INTELLIGENT ENGINE FOR HOUSES PRICES ESTIMATION

Master in Artificial Intelligence

M05 – Reproducibility and open science Mini project

Authors: Arthur CHERUBINI and Michelle Vanessa MEGUEP SAKAM

Professor: Dr. André Anjos

Table des matières

[I. INTRODUCTION 1](#_Toc51552985)

[II. STATE OF THE ART 2](#_Toc51552986)

[III. HYPOTHESES 3](#_Toc51552987)

[IV. DATABASE 4](#_Toc51552988)

[1. DATA QUALITY 4](#_Toc51552989)

[2. RELEVANT PARAMETERS SELECTION 4](#_Toc51552990)

[3. DATA PREPROCESSING 5](#_Toc51552991)

[VI. CODE STRUCTURE 7](#_Toc51552993)

# INTRODUCTION

The goal of this project is to develop an algorithm which can be used to predict house prices. In fact, several parameters can be taken into account in the evaluation of a property, such as the characteristics of the house, the street and the town in which it is located, the surrounding shops, the price of neighboring houses, etc.

In order to have accurate estimates, it is important to have a good database. If not all the parameters are selected to make predictions, it is important to make an analysis of which are the parameters which can be the more relevant for predicting the house price. To do so, the first part of the project is devoted to finding which are these relevant parameters by computing for example the Pearson correlation factor between parameters and target values. In addition, to make an analysis which is the most complete as possible, the performance and the predictive power of different algorithms such as linear regression, decision tree and random forest will be evaluated.

Nowadays it is capital that all the results published are reproducible and therefore it is important for us to ensure that all the results presented in this paper are reproducible.

# STATE OF THE ART

Predicting the house price is a well-known problem in the field of machine learning and it is often seen as a toy example. More particularly, the Boston House Prices is the dataset used in general to illustrate the problem. Originally, this dataset was presented in the [paper](https://www.researchgate.net/publication/4974606_Hedonic_housing_prices_and_the_demand_for_clean_air) from 1978 “Hedonic Housing Prices and the Demand for Clean Air” written by David Harrison and Daniel Rubinfeld. This paper investigates the methodological problems associated with the use of housing market data to measure the willingness to pay for clean air by using a dataset collected in Boston containing 506 samples with 14 features per sample. It must be said that the prediction of costs or values is a fairly widespread field of application for artificial intelligence. This is why this dataset was reused massively, and a lot of projects have already been carried out (cf. Kaggle competition). What is interesting is that it can be seen that it is possible to approach this problem from many different point-of-views.

In the scope of this project, the dataset used is not Boston House Prices, but a dataset inspired from it and which was presented in the [paper](https://doi.org/10.1080/10691898.2011.11889627): “Ames, Iowa: Alternative to the Boston Housing Data as End of Semester Regression Project” published by Dean De Cock in 2011 in the Journal of Statistics Education. The purpose of this dataset is purely for an educational point-of-view and was designed for a regression project for students to apply the skills they had learn. Similarly to the Boston House Prices problem, it is possible to find a lot of work done with the Ames dataset in Kaggle or other platforms.

As part of this project, one objective will be to predict house prices with different machine learning algorithms and compare their performance. The interests for such a prediction tool are well highlighted by [one](https://www.kimanalytics.com/single-post/2017/09/11/Predicting-House-Prices-with-Machine-Learning) of the numerous work which can be found online:

* It can be a tool used by people who would like to buy a house and they want to know whether the price corresponds to true house value.
* Or it can be a tool for anyone who wants to invest in real estate and make a big profit from it. This would allow them to invest the most money on the features that drive up house prices the most.

# HYPOTHESES

One of the most important steps of scientific method is to define which are the working hypotheses of a specific project. In our case we have two hypotheses, the first one which is kind of general while the second one is more precise:

* If the intrinsic value of a property is defined by different parameters such as the number of rooms or the total surface for example, then it must be possible to have an algorithm which can predict with good accuracy the price of a property.
* If some parameters are more meaningful than others to explain the price of a house, then by taking into account only these parameters the price of a house should be still predictable.

To verify the first hypothesis, what we will do is to give at the input of the machine learning algorithms the 80 variables we have and analysis the results to see whether the hypothesis has to be rejected or not. To verify the second hypothesis, we will first determine which are the variables having the highest correlation to the property price and we will use only those variables to predict the property price.

# DATABASE

As specified in the "State of the Art" section, our project will use the [Ames data set from Ames, Iowa](https://ww2.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt) database presented by Dean De Cock in his paper. This database is composed of 2930 observations with 80 features each, among which 23 are nominal, 23 are ordinal, 14 are discrete and 20 are continuous variables. The 80 variables focus on the quality and the quantity of many physical attributes of the property, such as the street, the house style, the year built, the type of road access to property etc. Most of the variables are exactly the type of information that a house buyer would want to know about a house. As a final note, the dataset contained sales that had occurred in Ames from 2006 and 2010.

