PROPOSAL

Amsterdam Rent Price Document 1

MARCH 1, 2022

RAMY ALASHABI
AI43

ABSTRACT:

In this challenge housing rent in Amsterdam from various sources will be analyzed to validate their authenticity among different platforms and try to predict future developments based on a variety of features.

Contents

1. Scope and purpose of this document

For a detailed explanation on how we get from point A to point B, Please check the Jupyter notebook. As this is only a proposal of the project, a general overview on how this can be achieved(i.e., what model should be used based on what evidence), but more detailed on why and what is the purpose of the project.

2. Domain understanding

Domain knowledge is experimental research that enhances your own understanding of the subject.

2.1. Background

Amsterdam is the capital city of the Netherlands, known for its stunning history, the city has a standing legacy since the 17th century (golden age). Amsterdam has a population of 821,752, as the population grows, the numerous possible factors that affects the rent are also growing. A few examples of this are area of residence, amount and size of bedrooms, bathroom facilities and internet access. The sheer number of different marketplaces for housing are increasing rapidly. On top of that, these marketplaces do not all work the same way and may have varying features.

Therefore, companies like funda and kamernet were developed but, those companies display only the houses available, and sometimes these houses do not have the desired features. Thus, the user can not know the estimated price of a house, due to it not yet existing or not yet being available. In addition to that, if manual estimations of the house price were to take place, that would take quite the time, and it could be hours or days. Therefore, using Artificial intelligence to predict the rental price of a house in Amsterdam, based on the important features is valuable for homeowners or real estate agencies. Hence, no matter what the features of the house are, type of housing or location, the agency is able to provide the information easily.

In this project we will be looking at a dataset about the renting price in Amsterdam where I will check the rent prices of the houses based on important factors such as area, bedrooms, toilet, internet, etc. As to why is this project challenging, that is due to the data being mostly categorical, where I will be able to show my abilities for DAIA, Machine learning, and by turning this into a business model, it is benefitable for sociality.

2.2. How do I proceed.

The challenge will be made in Python using Jupyter Notebook where I am going to identify which variables have an impact on the housing price, make a model that predicts the price, and validate it. To summarize what will be extensively elaborated in the chapters below, bullet points will be used. The usage of this is to keep the reader on track from point A to B.

1. The problem statement will be described
2. More extensive research in to the domain will be done, by conducting an interview.
3. The stakeholders of the project, goal of project, and the delivery schedule will also be mentioned below.
4. Main question, sub questions and extra questions that needed more research will also be discussed below.
5. Data sourcing and how the data was sourced will also be addressed below. As we are going to use 1 data set from kamernet and another from huurwoning. These datasets will be merged, and compared with the original dataset from kamernet, to study and investigate the reason on why enhancing datasets is necessary.
6. Cleaning data and EDA will also be discussed below.
7. Model selection as to why we select the specific model. As we are going to use 4 different models on 2 datasets, twice. Once with hyper parameter tuning and another with non-hyperparameter tuning. So, in total we are going to build 16 models and evaluate them.
8. The model that performs the best will be chosen and a PowerBI application will be created based on that model.
9. An interactive and feasible dashboard will be made, to target the target audience.
10. A 2nd interview with a target audience will be made and mentioned in the report document of the 4th phase of the AI methodology.

So in total, 6 documents will be delivered in the end and they are :

1. **Proposal document:** That is this document, marked on canvas as “Phase 1 Proposal.pdf”. It is the first document to read.
2. **Transcription of domain expert interview:** this is a transcription of the first interview done with a domain expert. It is marked on canvas as” Phase 1 landlord Interview ”. It is the second document to read, but also skippable.
3. **Jupyter notebook:** This is the notebook where it covers all the codes, extra extensive research and details. It also shows you how we got from point A to point B. It is marked on canvas as “Phase 2 and 3 challenge notebook iteration 8 ”. It is the third document to read.
4. **Report:** This document shows the powerBi buildup, why and how. It also explains how did this project started and how it will end, as a delivery phase. It

also reports with respect to the goal and the Ai methodology phases. It is marked on canvas as" Phase 4 report " It is the fourth document to read.

5. **Transcription of target audience interview:** This is a transcription of the 2nd interview that was done with the target audience. This document is marked on canvas as "Phase 4 target audience interview". This is the fifth document to read.
6. **powerBI application:** this is a power BI application. It is an interactive, feasible for the eyes dashboard, that can show the scams easily.

2.3. Problem statement describe the problem

In the current situation an individual could use multiple websites with different selections of available housing to find a potential future home. The sheer number of these different marketplaces are increasing rapidly. On top of that, these marketplaces do not all work the same way and may have varying features. This can make it hard for somebody looking for a house to make a financial calculation regarding what they will be paying in rent. Especially considering certain homes which are not available right now might become available, while in the meantime not being listed on a website. (Koole, 2022)

This can make it hard to accurately predict what one could be paying for future accommodation, making it hard to do proper budgeting, both for new tenants and existing tenants who might want to move into a bigger house. (Student Helpr, n.d.)

These factors combined with the relative lack of knowledge among international newcomers about the rules and rights of renting in the Netherlands lead to another possible gap within the market. (I Amsterdam, 2020) People who fall into this category can be easily preyed upon and exploited by potential landlords through for example the state of delivery, extra fees, security deposits and other hidden costs which may or may not be illegal in nature. (DutchNews, 2019)

Especially that according to a personal interview conducted with an expert within this domain, where it was mentioned that most landlords are in this only for commercial business and a way of making money, using the fact that there are huge demands in the housing market. Hence, rent price inflation. Therefore, people would start paying on how much they can pay and not for how much the room or house is worth. (Koole, 2022)

These proceedings can be financially taxing which could lead to further problems which can usually only be resolved with legal disputes. However, a possible way to help combat the existence of this problem could be to combine the information from various marketplaces to validate among competitors how the information lines up or misaligns. While this does not inherently solve the problem of exploitative landlords it could help spread awareness of the actual fee and features which might not be visible through other platforms. (Micaela Zaslabsky, 2019)

2.4. Interview domain research summary

To enhance my domain understanding of my AI challenge, I have interviewed Jeroen Koole. who is an entrepreneur that has business in technical software and hardware, who is also a landlord of two buildings that are rented to companies and 2 houses that he bought for his sons. According to Jeroen, the houses which he bought for his sons are also rented to other student. When asked how he decided the price range of the rooms he rented to the others student, the reply was that he just compared it to how much rent he used to pay for his daughter and placed his range based on that. As there are differences in the rooms size and features, the price was reduced or increased based on the features. Following that answer, I wondered if prices are controlled by the government, to which he didn't have an answer for. But it was mentioned that since that there are shortage in rooms for students, landlords start asking for huge prices. (Koole, 2022)

There are institutes like funda and kamernet that can help, where students can go to and negotiate. But on the report of Jeroen, even though it is nice to have institutes like these, it is always impossible to start negotiating over the price with out knowing your grounds. Which seems like a problem that needs to be solved. In addition to that, there was an additional problem that was discovered and that is the current market of housing. Nowadays, people seem to pay more than what the house is worth and that is because they can and they want it, as the current market is still in favor of selling market and not towards the renting market.

So what is the reason behind the market being so negative. According to Jeroen and some research done (D.Talsma, 2021), it can be concluded that not only the features of the house that has an effect on the price, but also, if we follow the supply and demand chain, it will be noticed that there is a low supply and a very huge demand. Moreover, since that the mortgage interest is low, people are not afraid to loan more money from the bank just because they can, to pay extra for a house which is not worth.

In the opinion of Jeroen, this AI project could not be useful for landlords, as most landlords see this as a commercial business and a way to make money. Hence, they won't use the AI model to give the best price that is in favor of the tenant. But it could be used for makelaars, as an objective tool that calculates a reasonable price with a margin as that is very important. He additionally mentions that this project is very useful and resourceful for starts to know the base price with some off estimations, before starting the negotiation over the price with housing institutes and to identify scammers as, comparing the base predicted amount with the required amount on any website. As what they say in the Netherlands "meten is weten." (Koole, 2022)

The main drive of this interview was to discover new facts, explanations, research and enhance my domain understanding and to know what I do not know about the housing prices.

2.5. Stakeholders

Stakeholders of this project are all parties that are involved in it. Those are the people who have further influence in navigating the direction of this project and the end deliveries. In addition to that, their input has a direct impact on the outcome. Therefore, reliable and consistent collaboration and communication between the stake holders is essential for this challenge. The stakeholders are listed below.

