

**ASSIGNMENT**

**MINING AND PREDICTIVE ANALYSIS**

**Student Name:**

**Student ID:**

**Module Code:** COMP 30044

**Professor:**

**Session:**

**TABLE OF CONTENTS**

Contents

[INTRODUCTION 1](#_Toc198428174)

[PROBLEM STATEMENT 1](#_Toc198428175)

[DETAILS OF DATASET 1](#_Toc198428176)

[PREPROCESSING 11](#_Toc198428177)

[REGRESSION MODELS 12](#_Toc198428178)

[PERFORMANCE METRICS 13](#_Toc198428179)

[CONCLUSION 16](#_Toc198428180)

[REFERENCES 16](#_Toc198428181)

# INTRODUCTION

In this project, I will show the use of two linear regression models, Linear Regression and Random Forest on a dataset consisting of several features about housing. The dataset contains several features that describe a house, and the price of the house is the target variable. Using the earlier mentioned models, I will prepare a housing price prediction algorithm that will use the same features from the dataset to predict the pricing of new houses. The project will see me describe the dataset in detail using existing functions, before performing preprocessing on it which will clean up the dataset. After this, the actual models are run, followed by analysis of its performance using various metrics to compare it against.

# PROBLEM STATEMENT

The goal of this assignment is to correctly predict the housing prices, based on an already available dataset consisting of multiple features about the house and its price. The dataset consists of 12 different features about the houses and another column that contains the pricing of every house on this dataset. The program will see me use some regression-based machine learning models that will be used to predict the house prices of new houses by training the model with this existing dataset.

# DETAILS OF DATASET

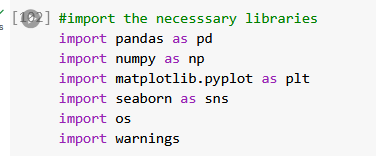
*Note: The dataset and python notebook have been uploaded to the GitHub link for this project. The link is available below.*

**GitHub Link:** [**mining-analysis/mining-assignment: A machine learning model that can predict house pricing, based on various features**](https://github.com/mining-analysis/mining-assignment)

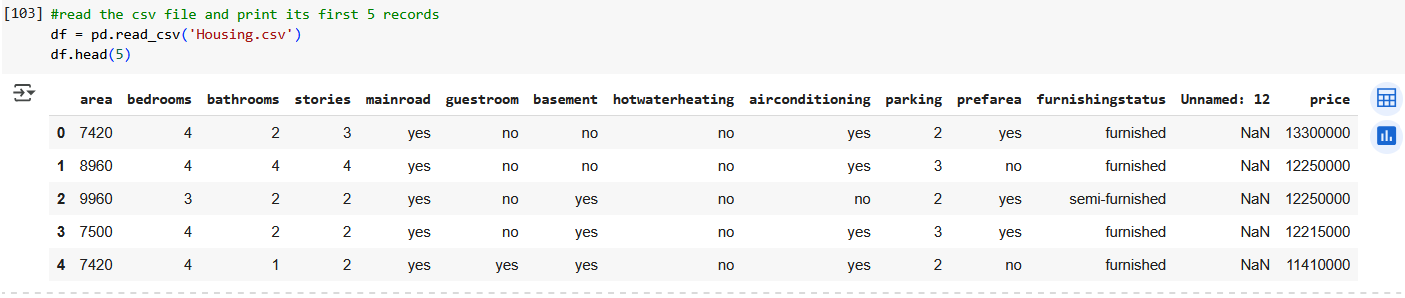
The dataset contains 545 records of various different houses with data on the following features about the house:

* **Area:** Lists the area covered by each house in square meters
* **Bedrooms:** Lists the number of bedrooms in each house
* **Bathrooms:** Lists the number of bathrooms in each house
* **Stories:** How many floors each house has
* **Main Road:** Whether a connection to the main road exists for each house or not
* **Guest Room:** Lists the number of guest rooms in each house
* **Basement:** Whether each house contains a basement floor or not
* **Hot Water Heater:** Whether water heating utility exists for each house or not
* **Air Conditioner:** Whether air conditioning exists for each house or not
* **Parking:** Whether parking space exists for each house or not
* **Pref Area:** Whether the house is located in a preferred (prime) area or not
* **Furnishing Status:** Whether the house is fully furnished with materials or not
* **Price:** Lists the price of each house

