Universidad de Barcelona

House Prices Prediction Project

//////////

September 2020

Agenda

1. Introduction

// Project description and goals

// Project structure and data sets.

2. Data analysis and processing
3. Regression Models

// Baseline Model applying Logistics Regression

// Random Forest Regressor Model

// Artificial Neural Network Model

4. Results & Conclusions
5. Closure: Contributors & Thanks

Project Description and Goals

Introduction

Project Description

- Applying advanced regression techniques to predict the house prices of the Ames city, a region of the United States.
- This case study is part of a Kaggle competition (House Prices: Advanced Regression Techniques). So, we created a working team (**bcnDataScience**) in the Leaderboard of competition to participate in it and test the theoretical knowledge learned during the course.
- We applied some data analysis processing methodologies, such as data cleaning, categorical data analysis and normalization; then, we built some regressions models in python based on Logistics Regression, Random Forest and Artificial Neural Networks to predict the sale price of the houses.
- Some regression metrics, such as R-squared and Mean-Squared-Error, have been applied to test the performance of the models.

Goals

- Find the best fitting model.
- Get the best results and score possible in the Kaggle competition.

Project Structure and data sets

Introduction

The project material is stored in a Github repository, which is called [Postgraduate-Project].

The Github repository is divided in two main folders:

- ***House Prices Prediction***: Here we store documents with explanations of the challenge and data downloaded from the Kaggle platform, models testing's and files with data.
- ***Kaggle Competition***: Here there are the Jupyter Notebooks with the last versions of the regression models and final predicted results that we delivered in Kaggle. Below you can find the details of the main notebooks:
  - House Prices Prediction: Logistics Regression & Random Forest
  - House Price Prediction: Artificial Neural Networks
  - Overview Kaggle Submissions & Results

Important! Subfolders with explanations have been included to saved the csv and pkl files generated by the models to classify correctly the data and avoid confusion.

Original Datasets

Data analysis and processing

The dataset that are part of the House Price Prediction challenge are composed of the following components:

Train dataset characteristics:

- 81 features (including independent and dependent variables) and a total of 1460 records.
- From the total of the variables, 38 features are numerical and the remaining 43 are categorical.
- The independent variable (the values to predict with the regression models) is the ***SalePrice*** (sale price of the houses).

Test dataset characteristics:

- 80 features (only independent variables) and a total of 1459 records.
- From the total of the variables, 37 features are numerical and the remaining 43 are categorical.
- We do not have the ***SalePrice*** variable in the test data because the goal of the challenge is to predict its values.

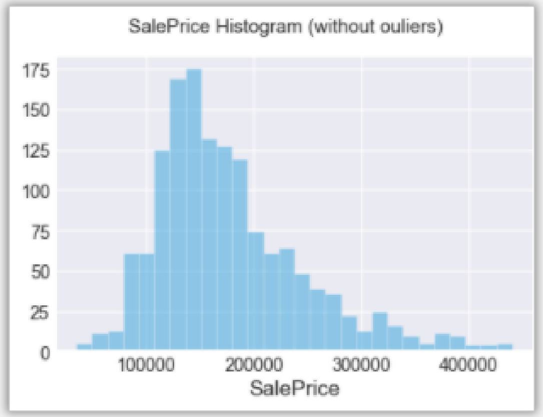
In order to build the best predictive model we need to analyze, clean and implement some adjustments in the original data. So, in the following slides we will explain the actions taken to achieve this objective.

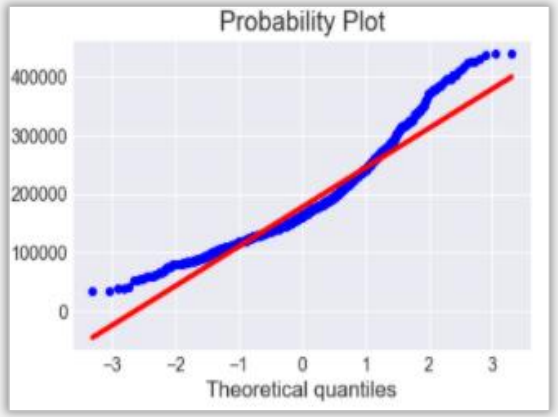
Data analysis and processing – Phase 1

SalePrice Distribution Analysis

- After analysing the data of the SalePrice variable, we realized that it did not follow a normal (gaussian) distribution, maybe due to outliers.
- We decided to remove the outliers (15 records) from the data to avoid inconsistencies when testing and predicting data.

