Regression on Ames Housing Dataset

The project's goal is to build two or models for a prediction task where the target is the Sale_Price feature from the famous Ames Housing Dataset.

Problems - What do we know?

Facing a dataset for the first can be challenging as we lack some knowledge:

- We are focusing on a particular geographical area, which we are not familiar with;
- We are focusing on a market we have no prior domain-knowledge of;

Therefore, we haven't a clue of how the market is going to behave, nor do we know what dictates a change in Sale_Price. While the first can be hardly tracked, we can find out about the latter two through data exploration and domain research. As we know, datasets rarely are perfect, so we must perform some transformations and checks.

Domain Research

As the name of the repository suggests, we are looking at a instances of houses which are located in *Ames, Iowa (USA). From Wikipedia we know that: "*Ames (/eɪmz/) is a city in Story County, Iowa, United States*". But this is not a sufficient insight for our goal, as the description lacks information about the real estate context. This means we might need a more general rule, such as house price assessment in the U.S.A.

After an extensive research I found out the main driving factors for house prices, generally speaking, are:

1. Neighborhood comps
2. Location
3. Home size and usable space
4. Age and condition
5. Upgrades and updates
6. The local market and economic change
7. Mortgage interest rate

The importance given to the first three points, is underlined by the article Cracking the Ames Housing Dataset with Linear Regression where the author states that *"[...] no two houses are exactly identical, and the basic idea of hedonic price modeling is that neighborhood-specific and unit-specific characteristics help determine house prices."*.

This is not far from the truth: most people value the neighborhood, more than the quality of the house itself as it can always be modified later.
However, under this scenario the dataset is not exhaustive. Some aspects such as the local market, and interest rates are hard to guess, although adding features and transformations can reinforce the importance of some features or even include new hidden trends when possible.

Ames Housing Dataset - What we know so far

As the referenced repository lacks of a detailed description, I went on looking for a more detailed version. The dataset presents 2930 instances, representing buildings which were sold in Ames, Iowa between 2006 and 2010. The number of features included are exactly 81, between 35 numerical and 46 non numerical variables. These features include information for each entry, about the structure of the house, its surroundings, access to the road, location and sale conditions etc...

The presence of many object features, brought some issues to the table:

- Representing quality scales, and labels is hard and usually is achieved through heuristics.
- The encoding can get messy for regular categorical variables, the increase in the number of features can lead to overfitting and increased variance in our predictions.
- The encoding of categorical features is strictly related to the model we are trying to build. For example, we cannot feed a one-hot-encoding of all the categorical features to a tree-based model as it would create a sparse decision tree.
- Miscellaneous features and their values might be hard to consider as they count only for some instances, which are usually outliers,
- Longitude and Latitude cannot be used to train models as they lead to overfitting, but cannot be excluded as they are needed for edges creation in case of graph based models.

Nevertheless, there are still plenty of numeric columns, which happen to be some of the most important ones as the domain research suggests. This is indeed a very complete dataset.
In fact, it is presented without any `Nan` values, which is not the case for other versions of the same dataset. This could mean the values were replaced with zeros, or the information was never provided to begin with. This resulted in an ambiguous version of the dataset and it is hard to tell whether some features should be kept a-priori.

Thanks to a notebook for a similar competition , I found out some more interesting information about the nature of the data.
First of all, some of the features are missing, but they are not fundamental for this task. However, the data has been gathered between 2006 and 2010 but there is no information regarding whether one house appears more than once. Even though this is not relevant from a training point of view, as the state of the house dictates the price, it can be a problem for prediction as two instances of the same house (before and after remodeling) can have a very big gap in terms of `Sale_Price`, while their features can be almost always the same.

Data Preprocessing:

During the preprocessing I have performed some changes based mostly of useless features and incoherent rows.

- Removed these features:
    - `Misc_Val`, `Misc_Feature` are not specific, nor they are valid for every entry. The result of a one-hot-encoding would be confusing and misleading
    - `Utilities`, `Functional`, `Condition_1` `Condition_2`, `Exterior_1st`, `Exterior_2nd`, `Foundation`, `MS_SubClass`, `Screen_Porch`, `Three_season_porch`, `Wood_Deck_SF`, `Open_Porch_SF` as there is low evidence, or only few classes have relevant instances, or they do not look relevant at all.
- Added some indicators variables.
- Removed * `BsmtFin_SF_1`, `BsmtFin_SF_2`, `BsmtFin_Type_1`, `BsmtFin_Type_2`, `Bsmt_Err` as they were incoherent for most entries.
- Removed some instances
    - In `MS_Zoning` we find buildings that are not residential and can be considered as confounders.
    - Other that were obviously not coherent
- Added some features: `LowQ_Total_Liv_Ratio`, `External_SF`, `Total_Bsmt_Fin_SF` (removed), `Bsmt_Err` (removed), `Bsmt`, `Total_SF`, `Bsmt_Total_Bath`, `Baths`,
- Checked the coherence of the remaining data, modified/substituted/imputed values when needed
- Explained what features could be considered for Binary, Ordinal encoding, Categorical at a later time (as many features can be interpreted as numeric values that can contribute to prediction).

Exploratory Data Analysis (EDA):

The EDA phase involved examining the distribution and relationships between variables to gain a better understanding of the data. Key observations from the EDA include:

- Price Distribution: The distribution of housing prices was found to be right-skewed, with a few properties having significantly higher prices compared to the majority.
- Correlation Analysis: Certain variables exhibited strong correlations with the sale price, such as the overall quality, living area square footage, and number of rooms. These variables are likely to have a substantial impact on housing prices.
- Categorical Variables: Neighborhood and dwelling type were found to be important categorical variables affecting house prices. Some neighborhoods had higher average prices, indicating their desirability.

Feature Engineering

To enhance the predictive power of the model, additional features were engineered from the existing dataset. This involved creating new variables based on domain knowledge, such as the age of the house or the total square footage. These engineered features aimed to capture important aspects that could influence housing prices.

Feature Selection

The feature selection process involved the correlation analysis which I used as basis to select a hypothetical feature subset. The most correlated features where also the ones coming from domain knowledge.

I have then performed an estimation of feature importance through an XGBoost on multiple types of importance and continued with a recursive feature selection through both XGBoost and a Random Forest.

From all the information gathered I decided to save 3 different dataset with some features in common and then I have passed those to the respective models.

XGBoost vs Elastic Net

For both regressors the routine was the same:

- Inference of best hyperparameters by GridSearchCV through RepeatedKFold
- Test of testing instances

The XGBoost regressor presented some interesting variance when it came to the final predictions, while the ElasticNet had the best results in MSE, RMSE and the lowest errors compared to XGBoost.

This brings us to the fact the dataset can really be fit for a parsimonious linear regression and the right transformations can obtain that.