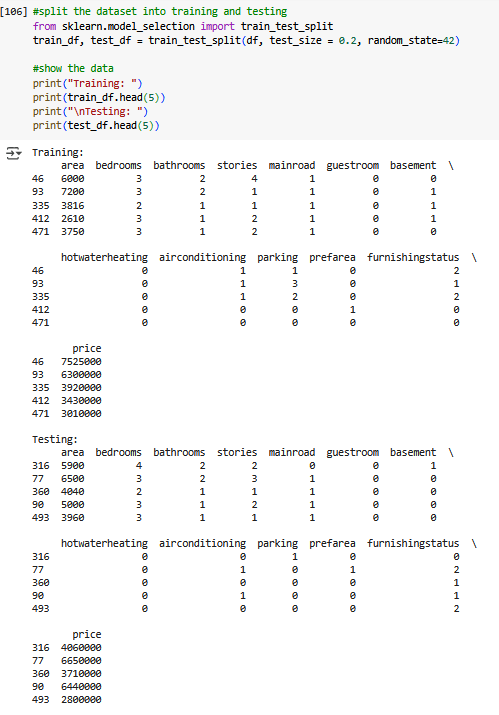
The machine learning model was prepared in Google Colab, a free online notebook tool on which python notebooks can be made. To begin with, I first uploaded the ‘housing.csv’ file online to be able to access its data.



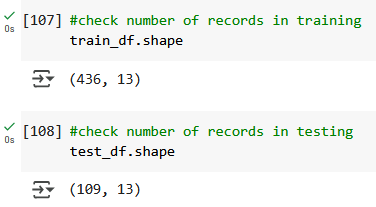
Firstly, the important libraries are imported into the project. Pandas is a library that is used for data manipulation and analysis in python. Numpy allows us to work with large, multi-dimensional arrays which contain a large amount of data in the form of matrices. Matplotlib consists of several plots(graphs) with which we can chart out various types of information in a visual manner. Seaborn is another library that is also used for visualization of data. OS allows us to interact with the operating system of the machine that the python notebook is running on. Warnings is a library that consists of features to deal with warnings that may pop up during the development of they python machine learning model (Gevorkyan et al., 2019).



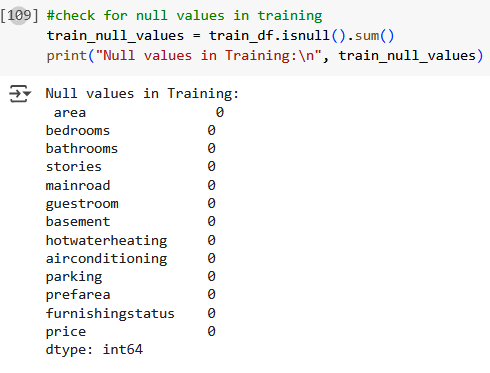
I first read the csv file using ‘pd.read\_csv(‘housing.csv’)’ command which analyzes the csv file. The head function, invoked in the next line, prints the first 5 rows of the dataset. I have started pre-processing here itself, but have kept the pre-processing section separate in this report.



The dataset is split into training and testing tests; the first 5 rows in each of the sets is printed after that.



The shape command lists the number rows and columns that are present in some kind of set. The training and testing sets’ shapes are printed here.

 A screenshot of a computer

AI-generated content may be incorrect.

