6908 | 6895 | 7375 | 6631 |
| 11068 | 11055 | 11206 | 10841 |
| 12848 | 12835 | 12856 | 12435 |
| 8738 | 8725 | 8673 | 8152 |
| 7368 | 7355 | 7284 | 6539 |
| 5598 | 5585 | 5500 | 5891 |
| 8228 | 8215 | 8224 | 7795 |
| 13838 | 13825 | 13398 | 12968 |
| 9128 | 9115 | 9095 | 9022 |
| 9888 | 9875 | 9706 | 9552 |
| 5718 | 5705 | 5655 | 6459 |
| 6178 | 6165 | 6311 | 6959 |
| 10468 | 10455 | 10611 | 10535 |
| 13188 | 13175 | 12727 | 11761 |
| 12128 | 12115 | 11993 | 11619 |
| 9668 | 9655 | 9701 | 9938 |
| 9578 | 9565 | 9518 | 9261 |
| 8778 | 8765 | 8859 | 8609 |
| 8738 | 8725 | 8810 | 8169 |
| 11448 | 11435 | 11315 | 10872 |
| 8098 | 8085 | 8275 | 9079 |
| 6438 | 6425 | 6562 | 7061 |
| 5018 | 5005 | 5324 | 5131 |
| 12478 | 12465 | 12717 | 11691 |
| 10548 | 10535 | 10377 | 9928 |
| 8348 | 8335 | 8352 | 8391 |

|  |  |  |  |
| --- | --- | --- | --- |
| 5038 | 5025 | 5064 | 4984 |
| 9568 | 9555 | 9555 | 10448 |
| 5468 | 5455 | 5357 | 5554 |
| 2748 | 2735 | 2874 | 3938 |
| 7618 | 7605 | 7536 | 7545 |
| 10858 | 10845 | 10606 | 10574 |
| 6668 | 6655 | 7007 | 7152 |
| 10068 | 10055 | 9978 | 9569 |
| 11778 | 11765 | 11335 | 10749 |
| 9788 | 9775 | 10094 | 10761 |
| 5488 | 5475 | 5680 | 5928 |
| 10088 | 10075 | 9820 | 10142 |
| 9788 | 9775 | 9755 | 8994 |
| 9388 | 9375 | 8834 | 8924 |
| 8528 | 8515 | 8616 | 8754 |
| 6118 | 6105 | 6243 | 6995 |

See full data: https://docs.google.com/spreadsheets/d/1CGmoXeZMpy2je7CHTKNC7FlXaGzPCWVj/ edit?usp=share\_link&ouid=118279275388236062240&rtpof=true&sd=true

# Code

**IMPORTING NECESSARY PACKAGES**

**from** scipy.spatial **import** distance

**import** pandas **as** pd

**import** numpy **as** np

**from** sklearn.linear\_model **import** LinearRegression

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.metrics **import** r2\_score, mean\_squared\_error, mean\_absolute\_error

**import** seaborn **as** sns

**from** math **import** sqrt

**import** math

# IMPORTING TRAINED DATASET

df = pd.read\_csv(r'/content/drive/MyDrive/Colab Notebooks/Copy of msc\_trainin g\_dataset.csv')

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| df | room bathroom kitchen | french\_door | backyard furnished | green\_paint |
| \  0 | 3 1 2 | 1 | 1 0 | 1 |
| 1 | 5 2 2 | 2 | 1 0 | 0 |
| 2 | 5 2 2 | 2 | 1 0 | 0 |
| 3 | 1 2 1 | 2 | 0 0 | 0 |
| 4 | 2 1 2 | 3 | 1 1 | 0 |
| ... | ... ... ... | ... | ... ... | ... |
| 2995 | 1 1 2 | 1 | 1 0 | 0 |
| 2996 | 1 1 2 | 3 | 0 1 | 1 |
| 2997 | 2 2 1 | 2 | 0 0 | 1 |
| 2998 | 4 1 1 | 2 | 1 1 | 1 |
| 2999 | 4 2 1 | 1 | 0 0 | 0 |
|  | solar\_power woodfloor | qlm\_security | club\_access price |  |
| 0 | 0 0 | 1 | 1 6835 |  |
| 1 | 0 0 | 1 | 1 9005 |  |
| 2 | 0 0 | 1 | 1 9005 |  |
| 3 | 0 1 | 1 | 0 5105 |  |
| 4  ... 2995 | 0 1  ... ...  1 0 | 1  ...  0 | 0 9105  ... ...  0 4825 |  |
| 2996 | 0 0 | 1 | 1 6755 |  |
| 2997 | 1 1 | 0 | 0 7565 |  |
| 2998 | 0 0 | 0 | 1 9135 |  |
| 2999 | 1 1 | 0 | 0 8955 |  |
| [3000 | rows x 12 columns] |  |  |  |

trained\_data = pd.DataFrame(df) trained\_data

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \ | room | bathroom | | kitchen | french\_door | backyard | furnished | | green\_paint |
| 0 | 3 | 1 | | 2 | 1 | 1 | 0 | | 1 |
| 1 | 5 | 2 | | 2 | 2 | 1 | 0 | | 0 |
| 2 | 5 | 2 | | 2 | 2 | 1 | 0 | | 0 |
| 3 | 1 | 2 | | 1 | 2 | 0 | 0 | | 0 |
| 4 | 2 | 1 | | 2 | 3 | 1 | 1 | | 0 |
| ... | ... | ... | | ... | ... | ... | ... | | ... |
| 2995 | 1 | 1 | | 2 | 1 | 1 | 0 | | 0 |
| 2996 | 1 | 1 | | 2 | 3 | 0 | 1 | | 1 |
| 2997 | 2 | 2 | | 1 | 2 | 0 | 0 | | 1 |
| 2998 | 4 | 1 | | 1 | 2 | 1 | 1 | | 1 |
| 2999 | 4 | 2 | | 1 | 1 | 0 | 0 | | 0 |
|  | solar\_power | | woodfloor | | qlm\_security | club\_access | | price | |
| 0 | 0 | | 0 | | 1 | 1 | | 6835 | |
| 1 | 0 | | 0 | | 1 | 1 | | 9005 | |
| 2 | 0 | | 0 | | 1 | 1 | | 9005 | |
| 3 | 0 | | 1 | | 1 | 0 | | 5105 | |
| 4 | 0 | | 1 | | 1 | 0 | | 9105 | |
| ... | ... | | ... | | ... | ... | | ... | |
| 2995 | 1 | | 0 | | 0 | 0 | | 4825 | |
| 2996 | 0 | | 0 | | 1 | 1 | | 6755 | |
| 2997 | 1 | | 1 | | 0 | 0 | | 7565 | |
| 2998 | 0 | | 0 | | 0 | 1 | | 9135 | |
| 2999 | 1 | | 1 | | 0 | 0 | | 8955 | |

[3000 rows x 12 columns]

# IMPORTING TEST DATASET

test\_data = pd.DataFrame(pd.read\_csv(r'/content/drive/MyDrive/Colab Notebooks

/Copy of msc\_testing\_dataset.csv')) test\_data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | room | bathroom | kitchen | french\_door | backyard | furnished | green\_paint |
| \ |  |  |  |  |  |  |  |
| 0 | 1 | 1 | 1 | 3 | 0 | 0 | 1 |
| 1 | 5 | 1 | 1 | 2 | 0 | 0 | 0 |
| 2 | 5 | 1 | 1 | 3 | 0 | 0 | 0 |
| 3 | 4 | 2 | 2 | 1 | 0 | 1 | 1 |
| 4 | 5 | 2 | 1 | 1 | 0 | 1 | 1 |
| .. | ... | ... | ... | ... | ... | ... | ... |
| 994 | 5 | 2 | 2 | 3 | 1 | 1 | 0 |
| 995 | 5 | 1 | 2 | 3 | 1 | 1 | 0 |
| 996 | 3 | 2 | 2 | 1 | 0 | 1 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 997 | 3 | 2 | 1 | 1 | 1 | 0 | 0 |
| 998 | 2 | 1 | 2 | 1 | 0 | 1 | 1 |
|  | solar\_power | woodfloor | qlm\_security | | club\_access | price | |
| 0 | 1 | 0 | 1 | | 0 | 5068 | |
| 1 | 0 | 0 | 1 | | 1 | 7658 | |
| 2 | 1 | 1 | 1 | | 1 | 11318 | |
| 3 | 0 | 0 | 1 | | 0 | 8858 | |
| 4 | 1 | 0 | 0 | | 1 | 11178 | |
| .. | ... | ... | ... | | ... | ... | |
| 994 | 0 | 0 | 0 | | 0 | 10088 | |
| 995 | 0 | 0 | 0 | | 0 | 9788 | |
| 996 | 1 | 0 | 1 | | 0 | 9388 | |
| 997 | 1 | 1 | 0 | | 0 | 8528 | |
| 998 | 0 | 0 | 0 | | 0 | 6118 | |

