# FDS Final Project

*House Pricings Understanding*

Michelangelo Saveriano
Paolo Falcone

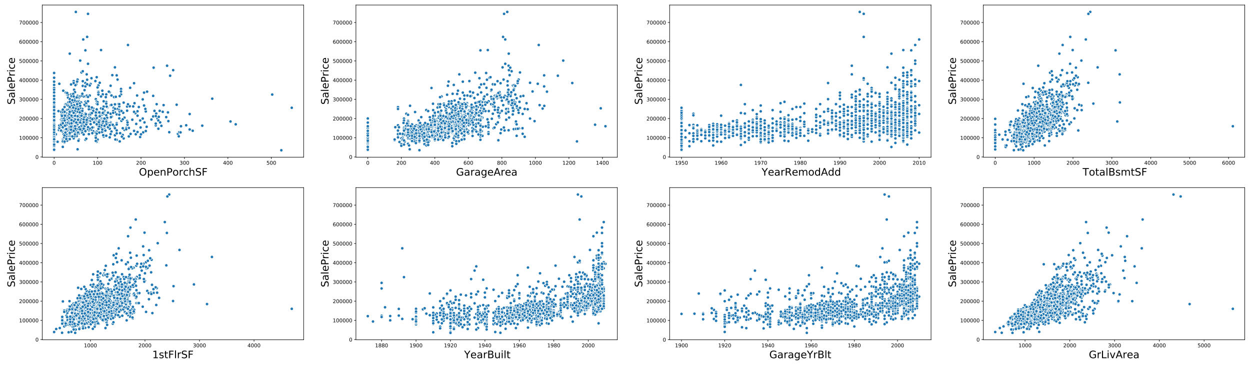
December 2021

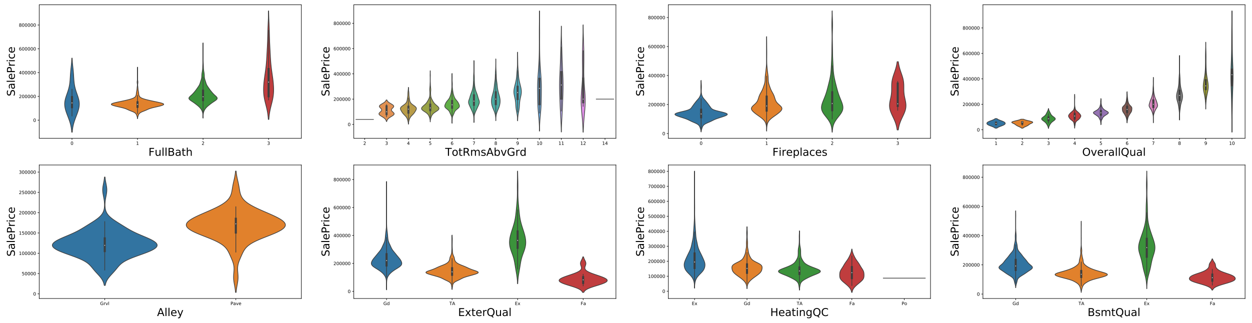
## Introduction

For this project we decided to take on the task of **predicting and understanding how an house price is established**, what it depends on and the most important features to predict one, we did this by partecipating in an on-going Kaggle Competition: Click here

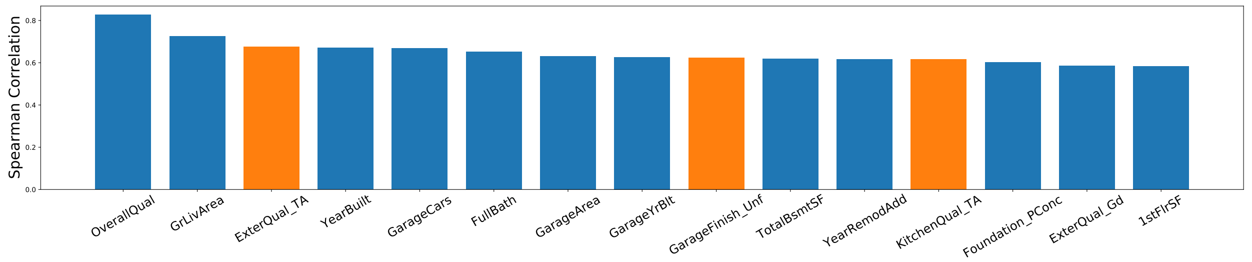
## Exploratory Data Analysis

The first thing we did was **understanding the dataset that was provided to us**, it was a csv with 1460 rows and 78 columns, each column representing a feature for a certain house (E.G Squared Feet). **We then did a plot analysis to get a grasp of what would later be the most important features for the final house price understanding**.





