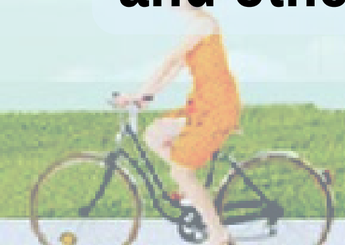

# PORTUGAL REAL ESTATE
# DATA SCIENCE PROJECT

ROY SHLOMO CHEN & YARIN AKIVA

# WHAT IS REAL ESTATE

Real estate is a term that encompasses the buying, selling, and renting of land and buildings.
It involves the physical land and buildings, as well as the rights and privileges associated with them.
Real estate can be residential, commercial, or industrial, and can include the purchase and sale of land, buildings, and other improvements.
Real estate can also include the management of rental properties and other services related to the industry.




# WHY?

In recent years, Jews who could prove that they were deported from Spain, could submit an application to the Portuguese authorities and the Jewish community living in Porto or Lisbon, and ask for citizenship in Portugal only by proving that they are descendants of those deported from Spain.

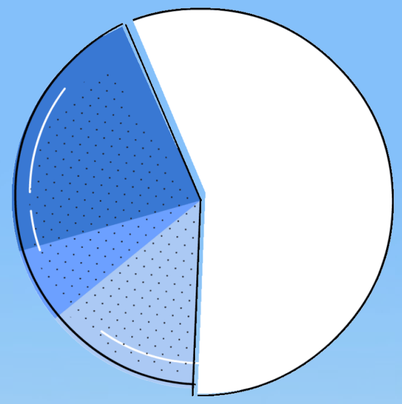

At the end of 2022, the Portuguese government decided to tighten the conditions for obtaining citizenship and a Portuguese passport for the descendants of those deported from Spain, and from now on you also have to show affinity to the state of Portugal.

# WHY?


One of these ways is to invest or purchase real estate in Portugal.

We decided to check whether it is possible to analyze the information on properties in Portugal using the Remax Portugal website, and to determine what is a good property for investment in Portugal.

The goal is to help those Jews who want to become citizens in Portugal and are now faced with the tightening of the conditions for obtaining citizenship



# RESEARCH QUESTION

# How to identify a good investment property in Portugal?

# MAIN PROCESS STEPS

1

## Obtaining Data

web scraping for collection , and obtaining the data

2

## Data Handling

Cleaning,formattin and filtering the data, removing duplicates data

3

## Exploring Data

Visualizing and understaing the data

4

## Machine Learning

Clustering the data in groups and modeling , and apply ML modeling on the data

5

## Interpretin Data

Presentation of data,understanding and delivering the results

# OBTAINING DATA



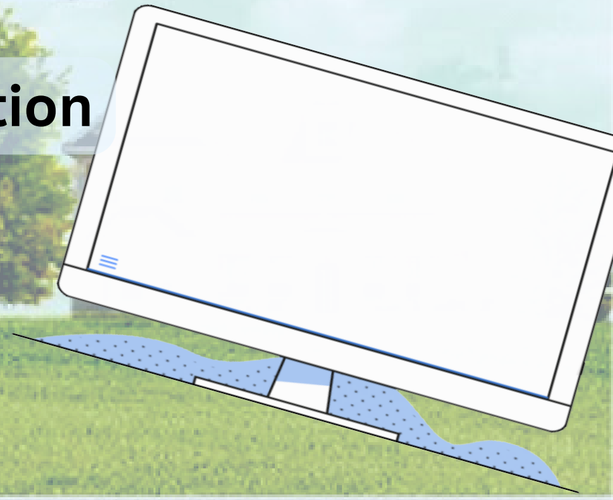
RE/MAX ®

In this step, we will collect information of properites in Portugal in various big cities

we will crawl along web pages and scrape information about properties features

Main tools : BeautifulSoup, Selenium

Source Data : https://www.remax.pt/en/



| | Price | Address | Prop_type | Size | Bath | Bed |
|---|---|---|---|---|---|---|
| | 255 000 € | Porto - Vila Nova de Gaia, São Felix da Mar... | Condo/Apartment | 105 m2 | 2 | 2 |
| | 198 500 € | Porto - Paços de Ferreira, Paços de Ferreira | Condo/Apartment | 132 m2 | 2 | 3 |
| | 138 000 € | Porto - Vila Nova de Gaia, Grijó e Sermonde | Condo/Apartment | 97 m2 | 2 | 2 |
| 3 | 225 000 € | Porto - Porto, Paranhos | Condo/Apartment | 89 m2 | 1 | 1 |
| 4 | 225 000 € | Porto - Porto, Paranhos | Condo/Apartment | 53 m2 | 1 | 1 |

```
#   Column     Non-Null Count   Dtype
0   Price      1452 non-null    object
1   Address    1452 non-null    object
2   Prop_type  1452 non-null    object
3   Size       1452 non-null    object
4   Bath       1452 non-null    object
5   Bed        1452 non-null    object
dtypes: object(6)
```

**To collect each property, we used Selenium because this is a dynamic scrapping,**

**After we passed the dynamic scrapping difficulty**

**We used BeatifulSoup to store the data in dataframes**

| | Price | Prop_type | Size | Bath | Bed | City | Division |
|---|---|---|---|---|---|---|---|
| 0 | 295000.0 | House | 140.0 | 2.0 | 3.0 | Porto | Lordelo do Ouro e Massarelos |
| 1 | 22500.0 | House | 46.0 | 0.0 | 2.0 | Baião | Ancede e Ribadouro |
| 2 | 485000.0 | House | 152.0 | 3.0 | 4.0 | Vila do Conde | Vila Chã |
| 3 | 1250000.0 | House | 527.0 | 0.0 | 10.0 | Porto | Aldoar |
| 4 | 67000.0 | House | 75.0 | 1.0 | 2.0 | Paços de Ferreira | Carvalhosa |
| ... | ... | ... | ... | ... | ... | ... | ... |
| 11660 | 135000.0 | Condo/Apartment | 22.0 | 1.0 | 0.0 | Braga | São Vicente |
| 11661 | 130000.0 | Condo/Apartment | 18.0 | 1.0 | 0.0 | Braga | São Vicente |
| 11662 | 139900.0 | Condo/Apartment | 110.0 | 2.0 | 3.0 | Barcelos | Barcelos |
| 11663 | 799000.0 | Condo/Apartment | 720.0 | 12.0 | 7.0 | Vieira do Minho | Guilhofrei |
| 11664 | 170000.0 | Condo/Apartment | 113.0 | 2.0 | 2.0 | Braga | (São José de São Lázaro e São João do Souto) |

11665 rows × 7 columns

```
Address       object
Prop_type     object
Size          object
Bath          object
Bed           object
dtype: object
```