[999 rows x 12 columns]

# CHECKING INFORMATION OF TEST AND TRAINED DATA

test\_data.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 999 entries, 0 to 998 Data columns (total 12 columns):

# Column Non-Null Count Dtype

- -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | room | 999 | non-null | int64 |
| 1 | bathroom | 999 | non-null | int64 |
| 2 | kitchen | 999 | non-null | int64 |
| 3 | french\_door | 999 | non-null | int64 |
| 4 | backyard | 999 | non-null | int64 |
| 5 | furnished | 999 | non-null | int64 |
| 6 | green\_paint | 999 | non-null | int64 |
| 7 | solar\_power | 999 | non-null | int64 |
| 8 | woodfloor | 999 | non-null | int64 |
| 9 | qlm\_security | 999 | non-null | int64 |
| 10 | club\_access | 999 | non-null | int64 |
| 11 | price | 999 | non-null | int64 |

dtypes: int64(12) memory usage: 93.8 KB

trained\_data.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 3000 entries, 0 to 2999

|  |  |  |  |
| --- | --- | --- | --- |
| Data  # | columns  Column | (total 12 columns):  Non-Null Count | Dtype |
| - | - |  |  |
| 0 | room | 3000 non-null | int64 |
| 1 | bathroom | 3000 non-null | int64 |

1. kitchen 3000 non-null int64
2. french\_door 3000 non-null int64
3. backyard 3000 non-null int64
4. furnished 3000 non-null int64
5. green\_paint 3000 non-null int64
6. solar\_power 3000 non-null int64
7. woodfloor 3000 non-null int64
8. qlm\_security 3000 non-null int64
9. club\_access 3000 non-null int64
10. price 3000 non-null int64 dtypes: int64(12)

memory usage: 281.4 KB trained\_data.head(20)

room bathroom kitchen french\_door backyard furnished green\_paint \ 0 3 1 2 1 1 0 1

1 5 2 2 2 1 0 0

2 5 2 2 2 1 0 0

3 1 2 1 2 0 0 0

4 2 1 2 3 1 1 0

5 5 1 2 1 0 0 1

6 3 1 1 3 1 0 0

7 1 1 1 1 0 0 0

8 5 1 1 2 0 0 0

9 3 1 1 2 0 0 0

10 4 1 1 2 0 1 0

11 5 1 1 2 1 0 1

12 2 1 2 1 1 0 1

13 5 1 2 2 1 1 0

14 5 2 2 2 1 0 0

15 1 1 2 2 0 0 0

16 5 1 2 3 0 0 0

17 5 2 1 2 0 1 0

18 1 2 2 2 1 0 0

19 4 2 1 2 1 0 0

solar\_power woodfloor qlm\_security club\_access price 0 0 0 1 1 6835

1 0 0 1 1 9005

2 0 0 1 1 9005

3 0 1 1 0 5105

4 0 1 1 0 9105

5 0 1 0 0 8995

6 1 0 0 0 6805

7 1 0 1 1 4935

8 1 1 0 0 9895

9 0 1 0 0 6365

10 1 1 0 1 11625

11 1 1 0 0 10825

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 12 | 0 | 0 | 0 | 0 | 4665 |
| 13 | 0 | 0 | 1 | 1 | 10705 |
| 14 | 1 | 1 | 0 | 1 | 11985 |
| 15 | 1 | 1 | 0 | 1 | 7125 |
| 16 | 1 | 1 | 0 | 1 | 11365 |
| 17 | 1 | 0 | 1 | 0 | 10745 |
| 18 | 1 | 1 | 0 | 0 | 7255 |
| 19 | 0 | 1 | 0 | 0 | 8225 |

# BASIC STATISTICS ON TRAINED DATA TO PROVIDE MORE UNDERSTANDING OF THE DATA

*#trained\_data.describe().transpose().round()*

**from** pandas.core.internals **import** concat

BasicStats = pd.concat([trained\_data.describe().transpose(),trained\_data.medi an()], axis = 1)

BasicStats.rename(columns={0: "Median"}, inplace=True)

BasicStats

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | count | mean | | std | min | 25% | 50% | \ |
| room | 3000.0 | 2.990000 | | 1.424281 | 1.0 | 2.0 | 3.0 |  |
| bathroom | 3000.0 | 1.489000 | | 0.499962 | 1.0 | 1.0 | 1.0 |  |
| kitchen | 3000.0 | 1.522000 | | 0.499599 | 1.0 | 1.0 | 2.0 |  |
| french\_door | 3000.0 | 1.998333 | | 0.813153 | 1.0 | 1.0 | 2.0 |  |
| backyard | 3000.0 | 0.490333 | | 0.499990 | 0.0 | 0.0 | 0.0 |  |
| furnished | 3000.0 | 0.488667 | | 0.499955 | 0.0 | 0.0 | 0.0 |  |
| green\_paint | 3000.0 | 0.485000 | | 0.499858 | 0.0 | 0.0 | 0.0 |  |
| solar\_power | 3000.0 | 0.495667 | | 0.500065 | 0.0 | 0.0 | 0.0 |  |
| woodfloor | 3000.0 | 0.512333 | | 0.499931 | 0.0 | 0.0 | 1.0 |  |
| qlm\_security | 3000.0 | 0.480667 | | 0.499709 | 0.0 | 0.0 | 0.0 |  |
| club\_access | 3000.0 | 0.499667 | | 0.500083 | 0.0 | 0.0 | 0.0 |  |
| price | 3000.0 | 8606.600000 | | 2216.248563 | 2235.0 | 7005.0 | 8615.0 |  |
|  | 75% | max | Median | | | | | |
| room | 4.0 | 5.0 | 3.0 | | | | | |
| bathroom | 2.0 | 2.0 | 1.0 | | | | | |
| kitchen | 2.0 | 2.0 | 2.0 | | | | | |
| french\_door | 3.0 | 3.0 | 2.0 | | | | | |
| backyard | 1.0 | 1.0 | 0.0 | | | | | |
| furnished | 1.0 | 1.0 | 0.0 | | | | | |
| green\_paint | 1.0 | 1.0 | 0.0 | | | | | |
| solar\_power | 1.0 | 1.0 | 0.0 | | | | | |
| woodfloor | 1.0 | 1.0 | 1.0 | | | | | |
| qlm\_security | 1.0 | 1.0 | 0.0 | | | | | |
| club\_access | 1.0 | 1.0 | 0.0 | | | | | |
| price | 10215.0 | 15035.0 | 8615.0 | | | | | |

