Big Data Analytics and Interactive Visualization

Predicting house prices using Apache Spark

February 2020

Environment Description

The used environment is a dockerised Spark container that includes python 3.7.3 and Spark.

Introduction

The goal of this assignment is to predict the price of a house based on given set of features. It is an attempt to a Kaggle competition titled "House Prices: Advanced Regression Techniques"[1].This is a regression task, meaning that the target variable is a numeric (continuous value), so instead of recognizing patterns we recognize trends.

Dataset

The dataset used in this project is the Ames Housing dataset compiled by Dean De Cock for use in data science education[2]. It describes the sale of individual residential property in Ames, Iowa from 2006 to 2010 and originally came directly from the Assessor's Office. The dataset prepared for this task contains 1460 observations and 80 explanatory variables (23 nominal, 23 ordinal, 14 discrete, and 20 continuous) involved in assessing home values. The variables describe almost every feature of a house, starting with the basic ones, like total living area, neighbourhood or year in which the house was built, to less relevant ones, like quality of a fireplace, area of a swimming pool or number of bathrooms in a basement. Since not every house has a swimming pool or a basement, this information is represented by null values. Moreover, in the dataset there are also unusual sale cases indicated by the feature 'SaleCondition'. If its value is different than "normal", the price the house was sold might not be representative. There are also some outliers in the dataset - big houses sold for a relatively low price.

Summary of Exploratory Data Analysis

The Exploratory data analysis starts with taking a look to the data set and checking how many missing values are present into the dataset. Once it has been done, some of the most important features are analyzed in order to understand their distribution. After that, based on some assumptions, many relevant features are returned and for them a correlation coefficient analysis is computed in order to understand how much they can influence the variance of the dependent variable. Once the defined most import features are returned, the data are standardized and the regression analysis is computed.

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	I
0	1	60	RL	65	8450	Pave	NA	Reg	Lvl	AllPub	...	0	NA	NA	NA	
1	2	20	RL	80	9600	Pave	NA	Reg	Lvl	AllPub	...	0	NA	NA	NA	
2	3	60	RL	68	11250	Pave	NA	IR1	Lvl	AllPub	...	0	NA	NA	NA	
3	4	70	RL	60	9550	Pave	NA	IR1	Lvl	AllPub	...	0	NA	NA	NA	
4	5	60	RL	84	14260	Pave	NA	IR1	Lvl	AllPub	...	0	NA	NA	NA	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62	7917	Pave	NA	Reg	Lvl	AllPub	...	0	NA	NA	NA	
1456	1457	20	RL	85	13175	Pave	NA	Reg	Lvl	AllPub	...	0	NA	MnPrv	NA	
1457	1458	70	RL	66	9042	Pave	NA	Reg	Lvl	AllPub	...	0	NA	GdPrv	Shed	
1458	1459	20	RL	68	9717	Pave	NA	Reg	Lvl	AllPub	...	0	NA	NA	NA	
1459	1460	20	RL	75	9937	Pave	NA	Reg	Lvl	AllPub	...	0	NA	NA	NA	

1460 rows × 81 columns

Figure 1: Dataset overview

[1] `https://www.kaggle.com/c/house-prices-advanced-regression-techniques`, access December 2019
[2] `http://jse.amstat.org/v19n3/decock.pdf`, access December 2019

Managing 'NaN' values

After having understood the dataset, the first thing that has to be done it is to check how many missing values there are. Taking a look to the dataset, it is possible seeing that there are a lot of missing values. However, not for all of them it means that the data have been wrongly recorded but that some houses don't have those facilities. In particular:

- NaN value for **PoolQC**, means that the house does not have the Swimming pool.
- NaN value for **Alley**, means that there is no alley access.
- NaN value for **MiscFeature**, means that there are no miscellaneous feature.
- NaN value for **Fence**, means that there is no fence.
- It is possible observe the same also for other twelves features.

However, there are three other features that instead have "real" missing values, meaning that something went wrong during the data recording. These features are:

- **LotFrontage**.
- **GarageYrBlt**.
- **MasVnrArea**.