# DATA HANDLING

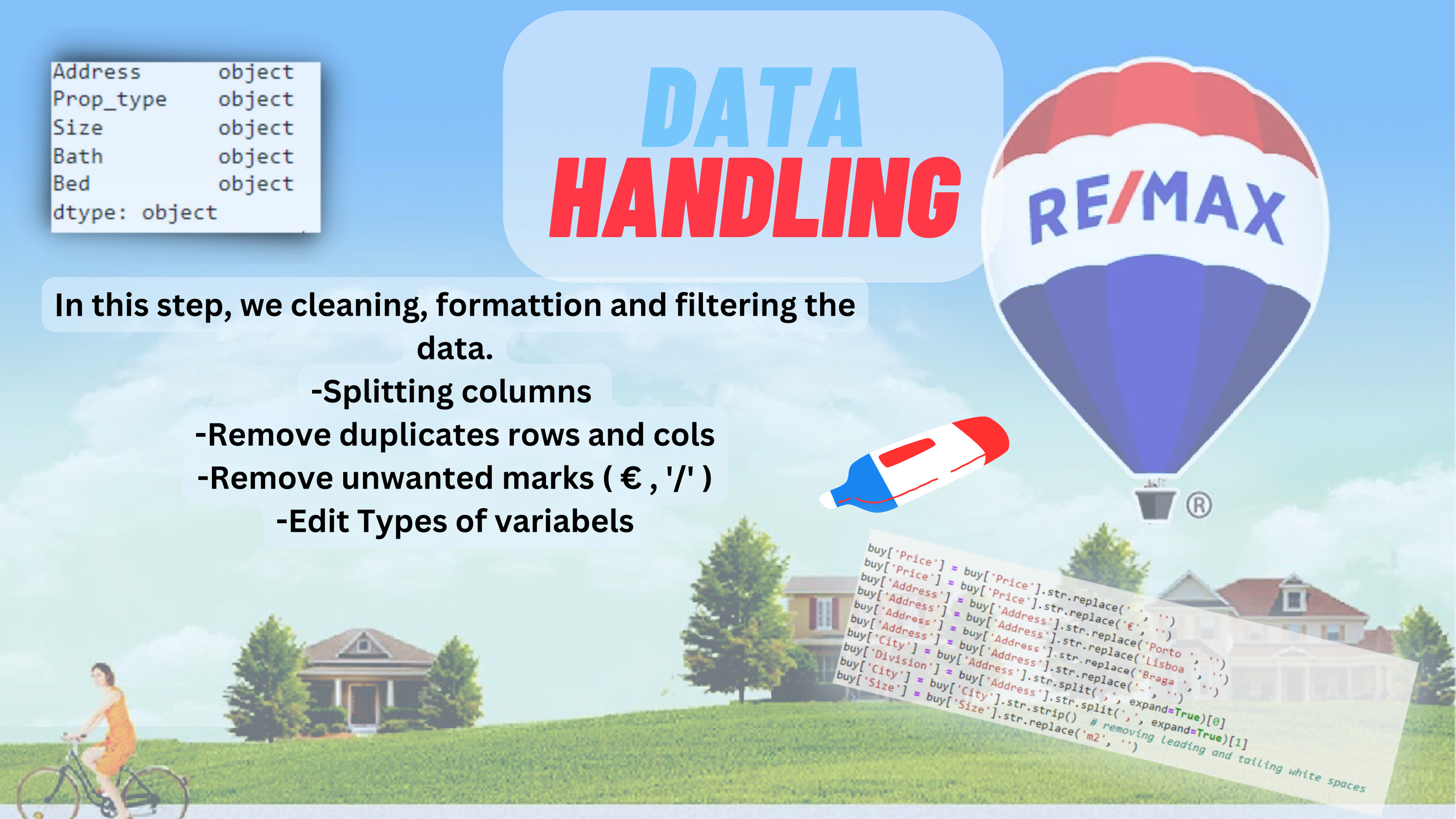
In this step, we cleaning, formattion and filtering the data.

-Splitting columns

-Remove duplicates rows and cols

-Remove unwanted marks ( € , '/' )

-Edit Types of variabels



```
buy['Price'] = buy['Price'].str.replace(' ', '')
buy['Price'] = buy['Price'].str.replace('€', '')
buy['Address'] = buy['Address'].str.replace('Porto ', '')
buy['Address'] = buy['Address'].str.replace('Lisboa ', '')
buy['Address'] = buy['Address'].str.replace('Braga ', '')
buy['Address'] = buy['Address'].str.replace('-', '')
buy['City'] = buy['Address'].str.split(',', expand=True)[0]
buy['Division'] = buy['Address'].str.split(',', expand=True)[1]
buy['City'] = buy['City'].str.strip()  # removing leading and tailing white spaces
buy['Size'] = buy['Size'].str.replace('m2', '')
```

**For each column we change the type from category to numeric , help in the next step and make more easiest to work on the data**

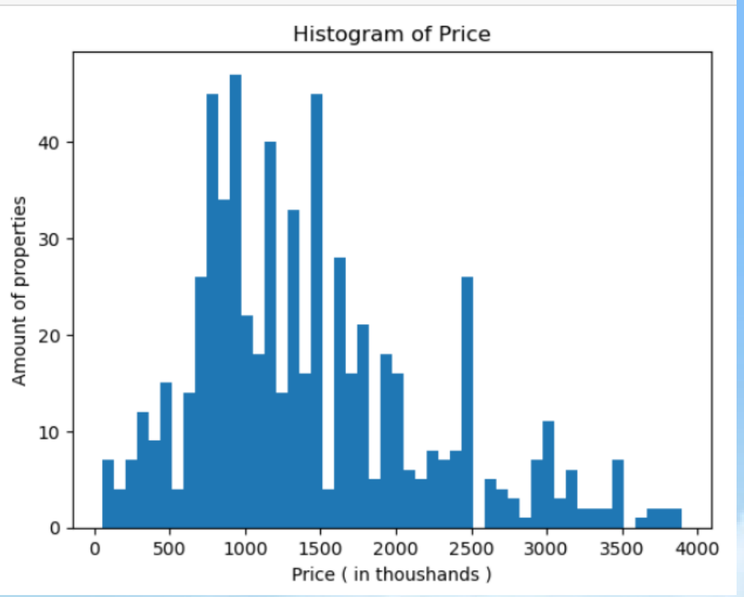
```
buy['Price'] = pd.to_numeric(buy['Price'], errors='coerce')
buy['Size'] = pd.to_numeric(buy['Size'], errors='coerce')
buy['Bath'] = pd.to_numeric(buy['Bath'], errors='coerce')
buy['Bed'] = pd.to_numeric(buy['Bed'], errors='coerce')
```

**Using replace map -**
**we change all the property type from category to numeric**
**'1' to House and '2' to apartment**

```
replace_map={'House':1,'Condo/Apartment':2}
buy2.replace(replace_map, inplace=True)
```

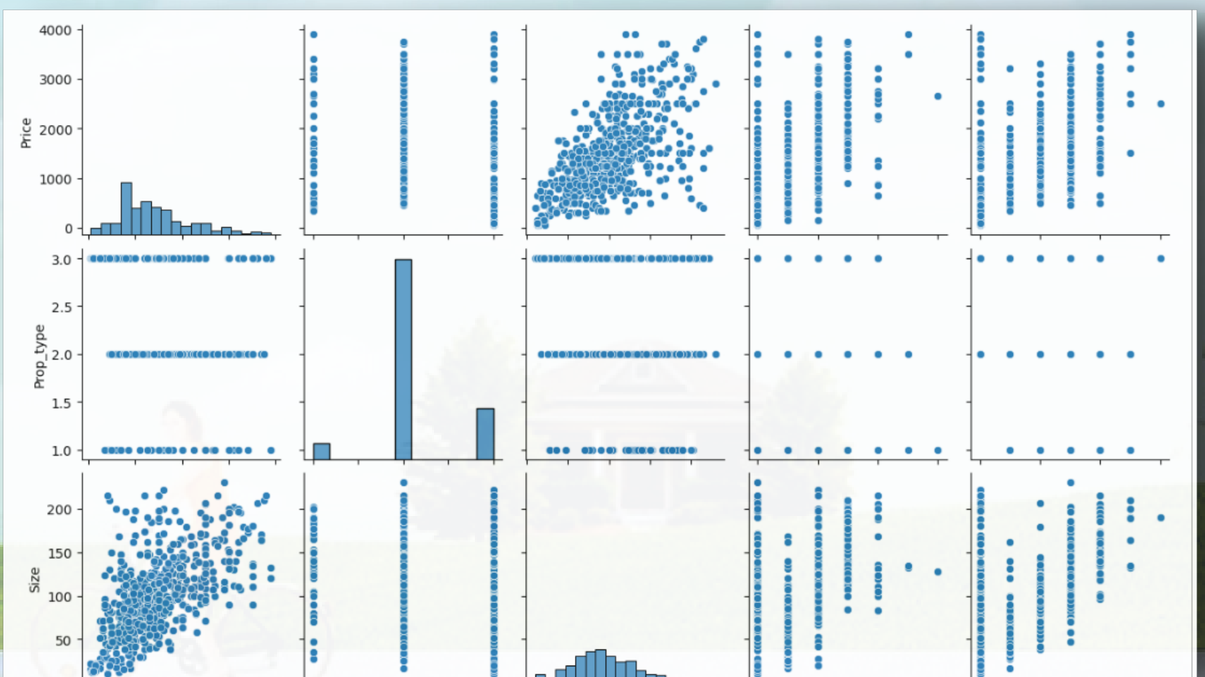
| Prop_type |
|---|
| House |
| House |
| House |
| House |
| House |
| ... |
| Condo/Apartment |
| Condo/Apartment |
| Condo/Apartment |
| Condo/Apartment |
| Condo/Apartment |