Stakeholder	Role	Contact info
Georgiana Manolache	Delivers support and feedback on machine learning models and evaluations	g.manolache@fontys.nl
José Huisman	Delivers support and feedback on DAIA and data understanding.	Jose.huisman@fontys.nl
Nick Welman	Delivers support and feedback on documentations.	n.welman@fontys.nl
Ramy Fuad	Delivering the challenge and documentations	r.alashabi@student.fontys.nl
Stakeholder who are interested (i.e., landlords, new students, real estate agency)	Users of the application	

2.6. Goal

The goal of this project is to forecast the housing rent prices for potential local tenants as well as international newcomers with respect to their Budget, financial plan, and preferred features to spread awareness about fraud, scams and to acknowledge the base amount within the housing market by building various regression models.

2.7. Main question & sub questions

What would be the estimated rent price for a tenant in Amsterdam given a housing with specific coordinates, housing square meter and tenancy agreement (i.e., shared, or individual)?

- What would be the estimated price for a current housing?
- What would be the estimated margin of the price?

Some of these questions like the main and sub questions will be answered after the modeling and comparing. The interview questions (Please look at the transcription), some answers to the questions weren't satisfying enough, therefore a research into reliable articles is a must, as a backup.

2.8. Questions that needs more research

Is renting monitored by government ?

According to an article made by Walter block, maximum rental control isn't that efficient as it leads to deterioration of housings, less repairs made, and less maintenance. (Block, 2021)

As stated by the Dutch government website, what is applied in the Netherlands is a rent ceiling, where there is a maximum rent amount based on the worth/ quality of the house. (Ministerie van Binnenlandse Zaken en Koninkrijksrelaties, 2022) Hence, a prediction of the price with an off estimation/ margin is helpful.

As stated by Jeroen, he displayed his houses on kamernet and facebook, this implies that landlords could include their desired rental price on non-governed websites. As kamernet validates their landlords before adding them to their system (kamernet, 2022). Hence less scamming landlords. This answers the sub question of the reliability of the data.

Based on the interview, a person can identify scams, by understanding the worth of the house first and the base price based on the worth, then compared to the offer. In that, sense this AI project can have a huge impact in society, by spreading awareness and finding the base ground to start a negotiation with.

As to if this project could have a negative impact on society, it was researched that every technology has its moral positive and negative sides and it all starts based on the maker himself, as for what purpose was his project designed. (Santa Clara University, 2017) As claimed by Jeroen, this project seems to only have a positive side to it and will be used as a tool to help us, but it depends on the access to the data it self how secure the data is. As this project takes the data from kamernet which is a reliable source, due to it verifying the legitimacy of landlords before displaying anything (kamernet, 2022). Unlike if the data was collected from facebook, where it is not governed and could be an easy platform for scams.

2.9. Delivery schedule

Date	Delivery
1/04/2022	Iteration 0
18/04/2022	Iterartion 1 draft
24/04/2022	Iteration 1 draft 2
25/04/2022	Iteration 1 draft 3
29/04/2022	Iteration 1 final delivery
8/05/2022	Masterpiece challenge

3. Data Sourcing

The dataset that is used in this project contains all information available on kamernet.nl. the site was crawled daily and added to the database. Disclaimer, I did not crawl the data. [1] This dataset will be enhanced with extra data set from Huurwoning which has the same columns and some extra features.

Below is an example of what the dataset contains.

3.1. The housing price dataset

The housing data set 62 columns and 46722 rows. Most of these columns are non-necessary data such as: link, ID or match gender. So, to stay in our scope I chose some columns which I think are necessary and they are:

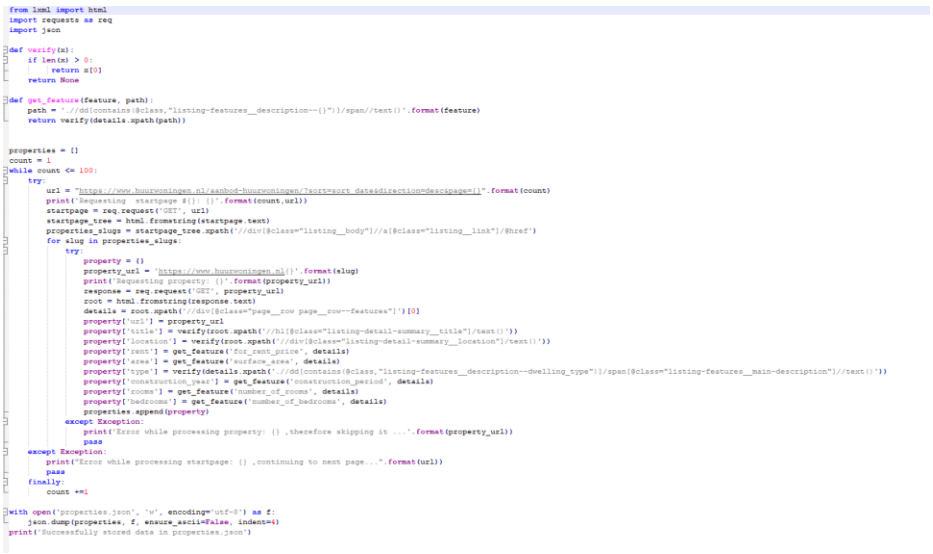
```
areaSqm          Area of the house in square meters
city             city
longitude        the longtitude of the house
latitude         the latitude of the house
toilet           if toilets is owned or shared
shower           if the shared or owned
smokingInside    if smoking inside is allowed or not
kitchen          if the kitchen is shared or owned
internet         if there is internet in the house or not
pets             if pets are allowed inside or not
living           if the general living room is owned or shared
propertyType     if the property is an apartment or a room
rent             rent price in Euro's
furnish          if the estate is furnished, semi furnished or not
```

3.2. Huurwoning dataset

The huurwoning data set consists of 9 columns and 1998 rows. Most of these columns are not necessary, therefore below is the columns that will be used:

Title	Title of the house which also contains the city name
Location	is a Zip code that shows where the house is located
Rooms	is the number of rooms of the house in total
Bedrooms	is the number of bedrooms of the house
Area	Is the area size of the house in square meters

This data set was crawled personally as an extra mile was taken, to learn how to crawl data as it can be seen below.



```
from lxml import html
import requests as req
import json

def verify(x):
    if len(x) > 0:
        return x[0]
    return None

def get_feature(feature, path):
    path = './/dd[contains(@class,"listing-features__description--{}")]/span//text()'.format(feature)
    return verify(details.xpath(path))


properties = []
count = 1
while count <= 100:
    try:
        url = "https://www.huurwoningen.nl/aanbod-huurwoningen/?sort=sort_dates&direction=desc&page={}".format(count)
        print('Requesting  startpage #{}: {}'.format(count,url))
        startpage = req.request('GET', url)
        startpage_tree = html.fromstring(startpage.text)
        properties_slugs = startpage_tree.xpath('//div[@class="listing__body"]//a[@class="listing__link"]/@href')
        for slug in properties_slugs:
            try:
                property = {}
                property_url = 'https://www.huurwoningen.nl{}'.format(slug)
                print('Requesting property: {}'.format(property_url))
                response = req.request('GET', property_url)
                root = html.fromstring(response.text)
                details = root.xpath('//div[@class="page__row page__row--features"]')[0]
                property['url'] = property_url
                property['title'] = verify(root.xpath('//h1[@class="listing-detail-summary__title"]/text()'))
                property['location'] = verify(root.xpath('//div[@class="listing-detail-summary__location"]/text()'))
                property['rent'] = get_feature('for_rent_price', details)
                property['area'] = get_feature('surface_area', details)
                property['type'] = verify(details.xpath('.//dd[contains(@class,"listing-features__description--dwelling_type")]/span[@class="listing-features__main-description"]//text()'))
                property['construction_year'] = get_feature('construction_period', details)
                property['rooms'] = get_feature('number_of_rooms', details)
                property['bedrooms'] = get_feature('number_of_bedrooms', details)
                properties.append(property)
            except Exception:
                print('Error while processing property: {} ,therefore skipping it ...'.format(property_url))
                pass
    except Exception:
        print("Error while processing startpage: {} ,continuing to next page...".format(url))
        pass
    finally:
        count +=1

with open('properties.json', 'w', encoding="utf-8") as f:
    json.dump(properties, f, ensure_ascii=False, indent=4)
print('Successfully stored data in properties.json')
```

Figure 1 crawling code

The figure above shows the code used to crawl the data from huurwoning.

