A logo for college computing

Description automatically generated

**Assessment Cover Page**

|  |  |
| --- | --- |
| *Student Full Name* | Denisse Guadarrama Tinoco |
| *Student Number* | 2024154 |
| *Module Title* | Machine Learning for Data Analysis |
| *Assessment Title* | CA1 |
| *Assessment Due Date* | 21th April 2024 |
| *Date of Submission* | 23th April 2024 |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

[Introduction: iii](#_Toc164859011)

[Objective: iii](#_Toc164859012)

[Problem Definition: iii](#_Toc164859013)

[Data Source: iii](#_Toc164859014)

[Execution: iv](#_Toc164859015)

[Therefore, we used tuning techniques such as GridSearchCV, this tool helps to find set of hyperparameters that result in the best performance of a model on a dataset. vii](#_Toc164859016)

[Conclusion: xii](#_Toc164859017)

[References: xii](#_Toc164859018)

# Introduction:

Predicting the price of a home could be very useful for anyone who is looking to buy or sell one.

With Machine Learning we can forecast very complex things having a reliable database, but for this Project we used a "simple" database to find more everyday answers such as: "how much will my house with 2 bathrooms, 3 bedrooms and that is in a rural area cost?" Being able to predict the price of a house makes what we have seen easier to understand.

In this project 3 different models will be used with a split of 20%, 25% and 30%:

* Random Forest Regression:

This model takes longer to execute because it creates multiple random decision trees, then averages the results of each one to generate a new result that leads to predictions.

* Linear Regression:

It is useful when you know the relationship between the independent and dependent variable have a linear relationship. It is important before trying to fit a linear model to the observed dataset, one should assess whether or not there is a relationship between the variables. Of course, this doesn't mean that one variable causes the other, but there should be some visible correlation between them.

* K-Neighbors Regressor:

It is one of the simplest models in machine learning. As it will use the entirety of the training dataset to find the “nearest neighbors” according to the average distance of the target variable values of these neighbors.

# Objective:

Predict the price of a house depending on 5 variables, finding the best model with the best split:

1. House Size
2. Number of bathrooms
3. Number of bedrooms
4. Type of Neighborhood
5. Year of construction

# Problem Definition:

Let's suppose we are a new real estate agency in X place, and we are starting to sell our first houses, but we have no idea how to sell it, however there is a database with the values of the houses sold, with this information I could find a price that according to the conditions of the house you could sell. Therefore, the main problem is to find a model that best predicts the price of a house.

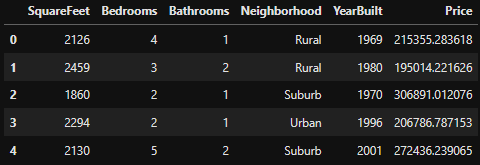
# Data Source:

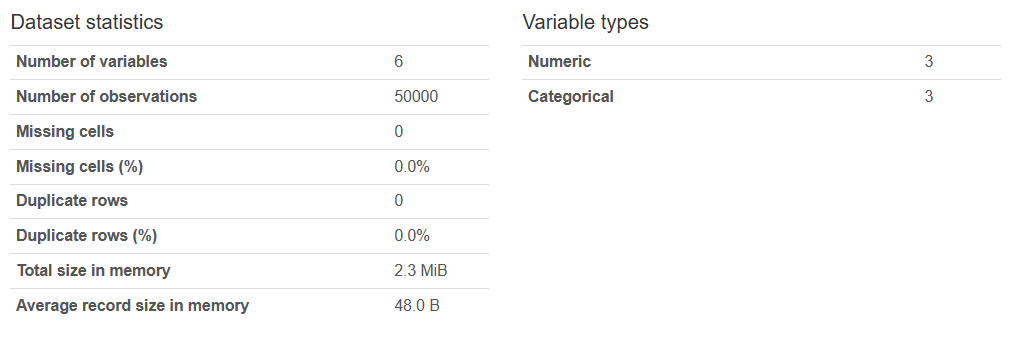
The URL database was taken: [Housing Price Prediction Data (kaggle.com)](https://www.kaggle.com/datasets/muhammadbinimran/housing-price-prediction-data)

# Execution:

The first step is to observe and clean our data, using statistical methods to know its distribution.