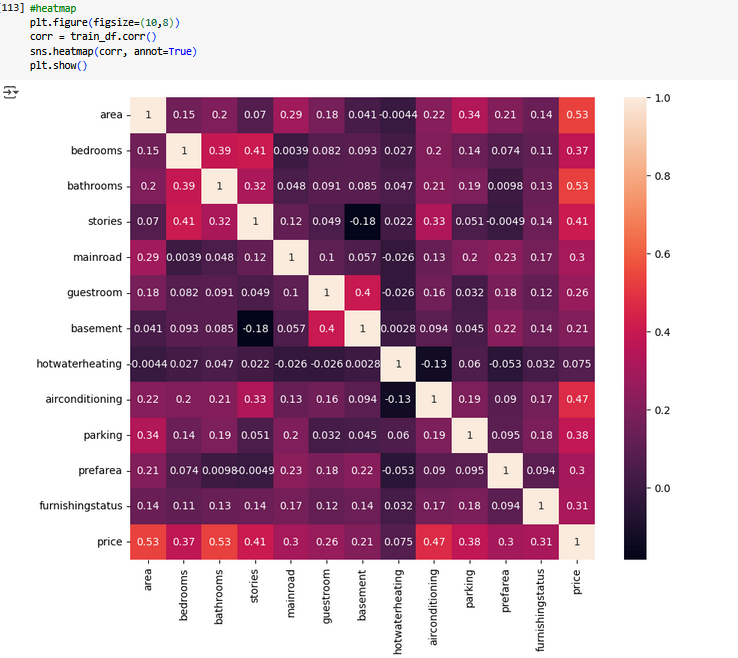
The ‘.isnull()’ function is run to check if either the training or testing sets have any kind of null values present in them. As shown, there exists no null value in either set (Stancin & Jovic, 2019).

 A screenshot of a computer

AI-generated content may be incorrect.

I also check what are the datatypes of each feature that are present in each set. As shown, both the training and testing dataset contain integer values.

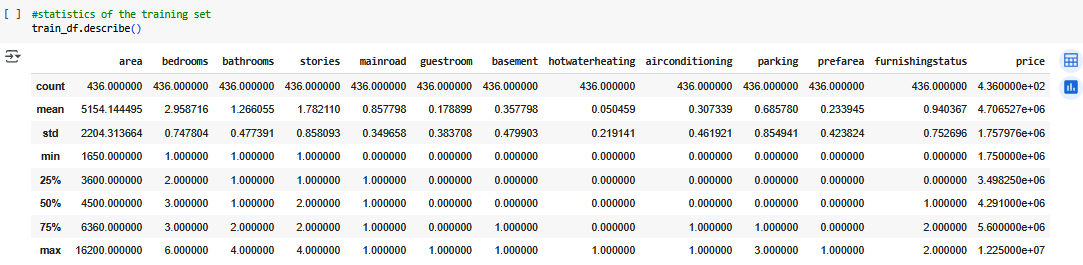
**Heatmap**



A heatmap is a technique that is used to create a visualization of some kind of data. It is used to understand the magnitude of values that exist in some kind of dataset. It uses some specific shades of colors to provide an easier understanding of the magnitude of each value with the colors changing from a gradient-like design. Numerical values are much easier to understand and compare using this method as the intensity of color change can signify how much or how little there is a difference between any two values in a dataset (Gu, 2022).

The heatmap shows the strongest correlation between the area and price features in the dataset. This can be understood as the price being directly proportional to the change in the amount of area covered by the house.

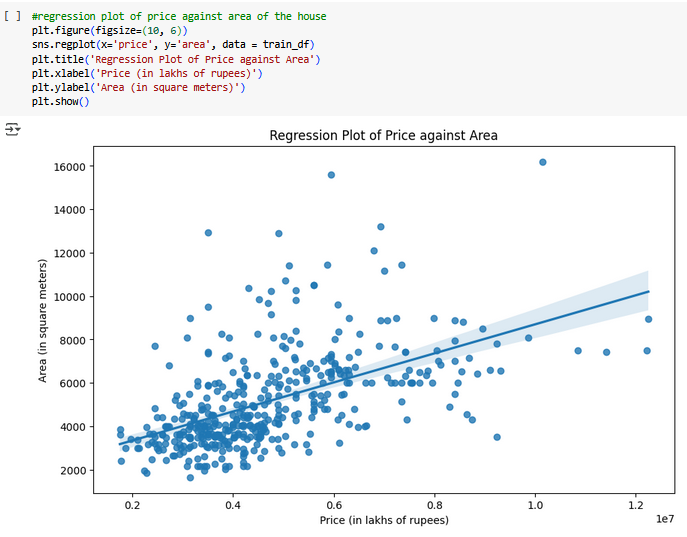
**Statistics**



The ‘describe()’ function provides the statistical value for each column in the dataset. The statistics shown include the following (Nagpal & Gabrani, 2019):

