Supervised Machine Learning: Regression – Project

IBM Machine Learning Professional Certificate

By: Bernardita Štitić

Date: November 26, 2021

TABLE OF CONTENTS

1. OBJECTIVE

In this project, the developed models will be focused on prediction so this will be the main objective of the analysis.

2. BRIEF DESCRIPTION OF THE DATA SET AND ITS ATTRIBUTES

The chosen data set for this project corresponds to a preprocessed version of the Ames housing data set (De Cock, 2011), which describes the sale of residential property in Ames, Iowa, from 2006 to 2010. This preprocessed version corresponds to the one produced in the project for the previous course in the program of the IBM Machine Learning Professional Certificate, specifically "Exploratory Data Analysis for Machine Learning" (EDA). Table 2.1 summarizes the information obtained using the `.info()` Pandas method on a Pandas DataFrame which contains the preprocessed data set.

Row index	Column name	Non-null values	Column data type
0	Fireplaces	2925	Float64
1	SalePrice	2925	Float64
2	Gr Liv Area	2925	Float64
3	Lot Area	2925	Float64
4	Garage Cars	2925	Float64
5	Total Bsmt SF	2925	Float64

Table 2.1. Relevant information of a Pandas DataFrame with the preprocessed version of the Ames housing data set.

As Table 2.1 implies, this data set consists of 2925 observations with 6 variables. In particular, there are 5 explanatory variables or features, which are related to the variables called Fireplaces, Gr Liv Area, Lot Area, Garage Cars and Total Bsmt SF of the original data set by De Cock (2011). The target is related to the original target called SalePrice, which corresponds to the sale price of the property. Moreover, a list with a brief description of each variable of the original data set was developed by Kuhn et al. (2020) and is available in the references section of this report.

Therefore, considering Table 2.1, it is possible to describe the following about the variables of the original data set whose names are shown in the table:

- Fireplaces: total number of fireplaces (Kuhn et al., 2020).
- Gr Liv Area: above grade living area in square feet (Kuhn et al., 2020);
- Lot Area: lot area in square feet (Kuhn et al., 2020);
- Garage cars: capacity of the garage in terms of cars (Kuhn et al., 2020);
- Total Bsmt SF: square feet of the basement area (Kuhn et al., 2020);

Furthermore, as implied by the previous list, Fireplaces and Garage cars are numerical, discrete variables. On the other hand, the remaining features of the list are numerical as well but continuous like the original target (SalePrice). It should be noted that originally the features Fireplaces, Gr Liv Area and Lot Area were int64 variables as shown in Appendix A. However, after certain data cleaning and feature engineering steps were performed in the EDA project these features were changed to float64 variables.

Moreover, the reason why the 6 columns names in Table 2.1 have been referred to as being related to the names of the original variables is because a logarithmic transformation was applied to 4 of the variables in Table 2.1 (3 features and the target). Therefore, technically, not all the variables of the preprocessed data set match the variables listed earlier exactly. This topic, as well as the data cleaning and feature engineering steps, will be addressed next.

3. DATA EXPLORATION, DATA CLEANING AND FEATURE ENGINEERING

This section is related to the steps taken when developing the project for the EDA course with the original housing data set by De Cock (2011). After these steps, the preprocessed data set which was used in this project was obtained.

3.1. Data exploration (EDA project)

In particular, for the data exploration phase, the following actions were taken with respect to the original data set, which consisted of 2930 observations and 82 variables:

1. Removal of outliers: as suggested by the data set author some outliers were initially removed. Specifically, observations whose value of the feature Gr Liv Area was larger

than 4000 were dropped. In consequence, a Pandas DataFrame with 2925 rows and 82 columns was obtained.

2. Feature selection: a subset of 5 features was chosen based on a selection made for a multiple regression model by the data set author (De Cock, 2011). The selected features were Fireplaces, Gr Liv Area, Lot Area, Garage cars and Total Bsmt SF.
3. Pair plot: a pair plot using Seaborn was made to visualize the distributions of the 5 numerical features and the target (histograms) as well as the dependencies between all variable pairs (scatter plots).
4. Box plot: a box plot of the features and the target was created using the .boxplot() Pandas method to quickly visualize any outliers.
5. Relevant statistics: using the Pandas .describe() method, relevant statistics of the chosen feature subset and the target were obtained, specifically the number of non-null values, the sample mean, the sample standard deviation, the minimum value, relevant percentiles (25^{th}, 50^{th} and 75^{th}) and the maximum value.

3.2. Data cleaning and feature engineering (EDA project)

After performing the activities described previously, the following steps were taken for the data cleaning and feature engineering phase:

1. Missing values: only the features Total Bsmt SF and Garage Cars had missing values, specifically one each. For each feature, the missing value was replaced with a 0, which was the minimum value for both Total Bsmt SF and Garage Cars. Implicitly, by doing so it was assumed that if the information was missing then most likely those properties did not have a basement or garage.
2. Skewness inspection: for this step, the .skew() Pandas method was employed like in one of the laboratories of the EDA course. This method returns an unbiased skew value, which is normalized by N-1 (https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.skew.html).
3. Logarithmic transformation: a logarithmic transformation was applied to all variables. In particular, like in the EDA course laboratories, the np.log1p transformation by NumPy was applied to each case (https://numpy.org/doc/stable/reference/generated/numpy.log1p.html). Also, similarly to the EDA course, histograms and bar plots were created to visually appreciate how the distribution of the 5 features and the target were affected by the transformation.

