Final Assignment supervised learning regression – Report

Main Objective

A **real estate** company in **Ames**, Iowa, has asked us to help them **study the home sales market**. Its objective is to find new homes to add to its portfolio, having these new houses characteristics that allow them to be sold at a high price and, thus, to increase the profits of the company. Despite having some experience in the sector and having a database with previous sales, they still do not know about what characteristics are really those that influence the price the most. Our **main objective** is to try to **understand** what these characteristics are, then to obtain **an interpretation of the price of a house by certain characteristics**.

To do this we are going to **look for the relationship between the price of the house and the rest of the characteristics using linear regression**, once the model is obtained, we can study which features have more weight over the price. The resulting model could also be used to predict prices for new observations.

Brief Description of data

The data set with which we are going to work is compound of **1379 rows and 80 columns**. Each **row** represents a different observation, in our case a **different sold house**. The **target** variable is the "**SalePrice**" which is the price the house was sold for. The other **79 columns are different characteristic of a house** such as '1stFlrSF' which is the first-floor surface (squared feet) or 'BedroomAbvGr' which is the number of bedrooms in the house, both are example of numeric features but there are also categorical ones like the 'PoolQC' which speak about if there are a pool or how is this pool or the 'Utilities' which are the different type of utilities available. Having a look on our data set we can find **37 numeric column and 43 categorical ones**. So, for sure we will need to **encoding these categorical features** before training our model. We also did not find **any null value** on our data, so we **don't need to deal with missing values**. Also, it will be a good idea to study the distribution of the features to make some transformation on them if needed.

Acknowledgments The data set was compiled by Dean De Cock for use in data science education. The Ames Iowa data set is a good alternative to the Boston data set for data scientist to learn hands-on skills on machine learning algorithms.

EDA Summary. Cleaning and feature engineering

As we already describe there are 1379 rows and 80 columns, of which the numeric column 'SalePrice' is the target variable. There are 37 numeric columns and 43 categorical ones. We also said that there are not null values on the data set so we are not dealing with missing values.

Having a look into the **numeric features** we would like to have a look into the feature **distribution**, studying how **skew** is to understand if there are any feature that need a logarithmic transformation or not. Furthermore, we are interested to find out if there is any **correlation** between the features to eliminate any possible duplicity that can lead to wrong predictions. Also having a look to identify **outliers** could be a good idea. We could be also interested in studying if there are any **interaction between these features** to add new ones, but it is something that we will dealing with during the model selection. So, for now we just create all the possible interactions with a polynomial transformation.

Regarding to the **categorical features** we need to **encode** them in order to be used by the models. So, we will apply a proper transformation for each one.

Despite that for a linear regression model the **standardization** does not make a difference we will apply it. The reason is that we are interested on the interpretability of the model so we will use also a **Lasso Regression** that deal with model complexity by taken to zero the coefficient of the features that have almost no influence on the price.

During the Exploratory data analysis, we find:

1) That there are **many columns with a high skew**, beside them the target value. We don't need to perform a transformation on the target variable so we leave it as is. We see that the **most of these features have a skew because most of the values are located at 0.0 which have an actual meaning**. For example, if the 'PoolArea' is equal to zero means that the house did not actually have a pool (as we can see in **figure 1**) **So we did not apply the logarithmic transformation on them**, and the rest of the values cannot be considered as outliers because take values on a range that could happen. Some of these columns clearly **improve after transformation** such a column is GrLivArea (see **figure 2**). Also, there are some columns that despite they have many 0.0 values the transformation makes a difference on the rest of the values, for example **see figure 3** for column 'WoodDeckSF'. We **drop columns that almost has all the rows in one single value,** such as MiscVal which has 1330 over 1379 so it is **not giving significant information**. Other features like this are PoolArea(1372/1379) , LowQualFinSF(1360/1379) and 3SsnPorch (1355/1379)
2) Regarding the **outliers**, the vast majority of the columns do not present anyone. Maybe only for column MSSubClass we could find values that can be considered as outliers, with 88 rows with values far from the mean. So, we can drop these rows in order to improve our data without losing too many data.