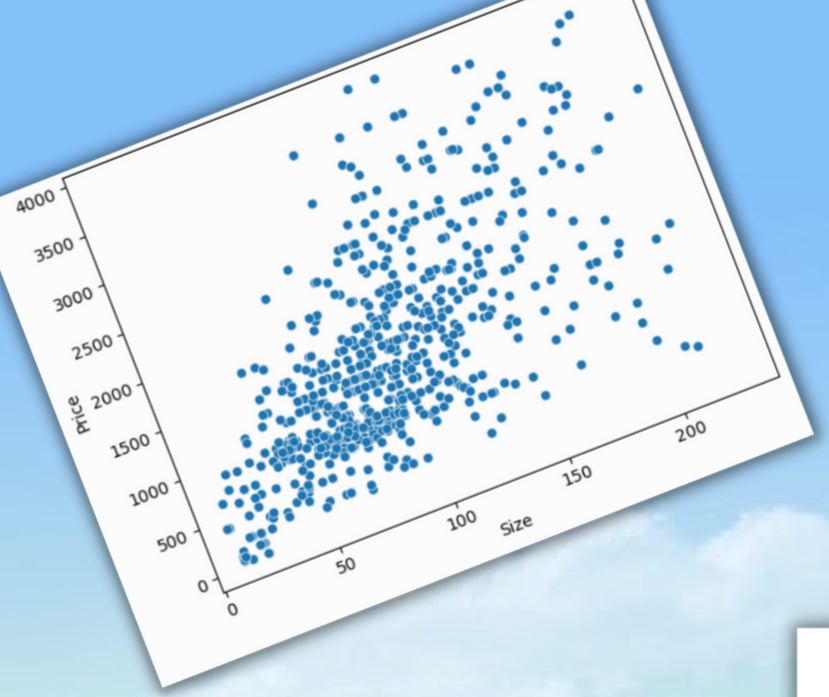
| Prop_type |
|---|
| 1 |
| 1 |
| 1 |
| 1 |
| 1 |
| ... |
| 2 |
| 2 |
| 2 |
| 2 |
| 2 |

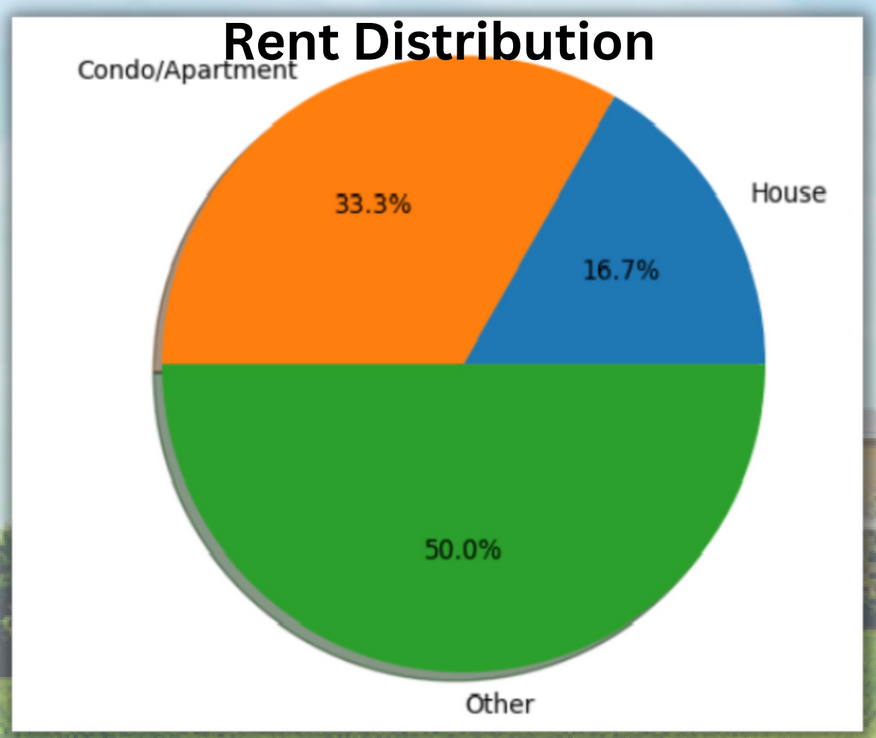



# EXPLORING DATA

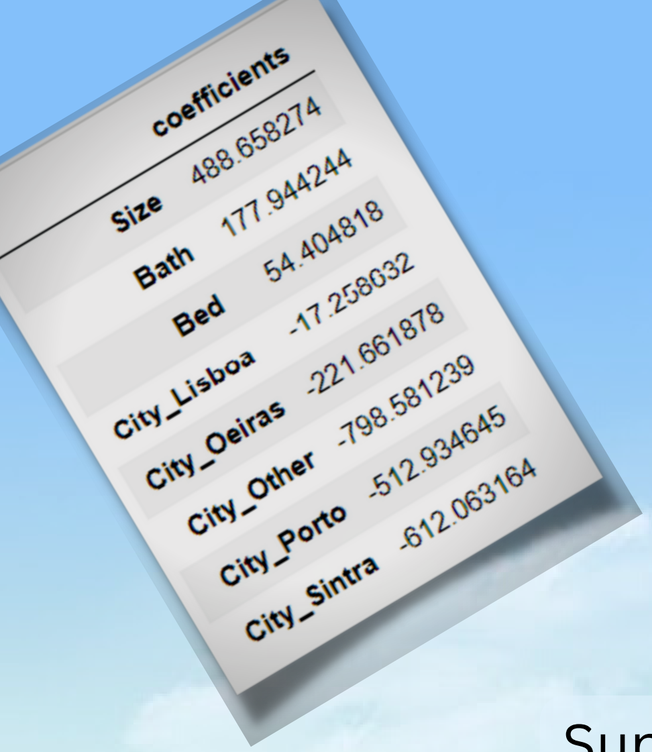
**We did some visualization with our data after all the cleanning with this EDA we did outliers treatment and remove unnecessary**





4000
3500
3000
2500
2000
1500
1000
500
0
Price
0
50
100
150
200
Size


Rent Distribution
Condo/Apartment
33.3%
House
16.7%
50.0%
Other


Price by Property Type
4000
3500
3000
2500
2000
1500
1000
500
0
Price
1
2
3
Prop_type