This is the visualization of 4 rows of the data set:

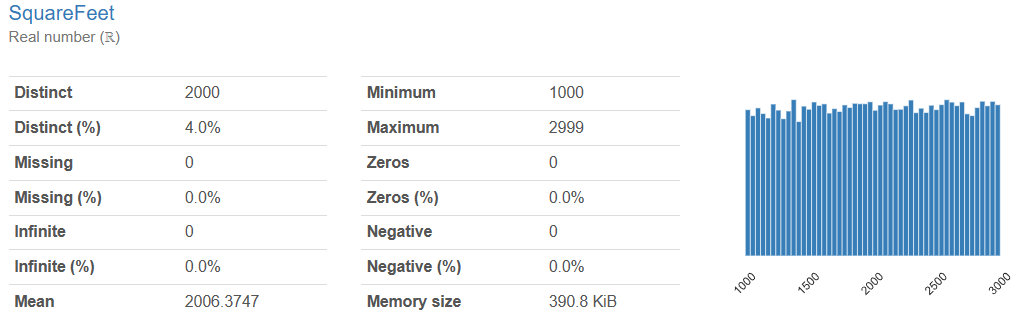


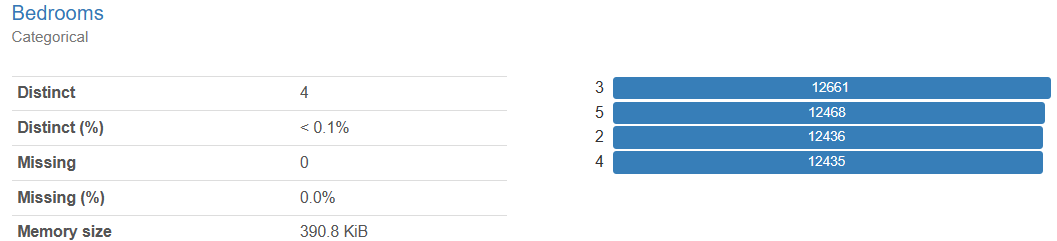


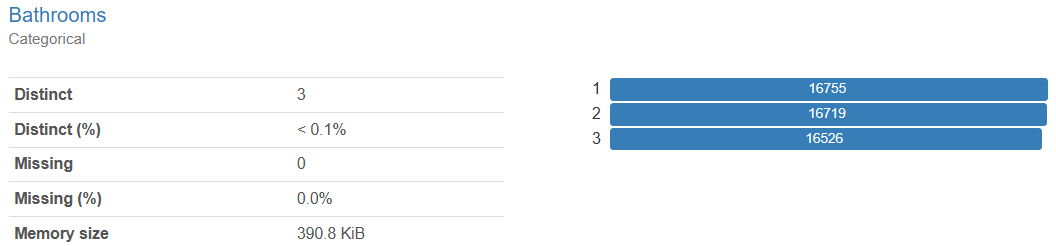
The data set contains 50,000 observations and 6 features.

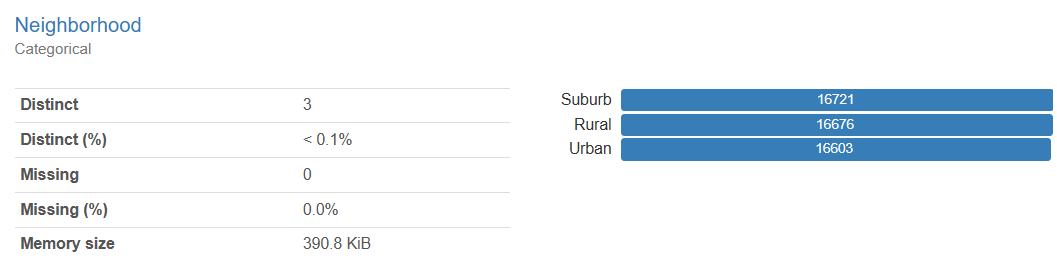
There is no missing data or duplicate rows.