4. Selection of logarithmic features: next, the skewness values were calculated again using the .skew() Pandas method once more. It was observed that the np.log1p transformation was beneficial except for the features Total Bsmt SF and Garage Cars. In the case of these two, the absolute value of the skewness increased. This was evidenced by the new skewness results and the behavior illustrated in the histograms and bar plots. Therefore, in the end, the transformation was applied only to the target and the other 3 features. The final skewness values of the preprocessed data set for this project are the ones highlighted in Figure 3.2.1.
5. New pair plot: a new pair plot using Seaborn was then made after applying the np.log1p transformation to the features Lot Area, Gr Liv Area, Fireplaces and the target SalePrice while leaving the features Total Bsmt SF and Garage Cars in their original form. This pair plot is shown in Figure 3.2.2.
6. New data exploration phase: in the same way as described earlier, box plots and relevant statistics were obtained for the transformed data set. The box plots are shown in Figure 3.2.3, Figure 3.2.4 and Figure 3.2.5 while the relevant statistics are summarized in Figure 3.2.6.

	Skew
Total Bsmt SF	0.395191
Fireplaces	0.235802
SalePrice	-0.041927
Gr Liv Area	-0.059228
Garage Cars	-0.221062
Lot Area	-0.534108

Figure 3.2.1. Skewness of the variables of the preprocessed data set.

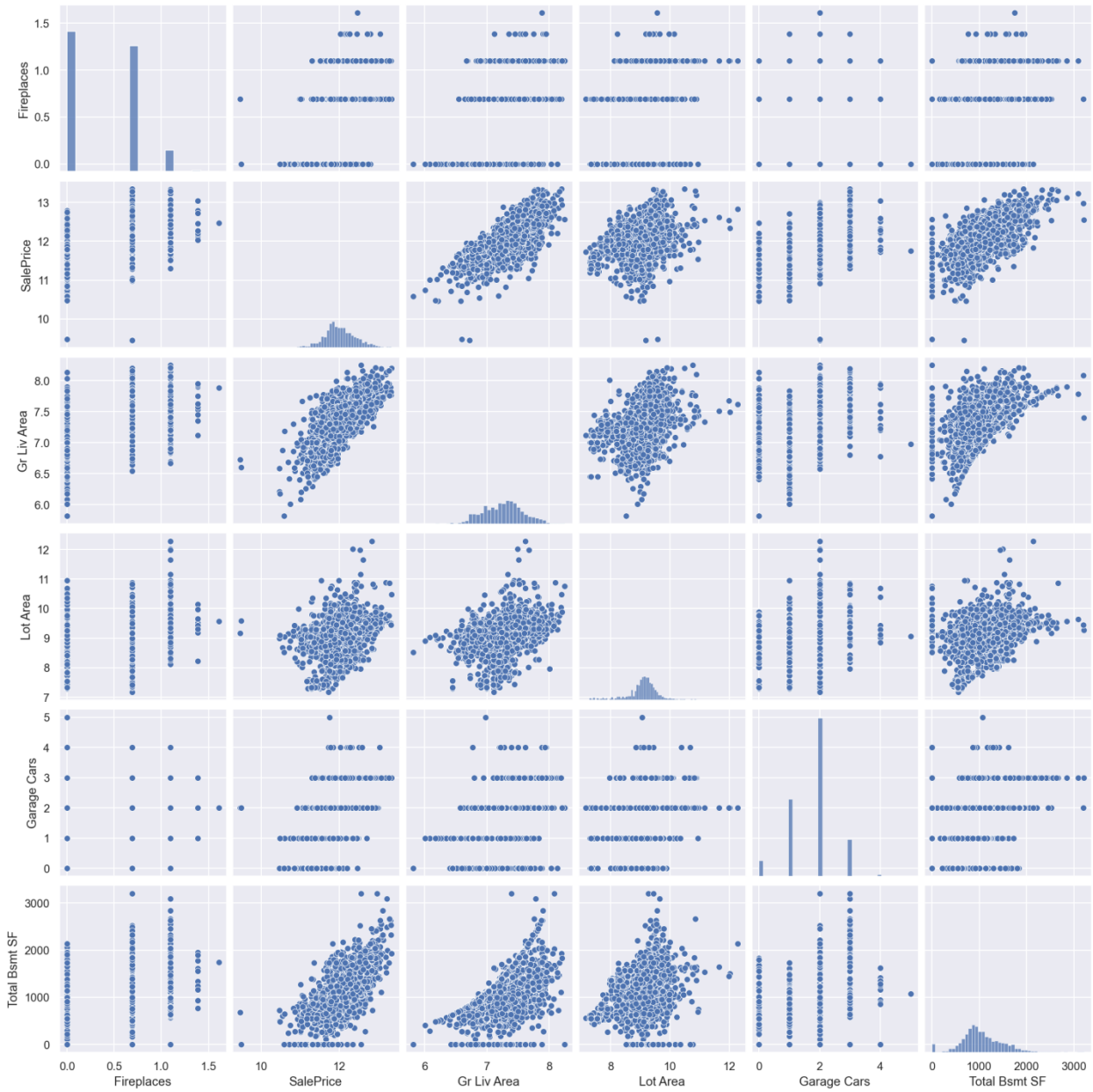

Figure 3.2.2. Pair plot of the preprocessed data set.