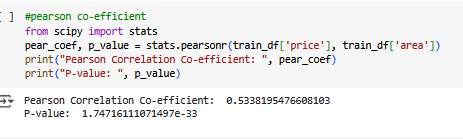
* **count:** Shows the number of non-null values in each column
* **mean:** Average value of each column
* **std:** Standard Deviation in each column (the spread of values)
* **min:** The minimum value in each column
* **25%:** The percentile; in other words, 25% of values in the column are below this value
* **50%:** The percentile; in other words, 50% of values in the column are below this value
* **75%:** The percentile; in other words, 75% of values in the column are below this value
* **max:** The maximum value in each column

**Regression Plot**



A regression plot shows the relationship between two variables that have some quantitative values associated with them. The regression line is the line that best fits the given model; in other words, the line covers most of the values within the given data. As seen in the above scatterplot, most of the values are closely related near the beginning of the scatterplot. They become more varied as the price and area increases (Calonico et al., 2015).

**Pearson Correlation Co-efficient and P-value**

****

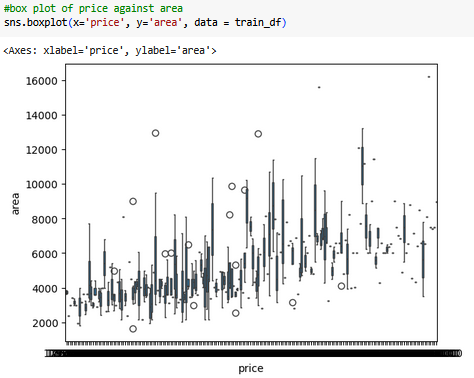
The pearson correlation coefficient is a score that measures the strength/proportionality of two different variables in a dataset, detailing their linear relationship. It is a value that ranges from (-1, 1). The score indicates how strong the linear correlation is between two features. In this case, the value indicates a somewhat strong ~0.533 score for the pearson score, which shows that the two features, price’ and ‘area’ are somewhat strongly linearly related to each other (Armstrong, 2019).

The p-value is an indicator of whether the correlation co-efficient itself is of significant importance; in other words, it tells if the pearson score is worth considering. The p-value indicates that the value is too high so it’s better to ignore the pearson score. In this case, it means that there is not enough reason to assume that the area covered by a home is proportional to its price increase (Komaroff, 2020).

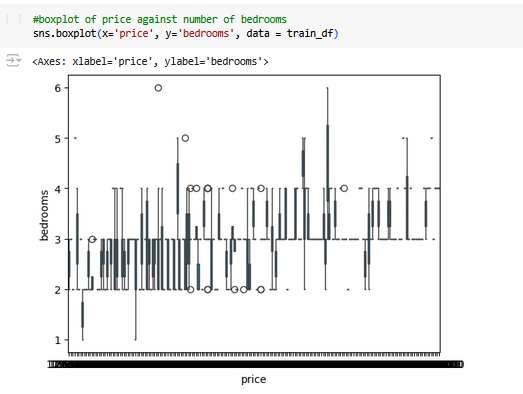
**Box Plot**

A box plot is a graphical visualization that shows the distribution of a dataset against a target variable. The box plot can be used to understand the median, quartiles, and outlier values when comparing among two variables. Box plots are useful for understanding when there are multiple features in a dataset and we want to compare the correlation between a particular feature and the target variable (Nuzzo, 2016).