4. Data Challenge

In this project, the enhanced data set will be used to build the model as well as the original dataset. So that both datasets can be compared, to deduce which one of them performs better.

Before we start with the EDA we would like to tidy the data and make sure it is clean, so that reliable and accurate information can be extracted. Then we will do EDA in the merged data and the original data. As during modeling phase we will compare between the metrics of the enhanced dataset and the original dataset.

Here I am going to explain how I am going to reach to my results, step by step.

4.1. Loading libraries

We first import the needed libraries numpy, which is used for numerical calculations. Pandas data analysis tools and sklearn for machine learning libraries.

```
import numpy as np
import pandas as pd
import sklearn as sk
import matplotlib
import matplotlib.pyplot as plt
from IPython.display import display
pd.set_option('max_columns', None)
```

Figure 2 libraries imported

4.2. Loading data

Here I load the data using read json, as the data I got was in a json file. The lines is set to true, to not read the non-sense data

```
hou3=pd.read_json(r'C:\Users\ennop\Documents\remy\challenge\properties.json', lines=True)
hou3.head(2)
```

```
hou3.info()
```

Figure 3 Reading json file

4.3. Cleaning data

Here we start by choosing the columns we want to run our study on

Figure 4 columns chosen

```
house=hou3[['areaSqm','city','longitude','latitude','toilet','shower','smokingInside','kitchen','internet',
            'pets','living','propertyType','rent','furnish','crawlStatus']]
```

After that we start looking for NA values

```
house.isnull().sum()
```

```
areaSqm            0
city               0
longitude          0
latitude           0
toilet           100
shower           100
smokingInside    100
kitchen          100
internet         100
pets             100
living           100
propertyType       0
rent               0
furnish            0
crawlStatus        0
dtype: int64
```

```
house[house.isnull().any(axis=1)]
```

Figure 5 searching for NA

As it can be seen there are a lot of missing values in this data set. How are we going to approach this? If we looked further in the data we will notice that some data are not filled in or called “unknown” and If we look at the crawlstatus columns we will see that it consists of uncrawled and crawled data. So let us drop the unavailable rows which crawling couldn’t get in this column and change the values of the missing data to NAN.

```
house['crawlStatus'].unique()
```

```
array(['done', 'unavailable'], dtype=object)
```

Figure 6 checking unique values

Using the following code we change to NAN and drop unavailable rows

```
house = house.replace({'': np.NaN, 'Unknown': np.NaN})
house = house.drop(house[house.crawlStatus == 'unavailable'].index)
house
```

Figure 7 replacing empy values to nan

If we look at the data now we notice that the number of NAN values have increased, how do we still deal with this problem.

```
house.isnull().sum()
```

```
areaSqm             0
city                0
longitude           0
latitude            0
toilet           7660
shower           7631
smokingInside       0
kitchen          7628
internet         8146
pets                0
living           8551
propertyType        0
rent                0
furnish           362
crawlStatus         0
dtype: int64
```

Figure 8 checking for NA

Lets try filtering the city for Amsterdam only and check again

```
house=house[house['city'].str.contains("Amsterdam")]
```

```
house.isnull().sum()
```

```
areaSqm             0
city                0
longitude           0
latitude            0
toilet            998
shower            982
smokingInside       0
kitchen           980
internet          989
pets                0
living           1124
propertyType        0
rent                0
furnish             4
crawlStatus         0
dtype: int64
```

Figure 9 Checking for NA after filtering

As it can be observed the NAN values have decreased. So, time to impute. Most of these data are missing completely by random, and that is because the data wasn't crawled completely or could be that the data was missing on the website. So, I would choose to impute using the mode as it is the most frequently repeated value. For every column

```
data=house
```

```
data['toilet'].mode()
data['toilet'].fillna('Shared',inplace=True)
data['toilet'].isnull().sum()
```

Figure 10 Filling missing data with median

So that we don't mess with our original data we save it in a new dataframe and do the calculations.

```
data.isnull().sum()
```

```
areaSqm          0
city             0
longitude        0
latitude         0
toilet           0
shower           0
smokingInside    0
kitchen          0
internet         0
pets             0
living           0
propertyType     0
rent             0
furnish          0
crawlStatus      0
dtype: int64
```

Figure 11 checking for NA

And here is the final na's checking.

4.4. EDA

4.4.1. Numerical EDA

We first start looking into the areaSqm column. What information can we get from there.

```
: data.describe()
```

:

	areaSqm	longitude	latitude	rent
count	8074.000000	8074.000000	8074.000000	8074.000000
mean	31.988358	4.888667	52.358590	970.569854
std	29.334260	0.053321	0.024571	513.987638
min	6.000000	4.771238	52.289948	1.000000
25%	12.000000	4.851282	52.347484	620.000000
50%	18.000000	4.887185	52.359754	795.000000
75%	48.000000	4.926215	52.374210	1250.000000
max	280.000000	5.018227	52.423914	5000.000000

Figure 12 data Description

Using the describe function, we notice that the minimum value for the areaSqm is 6, the max is 280 and the average is almost 32. Let's dive in further, to get more information.

```
binwidth= 8
plt.hist(data.areaSqm,bins=range(min(data.areaSqm),max(data.areaSqm)+binwidth,binwidth))
plt.show()
```

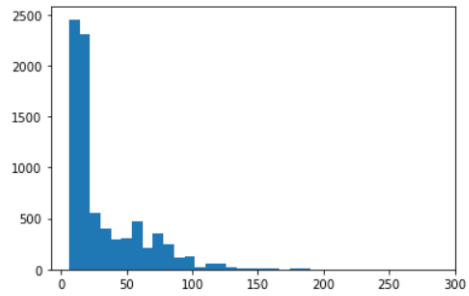


Figure 13 Area Sqm histogram

By plotting the histogram we can see that it is right skewd, which means that it could give more errors and overestimates the outcome variable. In addition to that, it can be seen that the concentration is mostly between 0 and 50, that's is our median, and this could tell us that most people could not afford big houses.

To visualize all numeric values using the seaborn library

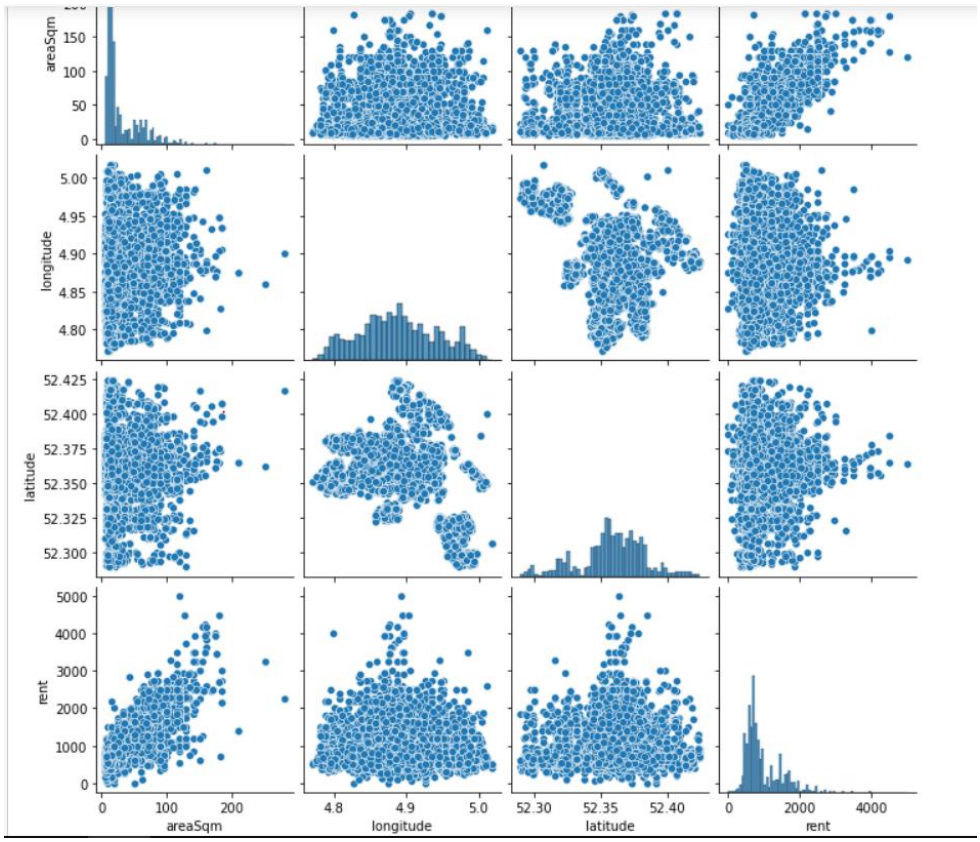


Figure 14 Numeric values visualziation seaborn

What can be told is that there is a positive correlation between the areaSqm and the rent.