trained\_data.median() room 3.0

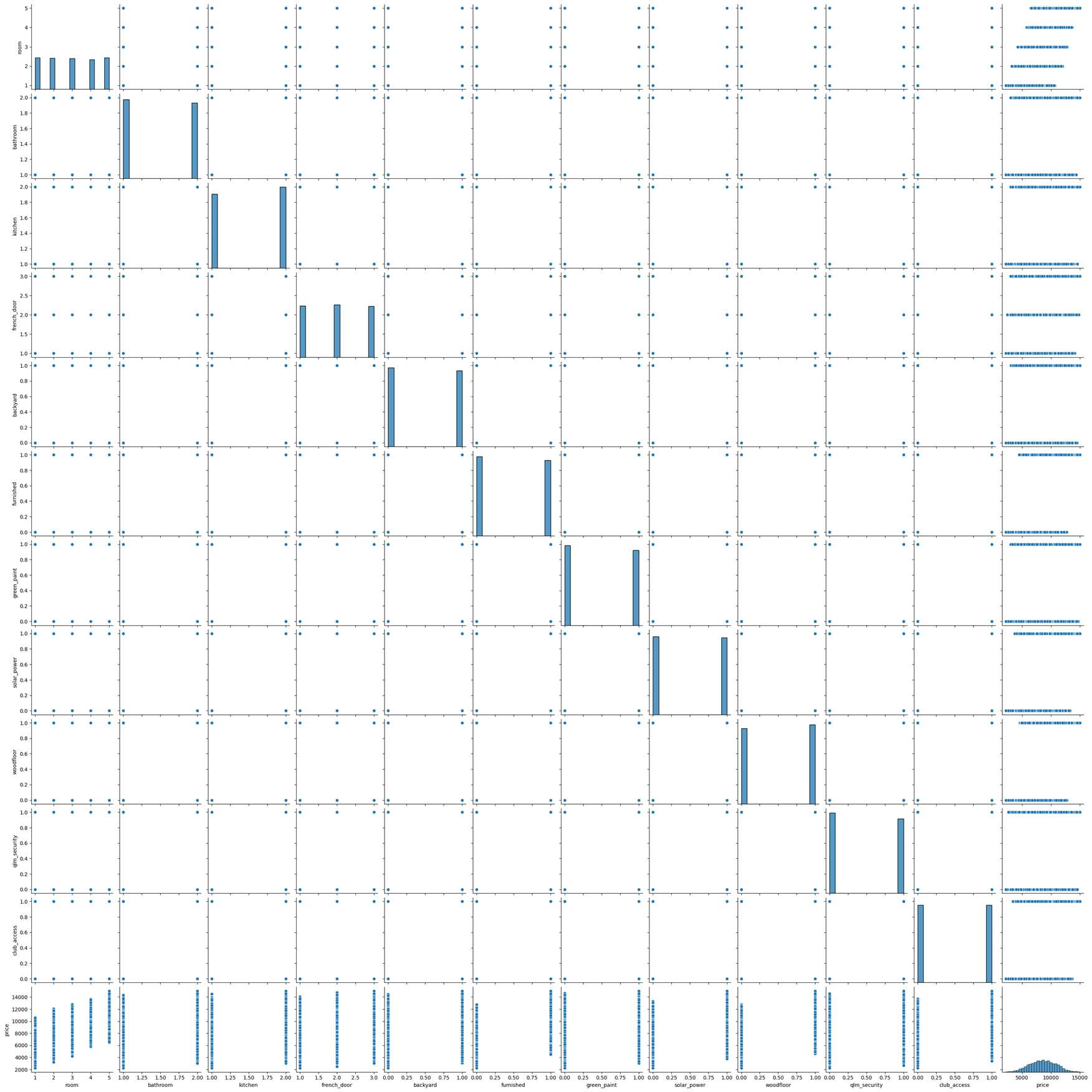
bathroom 1.0

|  |  |
| --- | --- |
| kitchen | 2.0 |
| french\_door | 2.0 |
| backyard | 0.0 |
| furnished | 0.0 |
| green\_paint | 0.0 |
| solar\_power | 0.0 |
| woodfloor | 1.0 |
| qlm\_security | 0.0 |
| club\_access | 0.0 |
| price | 8615.0 |
| dtype: float64 |  |

# DATA VISUALIZATION WITH PAIR PLOTS

sns.pairplot(trained\_data)

<seaborn.axisgrid.PairGrid at 0x7f9c4d12dcc0>



# CORRELATION ANALYSIS BETWEEN ALL THE FEATURES

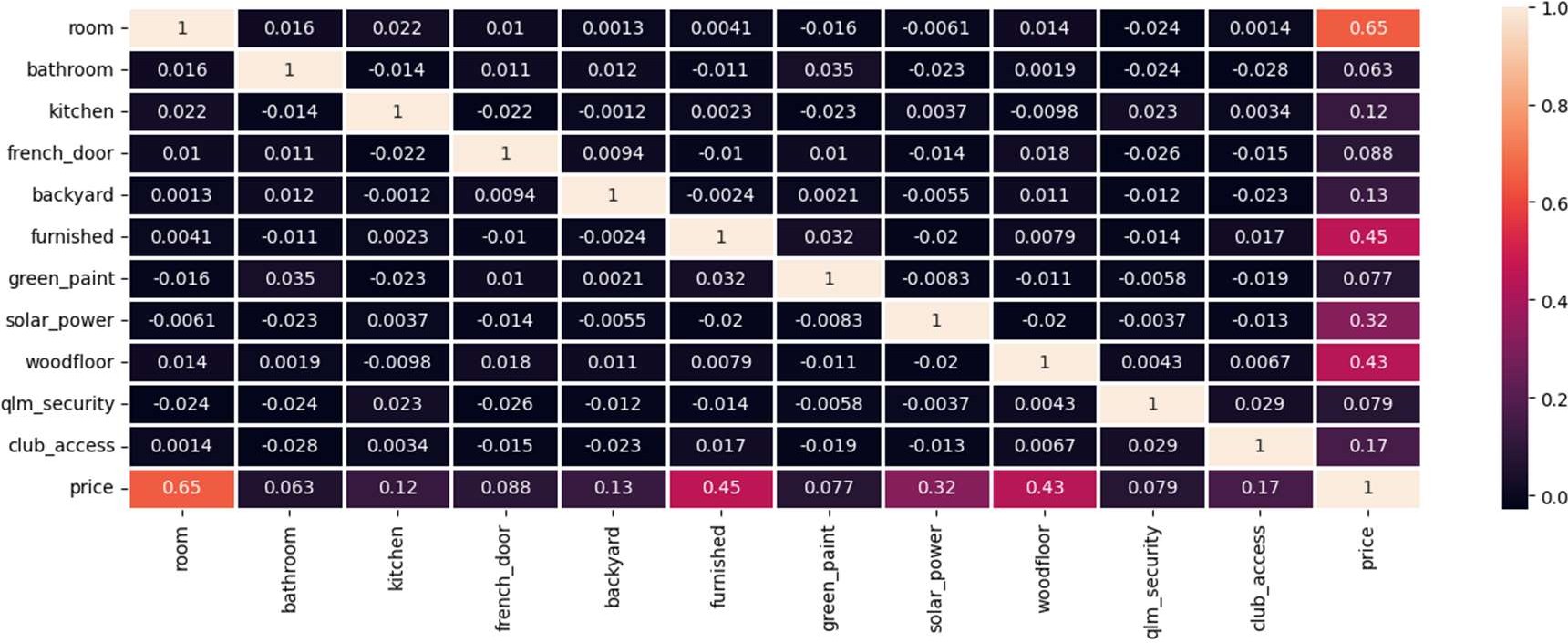
*#Manipulating the size of the figure* **import** matplotlib.pyplot **as** plt plt.figure(figsize = (16,5))

*#Correlation analysis between the features*

sns.heatmap(trained\_data.corr(), annot=True,linewidths=1

)

<Axes: >



*#Trained data columns*

trained\_data.columns

Index(['room', 'bathroom', 'kitchen', 'french\_door', 'backyard', 'furnished', 'green\_paint', 'solar\_power', 'woodfloor', 'qlm\_security', 'club\_access', 'price'],

dtype='object')

# MULTIPLE LINEAR REGRESSION MODEL

X=trained\_data[['room', 'bathroom', 'kitchen', 'french\_door', 'backyard', 'fu rnished',

'green\_paint', 'solar\_power', 'woodfloor', 'qlm\_security', 'club\_access']]

y=trained\_data['price']

|  |  |
| --- | --- |
| y |  |
| 0 | 6835 |
| 1 | 9005 |
| 2 | 9005 |
| 3 | 5105 |
| 4 | 9105 |

|  |  |
| --- | --- |
|  | ... |
| 2995 | 4825 |
| 2996 | 6755 |
| 2997 | 7565 |
| 2998 | 9135 |
| 2999 | 8955 |
| Name: | price, Length: 3000, dtype: int64 |

*#assigning variables to the train and test data to be fitted into the multipl e regression model*

