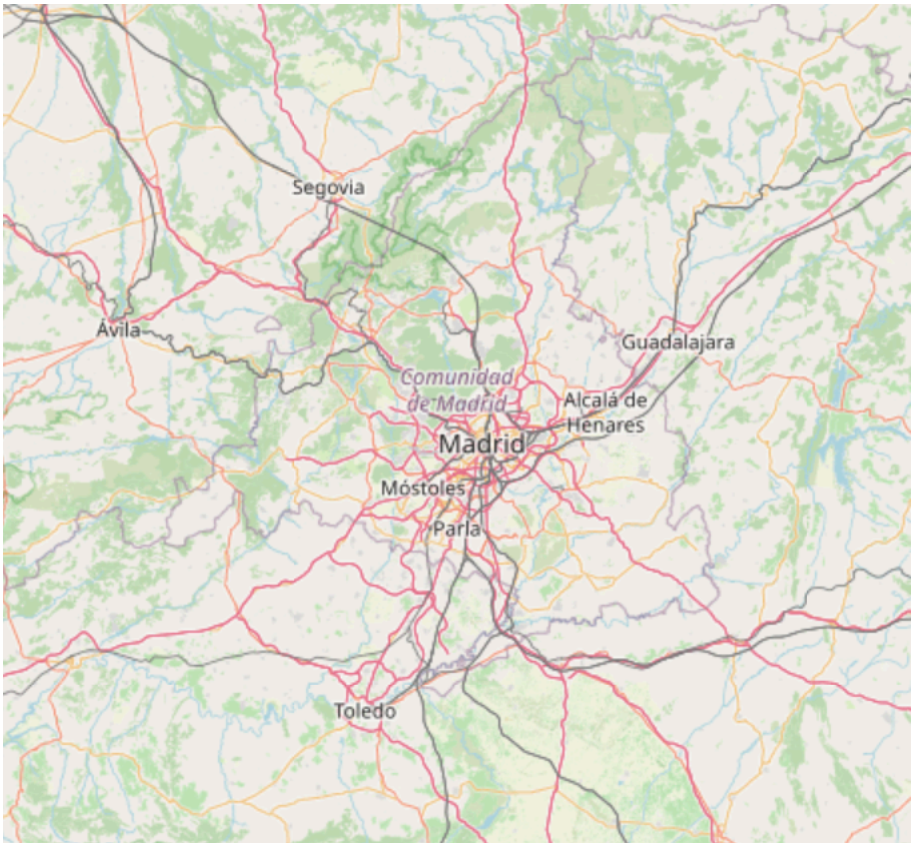
REPORT Xavier Martínez

1. Description of the problem

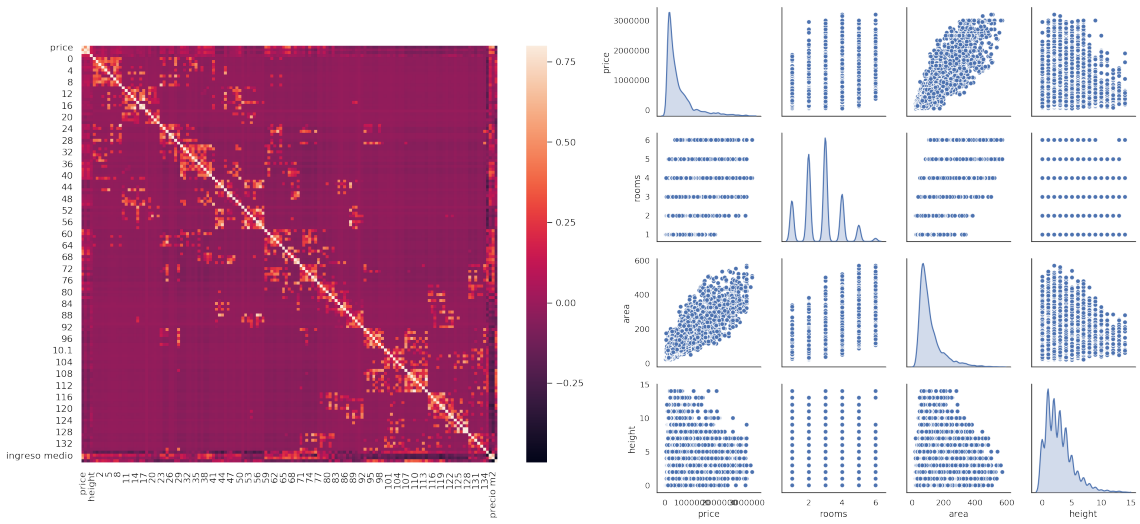
Now we are going to predict and analyze the price of buildings in Madrid. After doing a web scraping, we will show the data and basic statistics of it. We will use the tools studied during the course, such as seaborn, matplotlib or pandas.