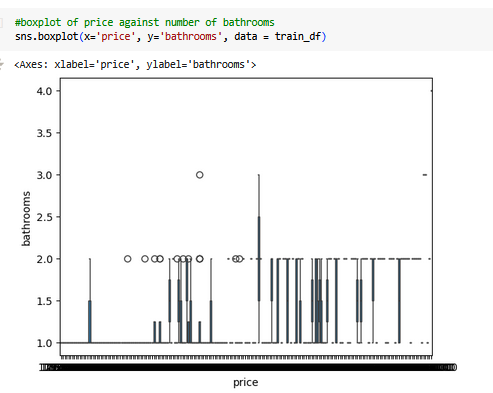
This box plot compares the price of the house against the area of the house. This plot suggests that as area of the house increases, the price of the house also increases.



This box plot compares the number of bedrooms in the house against its price. This does not show much of a correlation between the two variables.



This box plot compares the price of the house against the number of bathrooms in the house. There does not seem to be any relation between the price and number of bathrooms.



The box plot plots the price of the house against the number of floors in the house. Again, there does not seem to be a big correlation between the number of floors and the price of the house.

A screenshot of a computer screen

AI-generated content may be incorrect.

The box plot shows the price of the house against the number of guest rooms in the house. This shows one of the biggest variance, suggesting it to be very unlikely that both variables are related to each other in anyway.

A screenshot of a computer screen

AI-generated content may be incorrect.

This box plot checks the price of the house against the presence of a basement in the house. Again, there seems to be no correlation, whatsoever, between the two variables.

A screenshot of a bar code

AI-generated content may be incorrect.

This box plot shows the price of the furnishing status against the price of the house. The graph does seem to indicate some amount of correlation between those two variables.

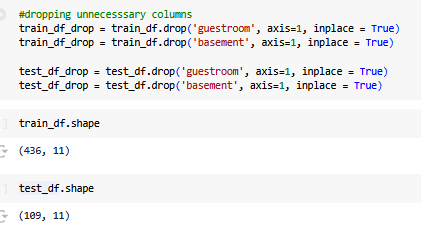
A screenshot of a barcode

AI-generated content may be incorrect.

# PREPROCESSING

Preprocessing is the technique of prepping the dataset to be implemented in a machine learning model to train it. Preprocessing ensures that unnecessary data is removed from the dataset to make it faster and more efficient to train the model and also reduces “noise” or bad data that could cause the model to make more wrong predictions than correct ones (García et al., 2016).

The ‘guestroom’ and ‘basement’ column are dropped from both the training and testing sets as they don’t contain any useful data for us to work with. The shape command shows the number of records in each set that we will be working with.



Once satisfied, we set up the training and testing sets; the x\_train contains all the features except the ‘price’, while the y\_train contains the ‘price’ column only

A computer code with black text

AI-generated content may be incorrect.

# REGRESSION MODELS

A regression model is a machine learning model that shows the relationship between some target variables and one or more independent variables. It is able to predict the value of the target variable using data received from the independent variables. A regression model is a good way to understand the proportionality of a variable to another variable, and if they are comparable or not (Cleveland et al., 2017).

I first import the required model library. The LinearRegression() function runs the model. The model is ran on the x\_train and y\_train datasets we split earlier, to train the model. The ‘train\_linear’ predicts housing based on this new training model.

A screenshot of a computer code

AI-generated content may be incorrect.

The Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are two metrics that are used to evaluate the performance of a regression model. It measures the difference between the predicted values from the model and the actual values from the dataset (Hodson, 2022).

MAE shows the average “Absolute Difference” between the values that were predicted and the actual values (Wang & Lu, 2018).

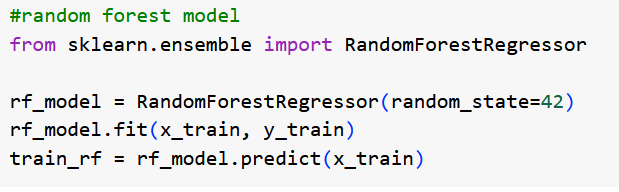
RMSE, meanwhile, shows the square root of the average of the squared errors from the training model. It is more sensitive to outlier values than MAE but is capable of displaying larger errors in a training model.