# MACHINE LEARNING

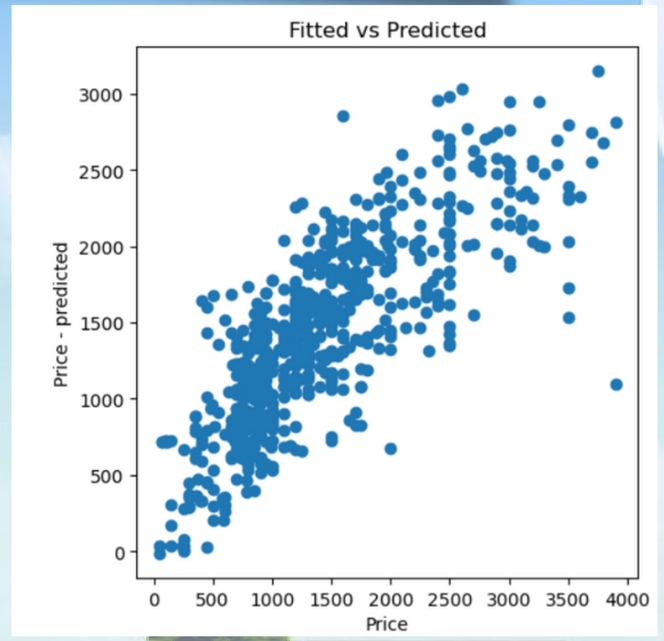
**Supervised Learning :**

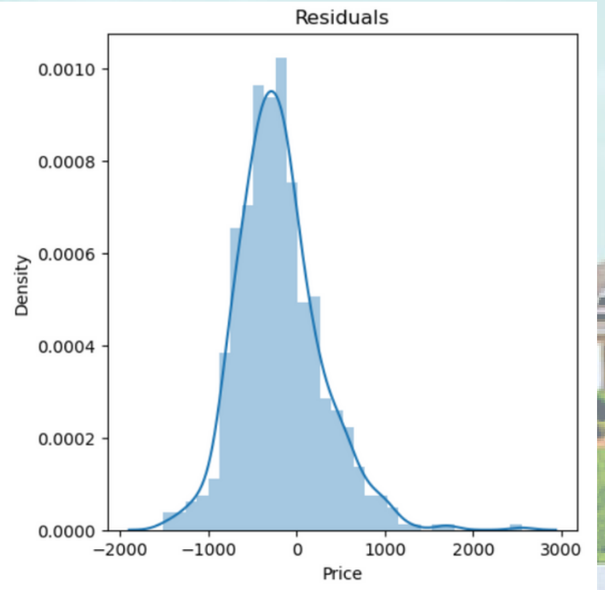
Supervised learning is a type of machine learning in which the computer is provided with labeled data and it is expected to generate a predictive model. This model is then used to make predictions about new data. Supervised learning can be used to classify data, predict outcomes, and identify patterns.

```python
# strat train the model , splitting to 20/80

X = rent2.drop(columns = ['Price'])
y = rent2['Price']

# 20% of the data will be used for testing and 80% will be used for training.
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2)

col = ['Size', 'Bath', 'Bed'] # used to scale data in the same range
scaler = RobustScaler().fit(X_train[col]) # RobustScaler() to scale the columns 'Size', 'Bath', and 'Bed' in the training set
X_train[col] = scaler.transform(X_train[col])
```

**Supervised Learning :**

we use 20% of the data to test
and 80% to train
our X column was the Bed,Size and Bath
and our Y column wash the Price




| | Size | Bath | Bed | City_Lisboa | City_Oeiras | City_Other | City_Porto | City_Sintra |
|---|---|---|---|---|---|---|---|---|
| **560** | 0.136364 | 0.0 | 0.5 | 1 | 0 | 0 | 0 | 0 |
| **601** | 0.827273 | 0.0 | 1.0 | 1 | 0 | 0 | 0 | 0 |
| **320** | -0.263636 | -1.0 | -0.5 | 1 | 0 | 0 | 0 | 0 |
| **374** | 2.027273 | -2.0 | -1.0 | 1 | 0 | 0 | 0 | 0 |
| **254** | 1.281818 | 1.0 | 0.5 | 0 | 0 | 1 | 0 | 0 |

```
# checking R2 for train/test
# The R2 score is a measure of how well the linear regression model fits the data

print(lm_scaled.score(X_train, y_train)) # This line prints the R2 score for the training dataset
print(lm_scaled.score(X_test, y_test)) # This line prints the R2 score for the testing dataset
```

```
0.6270767613029067
0.5546334818921739
```

**Supervised Learning :**

we test and train first the rent data
to understand how can we now
to calculete the predicted rent
while we give to the program the price property
and his details such as bed, bath, size.


**Supervised Learning :**

we start to train and test when we scaled the d
and after we did the same without scaling ( with
modeling on the Z-score )

```
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 453.2954578178126
Mean Squared Error: 354860.86853835644
Root Mean Squared Error: 595.7019964196497
```

**Supervised Learning :**

While we get our linear regression equation
we load the buy data
and fit the data to look like the same data
as the rent .
After we train the Price data
with the linear regression predict we got previously
and then we get the predicted rent column about the
buy data

```
X = buydf.drop(columns = ['Price'])
buydf['predicted_rent'] = lm.predict(X)
```

| | Price | Size | Bath | Bed | City_Cascais | City_Lisboa | City_Other | City_Porto | City_Sintra | predicted_rent |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 295000 | 140 | 2 | 3 | 0 | 0 | 0 | 1 | 0 | 1761.992029 |
| 1 | 22500 | 46 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 387.169896 |
| 2 | 485000 | 152 | 3 | 4 | 0 | 0 | 1 | 0 | 0 | 1831.693994 |
| 3 | 1250000 | 527 | 0 | 10 | 0 | 0 | 0 | 1 | 0 | 5211.002405 |
| 4 | 67000 | 75 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 766.781801 |
| 5 | 147000 | 475 | 4 | 4 | 0 | 0 | 1 | 0 | 0 | 4649.000758 |
| 6 | 280000 | 120 | 4 | 3 | 0 | 0 | 1 | 0 | 0 | 1631.450533 |
| 7 | 459000 | 149 | 8 | 3 | 0 | 0 | 1 | 0 | 0 | 2428.539470 |
| 8 | 150000 | 160 | 1 | 3 | 0 | 0 | 1 | 0 | 0 | 1545.632665 |
| 9 | 162000 | 463 | 0 | 5 | 0 | 0 | 1 | 0 | 0 | 4066.942106 |

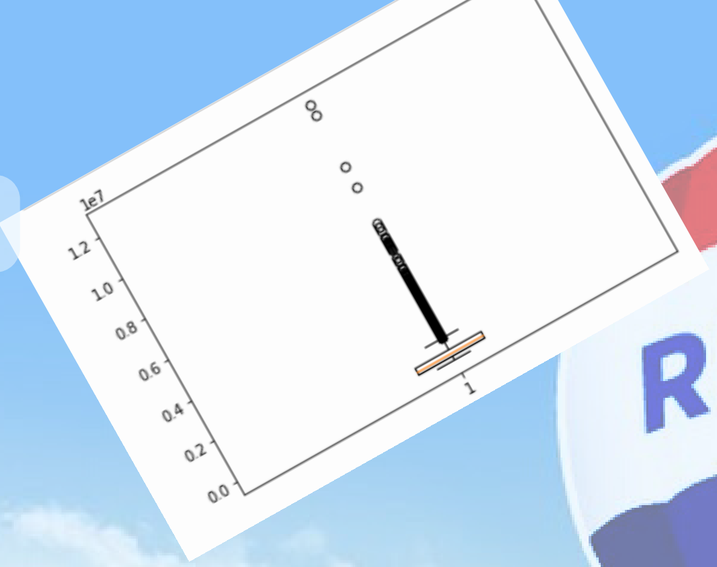
The last step
is to load the buy data after adding the predicted rent column.
we calculate with the Portugal formula to calculate the monthly mortgage and other monthly expense , such as insurance or property tax and make one column that shows , how much profit the property owner stay with at the end of the month.