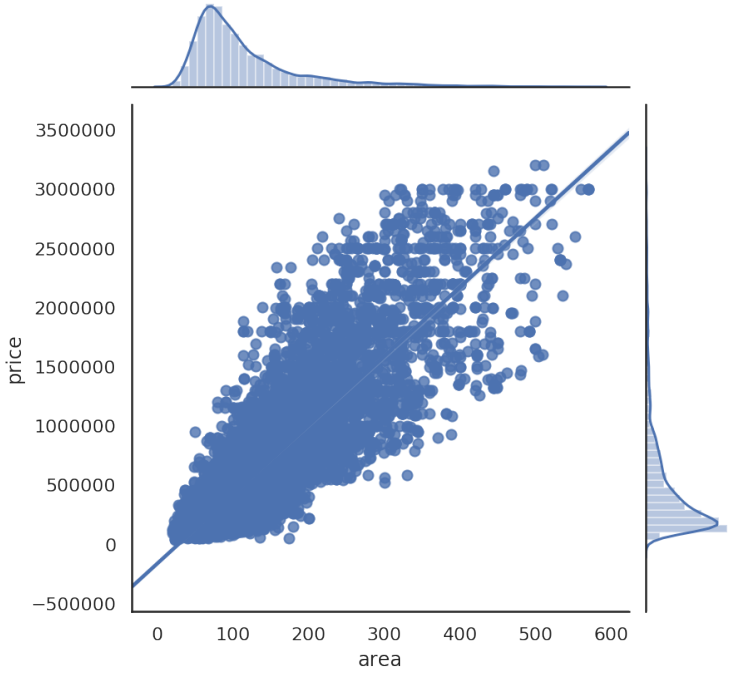
2. Description of the data

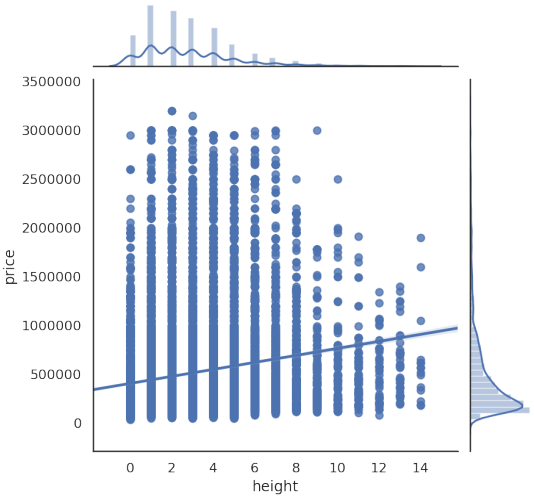
We have approx. 20.000 different buildings and we are going to predict the label, which is the price. We observe some basic characteristics and we observe that some of the features employed are the number of rooms, the crime rate per area, the height, the area and the numbers from 0 to 134 are the different neighborhoods of Madrid