As we can see from the plots above there are many features which show an heavy correlation with the final house price like the squared feet or the overall quality of a house. One thing that we noticed was that **even in highly correlated plots there were always a heavy presence of outliers**, this info would later be important to further refine our models.

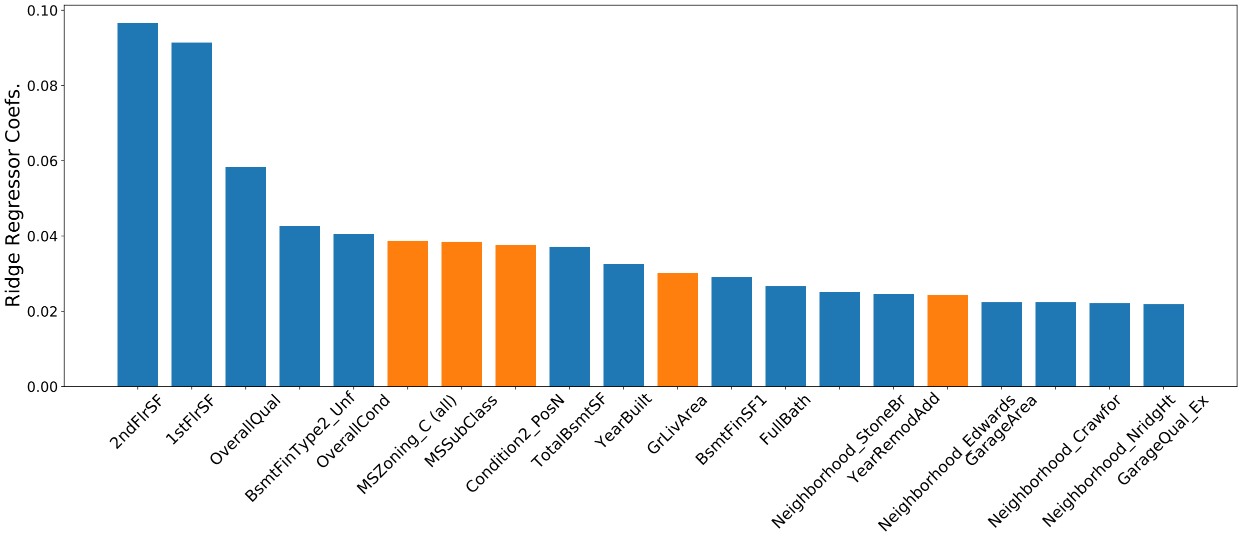


**Correlations**: the plot above shows **the** 15 **most correlated features** (blue are positive and orange are negative correlations). This helps us understand the dataset at its finest and also it quantifies how much each feature is linked to the price, telling us which are the most important features we should ask to an hypothetical users for a house price estimation. We used the *Spearman* $\rho$ as correlation measure since it's more robust and able to catch different types of correlations.

## Model Creation and Selection

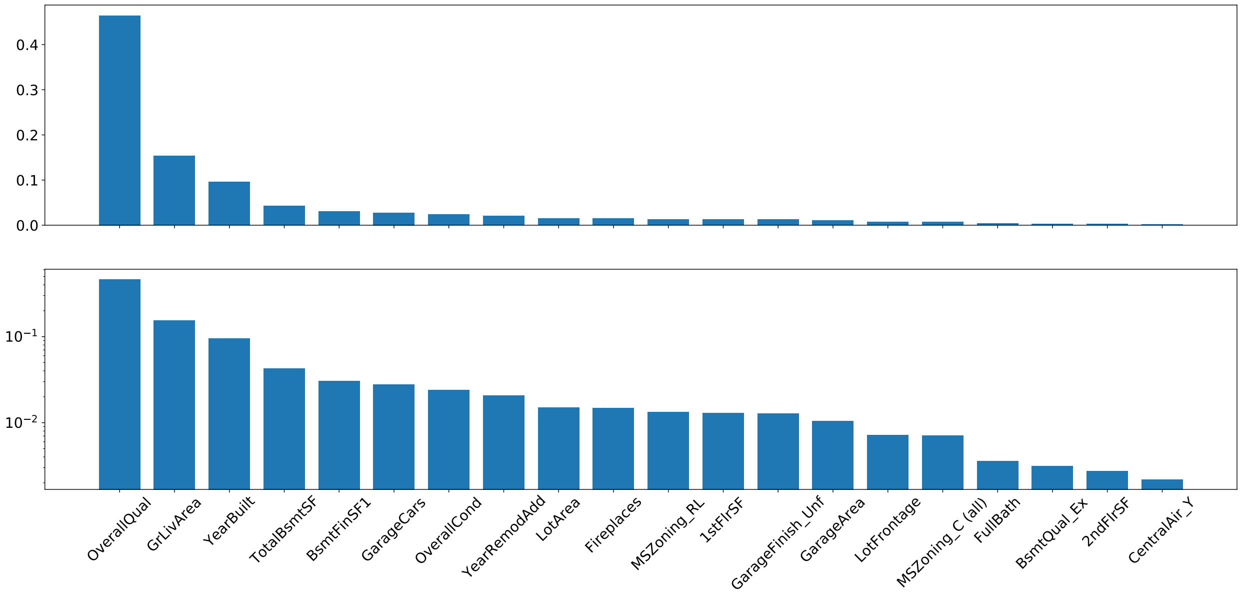
Here we present you the models we tested, the results we achieved and a way we found to optimize our best model thanks to an ablation study.