Results after dropping the outliers:





Skewness: 1.094946
Kurtosis: 1.276040

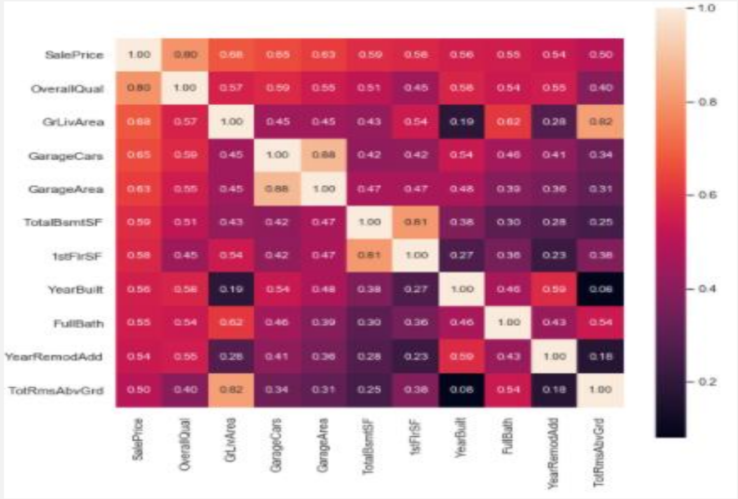
Missing & Null Values Analysis

- We identified a total of 13905 records with nan values that should be adjusted in the data.
- First, we removed those columns with more than 48% missing values (PoolQC, MiscFeature, Alley, Fence, MasVnrType).
- Then, for numerical variables we replaced the nulls with the median or mean (in case median was equal to 0), while for categorical variables we replaced missing values with the most occurring variable data.
- Finally, we dropped the 'Id' column from the training and test data-frames.

Data analysis and processing – Phase 2

Correlation Analysis

Top 10 variables with the highest correlation:



List of the removed variables with low correlation:

- ScreenPorch
- MoSold
- 3SsnPorch
- PoolArea
- BsmtFinSF2
- MSSubClass
- MiscVal
- BsmtHalfBath
- YrSold
- LowQualFinSF
- OverallCond

Categorical Variables Analysis

- Using the training set, we have analyzed the impact of the categorical variables on the price values.
- After checking the shape and the dimension of each of the categorical variables, we decided to implement the following adjustments:
  - Convert some categorical variables into dummy*, and,
  - Transform the remaining categorical variables into text to build a fully numerical baseline model.

*LotShape', 'LandContour', 'LandSlope', 'BldgType', 'ExterQual', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'CentralAir', 'KitchenQual', 'GarageFinish', 'PavedDrive

Final Results

Data analysis and processing

After finished the adjustments explained in the previous slides, we have two datasets composed of the following characteristics:

A training set composed of 100 features (independent and dependent variable) and 1324 records.
A test set composed of 99 features and 1459 records.

The total records of the training and test do not match because, keeping in mind the kaggle's rules, we must deliver submission files composed of 1459 records related to the predicted test data.

Now, we are ready to build the prediction models applying machine learning techniques.

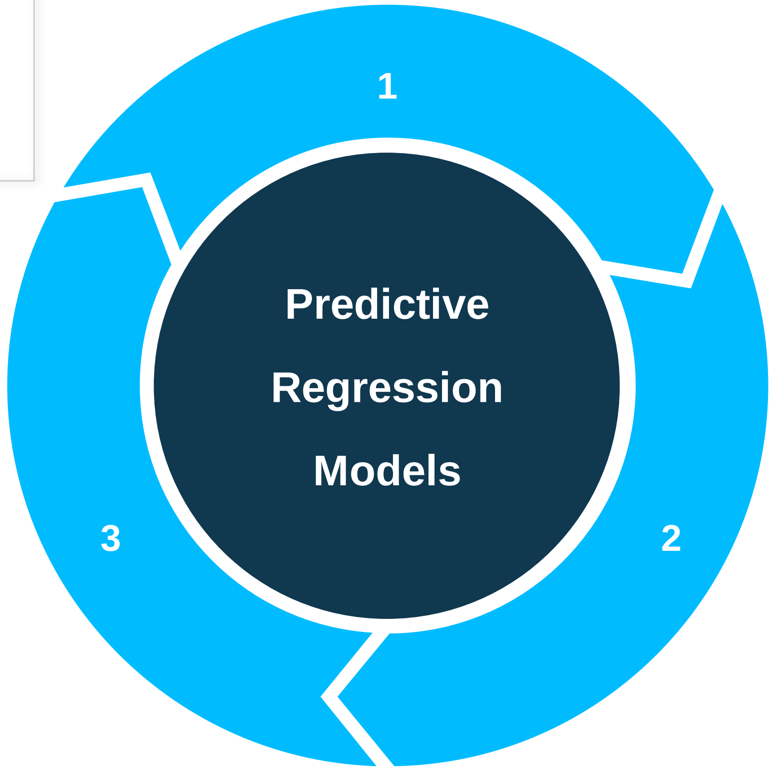
Important! We have to normalize the data to build the ANN model in order to avoid wrong effects in the model performance when predicting values. For this, we will use the standardization method called **Standard Scaler.**