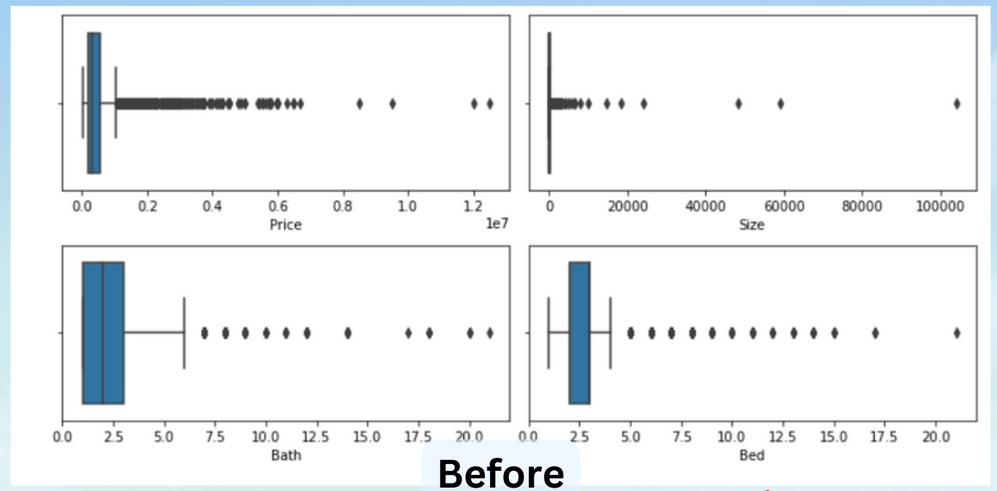
| | Price | Size | Bath | Bed | City_Cascais | City_Lisboa | City_Other | City_Porto | City_Sintra | Monthly_Profit |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 295000 | 140 | 2 | 3 | 0 | 0 | 0 | 1 | 0 | -286.157971 |
| **1** | 22500 | 46 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 231.344896 |
| **2** | 485000 | 152 | 3 | 4 | 0 | 0 | 1 | 0 | 0 | -1535.756006 |
| **3** | 1250000 | 527 | 0 | 10 | 0 | 0 | 0 | 1 | 0 | -3468.497595 |
| **4** | 67000 | 75 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 302.391801 |

## Eda and Outliers treatment:

We did again

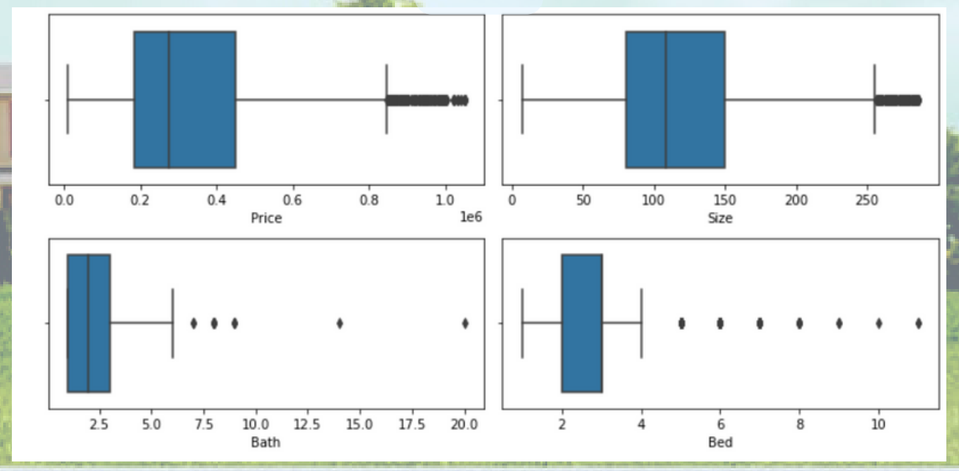
Eda and outliers treatment on the new data to make a cleanier data to work on with the machine learning modules.