X\_train,y\_train,X\_test,y\_test=X,y,test\_data[['room', 'bathroom', 'kitchen', ' french\_door', 'backyard', 'furnished',

|  |  |  |  |
| --- | --- | --- | --- |
| y\_test | 'green\_paint', 'solar\_power', 'woodfloor',  'club\_access']],test\_data['price'] | | 'qlm\_security', |
| 0 | 5068 | |  |
| 1 | 7658 | |  |
| 2 | 11318 | |  |
| 3 | 8858 | |  |
| 4 | 11178  ... | |  |
| 994 | 10088 |  | |
| 995 | 9788 |  | |
| 996 | 9388 |  | |
| 997 | 8528 |  | |
| 998  Name: | 6118  price, | Length: 999, dtype: int64 | |

*# Fit a multiple linear regression model*

LR=LinearRegression() LR.fit(X\_train,y\_train)

*# Make predictions on test data*

y\_pred = LR.predict(X\_test)

*#Regression model Coefficient*

Coefficie = pd.DataFrame(LR.coef\_.transpose(),X.columns,columns=['Coefficient '])

Coefficie.transpose() Coefficie

Coefficient

room 1000.0

bathroom 300.0

|  |  |
| --- | --- |
| kitchen | 500.0 |
| french\_door | 240.0 |
| backyard | 560.0 |
| furnished | 2000.0 |
| green\_paint | 370.0 |
| solar\_power | 1530.0 |
| woodfloor | 1890.0 |
| qlm\_security | 440.0 |
| club\_access | 730.0 |

*#fitting the train data into the regression model*

LR.fit(X\_train,y\_train)

*# Make predictions on test data*

y\_pred = LR.predict(X\_test)

*#printing the intercept*

print("Intercept: \n",LR.intercept\_)

*#Print the performance metrics*

print('R-squared: {:.3f}'.format(r2\_score(y\_test, y\_pred))) print('MSE: {:.3f}'.format(mean\_squared\_error(y\_test, y\_pred))) print('RMSE: {:.3f}'.format(sqrt(mean\_squared\_error(y\_test, y\_pred)))) print('MAE: {:.3f}'.format(mean\_absolute\_error(y\_test, y\_pred)))

Intercept:

195.00000000000182

R-squared: 1.000

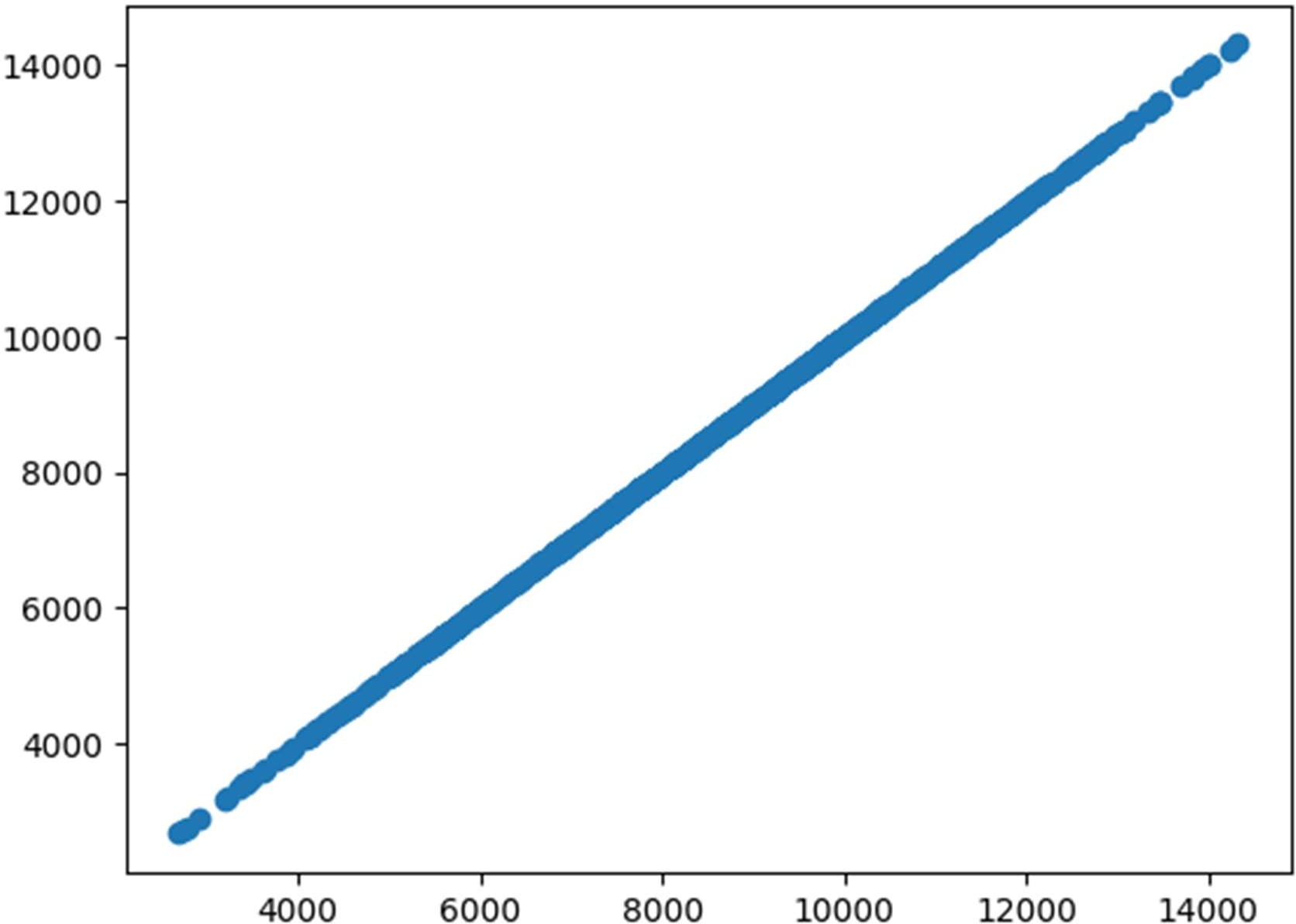
MSE: 169.000

RMSE: 13.000

MAE: 13.000

plt.scatter(y\_test,y\_pred)

<matplotlib.collections.PathCollection at 0x7f9c412342e0>



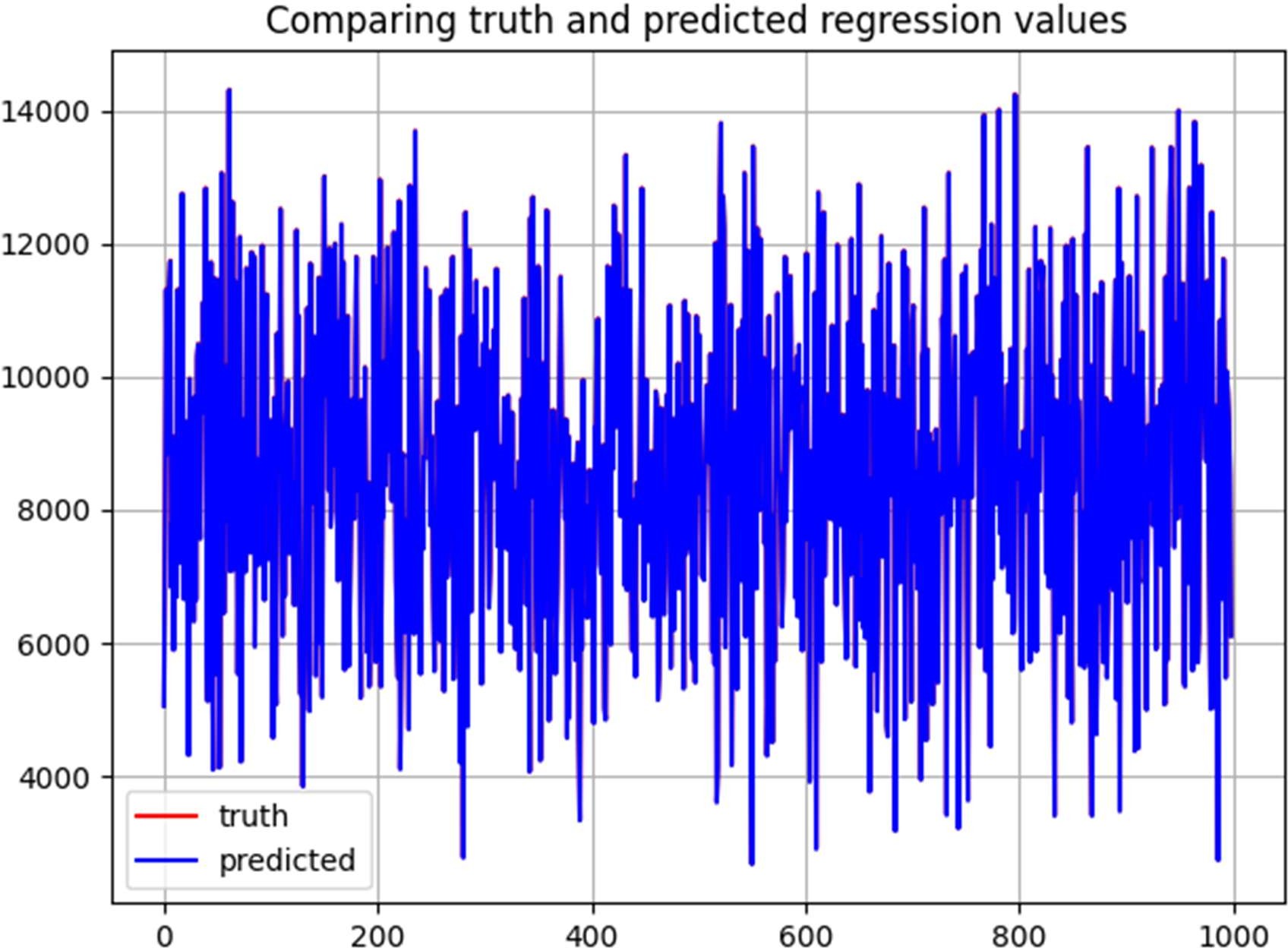
**def** make\_plot(truth, prediction): plt.plot(truth, color="red", label="truth")