**Ridge Regression**: The first model we instanciated as baseline is a Ridge Regressor *(Linear Regressor + L2 Penalty)*. Below we can see the top 20 coefficients associated to each input features.



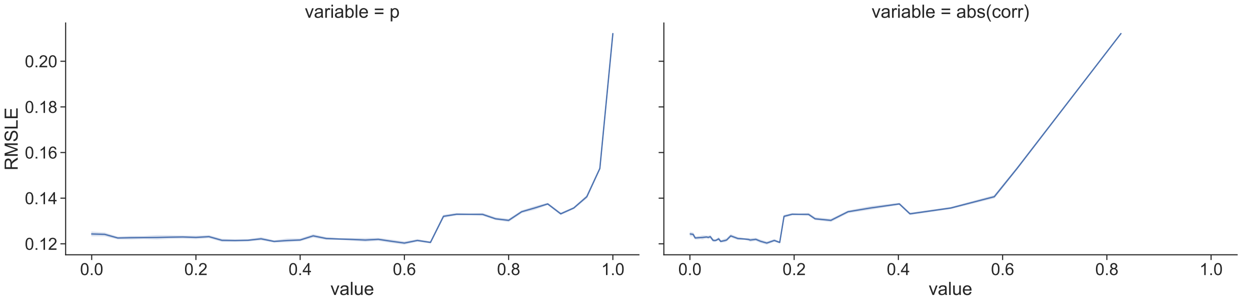
We can see how important features like the *Second Floor Squared Feet, First Floor Squared Feet, Overall Quality of the House* are for this linear model.

**Gradient Boosting**: The model we found that reaches the best performance is the Gradient Boosting and here we can see the top 20 most informative features, according to the *Gini importance*, for this model.



For this kind of model the most important feature is by far the *Overall Quality*, also note that the Gini importance is rapidly decreasing: few features are enough to esitamate and explain the final house price. This will be really useful when later we'll deploy our model.

**Ablation Study**: Once we had our class of models we decided to test what would happen when we reduce the amount of features we're giving as input. To do so we iteratively drop an increasing portion of it.



Our analysis reports that the $RMSLE$ is slightly decreasing until $p = 0.6$ and then is rapidly increasing. This means that dropping all the features such that their Spearman $\rho$ correlation with the log of the price is $-0.15 < \rho < 0.15$ gives us an optimized version of our model, completely detached form the less informative variables.

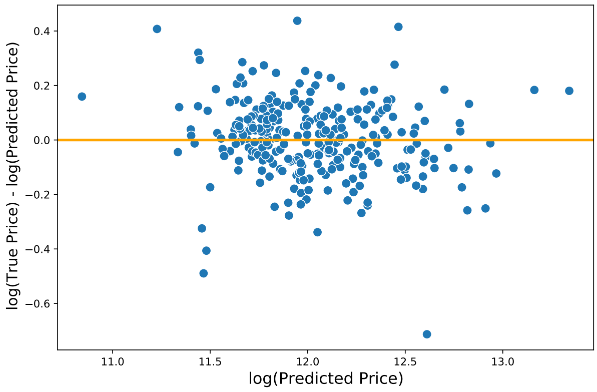
**Results**: the metrics we used to compare different models are $RootMeanSquaredLogError$ ($RMSLE$) and $RootMeanAbsoluteLogeError$ ($RMALE$).