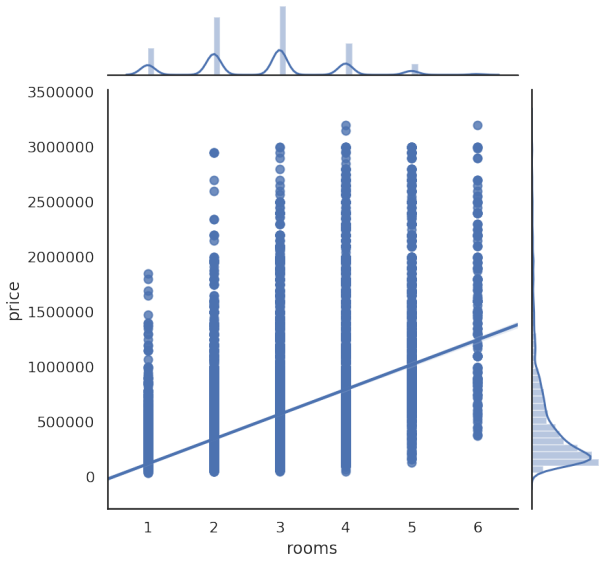
We observe the correlation and distribution of the features:

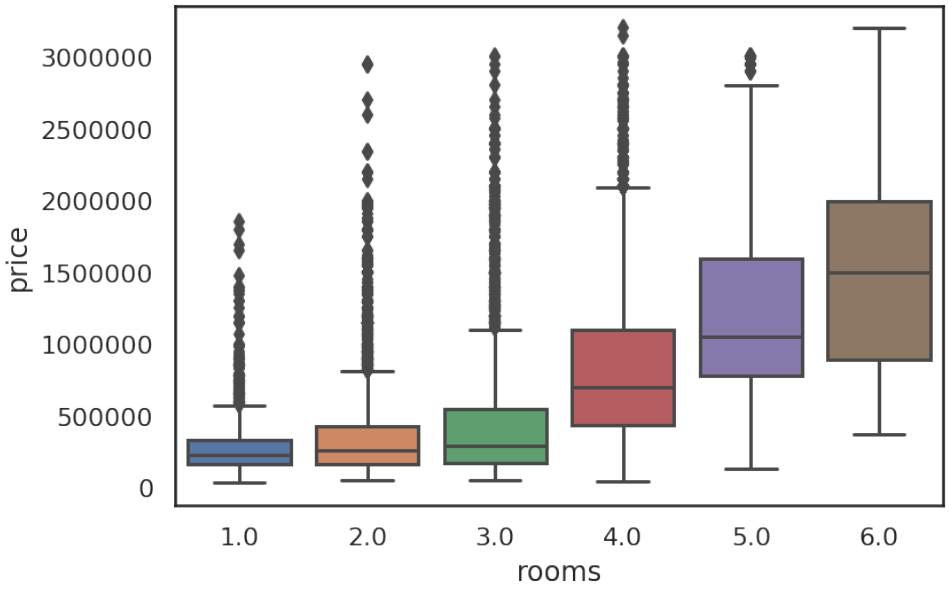


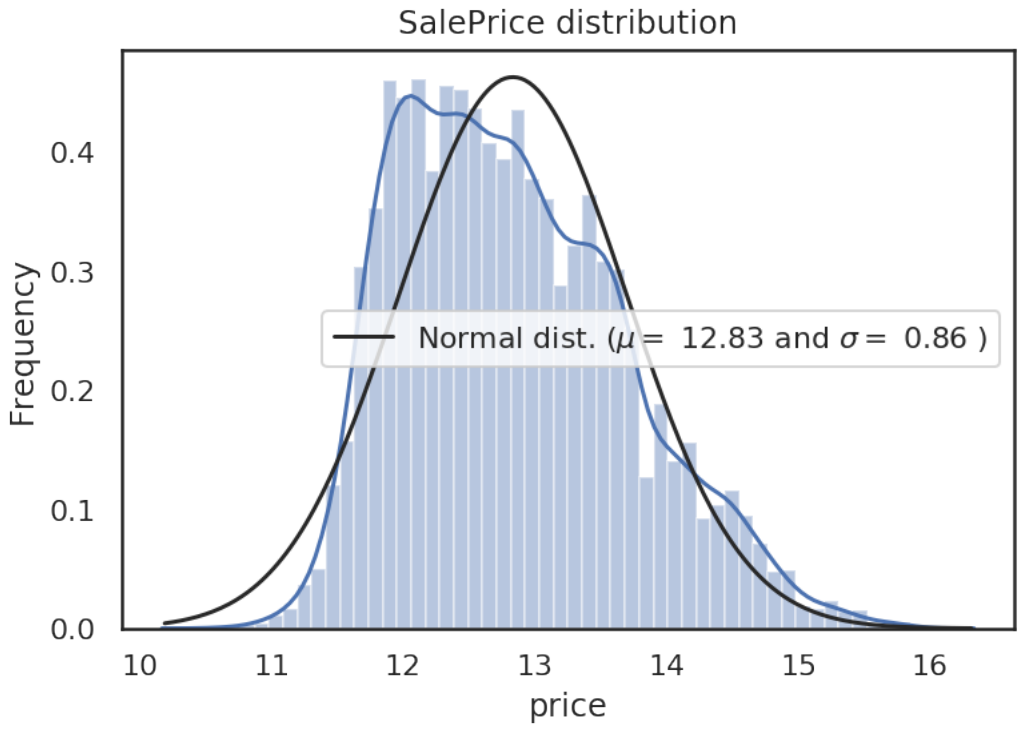
We observe the price respect to the area, the height, rooms..