Neighbourhood feature analysis

Taking a look to the Neighbourhood feature is a good approach because seems reasonable that there are some neighbourhoods better than others. For that reason, a bar-chart, showing the average price per each neighbourhood, is returned in the figure below.

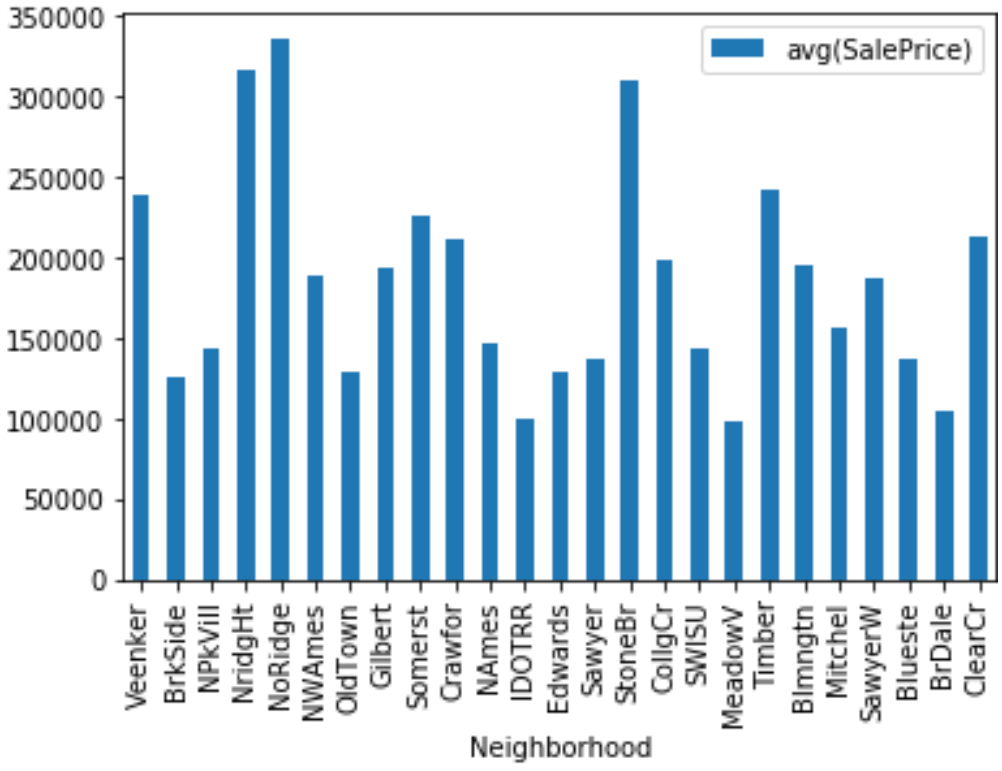

Figure 2: Average SalePrice per Neighbourhood

As it is possible seeing there are three different neighbourhoods that have a very high average price and other three neighbourhoods that instead have a low average price. Taking a further sight is possible seeing that there are four kind of neighbourhoods (high, medium-high, medium and low). However, this categorical feature will not be used to predicted the dependent variable, because it has to be encoded by using One-hot encoding or assigning ranks in order to transform it into a numerical feature. However using one-hot encoding or ranks will return to many features (in particular 25 features) and this will require to use other more techniques like (PCA) in order to reduce the dimensionality.

Bedrooms feature analysis

Another import feature could be the Bedrooms feature because seems to be reasonable that an house with more bedrooms should be bigger than an house with less bedrooms, and bigger house should be more expensive than a small one. The picture below shows a pie-chart where are reported the most frequent number of bedrooms. However, also this feature seems to be not so useful for predicting the house price, because there could be houses that have less bedrooms but more other rooms, like leaving room, kitchen, garage, and so on. For that reason it has been decided to exclude this one from the basket of best features used to build up the generalized function.