## 

## DATA QUALITY

It is important to make a difference between the nominal and ordinal parameters which are categorical parameters and the discrete/continuous parameters. This distinction has to be made for two reasons. First when specific data are missing it does imply the same thing whether the parameter is categorical or continuous/discrete. Then, the data preprocessing is not the same at all for these two kinds of data.

If a data is missing for a categorical variable it does not mean that we don’t have the information. Indeed, when looking to the documentation, we see that for example the variable “Garage Condition” can take the value “Excellent”, “Good”, “Typical/Average”, “Fair”, “Poor” or “No Garage”. However, when there is no garage instead of having a string mentioning this, there is simply no data. Therefore, for all the categorical data, when a value is missing, we simply replace it by the string “None”.

In the case of discrete/continuous variable, it is different because when a data is missing, it means that this information is not available. Therefore, when such a situation is encountered, the sample is removed.

## RELEVANT PARAMETERS SELECTION

To test our second hypothesis which is: “If some parameters are more meaningful than others to explain the price of a house, then by taking into account only these parameters the price of a house should be still predictable”, we have to preselect these meaningful parameters.

First, to simplify the analysis only discrete/continuous parameters are taken into account. Then, for each one of these parameters, the correlation existing between them and the house sale price is computed thanks to the Pearson correlation factor. Only the parameters with high correlation are kept.

#ADD PLOT CORRELATION MATRIX

## DATA PREPROCESSING

It has been mentioned above that the data preprocessing will not be the same whether the data are categorical or a discrete/continuous.

* discrete/continuous: It is well-known that normalizing the input data of a machine learning algorithm helps the training of such an algorithm. Therefore, we decided to apply to all the discrete/continuous variables a z-normalization:

If we represent the discrete/continuous parameters used to predict the price of the house as an 2D array where the number of lines is the number of observations and the number of columns is the number of discrete/continuous parameters of interest, then a mean and a standard deviation is computed for each columns. It means that concretely the mean and standard deviation are vectors with as much values as there are parameters.

It should be noted that the mean and standard deviation are calculated only on the training set. If we were to calculate them on the whole dataset, it would mean that we were using information from the test set, which should be avoided at all costs.

To note that the target price which is the sale price is a continuous variable and we also apply a z-normalization to it.

* categorical: The categorical data we have can’t be fed directly at the input of the machine learning algorithm under the current format. They have to be converted into a form that could be provided to ML algorithms. To do so, one hot encoding is used. Basically, a categorical integer is replaced by a vector of the size of the number of categories that this integer can take. A 1 is attributed to the category at the which the sample belongs to and 0 is attributed to all the other category. This is illustrated in the following example:

Let’s take again the variable “Garage Condition” which can take the value “Excellent”, “Good”, “Typical/Average”, “Fair”, “Poor” or “No Garage”. There are 6 possible categories for each samples, therefore at the input of the ML algorithm instead of having a single value we will have 6 values. If the category is “Excellent”, the vector will be . If the category is “Poor”, the vector will be .

The last part of the data preprocessing part is to concatenate the discrete/continuous variables with the categorical one. For each observation we will have the normalized discrete/continuous variables and the vectors after the one-hot encoding.

2. THE MODEL

# TALK ABOUT WHY WE HAVE CHOOSEN THE ALGORITHMS WE HAVE CHOOSEN

1. PREDICTIVE POWER COMPARAISON

In this section, we will compare the predictions obtained with linear regression algorithms with those obtained with decision trees and random forests.

## CODE STRUCTURE

To simplify the reproducibility of our experiments, a specific structure has to be followed for the code. Each block represented in the graph bellow is a generic piece of code which can be easily replaced by another block. All the toolchain is managed by a configuration file containing all the parameters which have to be set by the user, such as the protocol for splitting the data, details about the algorithms and especially which variables are used to predict the house prices. Here is a description about each block:

* Database: This is the part where the database is read and the data cleaning is done (remove samples where discrete/continuous variables is not defined and replaces by None all the non-defined value of categorical variables). This is also the block where the splitting into a train, a cross-validation (cv) and a test set is done accordingly to the protocol defined by the user. This block is very specific to each database and if now someone want to use the Boston House Prices dataset, a new block has to be created but all the other blocks composing the toolchain can be reused without modification.
* Data Preprocessing: In this block the discrete/continuous variables are normalized as well as the target value, and one hot encoding is applied to the categorical data. At the end, discrete/continuous variables and the categorical ones are concatenated together. At the output of this block, the data are ready to be used with machine learning algorithms.
* Algorithm: The algorithm block will be different for each machine learning algorithm tested (random forest, decision trees, etc). It is composed of two smaller blocks called Training and Testing. In the first one, a grid search is performed with the training and the cv set to determine the best model. Then, this best model is used with the test set in the Test block to evaluate the performance on unseen data.
* Analysis: The performance of different algorithms on the same testing set are compared here with the help of some graph or whatever tool needed.

Une image contenant capture d’écran

Description générée automatiquement

1. UNITS TESTS
2. CONCLUSION

## 