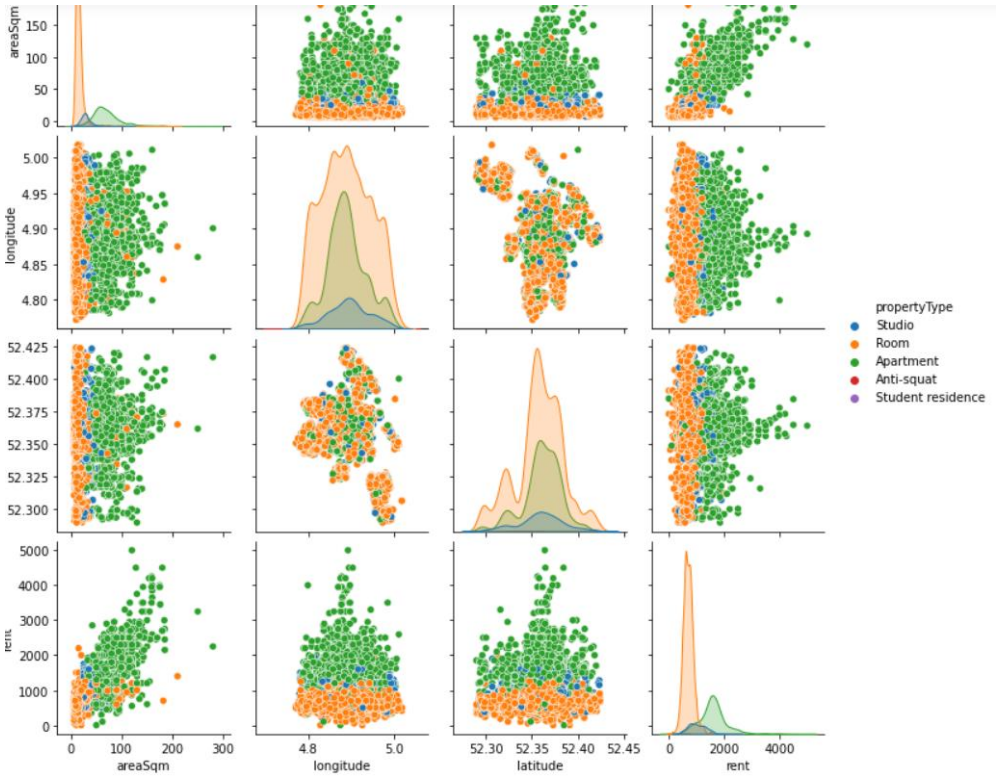


Figure 15 numeric value visuals with property type

What can also be seen between ''rent'' and ''areaSqm'' using the ''property type'' as a hue, is that apartments are on the higher end, studios are in the mid range and rooms are at the low range. Which is interesting as this can prove my hypothesis of area and property type has an impact on rent.

4.4.2. Categorical EDA

Using the box plot visualization, we can see some interesting EDA about the houses. Lets study them one by one.

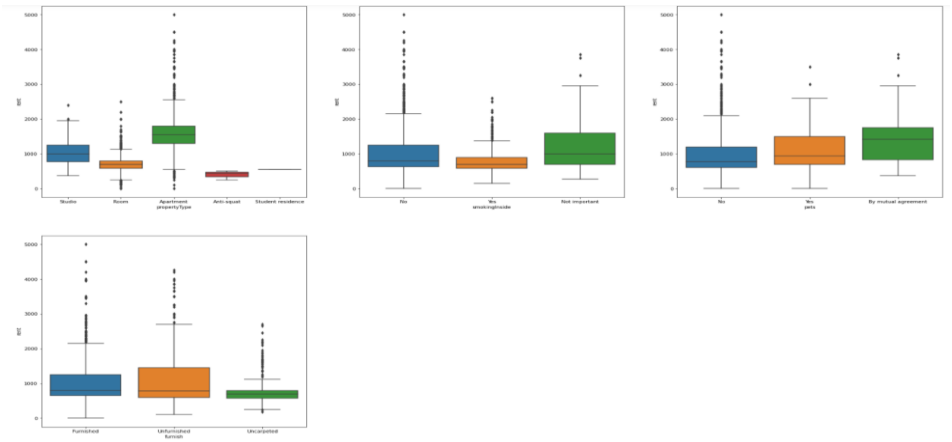


Figure 16 Categorical data box plot

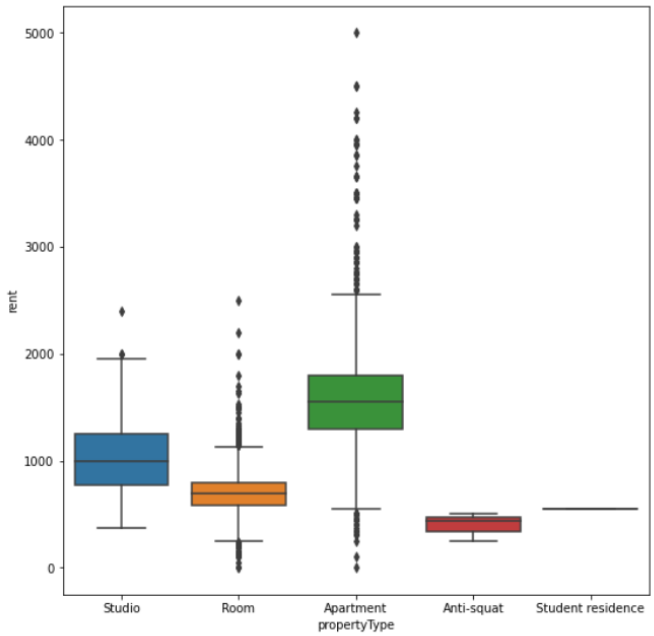


Figure 17 property type box plot

As it can be seen in the boxplot of the property type, we can see only a line for the student resident, which means that we don't have much information about it, but on the other hand, it seems that anti-squat houses are the cheapest and the less wanted and the median falls under 500. 50% of the data for apartments is concentrated on 1300-1800, the rest 50% is between the minimum and maximum of 800-2700 removing the interquartile range. For the rooms type, 50% of the data fall on 700-900 and the rest is between 400-1100 removing the interquartile range. For studios 50% of the data falls between 900-1100 and the rest is between 500-2000 removing the interquartile range. What is interesting here is that the apartments type seem to have a normal distribution following the formula Q3-Q2=Q2-Q1, with lots of outliers, but for the others we see that they are negatively skewd based on the formula of Q3-Q2<Q2-Q1.In addition to that what can be told from the box plots is that you can see the properties are over-lapping, which prevents the machine from differentiating between the different property types.

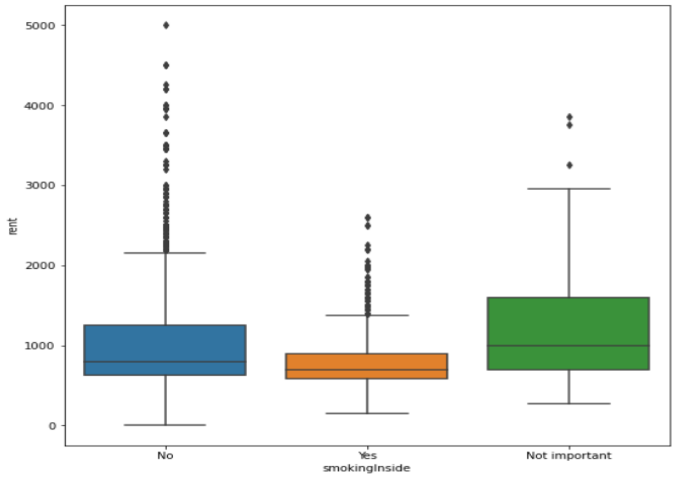


Figure 18 allowed smoking box plot

What can be seen here is that the houses with no smoking allowed have 50% of the data between 700-1100 and the rest is between the min and max, removing the interquartile range. Furthermore, it is negative skewd based on the formula of Q3-Q2<Q2-Q1. What is interesting is that the houses with yes to smoking inside has 50% of the data between 700-900 and it is negatively skewd. The not important category is positively skewd based on the formula Q3-Q2>Q2-Q1.