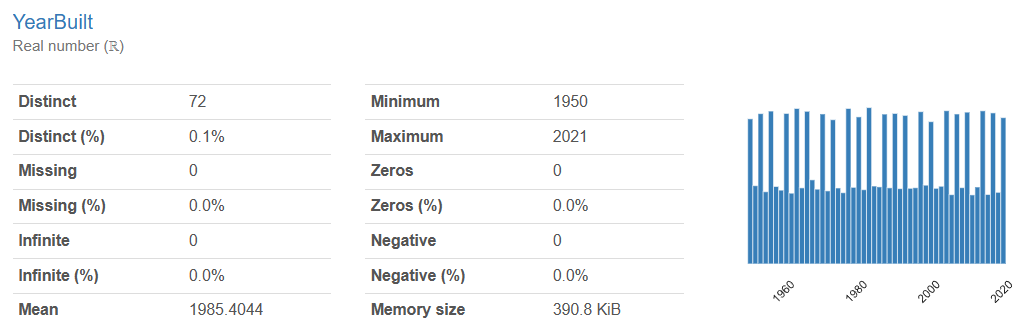
There are five variables:



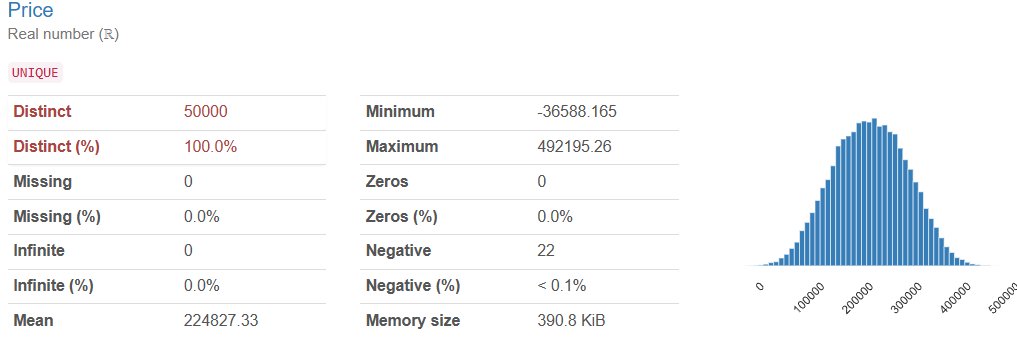






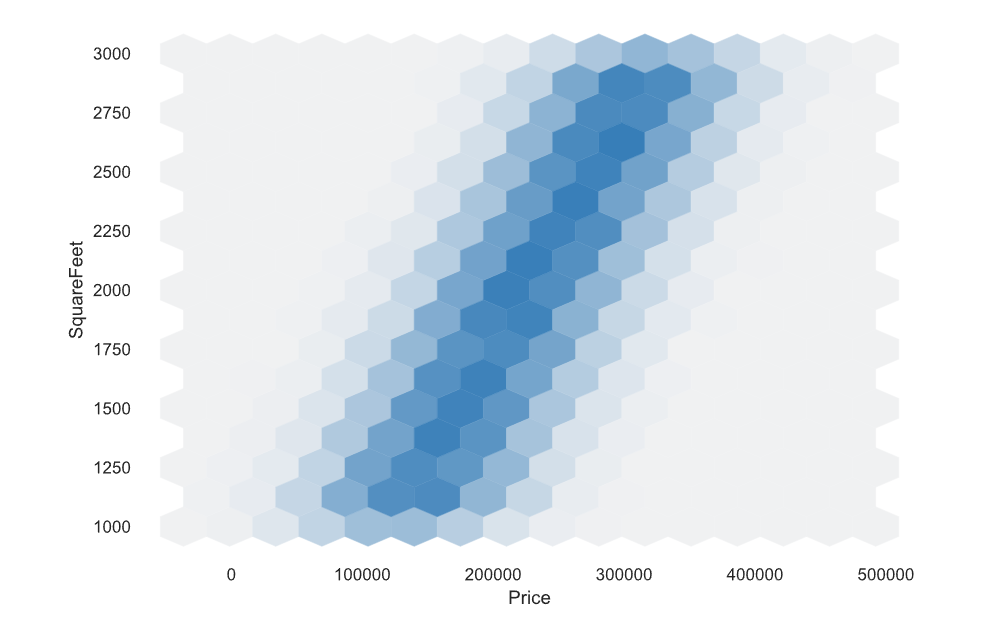


And the target variable:

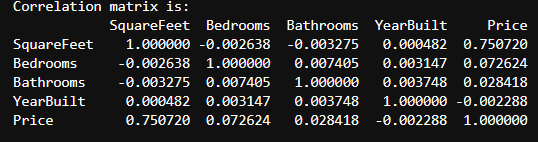


We can see that there are some outliers because there are prices less than 0. The mean, median and mode are not the same values for these outliers, however it can be observed that the price has a normal distribution.

In the following graph you can see the relationship between the squareFeet and price, the price increases as the size of the square increases.



Again, we see the correlation of the variables with the correlation matrix, only the squarefeet comes close to 1, this means that they have a linear relationship:

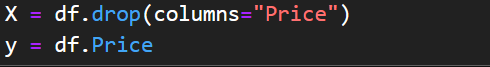


All of our columns are numeric, except for Neighborhood which contains 3 unique values: Rural, Suburb and Urban, so we are going to create dummies to create 3 new columns, which will be binary, this will make it easier to run our models, because if we just modified the Neighborhood column and assigned numerical values to each word, our model could assume that one value is better than another, therefore it is better to add new columns where only the data is represented as 1 or 0, it is easier to analyse.



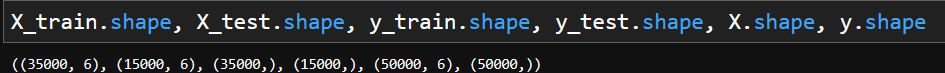
We removed the original column and one of the 3 dummies because with the first two we have all the information of the original.

Now we will separate the data in two, X will contain the dependent variables and Y will be the target variable.

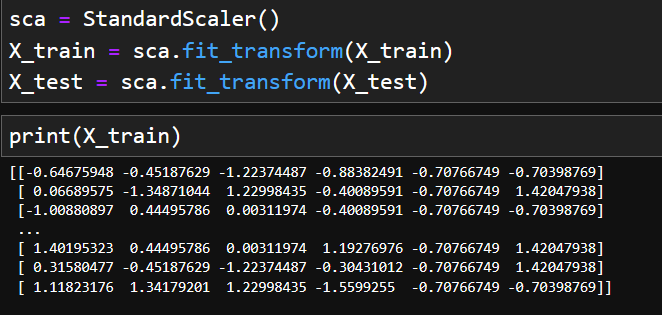


**Splitting Data into Training and Testing: 30%**

To start, let's divide the data into training and testing, where testing will be 30% of our data. This will make part of our data to train our model and the other part to test it:



As we saw above, our data has very different values, so let's scale them so that we all have a similar range.



The first model we will use is:

**Random Forest Regressor.**

This model is widely used to predict prices, so I found it suitable to use it, as it could find among all the possibilities, the best average result.

The first and second time I ran the model, I used 100 and 200 number of estimators, but the test score was low. Last time I changed it to 300 in n\_estimators but even then, the testing was still low, which could indicate that our model was overfitting, since the training score almost reached 100% but the testing was 51%.