3) Plotting the **correlation** matrix, we can spot possible highly correlated features (**figure 4**) Narrow down the column of the data frame keeping only the more correlated one we can perform a **pariplot** where we can **visualise better** these correlations. (**figure 5**). Finally, we found a strong correlation between YearBuilt and GarageYrBlt, so we **keep YearBuilt** being more representative. Also, there is a strong correlation between log_GarageArea and GarageCars, here **we keep log_GarageArea**. Also, there is a correlation between log_LotArea and log_LotFrontage but here we decide to **keep both to have a better idea of the shape of the lot**. Finally, MSSubClass seems to be correlated with log_LotArea and log_LotFrontage since we decide to keep the other, we will **dop MSSubClass**. So, since we are dropping MSSubClass after all we will not need to drop the rows with outliers on this column.
4) After **one hot encoding of categorical features and applying a polynomial transformation of degree 2 to the numeric features** the total number of **columns is 288**. Not all of these features are important but many of them will be dropped during the model selection.

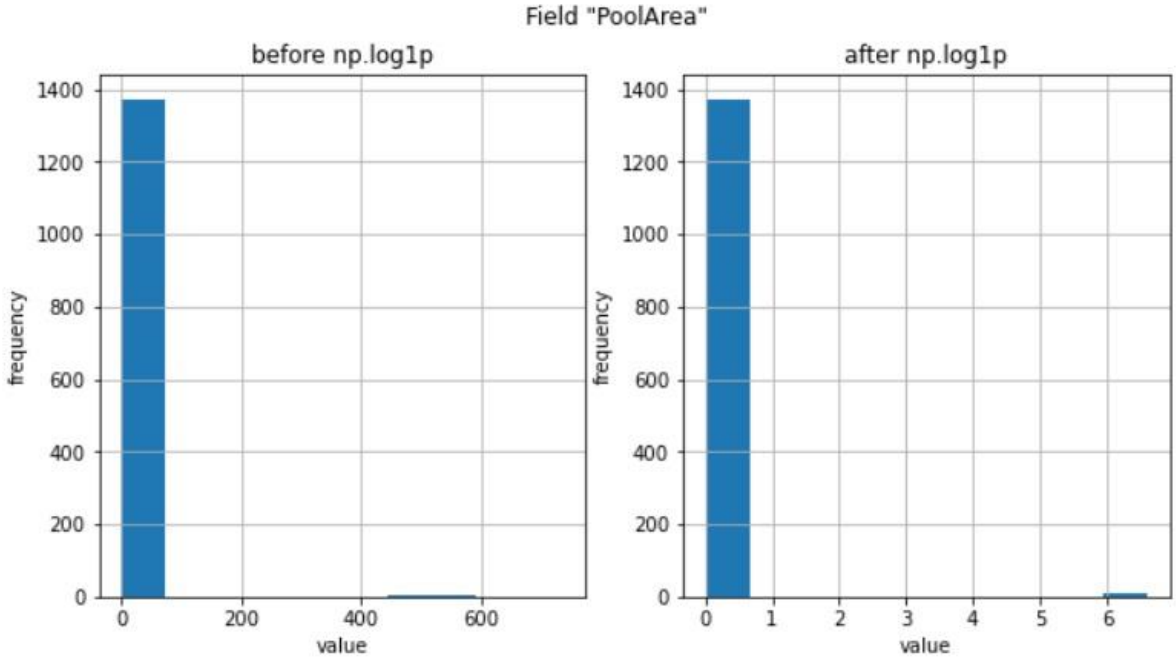

Figure 1. PoolArea before and after logarithmic transformation.