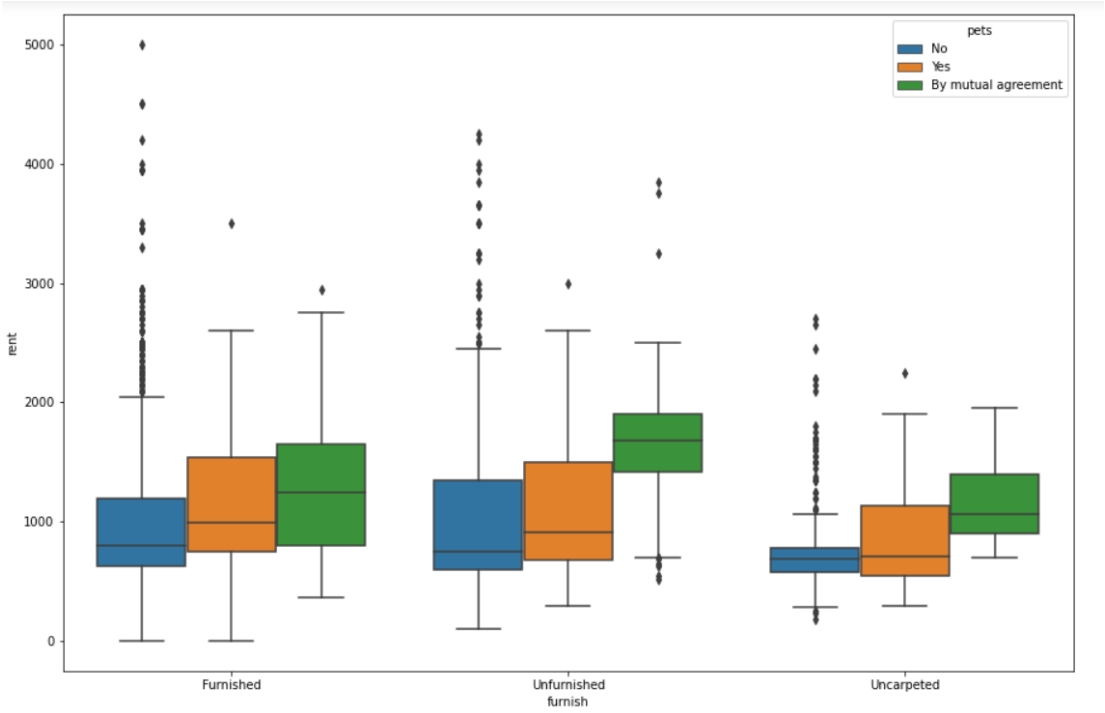


Figure 19 furnish box plot with pets

Now we look at the distribution of furnish houses, with pets as a hue. Furnished houses it can be seen that the price increases by pets, same goes for the other categories. Now if we compare the furniture types with no pets allowed, you can see that the uncarpeted houses are less wanted as it has a max of 1100 and 50% of the data is concentrated on 800-500. Comparing the furnished and unfurnished houses, it can be seen that 50% of the data of furnished houses is between 1100-700, mean while the unfurnished is between around 800-1200, this shows that the unfurnished houses could be more expensive than the furnished houses.

4.4.3. Map

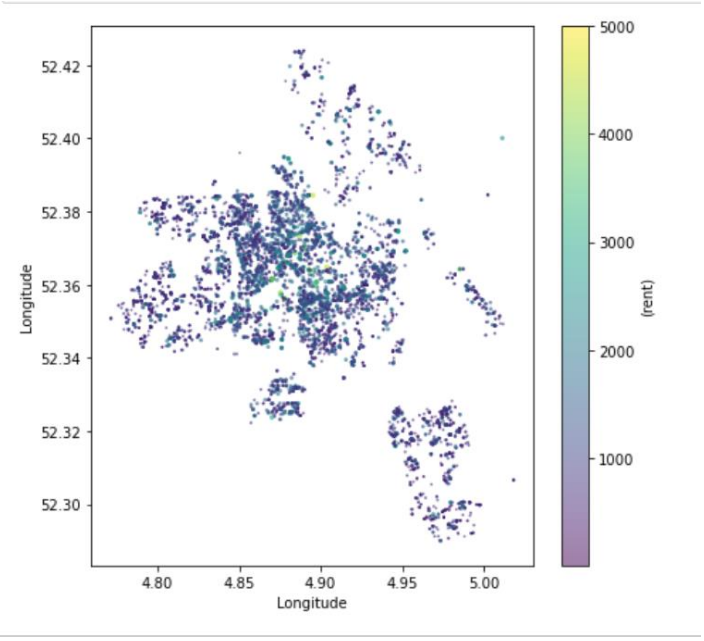


Figure 20 heat map of rent area

Looking at the map we can see that the further you get away from the city the less expensive the rent is.

4.5. Correlation heat map

4.5.1. Numerical correlation heat map

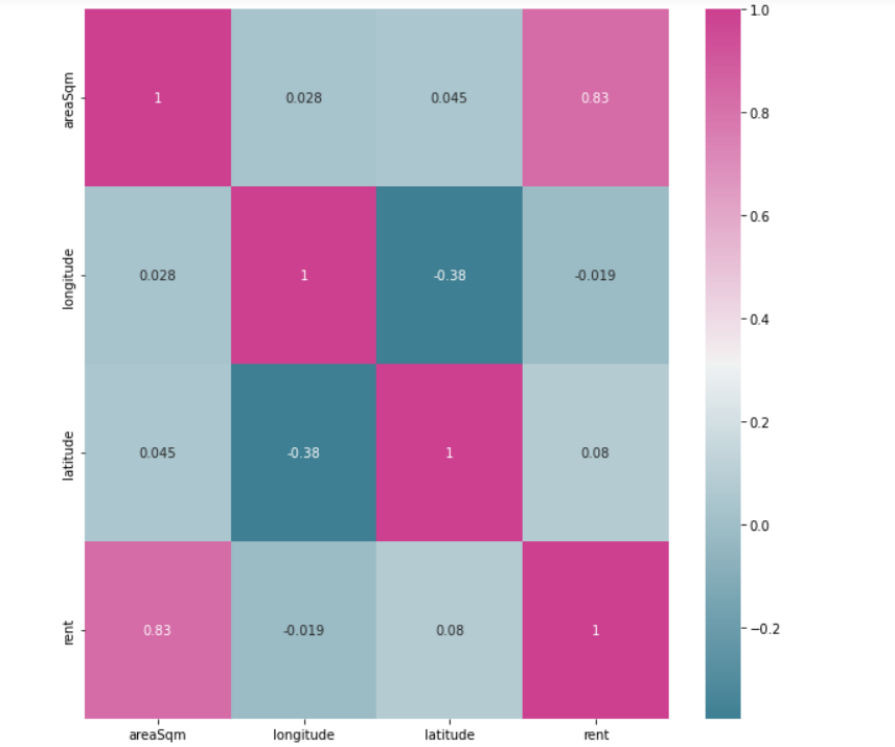


Figure 21 correlaton heat map

Through this heatmap, it can be observed that areasqm has a high correlation of 0.83 with the rent price.