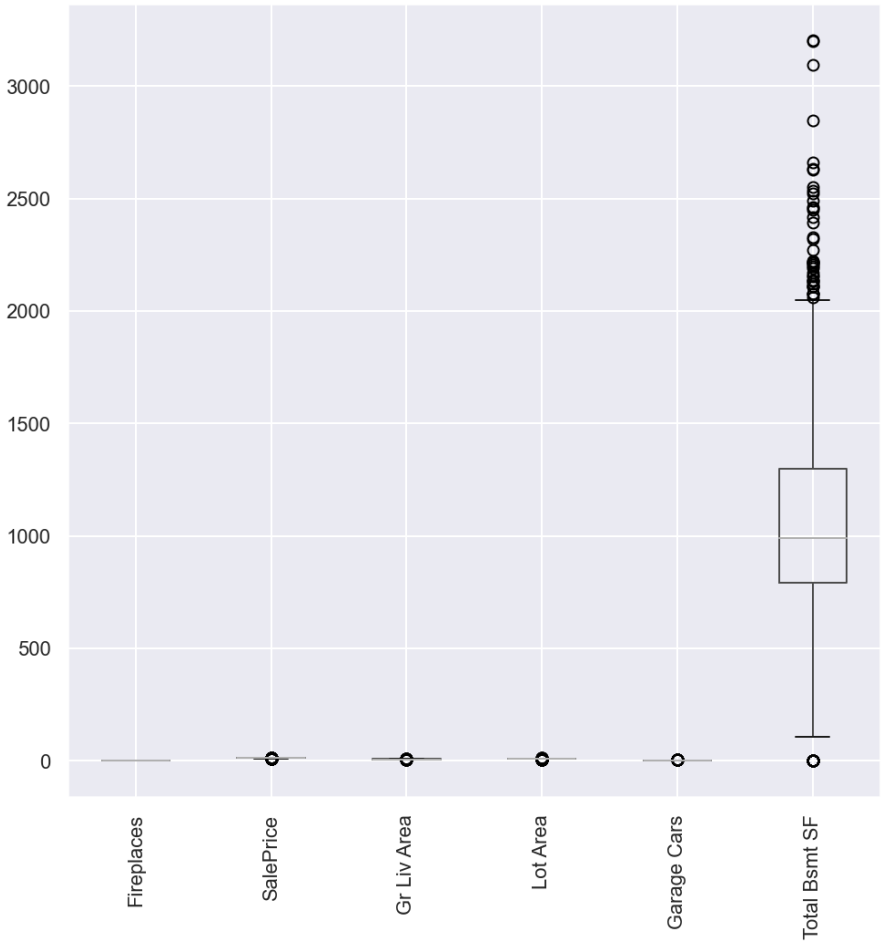

Figure 3.2.3. Box plot of the preprocessed data set.

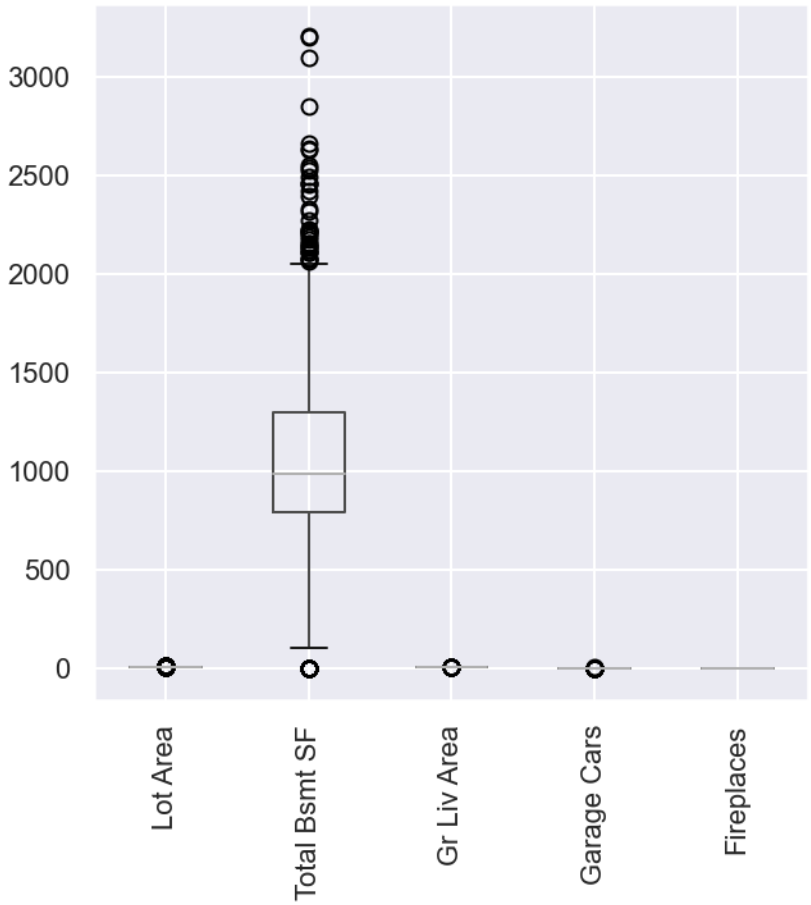

Figure 3.2.4. Box plot of the preprocessed data set (features only).