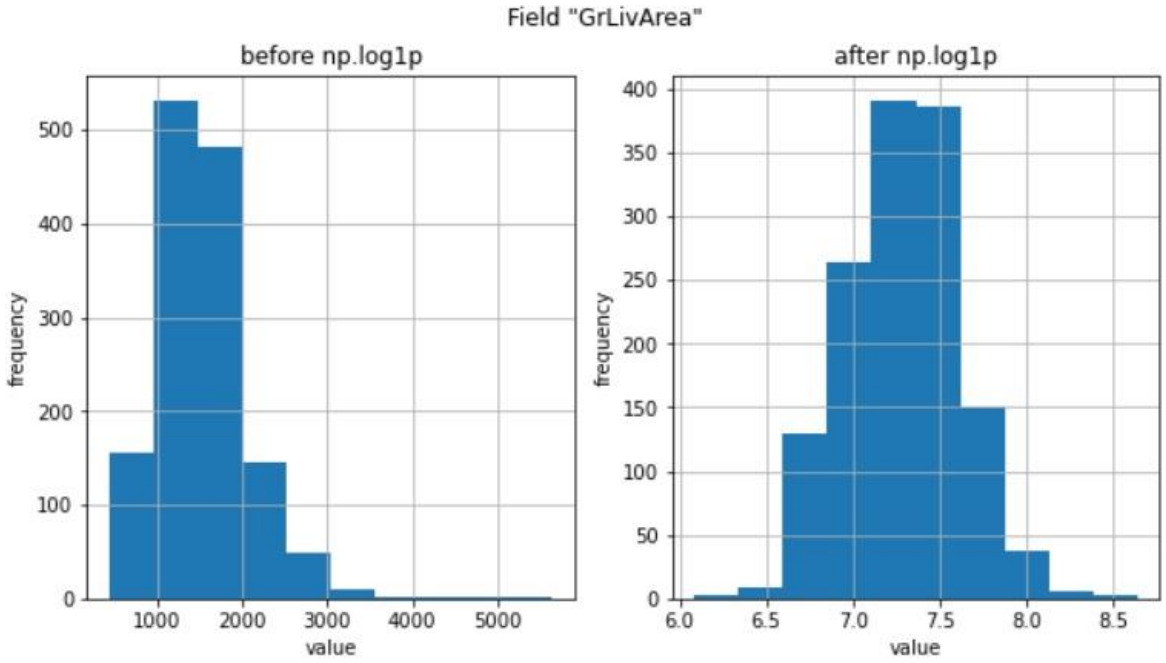

Figure 2. GrLivArea before and after logarithmic transformation.