4.5.2. Categorical correlation heat map

Using the Python library, we were able to plot the correlation heat map for also the categorical data.[3]

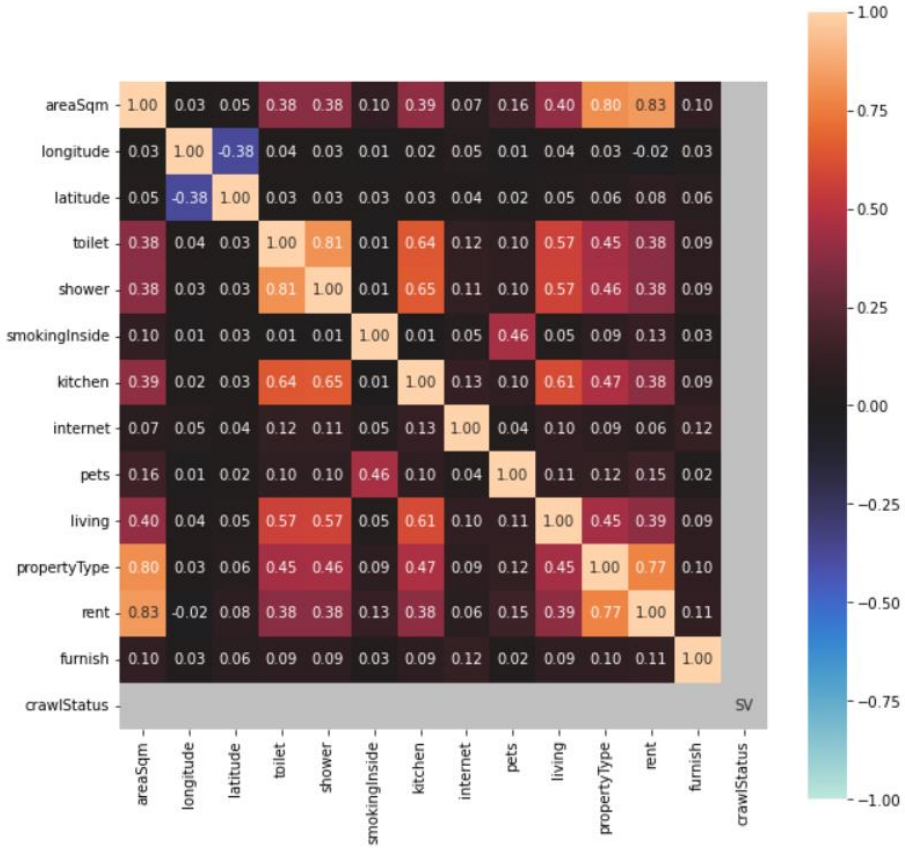
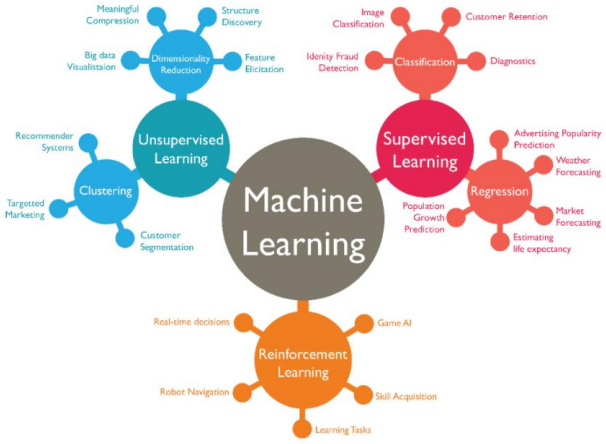


Figure 22 correlation heat map

It can be seen that rent has a high correlation with property type with a correlation of 0.77, it also has low correlation with furnish 0.11. Rent also has a negative correlation of -0.38 with longitude and latitude. Rent also has a low correlation between smoking inside and pets of 0.13 and 0.15.

4.6. Nature of problem

Depending on the nature of our target variable(rent) , and thus the nature of the problem at hand, a suitable type of algorithm needs to be chosen. Note that not every algorithm suits every problem. Some are better at this, and some are better at that. Choosing the "right tool for the job" is essential in obtaining high quality results. So referring the diagram below



Problem	Algorithm type
Is this A, B or C?	Classification
How many/much?	Regression
Is this true or false?	Logistic / Sigmoid
How is this organised?	Clustering
What should I do now?	Reinforcement learning

Figure 23 analytic approach

I would be choosing regression as it is a supervised learning, and we are trying to predict how much. Here are the different algorithms that will be used and compared with each other:

1. Linear regression
2. Ridge regression
3. Decision tree regression
4. Random forest regression

The reason to why we use regression in here, it is because it is a prediction of how much and so we use regression. Due to regression having multiple algorithms, we will use a couple of them and compare between them. Furthermore, we would do the modeling on the merged data set and the original dataset to compare. finally we will also do the modeling between both data sets with and without hyper parameter tuning.

The evaluation of the models will be done by using 3 different measurements, which are:

1. **RMSE:** which isa root mean square error, where it measures the error in a model in predicting quantifiable data. In simple words, it shows how much is the model over estimating or underestimating.
2. **MSE(Train and Test):** which is the mean squared error, where it measures the average squares of the error. That is the difference in averaged square of the estimated value and the actual value. The closer this value to zero is, the better the model is.
3. **R squared(Train and Test):** is a measure of fit that is used in regression models, which indicates the variation of a dependent variable that can be explained by an independent variable. The closer this value to 1 is, the better the accuracy is.

5. Data preparation

In here we will prepare the data sets to be fed into the model by hot encode both datasets the merged data and the original data for the reason of comparison. Since that most of the features that we have are categorical we are going to use a general approach that does not presume ordering. Therefore, we are going to use hot encoding with binary numbers were the instance either belongs to 0 or 1. That is also because that regression algorithm do not take objects or strings, it only accepts numerical values. Therefore, The best approach for our categorical values is to hot encode them to numerical with binary numbers were the instance either belongs to 0 or 1. Finally, we split the data before we start modeling, to prevent over fitting and to obtain a realistic evaluation of the model.

5.1. Merged data

The figure below shows how the hot encoding was done, as how the categorical variables changes to numerical and stays unique.

```
# here we map all to either 0 or 1
prepdata['owntoilet'] = prepdata['toilet'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
prepdata['sharedtoilet'] = prepdata['toilet'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
prepdata['notoilet'] = prepdata['toilet'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)

# here we map all to either 0 or 1
prepdata['ownshower'] = prepdata['shower'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
prepdata['sharedshower'] = prepdata['shower'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
prepdata['noshower'] = prepdata['shower'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)


# here we map all to either 0 or 1
prepdata['ownkitchen'] = prepdata['kitchen'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
prepdata['sharedkitchen'] = prepdata['kitchen'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
prepdata['nokitchen'] = prepdata['kitchen'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)



# here we map all to either 0 or 1
prepdata['ownliving'] = prepdata['living'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
prepdata['sharedliving'] = prepdata['living'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
prepdata['noliving'] = prepdata['living'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)




# here we map all to either 0 or 1
prepdata['kamer'] = prepdata['propertyType'].map(  { 'Room': 1, 'Apartment': 0} ).astype(int)
prepdata['appartement'] = prepdata['propertyType'].map(  {'Room': 0, 'Apartment': 1}  ).astype(int)


#prepdata['rooms1']  = huurwon['rooms'].map(   { '1.0': 1, '2.0': 0, '3.0':0, '4.0':0 } ).astype(int)
#prepdata['rooms2'] = huurwon['rooms'].map(  {'1.0': 0, '2.0': 1, '3.0':0, '4.0':0 }  ).astype(int)
#prepdata['rooms3'] = huurwon['rooms'].map(  {'1.0': 0, '2.0': 0, '3.0':1, '4.0':0 }  ).astype(int)
#prepdata['rooms4'] = huurwon['rooms'].map(  {'1.0': 0, '2.0': 0, '3.0':0, '4.0':1 }  ).astype(int)


prepdata
```

Figure 24 Hot enconding of the merged data set

```
from sklearn.model_selection import train_test_split
# in X we drop the columns which are already hot encoded and the non correlated columns too
X = prepdata.drop(['rent', 'toilet',
    'shower',
    'kitchen',
    'living',
    'propertyType','rooms'], axis=1)
y = prepdata['rent']
# we split the data in training and testing we choose also the 30% of the data to test it and 70% to tr
X_train, X_test, y_train, y_test = train_test_split(
     X, y, test_size=0.3, random_state=42
)
```

Figure 25 Preprocessing of the merged data

In X we drop the columns which are already hot encoded and the non-correlated columns too. Then we split the data in training and testing. we choose also the 30% of the data to test it and 70% to train it with a random state of 42 to start the internal random number generator and split the data randomly.