price
height
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
0
2
4
6
8
10
12
14


price
rooms
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
1
2
3
4
5
6


price
rooms
0
500000
1000000
1500000
2000000
2500000
3000000
1.0
2.0
3.0
4.0
5.0
6.0



3. Interested agents

It would be interesting for real estate investors, it will help to predict the price of the buildings

4. Methodology

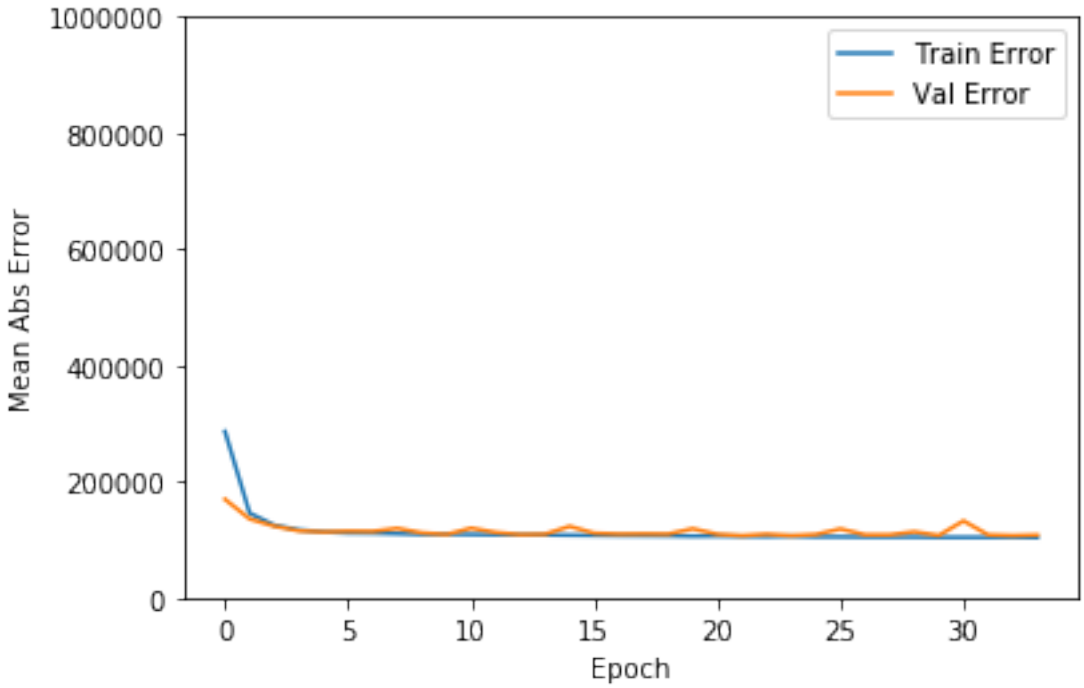
The description of the data is shown by using describe(). Also explanatory data analysis is observable and it is important to delete anomalies that could negatively affect the result of the prediction. The method used is neural networks