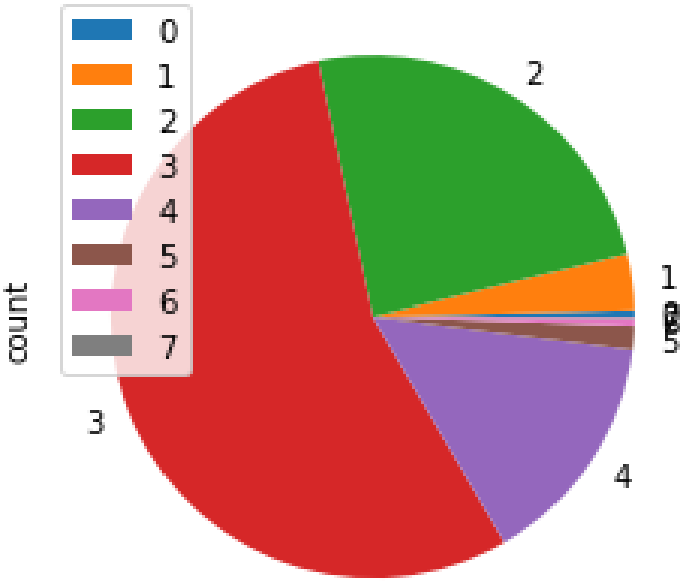

Figure 3: BedRooms Pie-Chart

Checking the distribtion of 'SalePrice'

Now it is necessary take a look to the target feature ('SalePrice'). For that reason a histogram is plotted to check the skweness in order to understand if the distribution is symmetric or not. The picture below shows the SalePrice distribution.

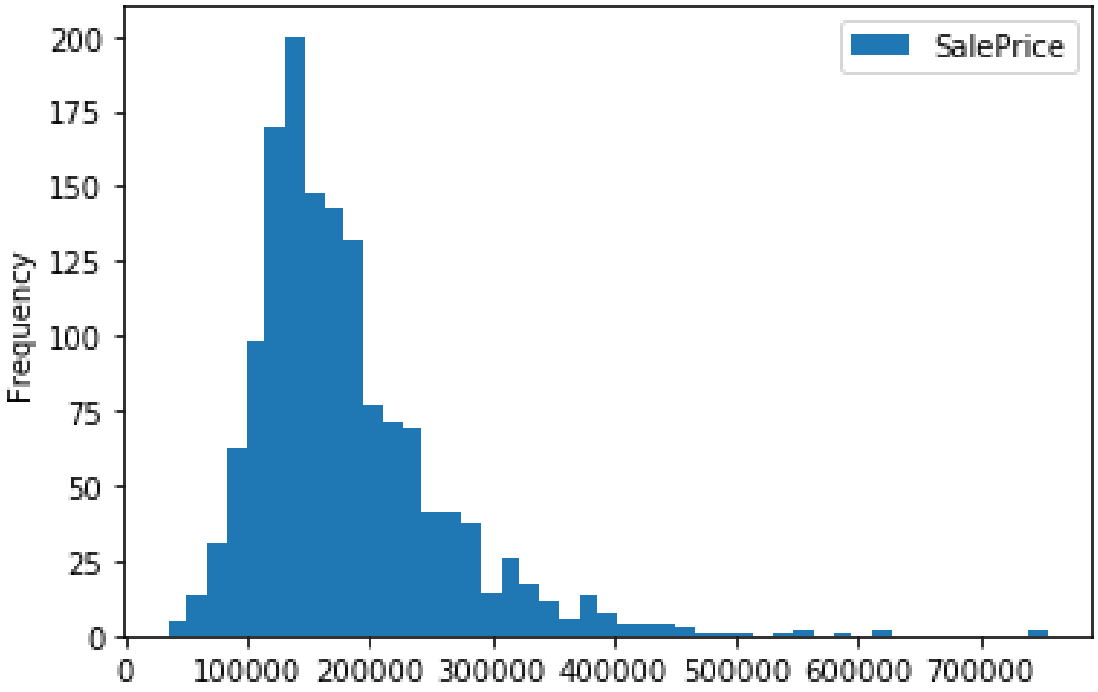

Figure 4: SalePrice Distribution

As we can see from the picture, the 'SalePrice' feature doesn't follow a normal distribution. In fact most of the data lie on the left side.