plt.plot(prediction, color="blue", label="predicted") plt.legend()

plt.grid()

plt.title("Comparing truth and predicted regression values") plt.tight\_layout()

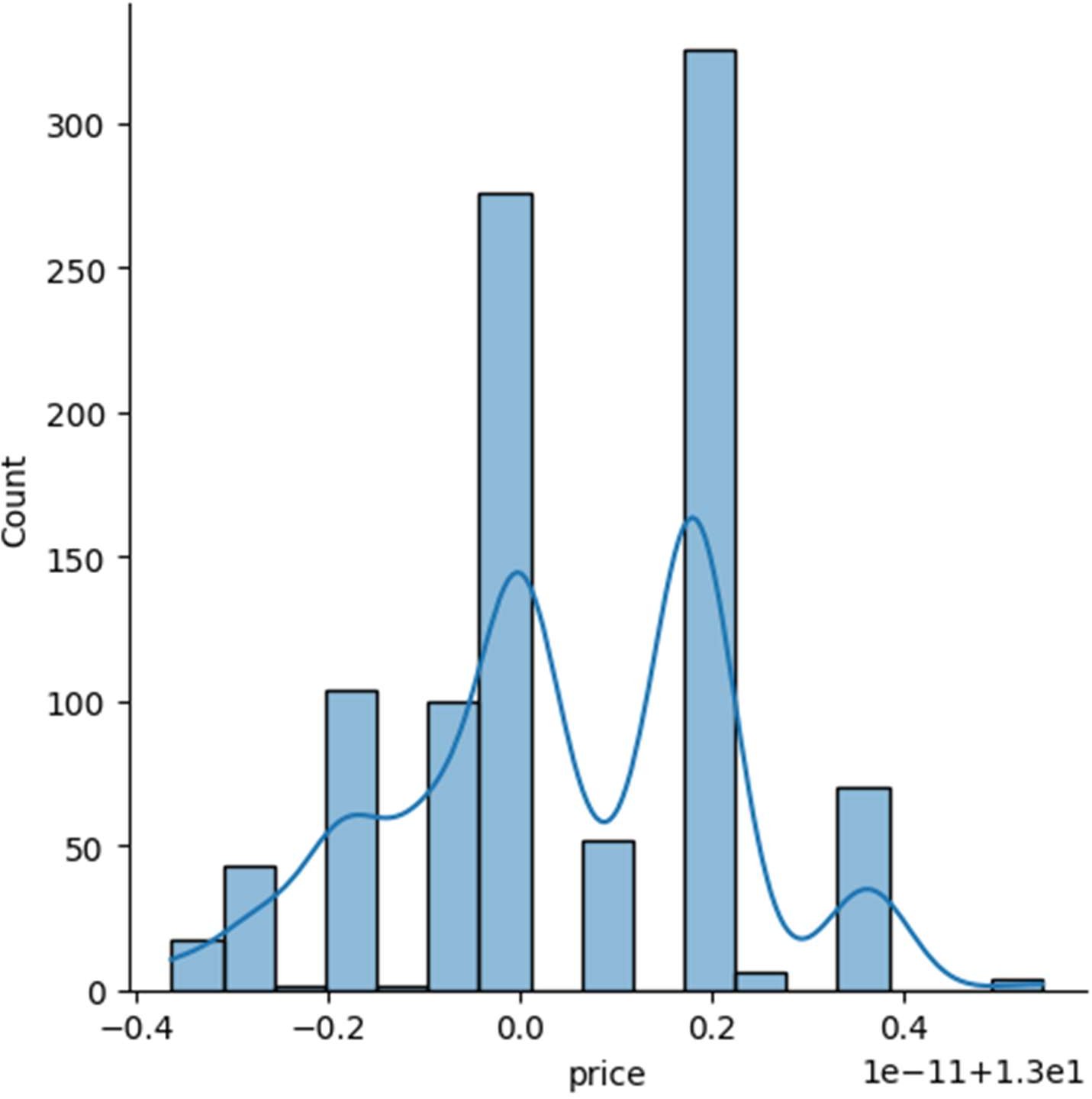
plt.show() make\_plot(y\_test,y\_pred)



*#Display the error distribution*

error = y\_test - y\_pred sns.displot(error, kde=True)

<seaborn.axisgrid.FacetGrid at 0x7f9c455bcd60>



*#converting the predicted price to a dataframe*

Price\_predictions=pd.DataFrame(y\_pred)

Price\_predictions

|  |  |
| --- | --- |
|  | 0 |
| 0 | 5055.0 |
| 1 | 7645.0 |
| 2 | 11305.0 |
| 3 | 8845.0 |
| 4 | 11165.0 |
| .. | ... |
| 994 | 10075.0 |
| 995 | 9775.0 |
| 996 | 9375.0 |
| 997 | 8515.0 |
| 998 | 6105.0 |

[999 rows x 1 columns]

**from** pandas.core.internals **import** concat

TESTDATAPREDICTIONS = pd.concat([test\_data,round(Price\_predictions)], axis = 1)

TESTDATAPREDICTIONS.columns

TESTDATAPREDICTIONS.rename(columns={0: "Predicted\_price"}, inplace=True) TESTDATAPREDICTIONS