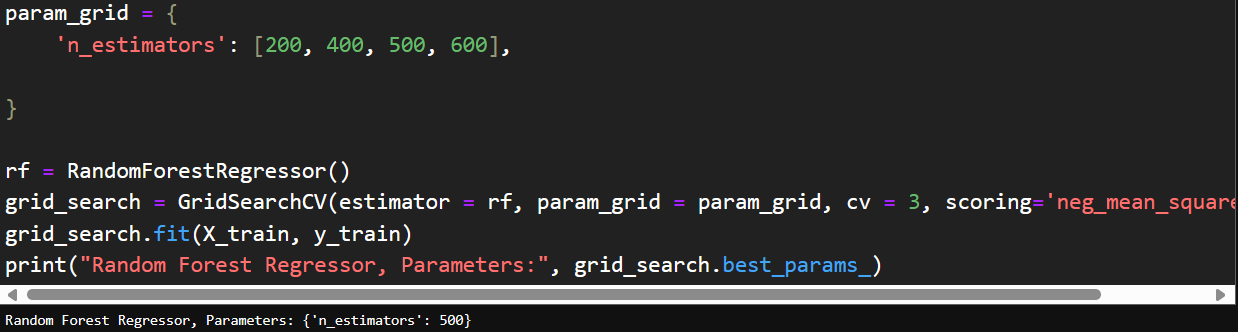
Train score: 93.30%

Test score: 51.41%

# Therefore, we used tuning techniques such as GridSearchCV, this tool helps to find set of hyperparameters that result in the best performance of a model on a dataset.

We ask to find the number of estimators with the lowest value of the mean-squared error in order to train our model.

It was given only 4 options, as it took too long to run this function with so many variables. The best option given was 500 estimators.



Now we used this information to run our model again with that hyperparameter and we got the following results:

Train score: 93.344%

Test score: 51.534%

The training score is still higher and the score only improved a little, from 51.41% to 51.53%

With the following metrics, we will be able to better evaluate the performance of our model's predictions with real values:

* MAE (Mean absolute error): “The MAE value itself indicates the average absolute error between predicted and actual values. The smaller the MAE, the better the model’s predictions align with the actual data”. (Ahmed, 2023)
* MSE (Mean Squared Error): “Represents the difference between the original and predicted values extracted by squared the average difference over the data set.” (DataTechNotes, 2019).

However, the effect of outliers in the data is most apparent with the presence of the square term in the MSE equation.

* RMSE (Root Mean Squared Error): “This is the most commonly used because it is measured in the same units as the target variable. The lower the RMSE, the better a model fits a dataset”. (BOBBITT, 2021)

Random Forest Regressor (test:30%):

MAE: 42254.42655540158

MSE: 2801065177.1177673

RSM: 52925.0902419426

R2 Score 0.5153488497639211

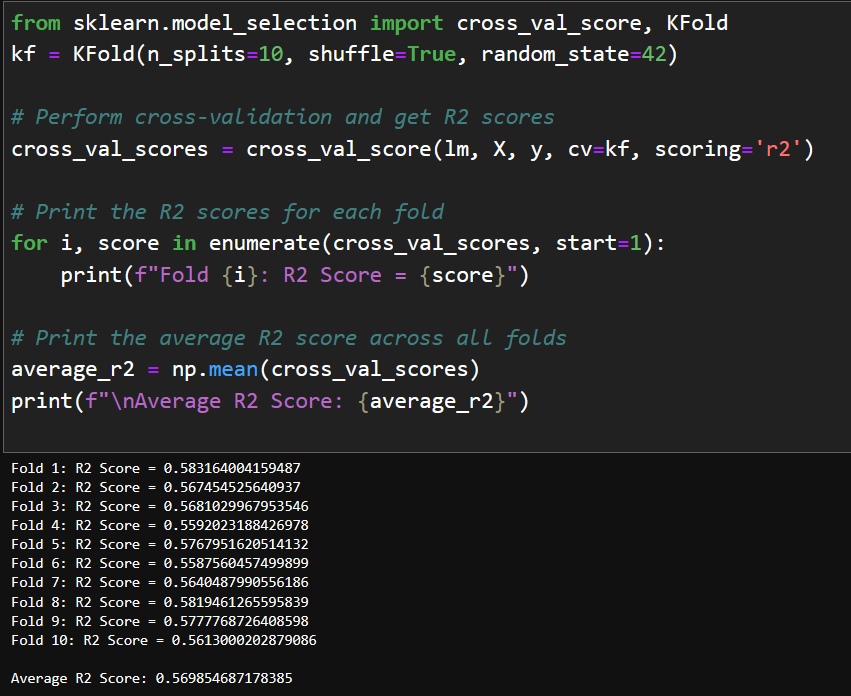
We can see that there is an error of 52,925 in the price of the predictions.