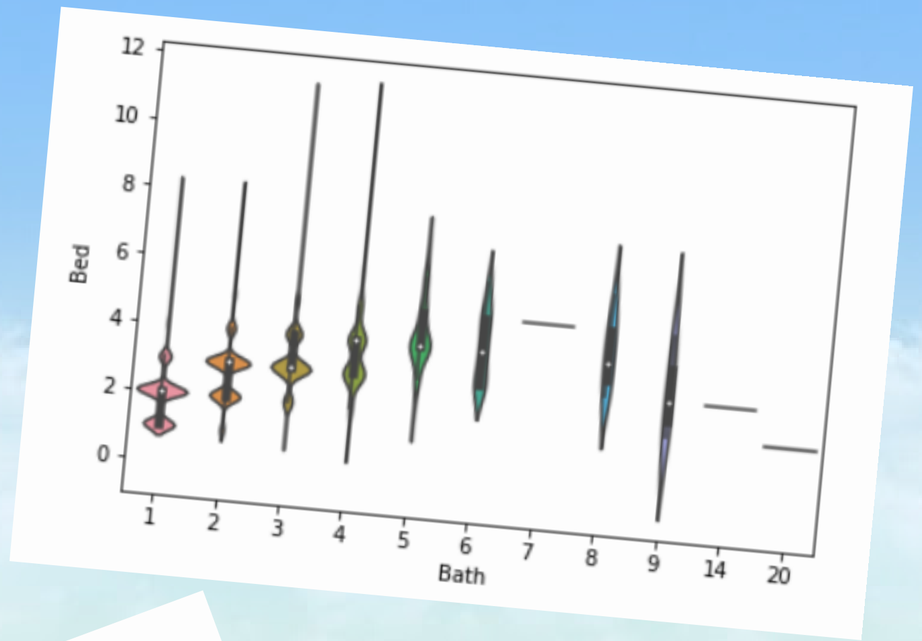



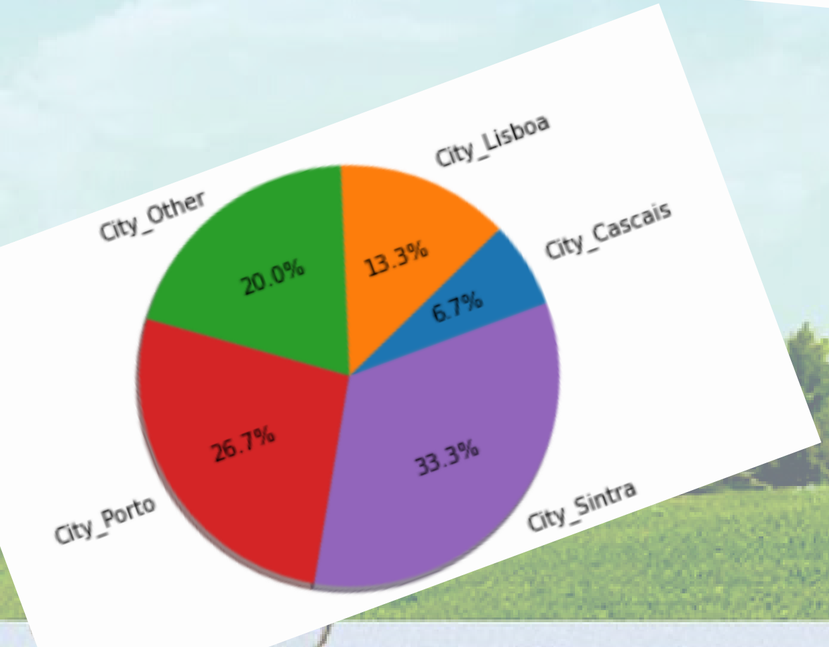

**Before**

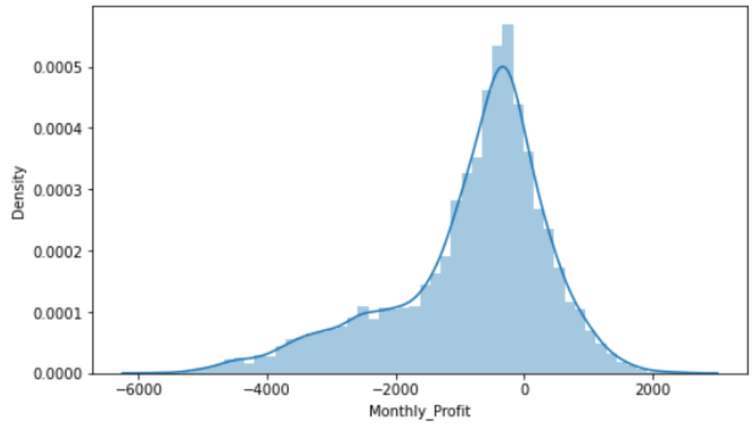
**After**

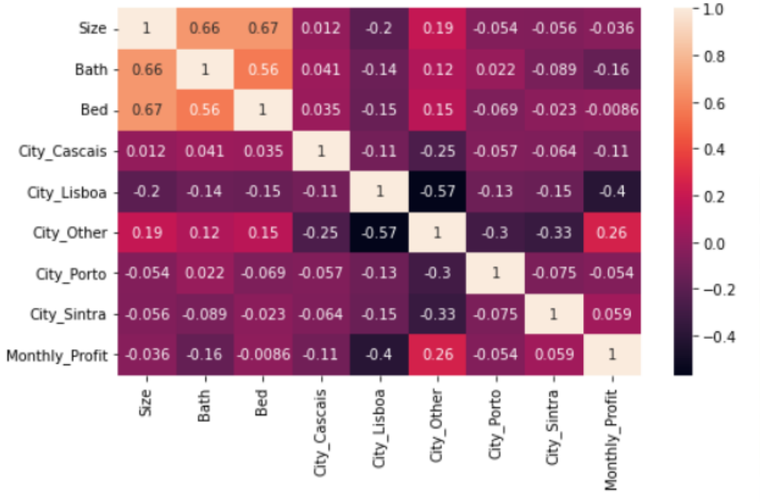


# Some of the visualization we get after cleanning the data



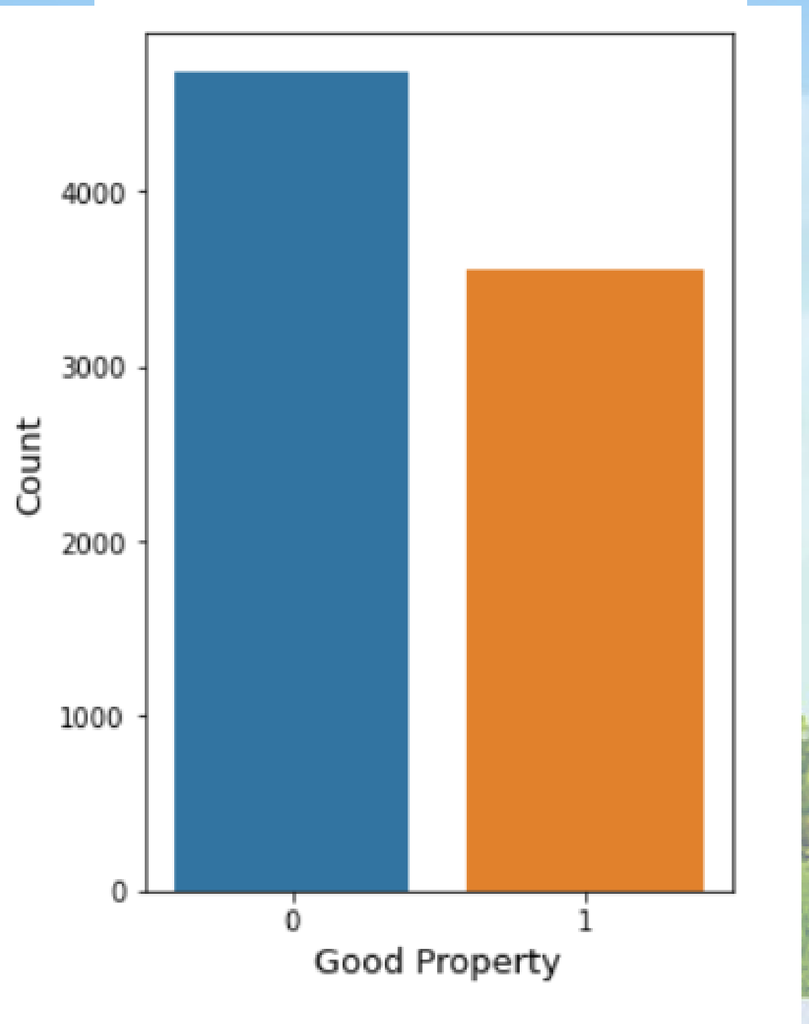



We decide that good property is a property that his owner doesn't lost more than 400 euro per month

```
not good: 56.99999999999999%
good: 43.0%
```




# Supervised Learning :

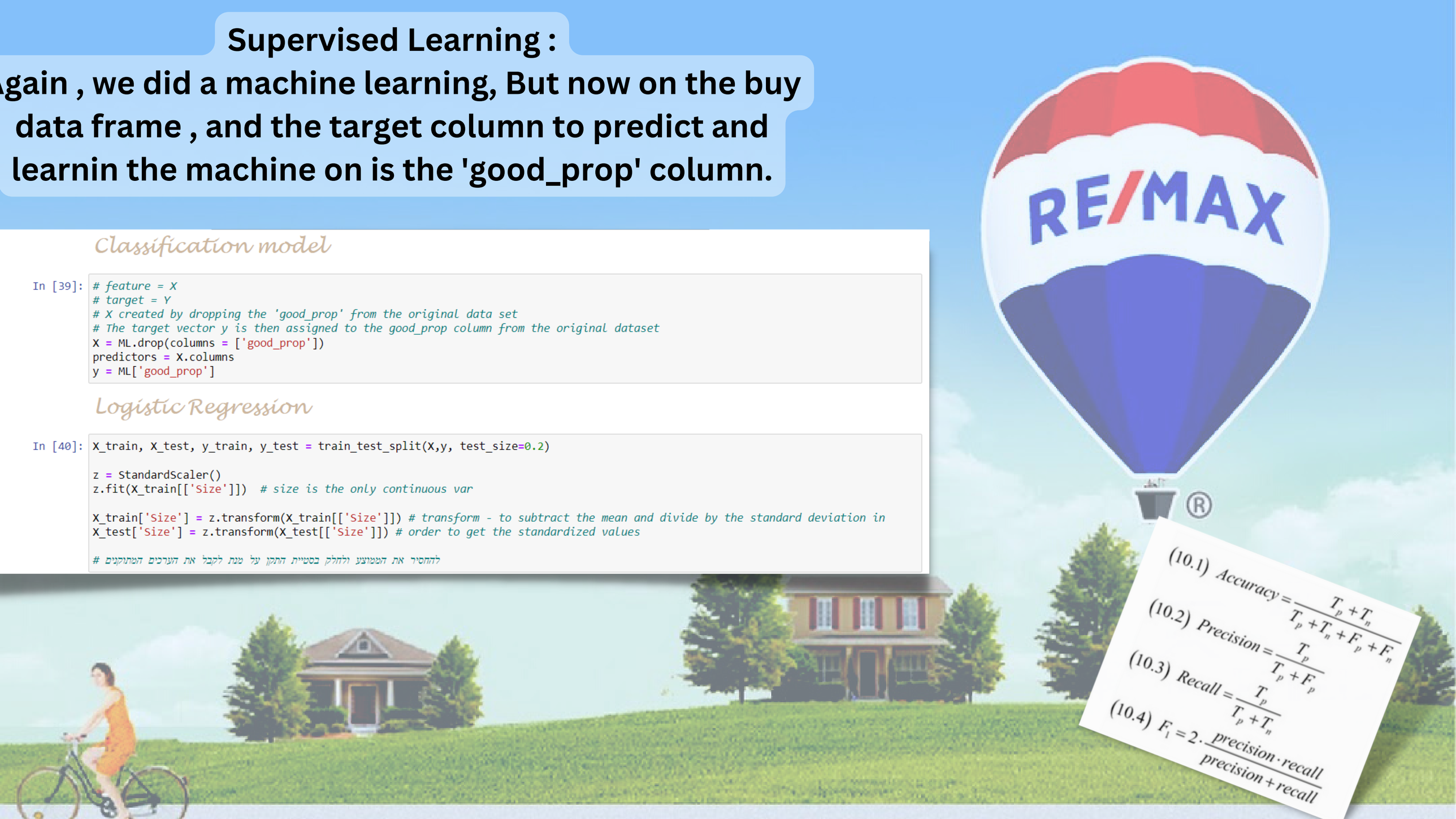
**Again , we did a machine learning, But now on the buy data frame , and the target column to predict and learnin the machine on is the 'good_prop' column.**