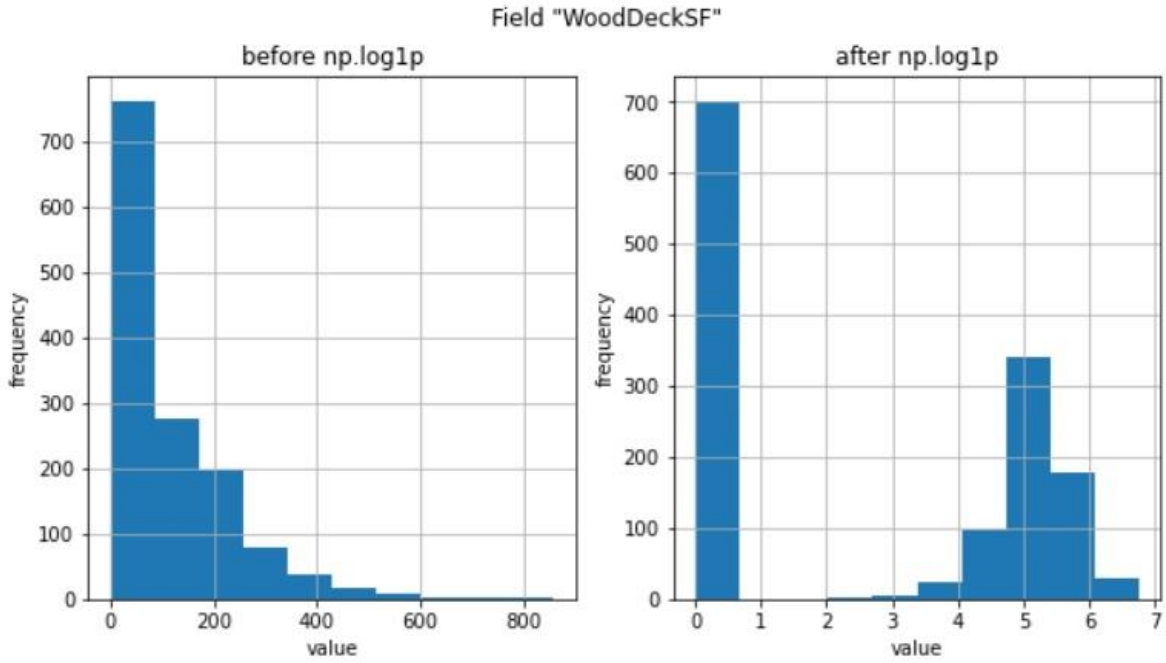

Figure 3. WoodDeckSF before and after logarithmic transformation.

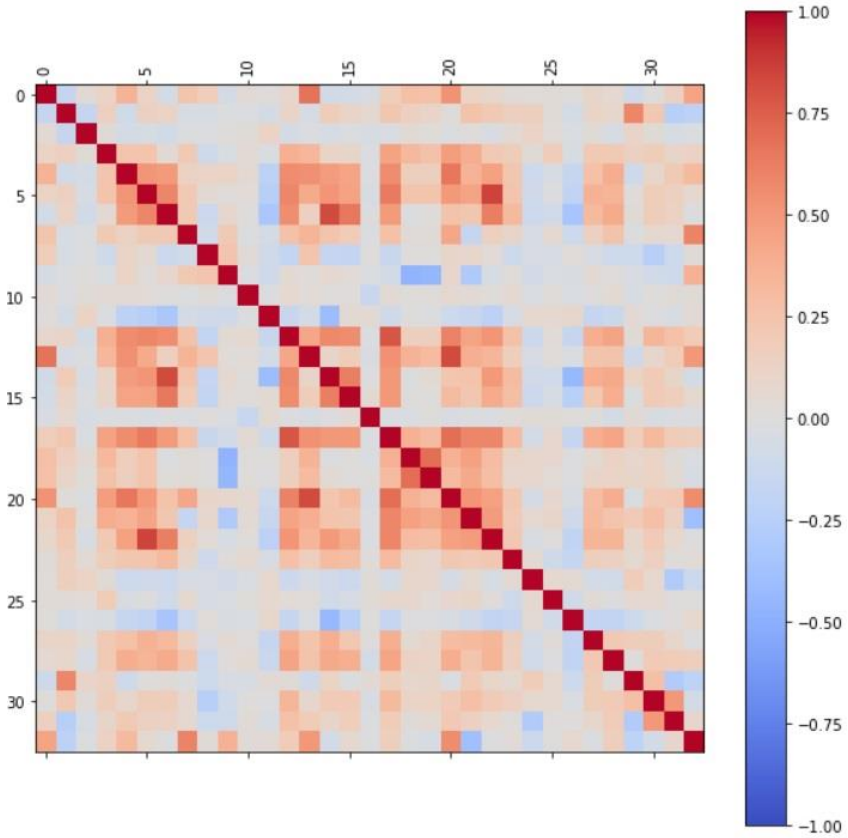

Figure 4. heatmap matrix correlation coefficients.