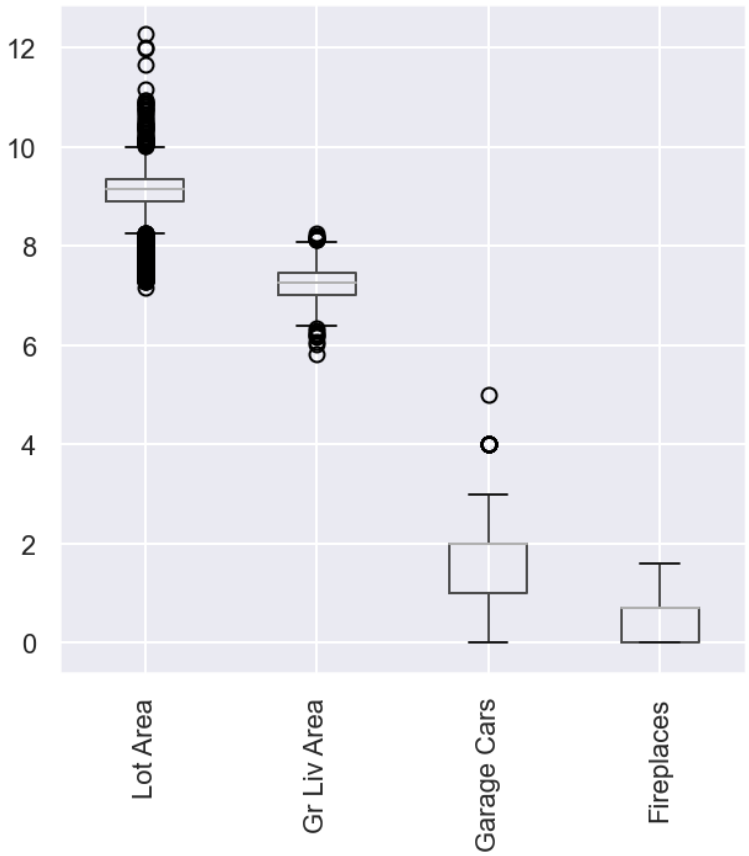

Figure 3.2.5. Box plot of the preprocessed data set (feature subset).

	count	mean	std	min	25%	50%	75%	max
Fireplaces	2925.0	0.389310	0.394534	0.000000	0.000000	0.693147	0.693147	1.609438
SalePrice	2925.0	12.019887	0.406013	9.456419	11.771444	11.982935	12.271397	13.345509
Gr Liv Area	2925.0	7.258784	0.320753	5.814131	7.027315	7.273786	7.462215	8.248267
Lot Area	2925.0	9.090148	0.508309	7.170888	8.914492	9.151545	9.351493	12.279537
Garage Cars	2925.0	1.764444	0.760405	0.000000	1.000000	2.000000	2.000000	5.000000
Total Bsmt SF	2925.0	1046.494359	421.482215	0.000000	792.000000	989.000000	1299.000000	3206.000000

Figure 3.2.6. Relevant statistics of the variables of the preprocessed data set.

3.3. Some of the suggested next steps which are applicable to this project

At the end of the report for the EDA project, the following steps were suggested for the preprocessed data set which was used in this project on regression:

- Based on the new pair plot, the dependency between the logarithmic transformation of SalePrice and Gr Liv Area (transformed as well) could benefit from a quadratic (polynomial) transformation. In consequence, polynomial features could be added. The scatter plot is illustrated in Figure 3.3.1.
- Based on the new pair plot, the relationship between the logarithmic transformation of SalePrice and Lot Area (also transformed) might benefit from a cubic transformation. The scatter plot shows an increase in prices, then a decrease in prices and finally another increase as the value of the independent variable grows. Therefore, like before, polynomial features could be incorporated into the transformed data set. The scatter plot is shown in Figure 3.3.2.
- Perform feature scaling depending on what Machine Learning algorithm is chosen due to the scale of the obtained values (Figure 3.2.6). An option is using standardization with StandardScaler() by Scikit-learn if Lasso or Ridge regression are chosen.

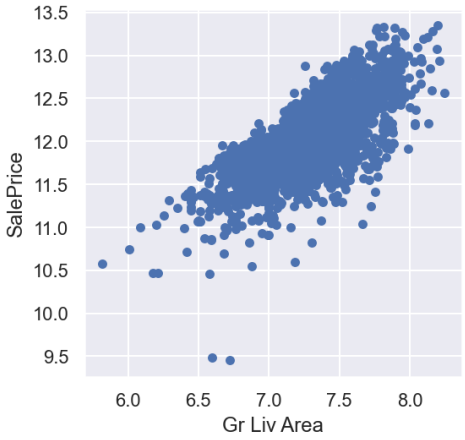

Figure 3.3.1. Scatter plot of the target (SalePrice) and the feature Gr Liv Area after applying np.log1p.

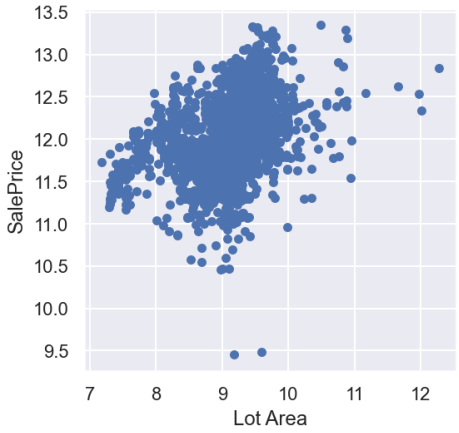

Figure 3.3.2. Scatter plot of the target (SalePrice) and the feature Lot Area after applying np.log1p.

4. LINEAR REGRESSION MODELS

In this section, the developed models will be addressed. In particular, like indicated in the project instructions, a simple linear regression model was first developed to serve as a baseline. Then, a linear regression model with polynomial features was developed. Finally, a Lasso and Ridge model were developed and also optimized for using GridSearchCV() by Scikit-Learn to search for a suitable value of alpha.

4.1. Simple linear regression (baseline)

For the simple linear regression model, the preprocessed data set from Section 2 was split using train_test_split() by Scikit-Learn with test_size=0.3 and random_state=0. This resulted in a training set of 2047 observations (5 features each) and a test set of 878 observations (5 features each as well). Then, an object was instantiated using LinearRegression() by Scikit-Learn and fitted to the training data. Later, the model was evaluated using the test set. Specifically, R^2 and MSE (Mean Square Error) values were calculated for both the training and test set. These results are summarized in Table 4.1.1 together with the values of Adjusted R^2 and $RMSE$ (Root Mean Square Error).