Overview of the construction process of the regression models

Regression Models

1 Baseline Model

We started creating simple model (which includes numeric variables and text variables transformed into dummy and numbers) applying logistics regression.



2 Random Forest Regressor Model

We improved the metrics results applying a more robust model to the baseline, the
Random Forest.

In addition, we included the cross-validation and grid search to look for those best parameters that provide us with the best possible result.

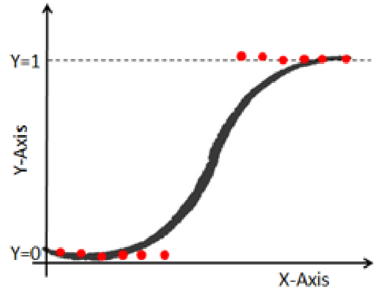
3 Artificial Neural Networks Model

Finally, we applyed Deep Leraning methods to build a strong model and try to improve the results achieved with the Random Forest.

To do this, we have normalized the data and built a neural network with 3 hidden layers, which has provided us with the best results in the competition.

Baseline Model applying Logistics Regression

Regression Models



Pilot Model 1

- We **only** considered **numerical variables**.
- Firstly, we included the variables with high correlation and then the remaining numerical variables.

Training Score: **0,5709**
Training MSE: **598.234.875,65**

Pilot Model 2

- To the initial model that only contained numeric variables, we included those **categorical variables transformed into dummy** that show a positive impact on the model.

Training Score: **0,7328**
Training MSE: **254.689.611,87**

Baseline Model

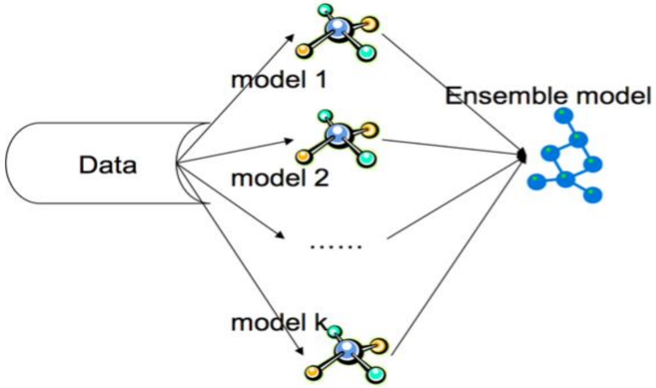
- Finally, we transformed the remaining categorical variables into numbers and added them to the previous model to build the reference model with Logistics Regression.

Training Score: **0,8747**
Training MSE: **114.541.500,30**

The objective is to improve this results with the Random Forest and Artificial Neural Networks models.

Random Forest Regressor Model

Regression Models



RF Baseline Model

- Starting from the Logistics Regression baseline model, we will create a simple Random Forest model and then we will improve it selecting those parameters that provide the best possible results.

Training Score: **0,9739**
Training MSE: **126.354.326,34**

K-Fold cross validation

- As we do not have an y_test to predict, we split the training set into data train (K-folds subsets) and data validation and fit then the model to check the performance.

Accuracy: **85.15 %**
Standard Deviation: **3.81 %**

Final RF Model

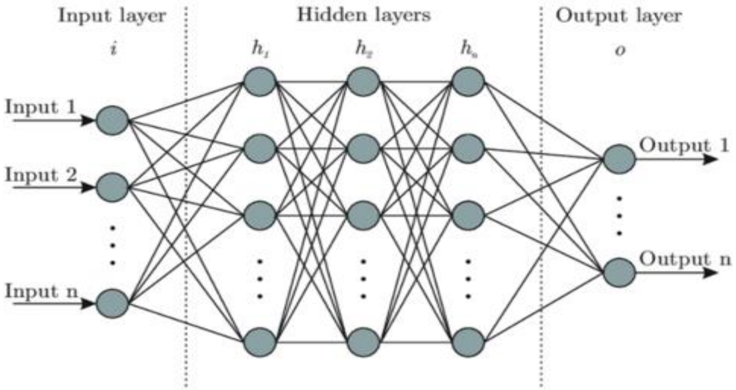
- We will tuning some hyper-parameters to optimize the model. First, we do a random search to find the value ranges with which we achieve a good score.
- Then, we apply the "Grid-Search" method to find the best parameters.