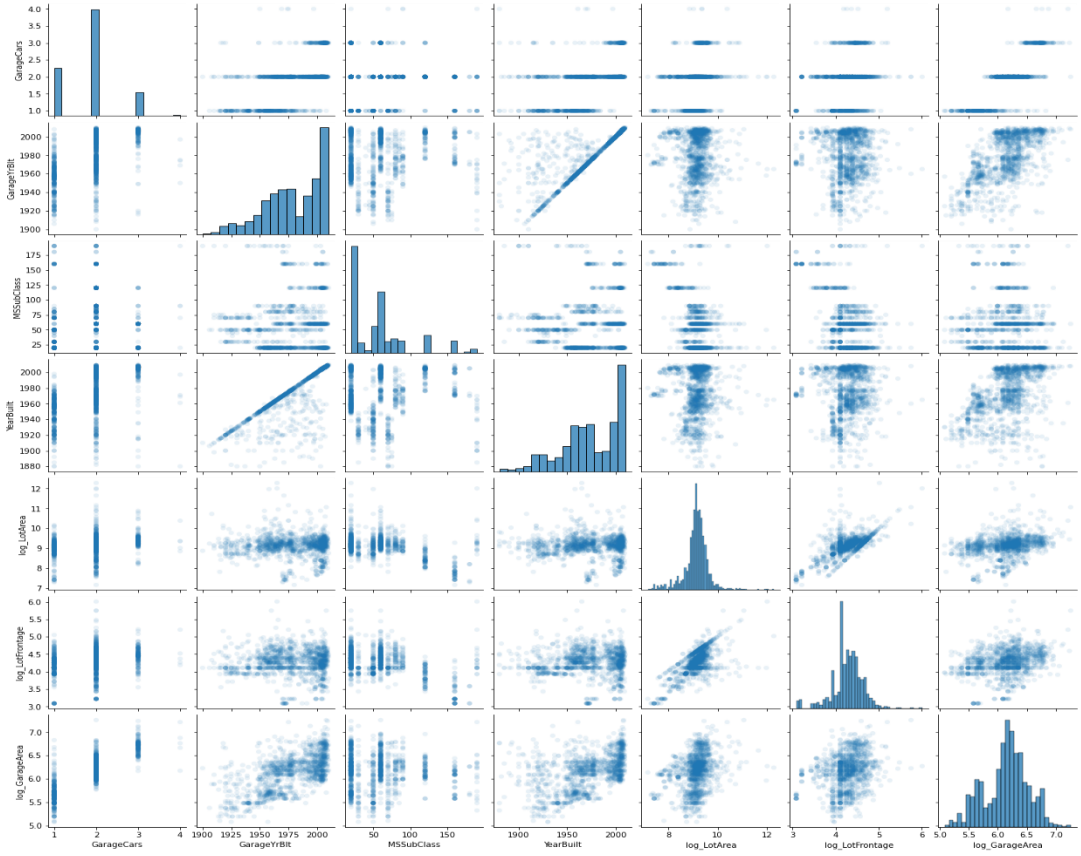

Figure 5. Pairplot with more correlated features.

Summary of Training the model

To work on the **different models**, we will use the same data set, which was **split into a train and a test sets** being the test set the 30% of the total. Having these two data sets we are able to measure de accuracy of each model.

We are going to **fit** our data **to a vanilla linear regression, to a ridge regression and to a lasso regression** to see which of them perform better. For **Ridge** and **Lasso** we will also need to find out which is the **optimal hyperparameter alpha** through **cross validation** method, the larger is the parameter the lower complexity the model.

To **compare** the performance of the models we calculate the **root mean squared error** (rmse) for each one. Being the best performance for the lowest value.

Statistics for each model:

1) **Linear Regression**: After training the model we calculate a **rmse = 51575057603295.92**. Any feature has been dropped, so we have 288 features.
2) **Ridge Regression**: After cross validation we found that the best value for the **hyperparameter alpha is 100**. For this value, we calculate a **rmse=31892.22**. Any coefficient was set to zero so we still have 288 features.
3) **Lasso Regression**: After cross-validation we found that the best value for the **hyperparameter alpha is 1000**. For this value, the model was fit and the prediction shows a **rmse = 32006.520** and the **non-zero coefficients = 77**.