5.2. original data

The figure below shows how the hot encoding was done, as how the categorical variables changes to numerical and stays unique.

```
# here we map alltoilet categories to columns conaining only  either 0 or 1
house['owntoilet'] = house['toilet'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
house['sharedtoilet'] = house['toilet'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
house['notoilet'] = house['toilet'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)

# here we map all shower categories to columns conaining only  either 0 or 1
house['ownshower'] = house['shower'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
house['sharedshower'] = house['shower'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
house['noshower'] = house['shower'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)


# here we map all kitchen categories to columns conaining only  either 0 or 1
house['ownkitchen'] = house['kitchen'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
house['sharedkitchen'] = house['kitchen'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
house['nokitchen'] = house['kitchen'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)



# here we map all living room categories to columns conaining only  either 0 or 1
house['ownliving'] = house['living'].map( {'Own': 1, 'Shared': 0, 'None': 0} ).astype(int)
house['sharedliving'] = house['living'].map( {'Own': 0, 'Shared': 1, 'None': 0} ).astype(int)
house['noliving'] = house['living'].map( {'Own': 0, 'Shared': 0, 'None': 1} ).astype(int)


# here we map all property types  categories to columns conaining only  either 0 or 1

house['studio'] = house['propertyType'].map( {'Studio': 1, 'Room': 0, 'Apartment': 0, 'Anti-squat': 0,
house['room'] = house['propertyType'].map(  {'Studio': 0, 'Room': 1, 'Apartment': 0, 'Anti-squat': 0,
house['appartement'] = house['propertyType'].map(  {'Studio': 0, 'Room': 0, 'Apartment': 1, 'Anti-squat
house['anti'] = house['propertyType'].map(  {'Studio': 0, 'Room': 0, 'Apartment': 0, 'Anti-squat': 1,
house['student res'] = house['propertyType'].map(  {'Studio': 0, 'Room': 0, 'Apartment': 0, 'Anti-squat


house
```

Figure 26 Hot encoding of Kamer net data

```
# in X we drop the columns which are already hot encoded and the non correlated columns too
X_orig = house.drop(['rent', 'toilet',
    'shower',
    'kitchen',
    'living',
    'propertyType'], axis=1)
y_orig = house['rent']
# we split the data in training and testing we choose also the 30% of the data to test it and 70% to tr
X_train_orig, X_test_orig, y_train_orig, y_test_orig = train_test_split(
     X_orig, y_orig, test_size=0.3, random_state=0
)
```

Figure 27 preprocessing of the original data from kamernet

In X we drop the columns which are already hot encoded and the non-correlated columns too. Then we split the data in training and testing. we choose also the 30% of the data to test it and 70% to train it with a random state of 42 to start the internal random number generator and split the data randomly.

So now we have hot encoded the data, so that every unique value per row has its own column and own Unique number as it can be seen above. We also preprocessed these data sets and prepared them for modeling where we will have to drop the categorical Object columns and use the hot encode.

6. Modeling

Now that the preparation is complete, the building of models commences. In here we are going to start the modeling and we are going to use various regression algorithms to find the most suitable one. The reason to why we use regression in here, it is because it is a prediction of how much and so we use regression. Due to regression having multiple algorithms, we will use a couple of them and compare between them. Furthermore, we would do the modeling on the merged data set and the original dataset to compare. finally we will also do the modeling between both data sets with and without hyper parameter tuning.

NOTE: Please check the jupyter notebook chapter 6, as if the modeling was included in this document, the document would be quite large.

In the notebook an extensive explanation, and investigation is done. Total of 16 models were created. That is because we have created 4 regression non tuned parameter models and 4 tuned regression models, that are tested on 2 different datasets. Therefore, 2 x 8=16.

7. Evaluation

In here we will compare the models and the datasets as which was better and elaborate more on why.

7.1. Kamernet dataset with default hyperparameters

:

	Modeltype	RMSE	R-squared(training)	R-squared(test)
0	Linear Regression	347.735454	0.778049	0.578251
1	Ridge Regression	340.275579	0.774040	0.596153
2	Decision Tree Regression	144.744347	0.990810	0.926927
3	Random Forest Regression	194.988024	0.967111	0.867391

Figure 28 Kamernet dataset models evaluation

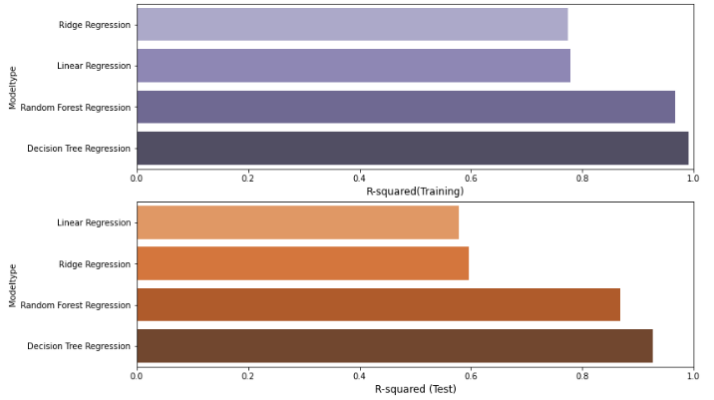


Figure 29 Barplot comparison of the training and testing R squared

It can be seen that when comparing the models of the merged dataset with default parameters, that the Decision tree and random forest has above 80% for testing r squared and almost 100% in training. This is very interesting as the random forest was expected to do better due to it preventing overfitting which implies that my decision tree model is overfit, based on the testing error is higher than the training. Because the model is trained well on these points but when new points are added it does worse.

Below we will explore more on this and compare.

7.2. Enhanced dataset hyper tuned parameters

	Modeltype	RMSE	R-squared(training)	R-squared(test)
0	Linear Regression	144.744347	0.778049	0.578251
1	Ridge Regression	322.488995	0.727277	0.637268
2	Decision Tree Regression	262.738022	0.869818	0.759230
3	Random Forest Regression	199.564857	0.937041	0.861093

Figure 30 enhanced dataset with tuned parameters evaluation

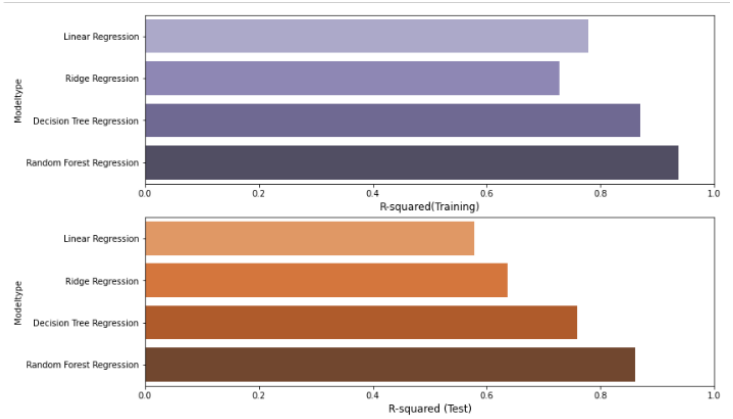


Figure 31 Barplot comparison of the training and testing R squared

This is very interesting as when tuned the random forest model did the best and better than the decision tree where it is above 80%. and that is due to the features of the random forest, as it disregards the noise of the data, which prevents overfitting, therefore a higher accuracy than the decision tree which is easily prone to overfitting. meanwhile if we compare between the ridge and linear regression we see that the ridge is performing better than the linear in testing and linear is performing better in training and that is due to the fact that our features are too much for it to handle so it overfits so ridge is recommended. that is because ridge and lasso are considered as regulation algorithms, which avoids over fitting.

Now if we compare between the non-tuned and tuned models we will notice that the non-tuned models are doing better than the tuned and that could be either due to the new parameters being worse than the default or that after tuning the model we made it balanced and not overfitted.