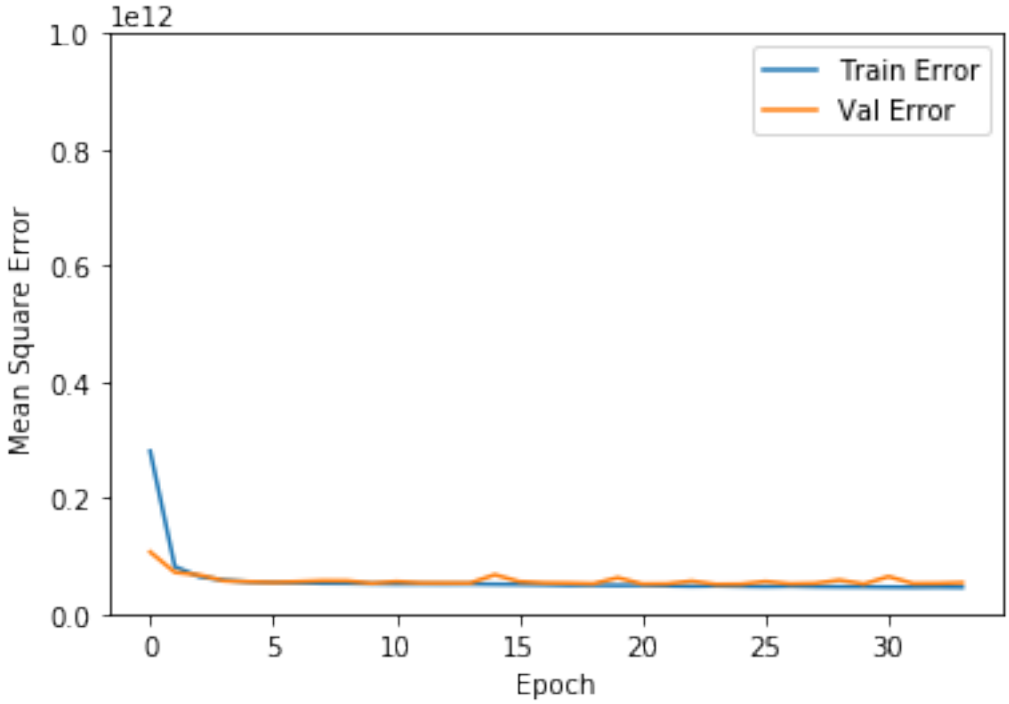
We split data into train and test, this data is normalized and the structure of the neural network could be observable in the following image:

```python
def build_model():
  model = keras.Sequential([
    layers.Dense(256, activation=tf.nn.relu, input_shape=[len(train_dataset.keys())]),
    layers.Dense(128, activation=tf.nn.relu),
    layers.Dense(128, activation=tf.nn.relu),
    layers.Dense(64, activation=tf.nn.relu),
    layers.Dense(64, activation=tf.nn.relu),
    layers.Dense(1)
  ])

  optimizer = tf.keras.optimizers.RMSprop(0.001)

  model.compile(loss='mean_squared_error',
                optimizer=optimizer,
                metrics=['mean_absolute_error', 'mean_squared_error'])
  return model
model = build_model()
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_12 (Dense)             (None, 256)               36608
_________________________________________________________________
dense_13 (Dense)             (None, 128)               32896
_________________________________________________________________
dense_14 (Dense)             (None, 128)               16512
_________________________________________________________________
dense_15 (Dense)             (None, 64)                8256
_________________________________________________________________
dense_16 (Dense)             (None, 64)                4160
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 65
=================================================================
Total params: 98,497
Trainable params: 98,497
Non-trainable params: 0
_________________________________________________________________
```