As we can see the model which **perform better is the Ridge regression** with the lowest rmse, while the **linear regression has the worst performance** with a high rmse. We can visualize how good each model is by plotting the actual values versus the predicted ones as we can see in the figure below (**figure 6**)

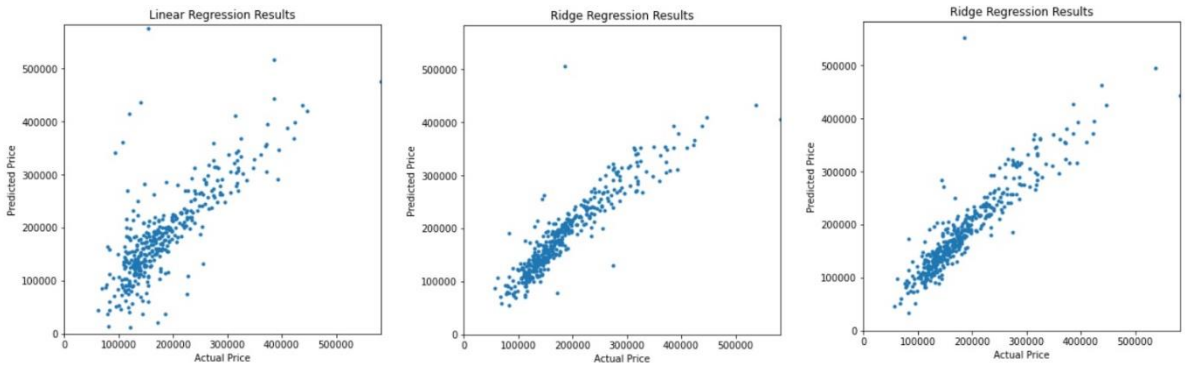

Figure 6. Predicted prices vs Actual prices for the three models.

Since we are interested on the interpretation of the model, we **recommend the Lasso regression**, which has the lower number of features and fits better our needs to achieve our main goal. **The lasso regression lowers the coefficients to zero when they have almost no**

contribution to the sale prices. Thus, we can identify the main characteristics of the model. For an alpha parameter equals to 1000 the complexity has been highly reduced to 77 features. This is still a large number but we can select the top ten features that contribute more to the price. (**table 1**)

feature	coefficient magnitude
log_LotFrontage	13843.054277
OverallCond	10572.951724
TotRmsAbvGrd	10549.705403
log_GrLivArea	10027.137923
YrSold	9683.186254
KitchenAbvGr log_OpenPorchSF	9391.420271
BedroomAbvGr log_LotArea	8771.543726
OverallCond YearBuilt	8502.427637
MoSold	7606.926308
log_OpenPorchSF	7203.692319

Table 1. Top ten features by contribution to the prices.

Key findings

After training the model we found that the features that contribute more to the price are the ones on table 1. With the **most important being the 'log_LotFrontage'** that is related to the size of the frontage lot. Also, from this table we can infer that the **total rooms** ('TotRmsAbvGrd'), the **habitable area** ('log_GrLivArea') or **interactions** between **kitchen area** ('KitchenAbvGr) **and open porch area** ('log_OpenPorchSF') are also relevant. So, we can improve our portfolio adding houses with these characteristics.

Next steps

The next step is **set the business according with the finding of the model** and **observe** what happen for a **period of time**. **Then** we can **measure** if these **actions have an impact in the company profit**. **If so**, we can say that **we have achieve our goal**. **If not means that we need to review our model**. In the former case, maybe we need **gather other kind of features or drop the ones that not make a sense after gaining knowledge** on the field. Then, we could train again the model and **repeat the process until we achieve our goal**.