7.3. Kamernet dataset hyper tuned parameters

	Modeltype	RMSE	R-squared(training)	R-squared(test)
0	Linear Regression	144.744347	0.778049	0.578251
1	Ridge Regression	261.582909	0.726389	0.743420
2	Decision Tree Regression	295.241200	0.658506	0.673143
3	Random Forest Regression	212.578894	0.961686	0.830549

Figure 32 Kamernet dataset with tuned parameters evaluation

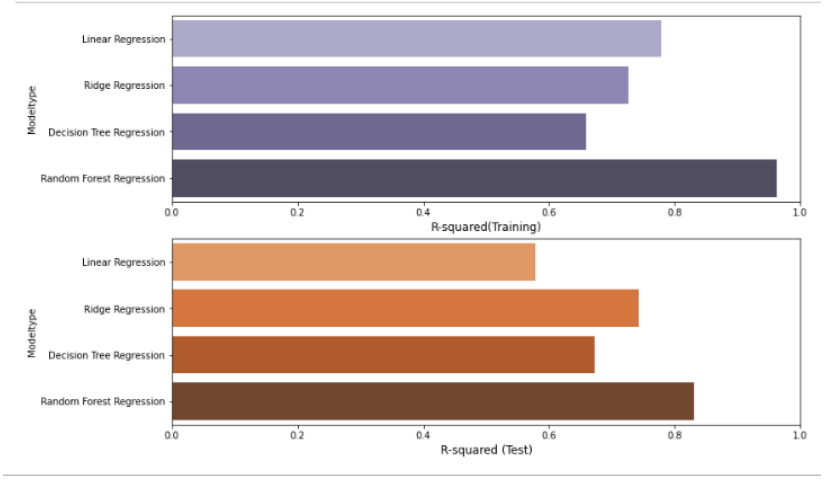


Figure 33 Barplot comparison of the training and testing R squared

Now if we compare the models of the tuned original dataset, we notice that in training the random forest does the best then comes the liner, ridge and decision tree. On the other hand, in the testing random forest has an r squared of 83% compared to the ridge which had almost 80%.

when tuned, the random forest model did the best and better than the decision tree where it is above 80%. and that is due to the features of the random forest, as it disregards the noise of the data, which prevents overfitting, therefore a higher accuracy than the decision tree which is easily prone to overfitting. meanwhile if we compare between the ridge and linear regression we see that the ridge is performing better than the linear in testing and linear is performing better in training and that is due to the fact that our features are too much for it to handle so it overfits so ridge is recommended. that is because ridge and lasso are considered as regulation algorithms, which avoids over fitting.

7.4. Enhanced dataset with default hyperparameters

	Modeltype	RMSE	R-squared(training)	R-squared(test)
0	Linear Regression	261.718786	0.726466	0.743154
1	Ridge Regression	261.736864	0.726466	0.743118
2	Decision Tree Regression	272.899142	0.993208	0.720741
3	Random Forest Regression	215.741352	0.969587	0.825470

Figure 34 enhanced dataset with non-tuned parameters evaluation

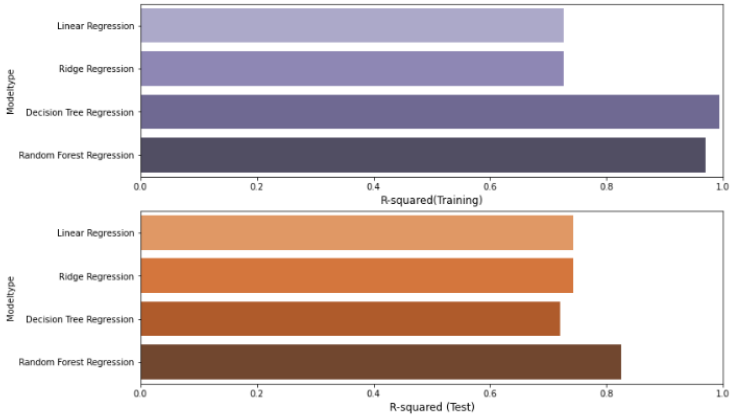


Figure 35 Barplot comparison of the training and testing R squared

What is interesting about this is that, when non tuned the model seems to have a high accuracy. if we compare between the decision tree and the random forest, we see that the decision tree has almost a 100% accuracy on training and around 70% on testing, and this is due to the algorithm being overfitted. on the other hand, the random forest has around 90% accuracy on training and around 80% on testing this is due to the nature of the random forest as it stops the model from overfitting.

8. conclusion

In conclusion, this project aims to spread awareness and predict the start base of the rent to help students and starts to protect themselves from scammers. For other interested parties, this project would be not interesting, due to the fact that the market now is in favor of selling, people pay more for what it is worth, low mortgage interests which makes it easier for people to loan more money from the bank. Therefore, This AI could not be used by landlords, as most of them do this for the money. In addition to that, the models and the datasets will be compared and a final conclusion of which dataset and which model are better to use, will be written in the 4th phase of the AI project methodology. Furthermore, based on the domain expert whom was interviewed for this project, this project is interesting and could have a huge positive impact on society as it will also reduce the time required for a tenant to do his research on the prices, offers a basic price based on the house 's worth as a start of price negotiations and finally, shows the price margin between the wanted price and the predicted price, as a way of predicting scams.

In addition to that, addressing the models, what can be seen is that, the difference between the datasets is that, the more data is included the higher the accuracy is. that can be seen above as the non-tuned original data set highest training score was 99% and the testing was 82%, measured in r squared. on the other hand the non-tuned enhanced data set highest training score was 99% and testing was 92%. This already shows that more data means more accuracy. Now, if we compare the datasets of the tuned original data set and tuned enhanced data set, we see that that the original data set has a highest score of training 96% and testing of 83%. compared to the enhanced data set which had a highest training of 93% and testing of 86%. This shows that the more data added the better the accuracy is. In addition to that, if we look at the performance and why does tuning the parameters matter, we will notice that the reason behind the accuracy is being high before tuning is because the algorithm was being overfitted. That can be seen in the Mean squared error of the algorithms. As before tuning, we notice that the testing MSE is higher than the training MSE, and this implies that the model was overfitted. After tuning the models, we noticed that the training MSE value was getting closer to the range of the testing MSE. An example of that is the decision tree algorithm of the original dataset. If we look at that section we will notice that before tuning the model the training MSE was 1785 and testing MSE was 74473, which clearly shows that this is overfitted. After tuning the model, we notice that the training MSE error is 89764 and the Testing MSE is 87167.Therefore, hyper tuning the parameters and enhancing the data are important factors in achieving high accuracy results.

According to the interview that were done with the target audience and the domain expert, this project can reduce the time taken to search for reliable houses, can predict the price worth of the house, with a price margin (i.e., RMSE), due to the model being overestimating, giving new coming student a base line for price negotiation, and shows the price margin between the predicted price and actual price to flag the houses as scam.

The next step is to add the predicted value column into the table. The table will be saved as a CSV and be used in a powerBI application. Please check phase 4 document.

9. Bibliography & references

(n.d.).

Block, W. (2021, 10 24). *Rent control*. Retrieved from Ecyclopedia : https://www.econlib.org/library/Enc/RentControl.html

D.Talsma. (2021, 07 07). *Renting, not buying: how the housing market is completely bogged down by failing government policies*. Retrieved from universonline: https://universonline.nl/nieuws/2021/07/01/renting-not-buying-how-the-housing-market-is-completely-bogged-down-by-failing-government-policies/

DutchNews. (2019, 09 13). *Scammed, sofa-surfing and stranded - finding a home as an international student*. Retrieved from DutchNews.nl: https://www.dutchnews.nl/news/2019/09/scammed-sofa-surfing-and-stranded-finding-a-home-as-an-international-student/

I Amsterdam. (2020, 06 09). *Overview of housing rights in the Amsterdam Area*. Retrieved from I Amsterdam: https://www.iamsterdam.com/en/living/settling-in/housing/rental-property/housing-rights

kamernet. (2022, 04 08). *how does it work* . Retrieved from Kamernet: https://kamernet.nl/en/how-does-it-work/a

Koole, J. (2022, 04 06). House renting. (R. Fuad, Interviewer)

Micaela Zaslabsky. (2019, 05 29). *Amsterdam Hangout*. Retrieved from What is The Cost of Living For a Student in Amsterdam?: https://amsterdamhangout.com/what-is-the-cost-of-living-for-a-student-in-amsterdam-2/

Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. (2022, 01 03). *Rented housing* . Retrieved from government of the netherlands : https://www.government.nl/topics/housing/rented-housing

Santa Clara University. (2017, 08 25). *The Relationship of Morality and Technology*. Retrieved from Markkula center for applied ethics at santa clara unicersity: https://www.scu.edu/ethics/all-about-ethics/the-relationship-of-morality-and-technology/

Student Helpr. (n.d.). *Living costs*. Retrieved from https://studenthelpr.com: https://studenthelpr.com/international-dutch-student-expenses

[1] *Netherlands Rent Properties*. (2020, 4 maart). Kaggle. Geraadpleegd op 4 maart 2022, van https://www.kaggle.com/juangesino/netherlands-rent-properties

Room for rent in Amsterdam €1095 | Kamernet. (2021). Kamernet.Nl. Geraadpleegd op 4 maart 2022, van https://kamernet.nl/en/for-rent/room-amsterdam/eosstraat/room-2026209

Deviations, O. T. S. (2018, 14 september). *An overview of correlation measures between categorical and continuous variables*. Medium. Geraadpleegd op 4 maart 2022, van

https://medium.com/@outside2SDs/an-overview-of-correlation-measures-between-categorical-and-continuous-variables-4c7f85610365