Type of set	R^2	Adjusted R^2	MSE [\$^{2}]	$RMSE$ [\$]
Training set	0.7434	0.7428	0.0426	0.2064
Test set	0.7588	0.7574	0.0390	0.1975

Table 4.1.1. Results of the simple linear regression model (baseline).

Furthermore, the values of the resulting coefficients as well as the intercept were obtained checking the .coef_ and .intercept_ attributes respectively (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html). The results were:

- Coefficients: 1.1195e-01, 4.9364e-01, 1.5376e-02, 1.5614e-01, 3.1177e-04.
- Intercept: 7.6522.

4.2. Linear regression with polynomial features

Next, a linear regression model was developed using polynomial features. In particular, polynomial features related to the logarithmically transformed Gr Liv Area and Lot Area were added to the original preprocessed data set with 2925 observations. Specifically, with respect to Gr Liv Area, polynomial features were generated up to a degree of 2, while for Lot Area the same was done but up to a degree of 3. Furthermore, no interaction terms were generated. This was done based on the suggestions highlighted in Section 3.3. Also, it should be noted that all polynomial features were obtained using PolynomialFeatures() by Scikit-Learn.

After this was performed, the data was split using train_test_split() by Scikit-Learn with test_size=0.3 and random_state=0 once more. The previous argument values ensured that the same partition from earlier was used when splitting the data into a training and test set. In this case, the resulting training set consisted of 2047 observations (8 features each) while the test set had 878 observations (8 features each as well). An object was then instantiated using LinearRegression() by Scikit-Learn and fitted to the training data. The same metrics that were calculated for the baseline model (Section 4.1) were then obtained for this case. These results are summarized in Table 4.2.1.

Type of set	R^2	Adjusted R^2	MSE [$\$^2$]	$RMSE$ [$]
Training set	0.7436	0.7426	0.0426	0.2064
Test set	0.7593	0.7571	0.0389	0.1972

Table 4.2.1. Results of the linear regression model trained with polynomial features.

Similarly to the baseline model, the values of the coefficients and the intercept were obtained with the .coef_ and .intercept_ attributes:

- Coefficients: 1.1245e-01, 1.5628e-01, 3.1200e-04, 9.6280e-02, 2.7567e-02, 1.2423e+00, -1.3078e-01, 4.5980e-03.
- Intercept: 5.2816.

4.3. Lasso regression

For both the Lasso and Ridge regression models, first GridSearchCV() was used to find a good value of alpha. The range of considered values was based on the available ranges in the Regularization and Gradient Descent Jupyter Notebook of the class. In particular, for Lasso regression, a NumPy array was created containing the values: 0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1 and 0.5. Regarding max_iter in the case of Lasso, this was fixed at a value of 10e5 maximum iterations.

Afterward, a Pipeline object using Pipeline() by Scikit-Learn was created. The pipeline consisted of two stages: a scaler object instantiated using StandardScaler() by Scikit-Learn and a Lasso regression object which was instantiated using Lasso() by Scikit-Learn as well. The search was then configured to cross-validate using 3 folds (cv=3) and to calculate the values of R^2 (scoring='r2'). The search was then launched using the training data.

For Lasso, the best value of alpha resulted to be 0.0001 with an average score of 0.7430. Then, with this value, a new Lasso object was instantiated and fitted to the training set, which was standardized using StandardScaler() before training using .fit_transform(). Moreover, the test set was also standardized (using .transform()). The same metrics from Section 4.1 and Section 4.2 were calculated for this model as well. The results are highlighted in Table 4.3.1.

Type of set	R^2	Adjusted R^2	MSE [$\$^2$]	$RMSE$ [$\$$]
Training set	0.7434	0.7424	0.0426	0.2064
Test set	0.7591	0.7569	0.0390	0.1975

Table 4.3.1. Results of the Lasso regression model trained with polynomial and standardized features.

The values of the coefficients and the intercept, which were obtained with the .coef_ and .intercept_ attributes, were:

- Coefficients: 0.0440, 0.1174, 0.1322, 0.0547, 0.1022, 0.0075, -0., -0.
- Intercept: 12.0198.

4.4. Ridge regression

For Ridge regression, the same steps considered to develop the Lasso regression model were performed with two minor changes. Firstly, the range of alpha values that was used for the grid search consisted of: 0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.3, 0.5, 1, 3, 5, 10, 15, 30 and 80. Secondly, the pipeline object had the same first stage from Section 4.3 whereas the second stage consisted of a Ridge regression object instead of a Lasso regression one. The Ridge object was instantiated using Ridge() by Scikit-Learn.

Like for Lasso, the search was launched using the training data. Once the process was over, the best value of alpha for Ridge resulted to be 15 with a score of 0.7432. Similarly to the case described in Section 4.3, a new Ridge object was instantiated and fitted to the training set, which was the same training set used to train the Lasso regression model. The same metrics from Section 4.1, Section 4.2 and Section 4.3 were calculated for the Ridge regression model too. The results are shown in Table 4.4.1.

Type of set	R^2	Adjusted R^2	MSE [$\$^2$]	$RMSE$ [$\$$]
Training set	0.7434	0.7424	0.0426	0.2064
Test set	0.7593	0.7571	0.0389	0.1972

Table 4.4.1. Results of the Ridge regression model trained with polynomial and standardized features.

Like for Lasso, the values of the coefficients and the intercept were obtained using the `.coef_` and `.intercept_` attributes and resulted to be:

- Coefficients: 0.0442, 0.1169, 0.1314, 0.0763, 0.0805, 0.0108, 0.0015, -0.0044.
- Intercept: 12.0198.