## Classification model

In [39]:
```
# feature = X
# target = Y
# X created by dropping the 'good_prop' from the original data set
# The target vector y is then assigned to the good_prop column from the original dataset
X = ML.drop(columns = ['good_prop'])
predictors = X.columns
y = ML['good_prop']
```

## Logistic Regression

In [40]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2)

z = StandardScaler()
z.fit(X_train[['Size']])  # size is the only continuous var

X_train['Size'] = z.transform(X_train[['Size']]) # transform - to subtract the mean and divide by the standard deviation in
X_test['Size'] = z.transform(X_test[['Size']]) # order to get the standardized values

# להחסיר את הממוצע ולחלק בסטיית התקן על מנת לקבל את הערכים המתוקנים
```



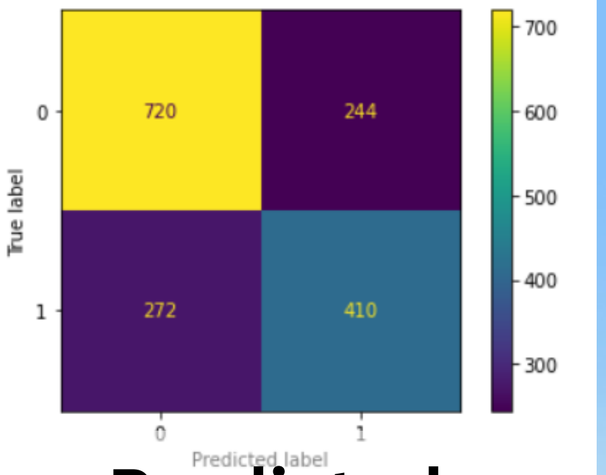

$$(10.1)\ Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$(10.2)\ Precision = \frac{T_p}{T_p + F_p}$$

$$(10.3)\ Recall = \frac{T_p}{T_p + T_n}$$

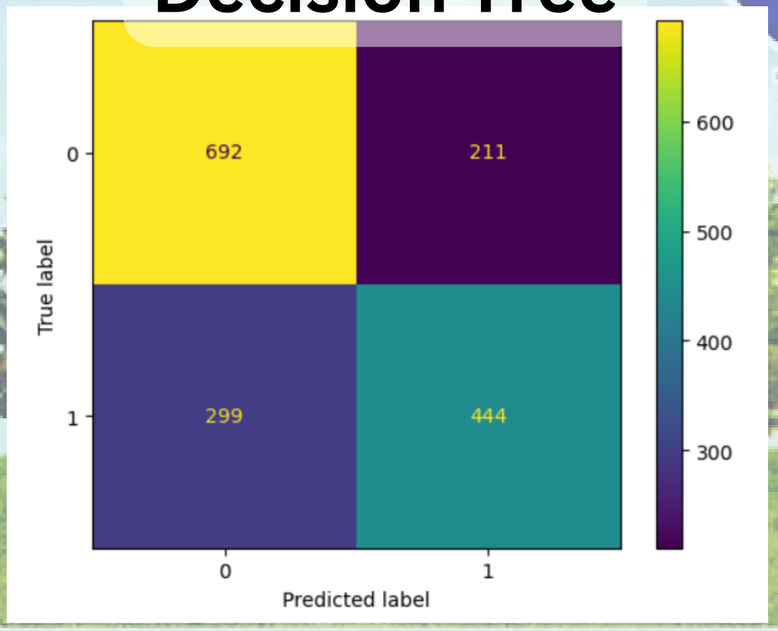
$$(10.4)\ F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

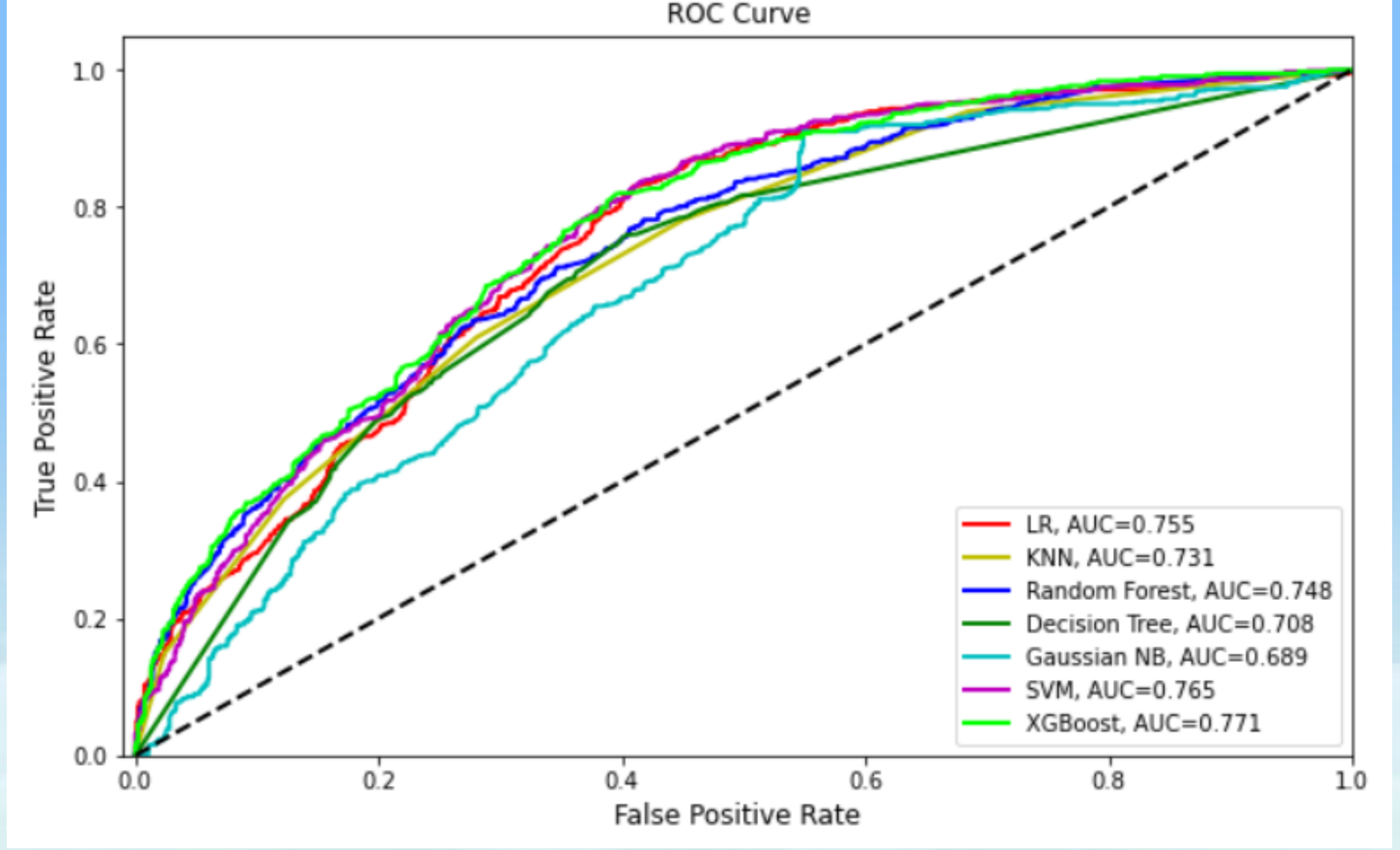
# MACHINE LEARNING RESULTS




**Predicted accuracy**

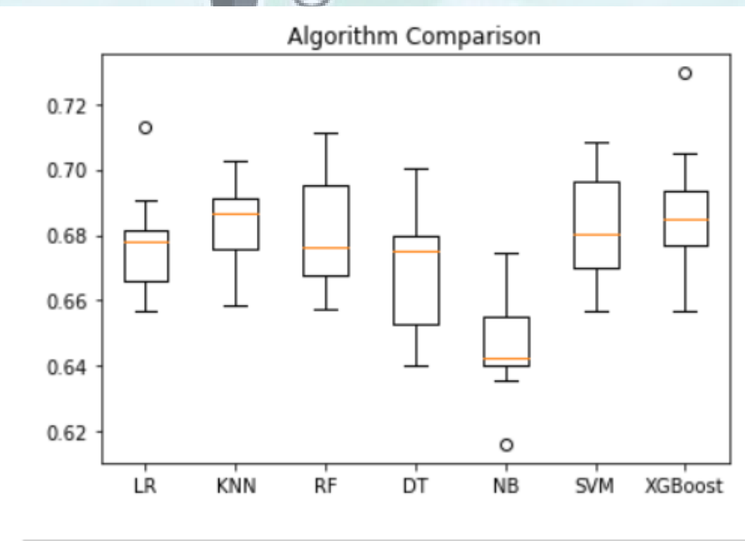
## Decision Tree






| | Model | Train Accuracy | Test Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|---|---|
| 0 | LR | 0.678 | 0.683 | 0.682 | 0.684 | 0.682 |
| 1 | KNN | 0.688 | 0.671 | 0.666 | 0.665 | 0.665 |
| 2 | Random Forest | 0.675 | 0.683 | 0.678 | 0.678 | 0.678 |
| 3 | Dec Tree | 0.673 | 0.665 | 0.659 | 0.653 | 0.654 |
| 4 | Gaussian NB | 0.646 | 0.651 | 0.713 | 0.679 | 0.644 |
| 5 | SVM | 0.685 | 0.697 | 0.697 | 0.700 | 0.696 |
| 6 | XGBoost | 0.684 | 0.688 | 0.683 | 0.681 | 0.682 |

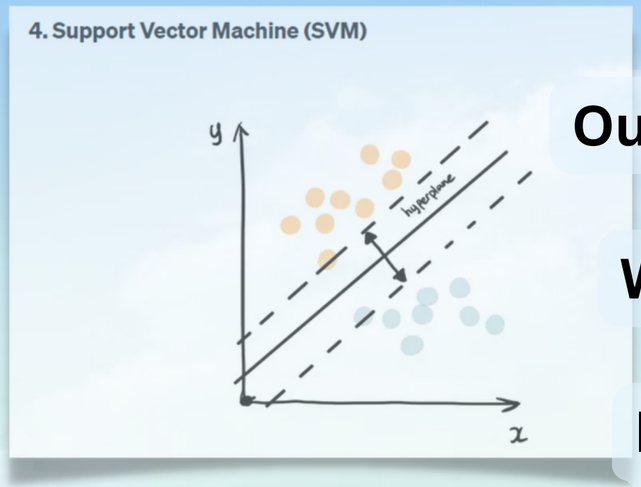
# SIX MODELS RESULTS



# FINAL CONCLUSION

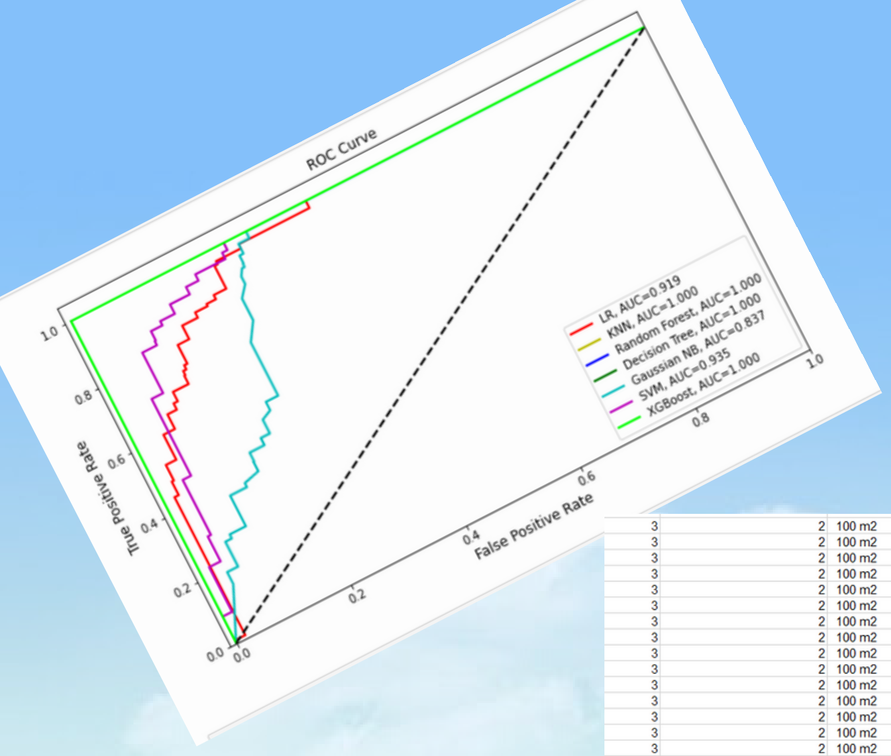
As we can see
he best score we get is from the
SVM model.

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | SVM | 0.685 | 0.697 | 0.697 | 0.700 | 0.696 |




Our conclusion is that it is indeed possible to identify a good investment property in Portugal.
We have indeed seen that machine learning models achieve quite good results.
It was also said that to improve the accuracy more information is needed which cannot be obtained
Apartment sale and rental site
such as the construction price, the quality of the neighborhood, average electricity and water prices for the property