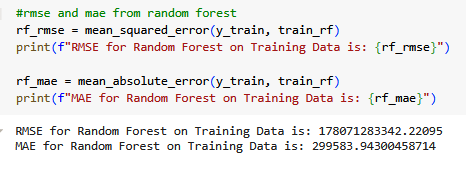
The linear model’s RMSE and MAE are listed below:

A computer screen shot of a program

AI-generated content may be incorrect.

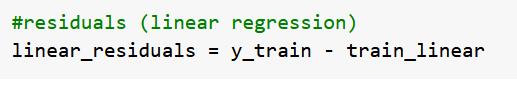
The same is repeated for the Random Forest Linear Regression model and its RMSE and MAE values are listed.





# PERFORMANCE METRICS

Performance metrics are the various metrics/solutions that exist to test the effectiveness of a machine learning model. These usually consist of some scores that can be used to measure the strength of a model (Padilla et al., 2020).



A graph of a distribution of residuals

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

Residuals are the numeric representation of the difference between the actual values and the values predicted by the model. Used for regression models, a residual is useful in understanding if a model is underestimating/overestimating the actual values and if the accuracy of the model is good enough. Residuals are good at understanding any patterns that show up in the trained model and find if there is any uneven spread of values (wrong values are getting predicted correctly) (Chen et al., 2018).

The bell-curve in the original distribution of residuals graph shows that values are quite normally spread out over the training model without too much bias to any particular value. The scatterplot shows that the spread of values is not very bad as even though a lot of values pass over the (-1, 1) threshold, there is a even spread value on either side, which means that values are being predicted on fairly good scale.

The same is repeated below for Random Forest model and its results have been given:

A close-up of a white background

AI-generated content may be incorrect.

A graph of a distribution of residuals

AI-generated content may be incorrect.

A screen shot of a graph

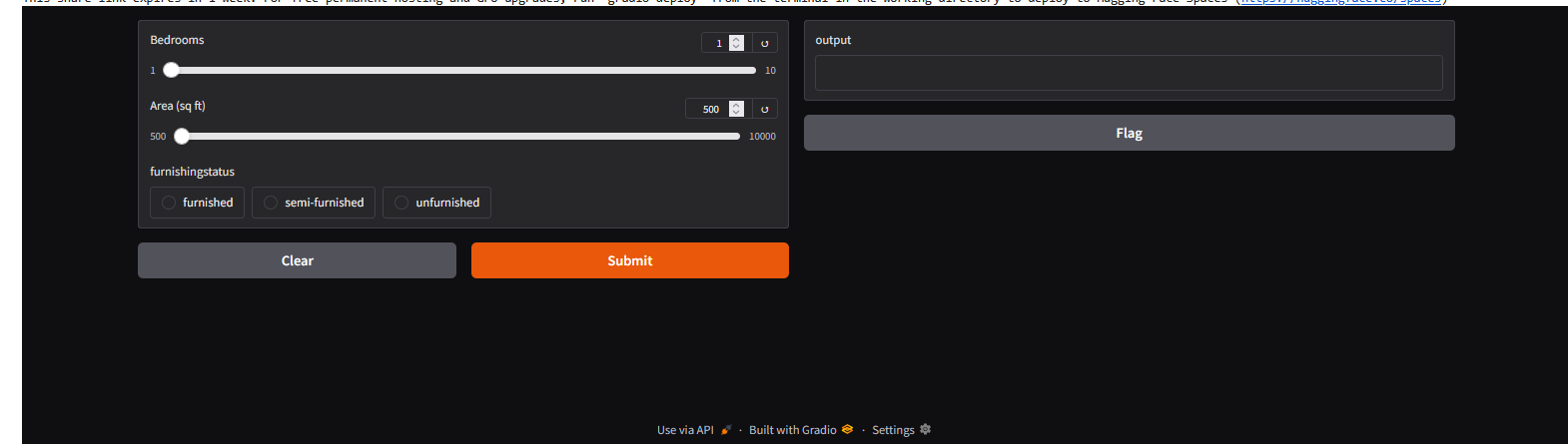
AI-generated content may be incorrect.