5. DISCUSSION AND MODEL RECOMMENDATION

Table 5.1 summarizes the results on the test sets of each model. This is highlighted since the baseline model was tested on the original test set, which consisted of 878 observations with 5 features each. Then, this test set was expanded by adding polynomial features to test the model that was trained using the training set with the new polynomial features, which were described in Section 4.2. Therefore, this second model was evaluated using a test set with the same 878 observations but with 8 features. Finally, the Lasso and Ridge models were evaluated on this test set as well but after standardizing it as explained in Section 4.3.

Type of result	Baseline model	Polynomial features	Lasso regression	Ridge regression
R^2	0.7588	0.7593	0.7591	0.7593
Adjusted R^2	0.7574	0.7571	0.7569	0.7571
MSE [$\$^2$]	0.0390	0.0389	0.0390	0.0389
$RMSE$ [$]	0.1975	0.1972	0.1975	0.1972

Table 5.1. Summary of testing results of the four developed regression models.

As shown in Table 5.1, it is interesting to note that despite the fact that the test sets were different the results were all very similar. Furthermore, the resulting values of R^2 and adjusted R^2 of each model were very close. This is an important result since it implies that, in general, the independent variables of each model actually added value to the regression models. However, when comparing the adjusted R^2 results of the baseline model and the one trained with the additional polynomial features, the adjusted R^2 metric fell marginally from 0.7574 to 0.7571. On the contrary, and as expected, the R^2 results increased from 0.7588

(baseline) to 0.7593 (second model). This behavior implies that, actually, adding polynomial features did not improve the goodness-of-fit of the second linear regression model by much.

Moreover, as seen in Section 4 for each developed model, the training and test set results did not differ by much. This implies that, at least based on the obtained results, there should be no issues with respect to overfitting in general. It is important to note that, when comparing the results of the models, as well as the results in Table 5.1, since technically there were 3 test sets (one for the baseline model, one for the second model and another one for the Lasso and Ridge models) comparisons should be made carefully.

Therefore, considering Table 5.1, the values of MSE and $RMSE$ can help evaluate the goodness-of-fit of each model with respect to a specific test set but not between test sets in a very straightforward way. In consequence, regarding these metrics, the baseline model, the second model and the Lasso and Ridge models should be evaluated as three different cases. Adjusted R^2 can help see whether adding new features to a given data set improve the model fit more than what would be expected by chance (Frost, n.d.), so the comparison between the results of the baseline model and the second model are relevant as discussed earlier.

Overall, between the baseline model and the second model, which was trained with the new polynomial features, the former is recommended over the latter due to the marginal decrease of the adjusted R^2 metric. Concerning the Lasso and Ridge regressions, since the data was standardized comparisons are not so straightforward with respect to the baseline and second models. However, between Lasso and Ridge, the value of $RMSE$ is slightly higher for Lasso. Also, for each of these two models, the results of R^2 and adjusted R^2 are quite close in each case so both models seem like relatively good initial options for the standardized data set with polynomial features.

However, again, since adding new polynomial features did not seem to have a very significant effect, the baseline model is still preferred if the objective is accuracy or prediction like in this project. In terms of explainability, Lasso might be the preferred option since it's a good way of performing feature selection and it performed relatively well as an initial starting point with respect to the standardized data set. Nevertheless, interpretation or explainability was not the objective of this project. In consequence, the recommended option that best suits the objective of this analysis is, again, the baseline model. Suggestions for improving it will be summarized in Section 7.

6. KEY FINDINGS AND INSIGHTS

In summary, after performing this analysis, the key findings and insights are:

- Adding polynomial features to the baseline model did not seem to improve the predictive capability of the initial model. In particular, test results of the adjusted R^2 metric decreased marginally when adding polynomial features from 0.7574 (baseline model) to 0.7571 (regression model trained with the new polynomial features).
    - As indicated, this suggests that, with the new features, the model fit did not improve by much. However, it is not entirely clear which specific polynomial features are the cause behind this marginal decrease. Consequently, it would be necessary to add the features one by one to obtain new results to properly identify this.
    - This might also suggest that, as a group, the initially chosen feature transformations might not be the most effective choice for the data set of this project. It might be interesting to experiment with other transformations for specific features, add interaction terms and/or optimize for a polynomial degree for the group of independent variables as a whole.
- In general, despite the training and test sets being different, it is possible to say that each model does not seem to have overfitting issues with respect to its respective data set. This is because, as seen in Section 4, training and test results did not differ too much in each case.
    - It is relevant to highlight that even though there were 3 different groups of training and test sets, the second and third groups were created from the first one by first adding polynomial features and then standardizing.
- Interestingly, overall results for all models were very similar, both for the training and test sets, as seen in Section 4 and Section 5. It would appear like each model, with respect to its data set, had similar predictive capabilities.
    - Explainability or interpretability could be quite different, as seen in the case of the Lasso regression model, which was the only one with two null coefficients. Moreover, it was the only regression model with null coefficients. However, as described earlier, interpretability was not the objective of this analysis.
        - These two null coefficients would suggest that the features $Lot\ Area^2$ and $Lot\ Area^3$ are not necessary.

- Since there would appear to be an absence of overfitting issues based on the results, it might not be necessary to experiment with regularization unless attempting to perform further feature selection with Lasso (for example).
- A possible absence of overfitting might suggest that, starting from the recommended model, other feature transformations could be attempted to improve predictive capability further. Furthermore, for simple linear regression, feature scaling might not add any predictive power as seen in the course.