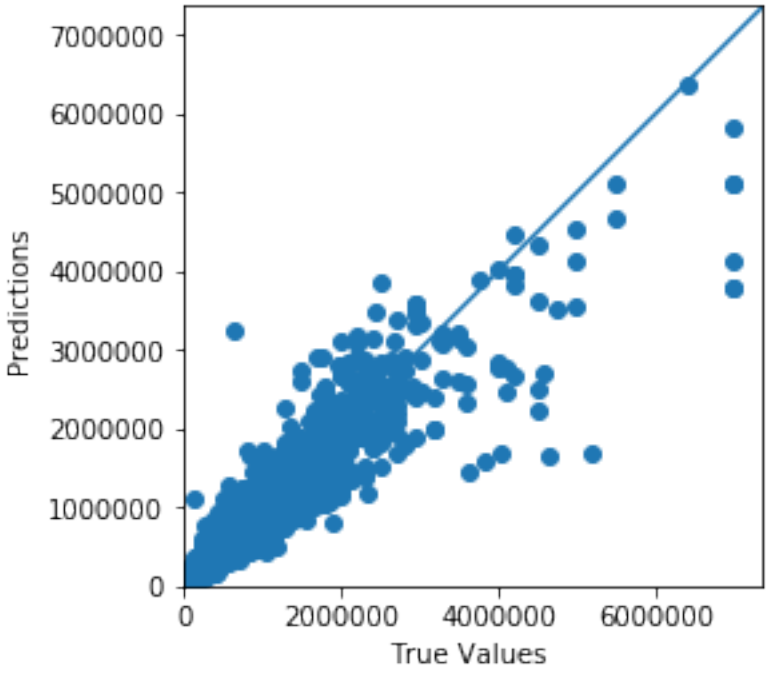
We observe the training of the neural network for both MAE and MSE:

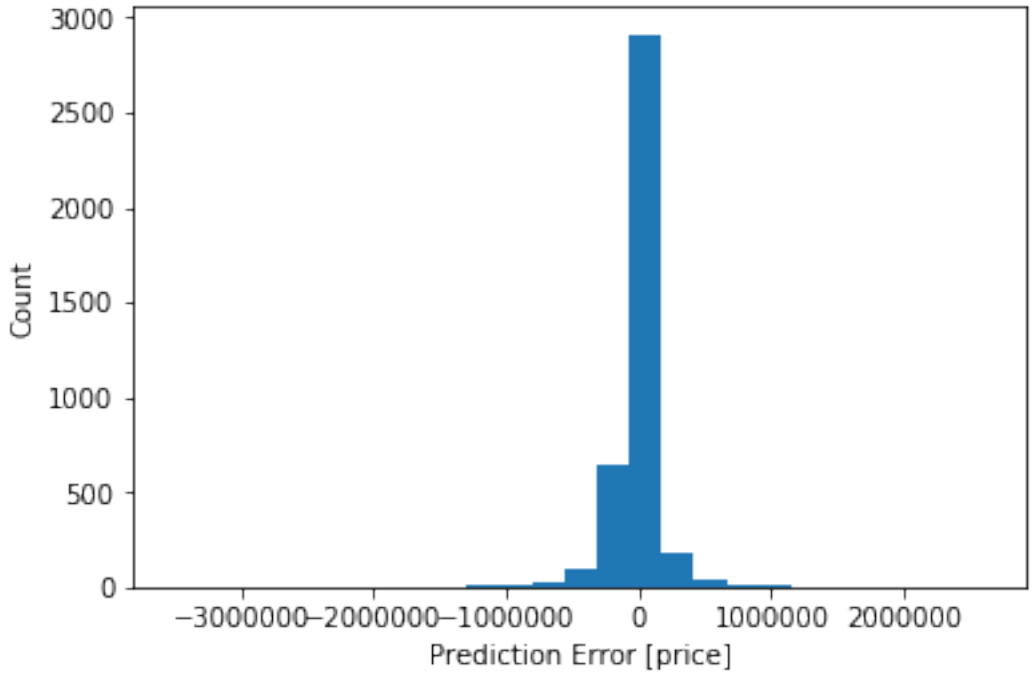

1000000
800000
600000
400000
200000
0
Mean Abs Error
0
5
10
15
20
25
30
Epoch
Train Error
Val Error


1e12
1.0
0.8
0.6
0.4
0.2
0.0
Mean Square Error
0
5
10
15
20
25
30
Epoch
Train Error
Val Error

5. Results

It is observable the prediction in comparison with the true values, the error is quite Gaussian and we have a good model that is able to predict the price of the houses.





6. Conclusion

The conclusion is that using web scraping and this kind of analysis we can predict the price per area in Madrid.