**USER INTERFACE**

A graphical user interface is something with which users can interact with to perform/execute some tasks on a computer. In this instance, I have used Gradio, an open-source framework, that allows me to create a graphical and interactive version of the machine learning model I just created, so that user can interact with Graphical User Interface (GUI), instead of having to run each cell in python notebook.

A white background with black text

AI-generated content may be incorrect.



The GUI here represents 3 options for the user to choose/set from. The user can set the number of bedrooms, the area of the covered by the house in square feet and the furnishing status of the house. Once all these values are set, the user can click the ‘Submit’ button and the model will generate an output result, which predicts the value of a new house, based on these set parameters.

# CONCLUSION

A screenshot of a computer code

AI-generated content may be incorrect.

From the results, we can conclude that Random Forest Regression model performs better than the Linear Regression model. This can understood to be, since Random Forest is more efficient at processing data in a dual manner that can also reduce time spent in training the model, as compared to Linear Regression model, which compares each value one-by-one with its direct counterpart, leading to higher rate of errors and training times. Overall, this proves that Random Forest is a good algorithm to create regression models and predict something on.

# REFERENCES

Armstrong, R. A. (2019). Should Pearson’s correlation coefficient be avoided? *Ophthalmic and Physiological Optics*, *39*(5), 316–327. https://doi.org/10.1111/OPO.12636

Calonico, S., Cattaneo, M. D., & Titiunik, R. (2015). Optimal Data-Driven Regression Discontinuity Plots. *Journal of the American Statistical Association*, *110*(512), 1753–1769. https://doi.org/10.1080/01621459.2015.1017578

Chen, W., Hribar, P., & Melessa, S. (2018). Incorrect Inferences When Using Residuals as Dependent Variables. *Journal of Accounting Research*, *56*(3), 751–796. https://doi.org/10.1111/1475-679X.12195

Cleveland, W. S., Grosse, E., & Shyu, W. M. (2017). Local Regression Models. *Statistical Models in S*, 309–376. https://doi.org/10.1201/9780203738535-8

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big Data Analytics 2016 1:1*, *1*(1), 1–22. https://doi.org/10.1186/S41044-016-0014-0

Gevorkyan, M. N., Demidova, A. V., Demidova, T. S., & Sobolev, A. A. (2019). Review and comparative analysis of machine learning libraries for machine learning. *Discrete and Continuous Models and Applied Computational Science*, *27*(4), 305–315. https://doi.org/10.22363/2658-4670-2019-27-4-305-315

Gu, Z. (2022). Complex heatmap visualization. *IMeta*, *1*(3), e43. https://doi.org/10.1002/IMT2.43

Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geoscientific Model Development*, *15*(14), 5481–5487. https://doi.org/10.5194/GMD-15-5481-2022

Komaroff, E. (2020). Relationships Between p-values and Pearson Correlation Coefficients, Type 1 Errors and Effect Size Errors, Under a True Null Hypothesis. *Journal of Statistical Theory and Practice*, *14*(3), 1–13. https://doi.org/10.1007/S42519-020-00115-6/METRICS

Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, 140–145. https://doi.org/10.1109/AICAI.2019.8701341

Nuzzo, R. L. (2016). The Box Plots Alternative for Visualizing Quantitative Data. *PM&R*, *8*(3), 268–272. https://doi.org/10.1016/J.PMRJ.2016.02.001

Padilla, R., Netto, S. L., & Da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. *International Conference on Systems, Signals, and Image Processing*, *2020-July*, 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130

Stancin, I., & Jovic, A. (2019). An overview and comparison of free Python libraries for data mining and big data analysis. *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2019 - Proceedings*, 977–982. https://doi.org/10.23919/MIPRO.2019.8757088

Wang, W., & Lu, Y. (2018). Analysis of the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) in Assessing Rounding Model. *IOP Conference Series: Materials Science and Engineering*, *324*(1), 012049. https://doi.org/10.1088/1757-899X/324/1/012049