Feature Selection

Now it is the time to try to understand which features could be taken into account to predict the dependent variable (SalePrice). Taking a look to the data description file, it has been possible to come up with these ten features:

- **OverallQual:** it represents the overall quality material used to build up the houses. It is intuitive that an house that has been built up by using excellent materials (like parquet for the pave instead of the carpet) should be more expensive than one with low quality material.
- **YearBuilt:** shows the original construction date. New houses should be more expensive than old ones (of course it is not always like that).
- **YearRemodAdd:** when the advertisement has been removed from the market. This could be very important because, due to the financial crack related to the subprime loans of 2007, the houses´ prices drastically went down after this year. For that reason houses that have been sold before 2007 should have an higher price, taking fixed the other conditions.
- **TotalBsmtSF:** total basement surface. An house with a basement should be more expensive that a one that do not have one.
- **1stFlrSF:** an house that have a second floor should be more expensive than one that is not equipped with it.
- **GrLivArea:** houses with a wide garage area should be more expensive than houses with lower or even without garage area.
- **FullBath:** the feature represents the overall bathroom quality. Higher is the quality, higher should be the impact on the price.

- **TotRmsAbvGrd:** houses with more rooms should be bigger than houses with less rooms. And a bigger house should usually be more expensive.
- **GarageCars:** how many cars the garage can host.
- **GarageArea:** garage surface. Probably these last two features could be related with each other meaning that there is the presence of multicollinearity.

Pearson Correlation Coefficient

In order to understand if these selected features are useful to predict the houses prices, the Pearson correlation coefficient is returned. The features are considered significantly relevant if they have a Pearson correlation coefficient greater than 0.6. The correlation coefficient can be expressed by the formula:

$$r_{xy} = \sum_{i=1}^{n} \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$

The table below shows the correlations between selected features and target feature.

Table 1: Feature Correlation Coefficients Results

feature name	**correlation coefficent**	R^2
OverallQual	0.790982	0.6256
GrLivArea	0.708624	0.5021
GarageCars	0.640409	0.4101
GarageArea	0.623431	0.3887
TotalBsmtSF	0.613581	0.3765
1stFlrSF	0.605852	0.3671
FullBath	0.560664	0.3143
TotRmsAbvGrd	0.533723	0.2849
YearBuilt	0.522897	0.2734
YearRemodAdd	0.507101	0.2572

At the end, 6 features have been chosen as the the most relevant to predict the house prices:

Table 2: Feature Correlation Coefficients Results

feature name	**correlation coefficent**	R^2
OverallQual	0.790982	0.6256
GrLivArea	0.708624	0.5021
GarageCars	0.640409	0.4101
GarageArea	0.623431	0.3887
TotalBsmtSF	0.613581	0.3765
1stFlrSF	0.605852	0.3671

Standardizing and splitting the data

Before running machine learning models, it is necessary to standardize and split the data into training and test set. It has been decided to use 70% of the data has training set and 30% of the data as test set.

Machine Learning Models

Linear Regression an Decision Tree are the two selected models used to perform the regression analysis. In particular, for Decision Tree a MaxDepth (how much in the depth the tree as to be built), minInstancesPerNode (minimum number of samples that have to be in every child node in order to make the split) and maxBins(max number of bins used for splitting features) has been personalized.

Results & Conclusions

The RMSE (Root Mean Square Error) has been chosen as yardstick because add normalization to the SSE (Sum of Square Error) and moreover return the value in the same scale of the predictions because it is performed by computing the square root of the MSE (Mean Square Error).

As it is possible seeing from the table below, the Decision Tree Regressor is able to generalize better than the Linear Regression. For that reason the Decision Tree Regressor has been chosen as the predictive model.

Table 3: Model Comparison

model	**RMSE**
Linear Regression	39511.1
Decision Regressor Tree	35822.4