*#student\_df\_1.rename(columns={"id": "ID"}, inplace=True) #student\_df\_1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \ | room | bathroom | | kitchen | french\_door | backyard | furnished | | green\_paint |
| 0 | 1 | 1 | | 1 | 3 | 0 | 0 | | 1 |
| 1 | 5 | 1 | | 1 | 2 | 0 | 0 | | 0 |
| 2 | 5 | 1 | | 1 | 3 | 0 | 0 | | 0 |
| 3 | 4 | 2 | | 2 | 1 | 0 | 1 | | 1 |
| 4 | 5 | 2 | | 1 | 1 | 0 | 1 | | 1 |
| .. | ... | ... | | ... | ... | ... | ... | | ... |
| 994 | 5 | 2 | | 2 | 3 | 1 | 1 | | 0 |
| 995 | 5 | 1 | | 2 | 3 | 1 | 1 | | 0 |
| 996 | 3 | 2 | | 2 | 1 | 0 | 1 | | 1 |
| 997 | 3 | 2 | | 1 | 1 | 1 | 0 | | 0 |
| 998 | 2 | 1 | | 2 | 1 | 0 | 1 | | 1 |
| e | solar\_power | | woodfloor | | qlm\_security | club\_access | | price | Predicted\_pric |
| 0 | 1 | | 0 | | 1 | 0 | | 5068 | 5055. |
| 0 |  | |  | |  |  | |  |  |
| 1 | 0 | | 0 | | 1 | 1 | | 7658 | 7645. |
| 0 |  | |  | |  |  | |  |  |
| 2 | 1 | | 1 | | 1 | 1 | | 11318 | 11305. |
| 0 |  | |  | |  |  | |  |  |
| 3 | 0 | | 0 | | 1 | 0 | | 8858 | 8845. |
| 0 |  | |  | |  |  | |  |  |
| 4 | 1 | | 0 | | 0 | 1 | | 11178 | 11165. |
| 0  ..  . 994 | ...  0 | | ...  0 | | ...  0 | ...  0 | | ... 10088 | .. 10075. |
| 0 |  | |  | |  |  | |  |  |
| 995 | 0 | | 0 | | 0 | 0 | | 9788 | 9775. |
| 0 |  | |  | |  |  | |  |  |
| 996 | 1 | | 0 | | 1 | 0 | | 9388 | 9375. |
| 0 |  | |  | |  |  | |  |  |
| 997 | 1 | | 1 | | 0 | 0 | | 8528 | 8515. |
| 0 |  | |  | |  |  | |  |  |
| 998 | 0 | | 0 | | 0 | 0 | | 6118 | 6105. |
| 0 |  | |  | |  |  | |  |  |

[999 rows x 13 columns]

# APPLICATION OF RANDOM FOREST

*#PREDICTION ON THE SPLITTED DATA*

**from** sklearn.ensemble **import** RandomForestRegressor

*#X\_train2, X\_test2, y\_train2, y\_test2 = train\_test\_split(X, y, test\_size=0.2, random\_state=42)*

random\_forest = RandomForestRegressor().fit(X\_train, y\_train) random\_forest.fit(X\_train, y\_train)

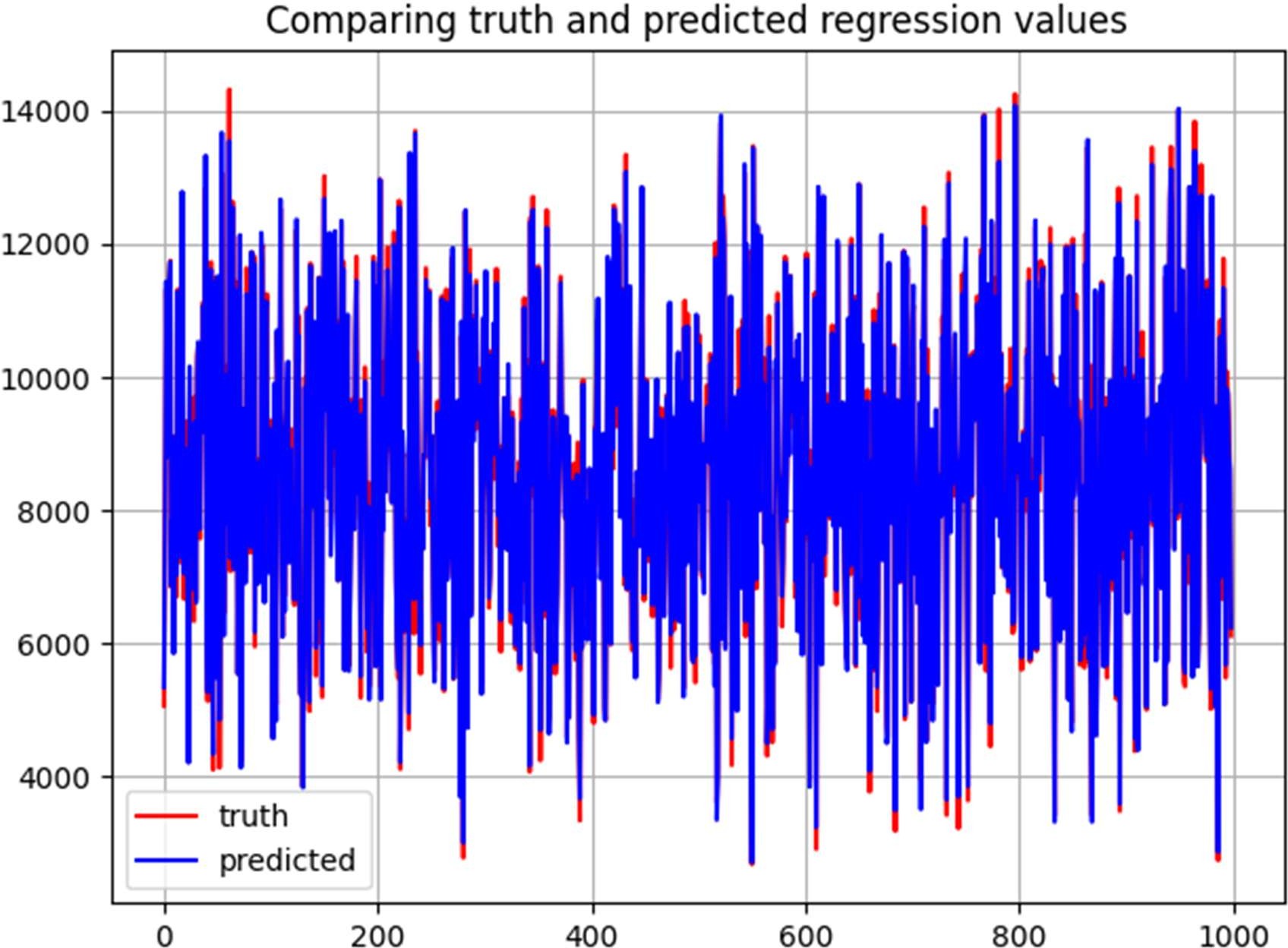
y\_pred2 = random\_forest.predict(X\_test) print('Test R2 score:', r2\_score(y\_test, y\_pred2))

print('Test MSE score:', mean\_squared\_error(y\_test, y\_pred2)) print('RMSE: {:.3f}'.format(sqrt(mean\_squared\_error(y\_test, y\_pred2)))) print('MAE: {:.3f}'.format(mean\_absolute\_error(y\_test, y\_pred2)))

Test R2 score: 0.9894378057950831 Test MSE score: 51904.21548548549 RMSE: 227.825

MAE: 171.646

make\_plot(y\_test,y\_pred2)