The specific one being tested, and more

# OUR FAILURES



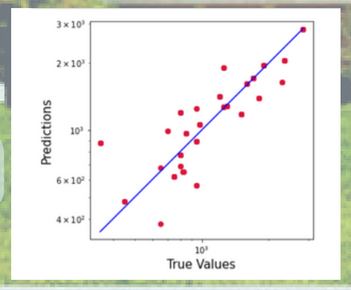

| | | | | | | |
|---|---|---|---|---|---|---|
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 62 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 63 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 64 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 65 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 66 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 67 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 68 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 69 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 70 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 71 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 72 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 73 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 74 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 75 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 76 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 77 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 78 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 79 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 80 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 81 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 82 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 83 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 84 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 85 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 86 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 87 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 88 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 89 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 90 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 91 |
| 3 | 2 | 100 m2 | Condo/Apartment | Braga -, Braga, Braga (Maximinos, S | 198 500 | 92 |

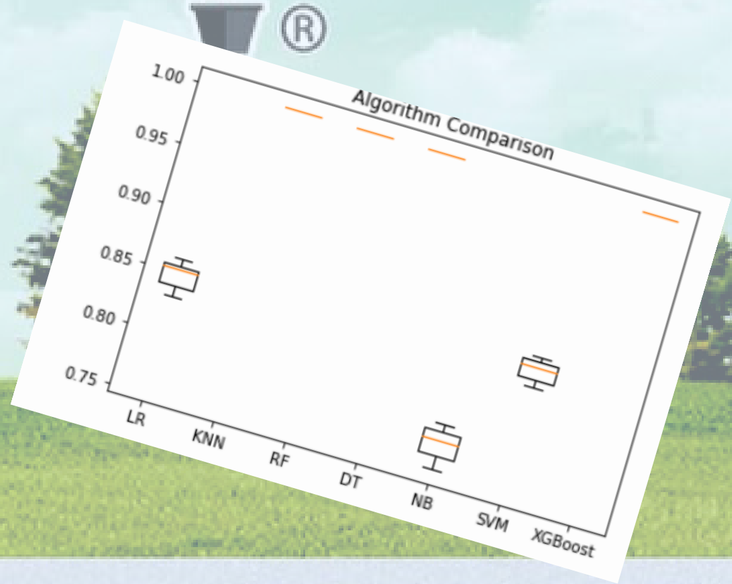
## overfitting and bias

```
LR: 0.847653 (0.009985)
KNN: 1.000000 (0.000000)
RF: 1.000000 (0.000000)
DT: 1.000000 (0.000000)
NB: 0.776267 (0.012458)
SVM: 0.854606 (0.008567)
XGBoost: 1.000000 (0.000000)
```

## scrapping problems

## bad linear regression equation




stackoverflow
OUR
ASSISTANTS
RE/MAX
documentation
OpenAI
Medium
GPT-3

# THANK YOU