Also, considering that the recommended model in Section 5 was the baseline model since it better met the objective of this project, the preprocessed data set used to train it is of particular interest. Regarding this data set, there are some important findings that could help improve prediction results in the future. In particular, some relevant findings and insights are:

- The pair plot obtained after the application of `np.log1p` (Figure 3.2.2) shows dependencies between feature pairs that might be of concern. This should be addressed later.
  - Example: the scatter plot between `Lot Area` and `Gr Live Area` shows that correlation in this case might be important.
- The pair plot also illustrates that, regarding the dependency between `SalePrice` and `Total Bsmt SF`, there is a concentration of points along a vertical line where `Total Bsmt SF=0`. Also, the scatter plot suggests non-linear behavior.
- After applying the logarithmic transformation, the box plots show that there are still data points located past the whiskers.
  - Specifically, there are values beyond the 75^{th} quartile + 1.5 IQR and below the 25^{th} quartile – 1.5 IQR, where IQR is the Interquartile Range.
- Relevant statistics calculated after applying the transformation show that, considering the features, the variable `Total Bsmt SF` has a maximum order of magnitude of 10^3.
  - The other features reach a maximum order of magnitude of 10^2 or 10^1.

7. NEXT STEPS

The (recommended) baseline model should be considered as a starting model, which should be developed further to improve prediction. In particular, this is because, as seen in this analysis, a possible flaw of the model is that the added group of polynomial features might not be the most effective choice. Moreover, the data set possibly requires more processing. Furthermore, another possible flaw is that, for example, linear regression assumptions should be verified in depth using relevant hypothesis testing when required (this will be explained soon at the end of this section). Ultimately, the steps to continue this analysis are related both to the preprocessed data set of this project (Section 2) and the baseline model itself. Therefore, based on the analysis presented so far, some relevant next steps include:

- Increase model complexity gradually by adding the polynomial features of this analysis one by one. As indicated in Section 6, it is not clear which of the considered polynomial features marginally decrease the value of the adjusted R^2 metric.
- Once the previous step has been completed, it would be relevant to consider other feature transformations. The pair plot in Figure 3.2.2 shows evidence of other non-linear dependencies of interest as highlighted in Section 6.
    - Example: the dependency between SalePrice and Total Bsmt SF could benefit from a square root transformation. Furthermore, concerning this particular dependency, there is also the concentration of points along the vertical line where Total Bsmt SF=0 as indicated in Section 6.
        - It could be worth trying to use a dummy variable for this behavior.
- Once the previous steps are performed, it could be worth studying whether interaction terms could improve the predictive power of the model.
    - Considering this, it could be useful to optimize for a polynomial degree using cross-validation to capture (possibly) relevant interaction terms.
- Address the topic of outliers further, which will require considering whether the data points observed in the box plots are true outliers. Moreover, it should then be decided whether it will be truly necessary to eliminate them.
- It is recommended to use a validation set to compare different models and leave a holdout or test set aside for final evaluation.
- Once the previous steps are completed, Lasso could be used to perform further feature selection if required.

    - In this case, it will be necessary to scale the features due to their different scales (as seen in this analysis).
- It is highly advisable to check that the 7 OLS (Ordinary Least Squares) assumptions for linear regression are met (Frost, n.d.).
    - It is important to study the residuals of the final regression model (Frost, n.d.) and test for the statistical significance of its coefficients (for instance, with One Sample T-Student tests).
        - The residuals of the baseline model were explored in this project, as well as the related histogram of the residuals. These are shown in Figure 7.1 and Figure 7.2 respectively. It would appear like the behavior is a good starting point.
        - The average of the residuals was also calculated and resulted to be -1.1108e-16, which is very close to zero.
    - It will be necessary to study the correlation between feature pairs that appear to be strongly correlated using pair plots and correlation coefficients, like possibly the Pearson coefficient. In this case, it will be necessary to make sure the assumptions for Pearson are met to perform hypothesis testing on the coefficients.
        - It is also necessary to study the correlation between the residuals and the independent variables (Frost, n.d.).
    - Relevant hypothesis testing should be performed when required to check for assumptions.

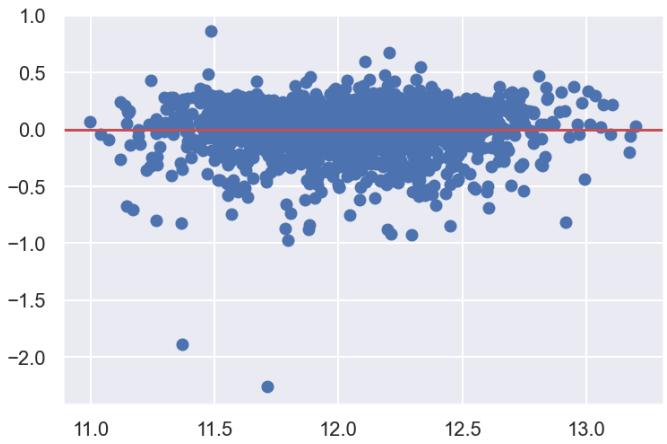

Figure 7.1. Residual plot of the baseline model (training set).

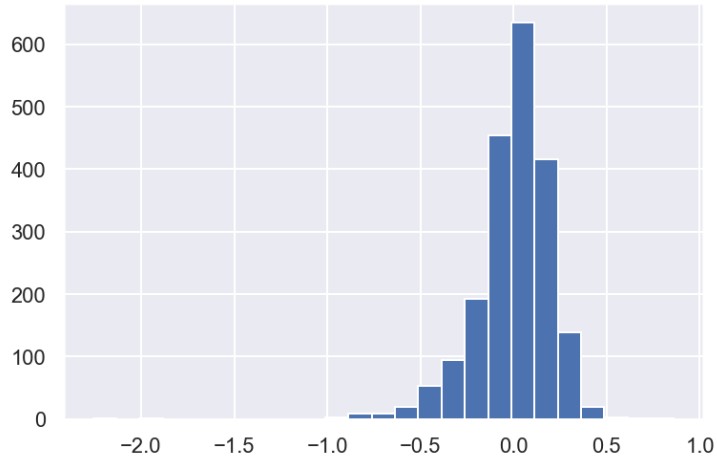

Figure 7.2. Histogram of residuals of the baseline model (training set).