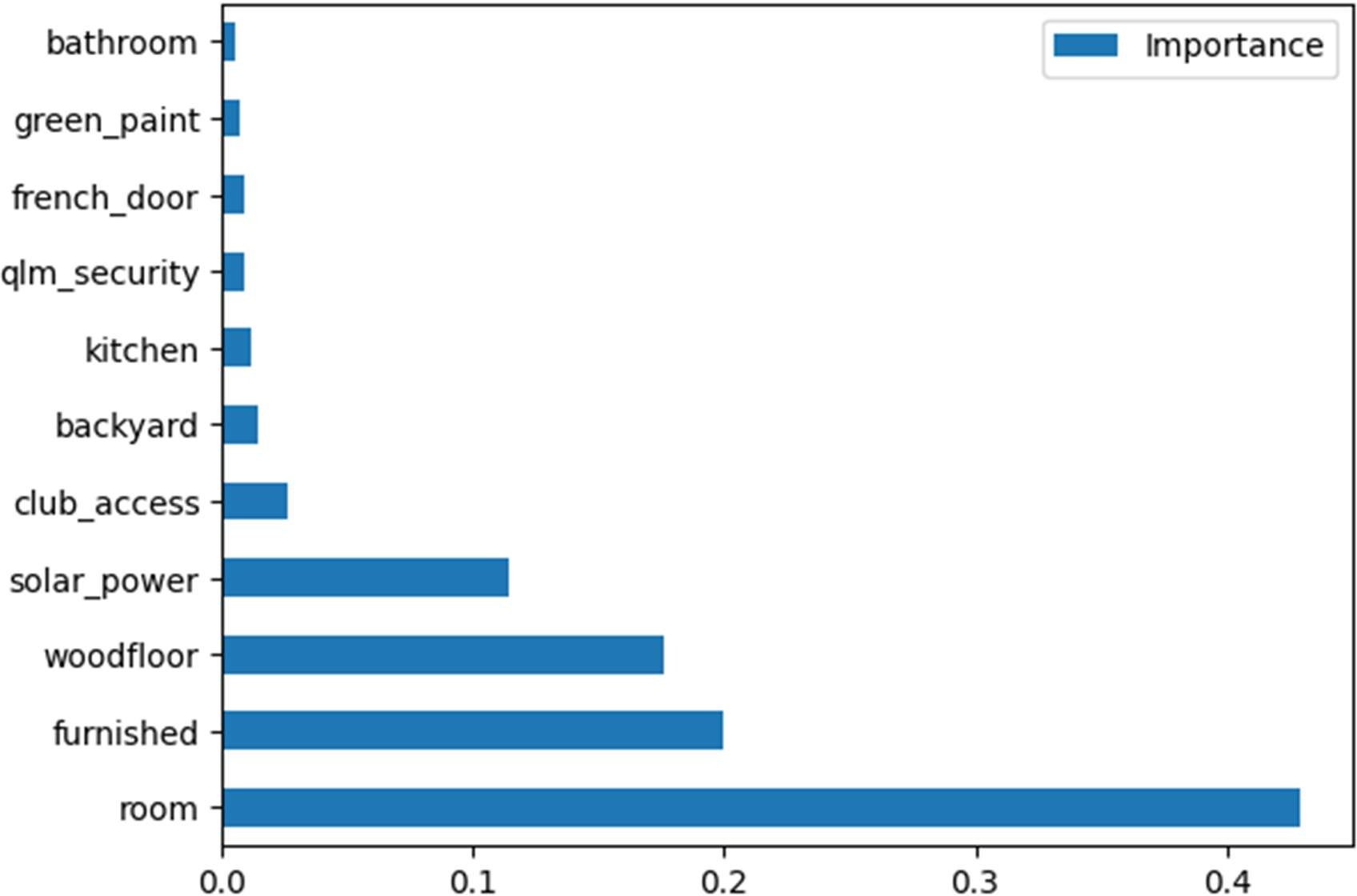
# Feature Importance

feature\_importances = pd.DataFrame(random\_forest.feature\_importances\_, index= X\_train.columns,columns=['Importance']).sort\_values('Importance',ascending= F alse)

feature\_importances

|  |  |
| --- | --- |
|  | Importance |
| room | 0.429179 |
| furnished | 0.199253 |
| woodfloor | 0.176120 |
| solar\_power | 0.114433 |
| club\_access | 0.026048 |
| backyard | 0.014553 |
| kitchen | 0.011144 |
| qlm\_security | 0.008883 |
| french\_door | 0.008842 |
| green\_paint | 0.006823 |
| bathroom | 0.004723 |

feature\_importances.plot(kind='barh') plt.show()



# OUT-OF-BAG (OOB) Feature Importance

*# RF Model*

randomforestoob = RandomForestRegressor(random\_state=42, oob\_score=True)

*# RF model fit with OOB*

randomforestoob.fit(X\_train, y\_train)

*# Evaluate OOB score*

print('OOB R2 score:', randomforestoob.oob\_score\_)

print('OOB MSE score:', mean\_squared\_error(y\_train, randomforestoob.oob\_predi ction\_))

*#pd.DataFrame(random\_forest.feature\_importances\_, index=X\_train2.columns,colu mns=['Importance']).sort\_values('Importance',ascending= False)*

*# Obtain feature importances and plot them*

importancesoob = pd.DataFrame(randomforestoob.feature\_importances\_,index=X\_tr ain.columns,columns=['Importance']).sort\_values('Importance',ascending= False

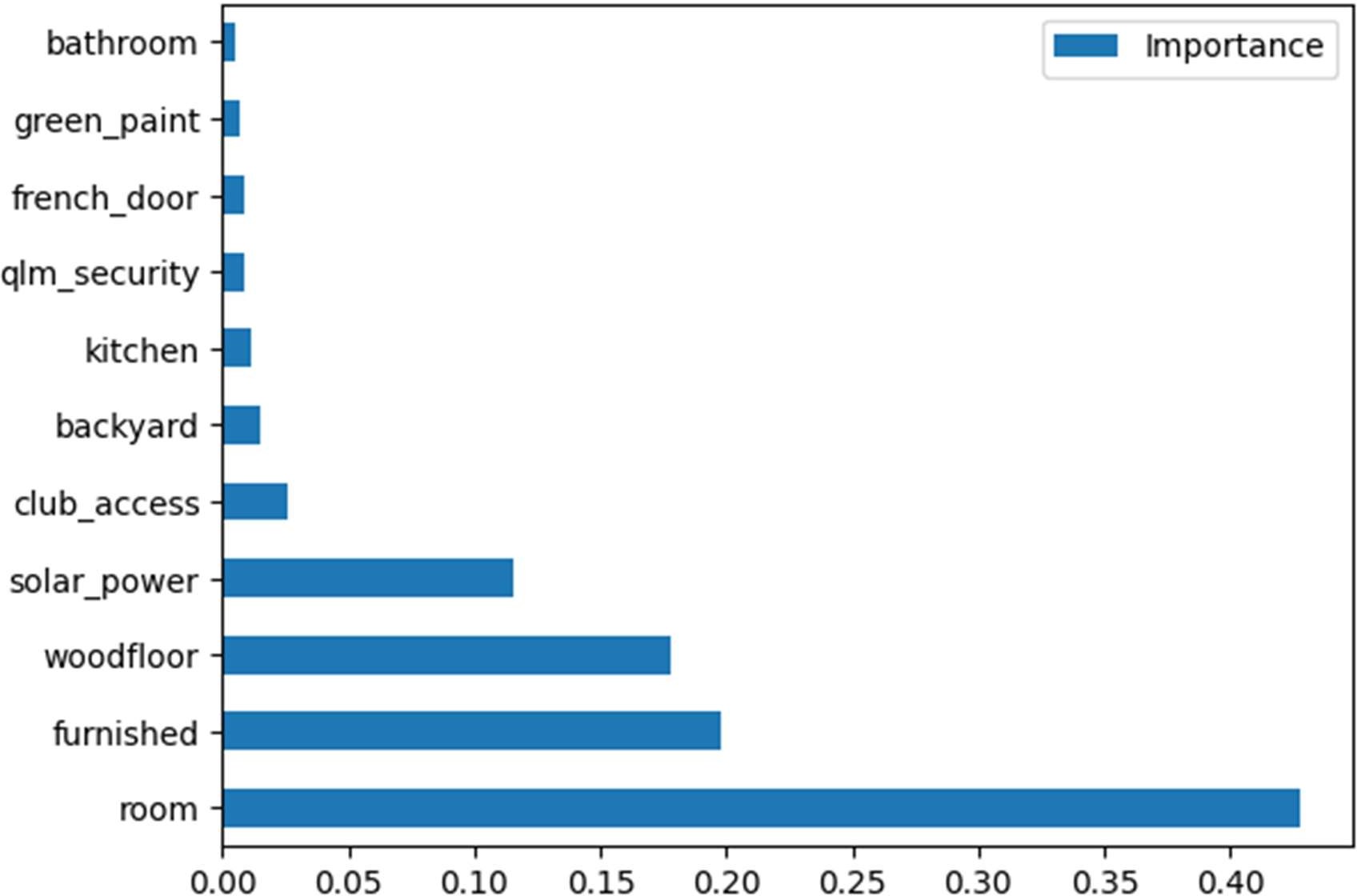
)

print(importancesoob) importancesoob.plot(kind='barh') plt.show()

OOB R2 score: 0.9888252925267629 OOB MSE score: 54869.15957536236

Importance

|  |  |
| --- | --- |
| room | 0.427519 |
| furnished | 0.197675 |
| woodfloor | 0.178273 |
| solar\_power | 0.115170 |
| club\_access | 0.025911 |
| backyard | 0.014622 |
| kitchen | 0.011362 |
| qlm\_security | 0.008978 |
| french\_door | 0.008892 |
| green\_paint | 0.006909 |
| bathroom | 0.004688 |