The next model is the:

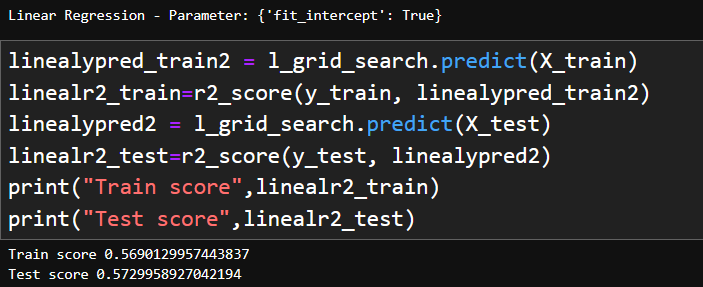
**Linear Regression**

We are going to use cross validation method to give a better understanding of model performance, that means that we are going to test our model with different Train Data and Test Data, in order to improve our model and avoid believing that the first Train Data is representative of all the data.

We used 10 and 5 splits. The average results of each Fold were better with 5 (57%) than with 10 (56.98%), so we will use 5 for the number of cross-validations



We also looked for the hyperparameter that turned out to be the default one and the results are as follows:



It performed better in the test.

The last model is the:

**K Nearest Neighbors**

This is one of the simplest models in machine learning. It doesn’t make any assumptions about the data distribution and the prediction value is calculated by averaging the target variable values of these neighbors.

We run the model with the default parameters and without cross validation and the train score is 65.6% and the test score is 48.33%.

Now we are looking to improve our model:

****

We use the 69 and uniform hyperparameters and the performance metrics are:

KNeighborsRegressor (test:30%):

MAE: 40315.6529065227

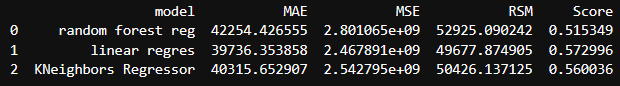
MSE: 2542795305.3622947

RSM: 50426.137125128815

Score: 0.5600357036936865

The test score improved and now is very similar to the train score (56.8%)

Here are the metrics of the 3 models with a 30% split:



The best to use is Linear Regression, it has a higher score and the RSM is lower, which means that on average you will see an error in the prediction of $49k

**Splitting Data into Training and Testing: 25%**

We directly search for the hyperparameter for our models and these are the results:

**Random Forest Regressor:**

Parameters: {'n\_estimators': 500}

Train: 0.9332155104078699 (93.3%)

Test: 0.5184008431963774 (51.8%)

**Linear Regression:**

Parameter: {'fit\_intercept': True}

Train score 0.5689513262542885 (56.89%)

Test score 0.5739334454304839 (57.39%)

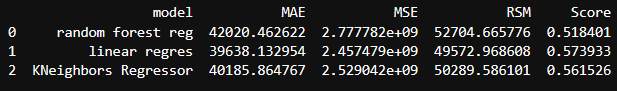
**K Nearest Neighbors:**

Parameters: {'n\_neighbors': 59, 'weights': 'uniform'}

Train score 0.5707451498568521 (57.07%)

Test score 0.5615261344423637 (56.15%)

Here are the metrics of the 3 models with a 25% split:



The best to use is Linear Regression, it has a higher score and the RSM is lower again.

**Splitting Data into Training and Testing: 20%**

**Random Forest Regressor:**

Parameters: {'n\_estimators': 600}

Train: 0.9331533085512287 (93.31%)

Test: 0.5217038189561088 (52.17%)