Training Score: **0,999**
Training MSE: **195.08,86**

The result achieved in the MSE is really low, considering that the maximum price of the houses oscillated in 500,000 USD in the training test. We will try to improve it with the ANN.

Artificial Neural Network Model

Regression Models



- Starting from the Logistics Regression baseline model, we will try to improve the results building a robust with an ANN.
- It is necessary to normalize the values to avoid the effect of incorrect influences when predicting values..

Artificial Neural Network Architecture

Input Layer

- A total of 94 input variables.

Hidden Layers

- A total of 3 hidden layers.
- 50 Units and the ‘relu’ Activation Function.

Output Layer

- Units = 1
- No Activation Function.

Fitting the Artificial Neural Network

Compiling the ANN

- Optimizer = 'adam'
- Metrics = 'mean_squared_error'.

Fitting the ANN

- batch_size = 32; epochs = 200,
- Validation_split = 0.2 (valiation data)

Results:

R-Squared = 0.9146
MSE = 0.0853

The results are calculated based on the training predicted values.

Outcomes Models Summary

Results and Conclusions

	Logistics Regression	Random Forest Regressor	Artificial Neural Networks
R-Score*	0,976	0,9999	0,9412
Mean-Squared-Error*	114,541,500	19.508,86	0,058
Root-Mean-Squared-Error*	N/A	139,67	0,2424
y_pred_train	208500; 181500; 223500 ... 266500; 142125; 147505	208500; 181500; 223500 ... 266500; 142125; 147505	208105.67; 181104.62; 224340.47 ... 262125.66; 141739.89 163646.03
y_pred_test	176000, 158000, 190000 ... 153337, 134500, 187500	125465.91; 157284.68; 182329.08 ... 158148.56; 123950.85; 222635.03	106.053,51; 186.427,40; 203.674,38 ... 148.035,94; 136.759,94; 232.750,22
Kaggle Score	0,30963	0,16192	0,15832

*The resultas are calculated on based of the predicted values.

Conclusions

- We have tried to find the best fitting model to predict the house pricing for a specific location in the United States applying machine learning techniques. This challenge is part of the "House Prices Prediction: Advanced Regression Techniques“ Kaggle competition .

- The first step was cleaning the data sets, both the training set and the test set, from missing and non-available values. Then, we have eliminated the features with the lowest correlation with the price variable. Later, using the training set, we have analyzed the impact of the categorical variables on the price values to check the distribution and distinguish the features that should be analyzed and what should be processed as dummy variables if needed.

- Once the data was cleaned, we start to build the prediction models. To diversify our analysis, we have chosen three supervised learning techniques: logistic regression, random forest and deep learning. We chose the logistic regression analysis to obtain a baseline model and to identify all concerns and metrics related to the price housing prediction. Afterward, we implemented the random forest model and the artificial neural network model.

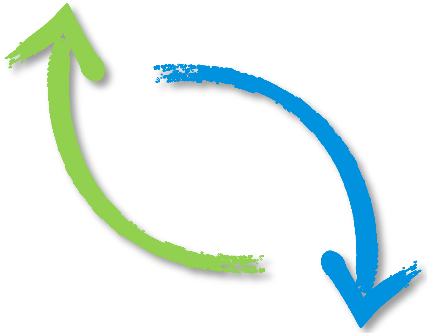
- The results obtained based on the metrics R-score and mean-square-error reveal that all models enable house price prediction. The best predictor is the deep learning model, followed by the random forest model and logistic regression models, respectively.

- Finally, to verify the performance of our prediction models, we adapted and submitted the corresponding notebooks to the Kaggle competition. As expected, the performance of the deep learning and random forest models overcomes the performance of the logistic regression model base on the Kaggle ranking. Specifically, our Kaggle working group ("bcnDataScience“) have obtained a best score of 0.15832 belonging to the deep learning model, and is located in the rank 3391 out of 4886 teams.

¿Quienes han contribuido en este proyecto?

Lorena Méndez

Nuria Sánchez



Christian Tipantuña

Theresa Kothe

¡Gracias!