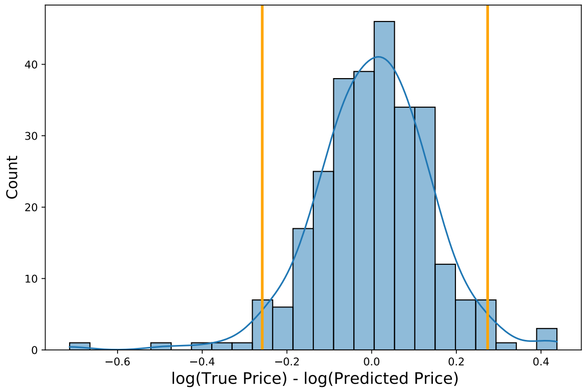
| Model | $RMSLE$ | $RMALE$ |
|---|---|---|
| Ridge | 0.196 | 0.305 |
| kNN | 0.199 | 0.368 |
| SVR | 0.192 | 0.342 |
| Random Forest | 0.143 | 0.312 |
| Gradient Boosting | 0.124 | 0.300 |
| **Gradient Boosting + Feats. Removed** | **0.119** | **0.294** |

As we can see the the combination **Gradient Boosting** and **Features Removed** outperforms all the model we tested.

## House Price Evaluation

In this part we want to **deploy our model** to estimate for some given house features a possible price and test if the user price is close to our prediction for the information given.
We did this by training our best performing model (Gradient Boosting) on a small subset of the features resulting in a $RMSLE = 0.139$ and a $RMALE = 0.323$, this means that although we used only 16 features the final metrics don't degrade too much. After this we would ask the user to input said variables and then use our model to predict the house price; instead of spitting out a sigle value (which isn't realistic) **we gave a price interval which is based off our model error**.
The confidence interval we used is the 95% Empirical CI evaluated on the test residuals where $residual = log(TruePrice) - log(PredPrice)$ and the $PredPrice$ are the predictions of the model trained using only the most correlated features.

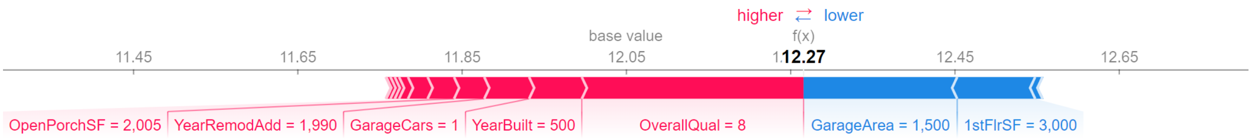




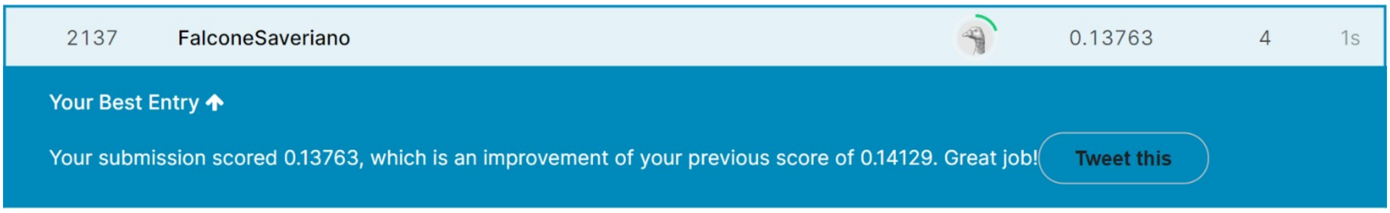
| Feature | Value | Feature | Value | Feature | Value | Feature | Value |
|---|---|---|---|---|---|---|---|
| TotalBsmtSF | 1000 | OpenPorchSF | 2005 | FullBath | 3 | OverallQual | 8 |
| GrLivArea | 0 | 1stFlrSF | 3000 | TotRmsAbvGrd | 2 | ExterQual_Gd | True |
| GarageArea | 1500 | GarageYrBlt | 2005 | Fireplaces | 1 | Foundation_PConc | True |
| YearBuilt | 500 | YearRemodAdd | 1990 | GarageCars | 1 | HeatingQC_Ex | True |

The 95.0% of the residuals are within the interval $[-0.246, 0.218]$.

**Example**: Giving as input to the model the sample we can see in the table above we have that the predicted price is 146118 with 95.0% CI $= [112856, 188647]$ and we can also visualize, from the plot below, as the different features contribute to the final prediction.



## Kaggle Competition Results



2137 FalconeSaveriano 0.13763 4 1s

Your Best Entry ↑

Your submission scored 0.13763, which is an improvement of your previous score of 0.14129. Great job!

We can see from the image above that with the final submission we managed to improve of 1.5k position, securing a solid 2137th place. **This got us in the top 35%** of the leaderboard just 0,025 away from top 100.

## Conclusion

We think that in order to make this model truly deployable the predictions should be more accurate, to do so few more things that could be done, for instance: features engineering, testing other models like Neural Networks, acquisition of new and more expressive features, etc.