REFERENCES

De Cock, D. (2011). Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project. *Journal of Statistics Education, 19*(3). http://jse.amstat.org/v19n3/decock.pdf

Frost, J. (n.d.). 7 Classical Assumptions of Ordinary Least Squares (OLS) Linear Regression [Blog entry]. Statistics by Jim. https://statisticsbyjim.com/regression/ols-linear-regression-assumptions/

Frost, J. (n.d.). Check Your Residual Plots to Ensure Trustworthy Regression Results! [Blog entry]. https://statisticsbyjim.com/regression/check-residual-plots-regression-analysis/

Frost, J. (n.d.). How to Interpret Adjusted R-Squared and Predicted R-Squared in Regression Analysis [Blog entry]. Statistics by Jim. https://statisticsbyjim.com/regression/interpret-adjusted-r-squared-predicted-r-squared-regression/

Kuhn, M., Perepolkin, D., & RStudio. (2020, June 23). Package 'AmesHousing'. CRAN. https://cran.r-project.org/web/packages/AmesHousing/AmesHousing.pdf

APPENDICES

Appendix A. Column information of the original data set by De Cock (2011)

The variables in the subset used for this project have been highlighted on the list below to find them more quickly. The following information was obtained after calling the .decribe() method on a Pandas DataFrame with the original housing data set (De Cock, 2011).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 82 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
0   Order            2930 non-null   int64
 1   PID              2930 non-null   int64
 2   MS SubClass      2930 non-null   int64
 3   MS Zoning        2930 non-null   object
4   Lot Frontage     2440 non-null   float64
 5   Lot Area         2930 non-null   int64
 6   Street           2930 non-null   object
7   Alley            198 non-null    object
 8   Lot Shape        2930 non-null   object
 9   Land Contour     2930 non-null   object
10  Utilities        2930 non-null   object
11  Lot Config       2930 non-null   object
12  Land Slope       2930 non-null   object
13  Neighborhood     2930 non-null   object
14  Condition 1      2930 non-null   object
15  Condition 2      2930 non-null   object
16  Bldg Type        2930 non-null   object
17  House Style      2930 non-null   object
18  Overall Qual     2930 non-null   int64
19  Overall Cond     2930 non-null   int64
20  Year Built       2930 non-null   int64
21  YearRemod/Add    2930 non-null   int64
22  Roof Style       2930 non-null   object
23  RoofMatl         2930 non-null   object
24  Exterior 1st     2930 non-null   object
25  Exterior 2nd     2930 non-null   object
26  MasVnr Type      2907 non-null   object
27  MasVnr Area      2907 non-null   float64
28  Exter Qual       2930 non-null   object
29  Exter Cond       2930 non-null   object
30  Foundation       2930 non-null   object
31  Bsmt Qual        2850 non-null   object
32  Bsmt Cond        2850 non-null   object
33  Bsmt Exposure    2847 non-null   object
34  BsmtFin Type 1   2850 non-null   object
35  BsmtFin SF 1     2929 non-null   float64
```

```
36  BsmtFin Type 2   2849 non-null   object
37  BsmtFin SF 2     2929 non-null   float64
38  BsmtUnf SF      2929 non-null   float64
39  TotalBsmt SF    2929 non-null   float64
40  Heating           2930 non-null   object
41  Heating QC        2930 non-null   object
42  Central Air       2930 non-null   object
43  Electrical        2929 non-null   object
44  1st Flr SF        2930 non-null   int64
45  2nd Flr SF        2930 non-null   int64
46  Low Qual Fin SF  2930 non-null   int64
47  Gr Liv Area       2930 non-null   int64
48  Bsmt Full Bath   2928 non-null   float64
49  Bsmt Half Bath   2928 non-null   float64
50  Full Bath         2930 non-null   int64
51  Half Bath         2930 non-null   int64
52  BedroomAbvGr    2930 non-null   int64
53  KitchenAbvGr    2930 non-null   int64
54  Kitchen Qual      2930 non-null   object
55  TotRmsAbvGrd    2930 non-null   int64
56  Functional        2930 non-null   object
57  Fireplaces        2930 non-null   int64
58  Fireplace Qu      1508 non-null   object
59  Garage Type       2773 non-null   object
60  GarageYrBlt    2771 non-null   float64
61  Garage Finish     2771 non-null   object
62  Garage Cars       2929 non-null   float64
63  Garage Area       2929 non-null   float64
64  Garage Qual       2771 non-null   object
65  Garage Cond       2771 non-null   object
66  Paved Drive       2930 non-null   object
67  Wood Deck SF      2930 non-null   int64
68  Open Porch SF     2930 non-null   int64
69  Enclosed Porch    2930 non-null   int64
70  3Ssn Porch        2930 non-null   int64
71  Screen Porch      2930 non-null   int64
72  Pool Area         2930 non-null   int64
73  Pool QC           13 non-null     object
74  Fence             572 non-null    object
75  Misc Feature      106 non-null    object
76  Misc Val          2930 non-null   int64
77  Mo Sold           2930 non-null   int64
78  Yr Sold           2930 non-null   int64
79  Sale Type         2930 non-null   object
80  Sale Condition    2930 non-null   object
81  SalePrice         2930 non-null   int64
dtypes: float64(11), int64(28), object(43)
memory usage: 1.8+ MB
```