**Linear Regression:**

Parameter: {'fit\_intercept': True}

**Train score 0.5688921898995294 (56.88%)**

**Test score 0.5755662287751139 (57.55%)**

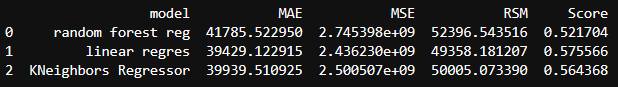
**K Nearest Neighbors:**

Parameters: {'n\_neighbors': 60, 'weights': 'uniform'}

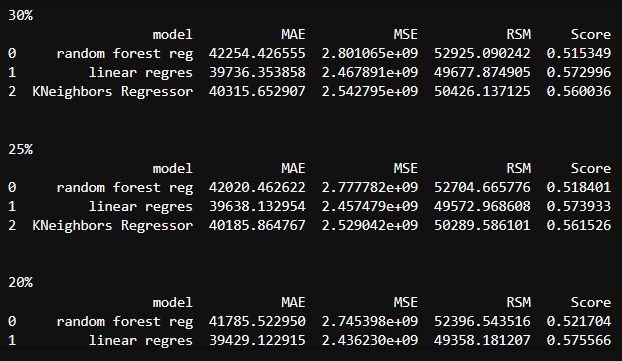
Train score 0.5709318951014181 (57.09%)

Test score 0.5643679996958288 (56.43%)

Here are the metrics of the 3 models with a 20% split:



Finally, we can compare the 3 splits with the metrics of the 3 models:



# Conclusion:

I can see that the lower the % given to the test, the better the results will be, with minor errors and higher scores. It's important to use hypermeters because this helped our models work much better. I had also tried another model like Lasso but the results were the same as Linear Regression. The most important variable that has the greatest relationship with price is size (square feet), which, as seen in the matrix, is the one with a correlation close to 1 The Linear Regression model has similar performance on training and testing The Random Forest Regressor model had good results in training set close to 1, but in testing set the results dropped, so it could indicate an overfitting The K Neighbors Regressor model had a better performance without the hyperparameters in the training set, but in testing set they were less than 50%, when applying the hyperparameters both results were more similar. In the end, it can be concluded that comparing the metrics, the best model to predict house prices in this dataset is the Linear Regression model

# References:

Ahmed, M.W. (2023). *Understanding Mean Absolute Error (MAE) in Regression: A Practical Guide*. [online] Medium. Available at: <https://medium.com/@m.waqar.ahmed/understanding-mean-absolute-error-mae-in-regression-a-practical-guide-26e80ebb97df>.

Beheshti, N. (2022). *Random Forest Regression*. [online] Medium. Available at: <https://towardsdatascience.com/random-forest-regression-5f605132d19d>.

Brownlee, J. (2020). *Hyperparameter Optimization With Random Search and Grid Search*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/>.

Github.io. (2020). *Data Science III with python (Class notes) - 4  Hyperparameter tuning*. [online] Available at: <https://nustat.github.io/STAT303-3-class-notes/Hyperparameter%20tuning.html> [Accessed 24 Apr. 2024].

Mastrandrea, G. (2022). *Correlation matrix, demystified*. [online] Medium. Available at: <https://towardsdatascience.com/correlation-matrix-demystified-3ae3405c86c1>.

MLK (2023). *Random Forest Regression in Python Sklearn with Example*. [online] MLK - Machine Learning Knowledge. Available at: <https://machinelearningknowledge.ai/random-forest-regression-in-python-sklearn-with-example/>.

Rout, A.R. (2020). *ML - Advantages and Disadvantages of Linear Regression*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/ml-advantages-and-disadvantages-of-linear-regression/>.

Shi, A. and K. (2023). *Nearest Neighbors Regressors — A Visual Guide*. [online] Medium. Available at: <https://towardsdatascience.com/nearest-neighbors-regressors-a-visual-guide-78595b78072e> [Accessed 24 Apr. 2024].

Stephen Allwright. (2022). *Using cross\_val\_score in sklearn, simply explained*. [online] Available at: <https://stephenallwright.com/cross_val_score-sklearn/>.

www.keboola.com. (2020). *The Ultimate Guide to Random Forest Regression*. [online] Available at: <https://www.keboola.com/blog/random-forest-regression>.

Zach (2021). *MSE vs. RMSE: Which Metric Should You Use?* [online] Statology. Available at: <https://www.statology.org/mse-vs-rmse/>.