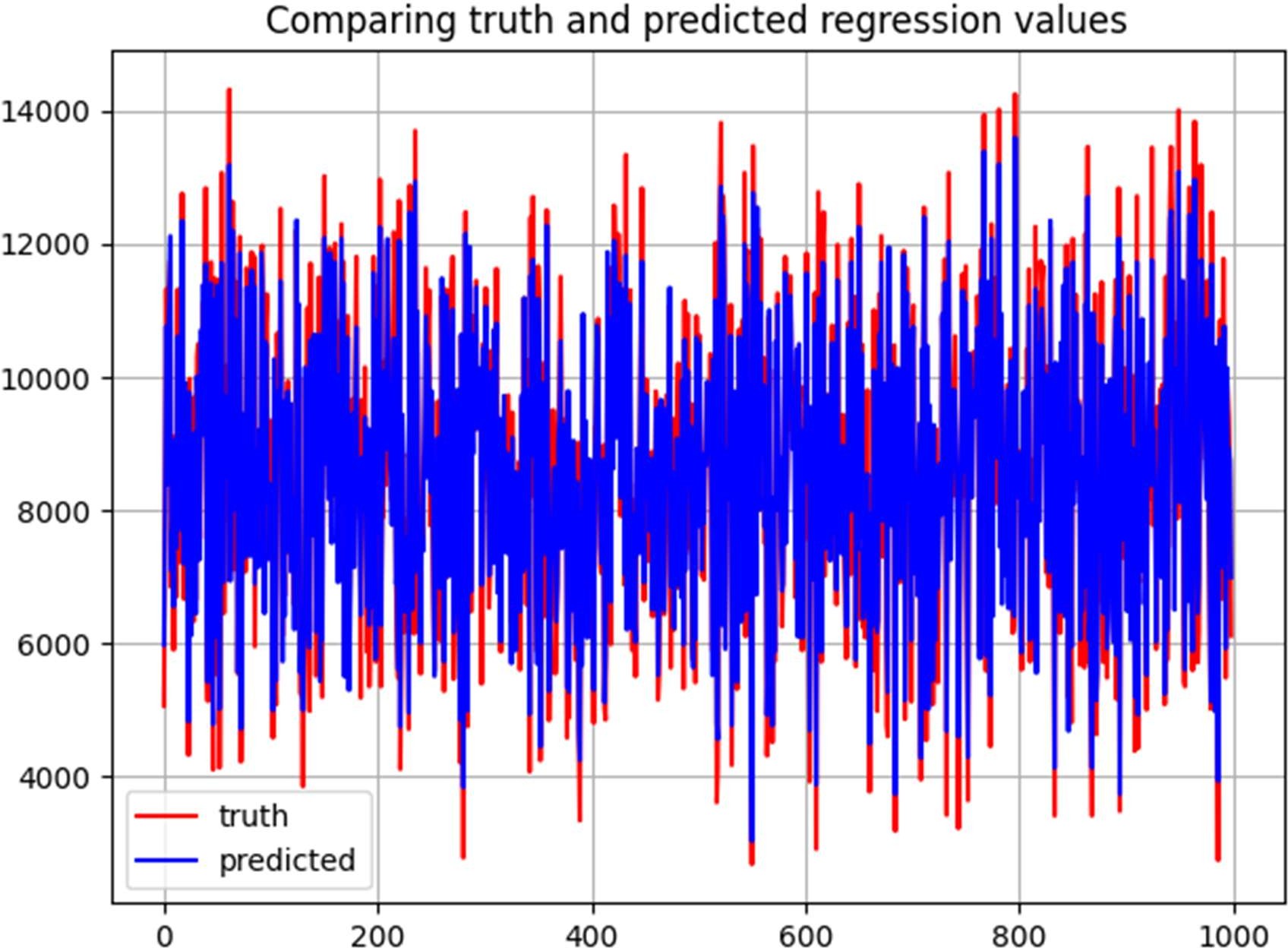
# Using KNeighborsRegressor

**from** sklearn.neighbors **import** KNeighborsRegressor neigh = KNeighborsRegressor(n\_neighbors=7) neigh.fit(X\_train,y\_train)

knn\_predicted = neigh.predict(X\_test) print(neigh.score(X\_train,y\_train)) print(mean\_absolute\_error(y\_test, knn\_predicted)) make\_plot(y\_test, knn\_predicted)

0.9471383275821861

471.3446303446304



# PRICE PREDICTIONS WITH MULTIPLE LINEAR REGRESSION VS RANDOM FOREST

*#TESTDATAPREDICTIONS*

Randomforestpricepridiction=pd.DataFrame(y\_pred2) Randomforestpricepridiction

TESTDATAPREDICTIONS

knn\_price=pd.DataFrame(knn\_predicted)

ComparisonMLR\_RF = pd.concat([TESTDATAPREDICTIONS,round(Randomforestpriceprid iction)], axis = 1)

*#TESTDATAPREDICTIONS.columns*

ComparisonMLR\_RF.rename(columns={0: "RF\_PRICEPREDICT","Predicted\_price": "MLR

\_PRICEPREDICT"}, inplace=True)

Final=pd.concat([ComparisonMLR\_RF,round(knn\_price)], axis = 1) Final.rename(columns={0: "Knn\_PRICEPREDICT"}, inplace=True)

Final

room bathroom kitchen french\_door backyard furnished green\_paint

\

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 3 | | 0 | 0 | 1 |
| 1 | 5 | 1 | 1 | 2 | | 0 | 0 | 0 |
| 2 | 5 | 1 | 1 | 3 | | 0 | 0 | 0 |
| 3 | 4 | 2 | 2 | 1 | | 0 | 1 | 1 |
| 4 | 5 | 2 | 1 | 1 | | 0 | 1 | 1 |
| .. | ... | ... | ... | ... | | ... | ... | ... |
| 994 | 5 | 2 | 2 | 3 | | 1 | 1 | 0 |
| 995 | 5 | 1 | 2 | 3 | | 1 | 1 | 0 |
| 996 | 3 | 2 | 2 | 1 | | 0 | 1 | 1 |
| 997 | 3 | 2 | 1 | 1 | | 1 | 0 | 0 |
| 998 | 2 | 1 | 2 | 1 | | 0 | 1 | 1 |
|  | solar\_power | woodfloor | | qlm\_security | | club\_access | price | \ |
| 0 | 1 | 0 | | 1 | | 0 | 5068 |  |
| 1 | 0 | 0 | | 1 | | 1 | 7658 |  |
| 2 | 1 | 1 | | 1 | | 1 | 11318 |  |
| 3 | 0 | 0 | | 1 | | 0 | 8858 |  |
| 4 | 1 | 0 | | 0 | | 1 | 11178 |  |
| .. | ... | ... | | ... | | ... | ... |  |
| 994 | 0 | 0 | | 0 | | 0 | 10088 |  |
| 995 | 0 | 0 | | 0 | | 0 | 9788 |  |
| 996 | 1 | 0 | | 1 | | 0 | 9388 |  |
| 997 | 1 | 1 | | 0 | | 0 | 8528 |  |
| 998 | 0 | 0 | | 0 | | 0 | 6118 |  |
|  | MLR\_PRICEPREDICT | | RF\_PRICEPREDICT | | Knn\_PRICEPREDICT | | | |
| 0 | 5055.0 | | 5346.0 | | 5981.0 | | | |
| 1 | 7645.0 | | 7717.0 | | 7901.0 | | | |
| 2 | 11305.0 | | 11432.0 | | 10762.0 | | | |
| 3 | 8845.0 | | 8816.0 | | 8391.0 | | | |
| 4 | 11165.0 | | 11090.0 | | 10556.0 | | | |
| .. | ... | | ... | | ... | | | |
| 994 | 10075.0 | | 9820.0 | | 10142.0 | | | |
| 995 | 9775.0 | | 9755.0 | | 8994.0 | | | |
| 996 | 9375.0 | | 8834.0 | | 8924.0 | | | |
| 997 | 8515.0 | | 8616.0 | | 8754.0 | | | |
| 998 | 6105.0 | | 6243.0 | | 6995.0 | | | |

[999 rows x 15 columns]

Final.to\_excel(r'/content/drive/MyDrive/Colab Notebooks/price projection2.xls x', index=False)