Two page project summary

More in-depth analysis can be found in the Jupyter notebook.

Data

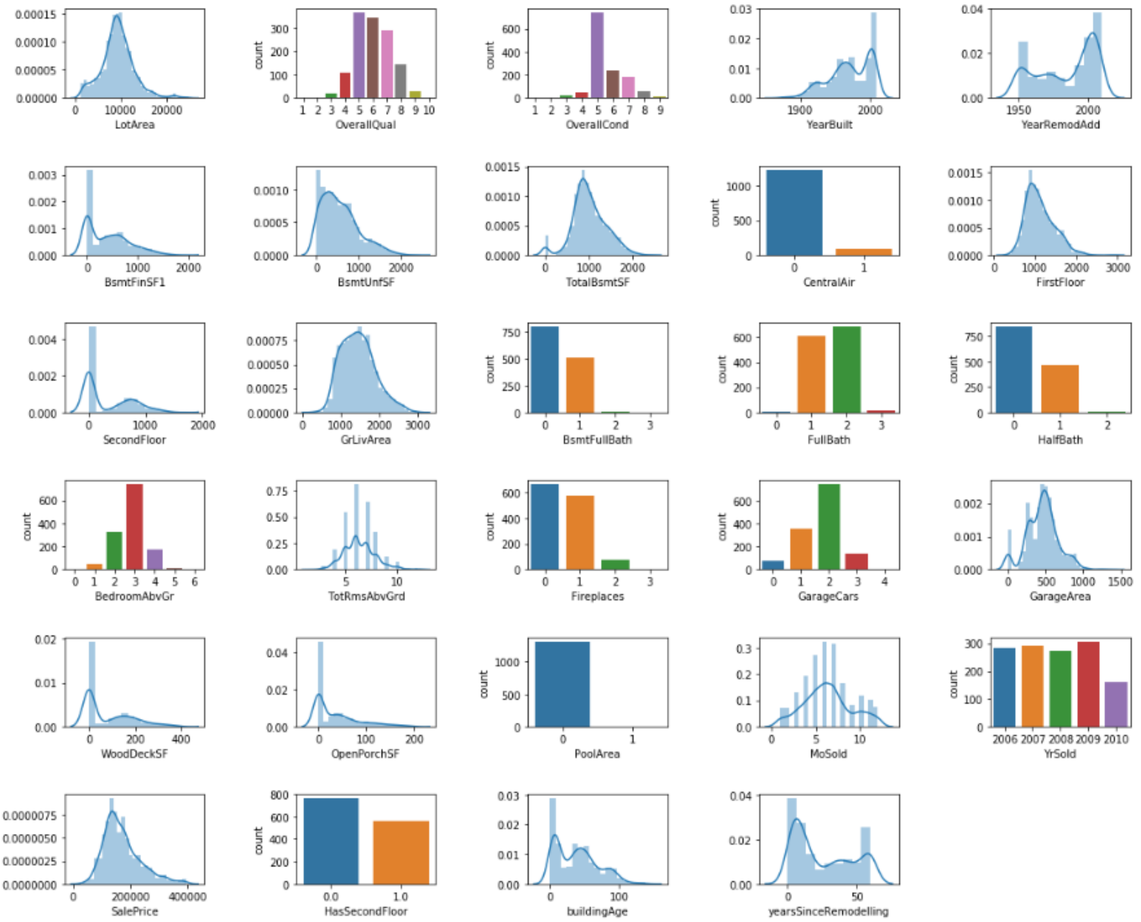
The data describes houses in the city of Ames in Iowa, USA. It consists of 1 460 observations and 81 variables. Each observation represents a different building. Judging by the city population (66 000), it can be assumed that the houses in this dataset represent a significant portion of all houses in this city. This dataset was created by taking all sales in the 2006 to 2010 period, and exported to Kaggle.

Data manipulation

Since the dataset consist of a mixture of continuous, rank and categorical variables, we needed to start with some feature selection. After taking a quick look at the data, we decided to: drop all categorical variables with more than 2 levels, convert all remaining to binary. If any of them exhibited strong skewness, it was also removed. This allowed us to avoid using a large number of dummy variables. We also run a check on all columns, and dropped all that had a significant number of missing data.

EDA

After data manipulation, we are left with 19 continuous and binary variables. All of them have good distributions, many displaying significant normality.



Some of the variables are correlated, both positively and negatively. This will need to be removed to perform OLS analysis, as leaving them can lead to unexpected results.

Machine Learning Models

We decided to compare 4 machine learning methods – OLS, Ridge, Lasso and ElasticNet. We started with running basic OLS on all variables to get a glimpse into the data. We then run the OLS using cross validation, but it did not influence the results.

As a next step, we constructed a list on features that were the most significant in the previous OLS model. We also removed correlated variables. We then proceeded to run another OLS on this smaller set of features.

After this, we moved to more advanced models. We run Ridge, Lasso and ElasticNet on cross validation to estimate the alpha and ratio parameters.

In the end, we run all the final models and compared their predictive power using R^2. We also wrapped them into functions, and performed a simple 100 iteration time estimation using the timeit library.

Findings

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-4.957e+04	6417.088	-7.725	0.000	-6.22e+04	-3.7e+04
LotArea	1.5667	0.215	7.295	0.000	1.145	1.988
OverallQual	1.492e+04	838.468	17.791	0.000	1.33e+04	1.66e+04
OverallCond	4159.2432	759.970	5.473	0.000	2668.345	5650.141
buildingAge	-358.2261	39.446	-9.082	0.000	-435.610	-280.842
yearsSinceRemodelling	-278.6067	45.900	-6.070	0.000	-368.652	-188.561
BsmtFinSF1	20.9782	1.846	11.363	0.000	17.356	24.600
TotalBsmtSF	20.7353	2.294	9.040	0.000	16.236	25.235
CentralAir	8327.7889	3065.714	2.716	0.007	2313.513	1.43e+04
GrLivArea	42.4448	2.115	20.072	0.000	38.296	46.593
Fireplaces	6592.1161	1258.318	5.239	0.000	4123.565	9060.668
GarageArea	33.6709	4.233	7.955	0.000	25.367	41.975
WoodDeckSF	19.6726	6.912	2.846	0.004	6.112	33.233
OpenPorchSF	53.3614	15.548	3.432	0.001	22.859	83.864

In the OLS model, we were left with 13 significant variables. All of them seem to be in line with the general wisdom – the price of the house is dependent on the area of the lot and the house, its condition and quality, its age and the time since a major remodeling. Having a central air conditioning and a fireplace seem to increase the value of the house substantially. Interesting are the wood deck and open porch variables – each square feet seem to increase the values substantially.

	R^2 value	Time to run in seconds
OLS - correlation among variables	0.8687	0.096
OLS - with feature selection	0.8625	-
Ridge	0.8618	0.116
Lasso	0.8598	0.211
ElasticNet	0.8616	0.474

The R^2 values of the estimated models are very similar. We were not able to achieve a significant increase in predicting power. But this is not the main feature of Ridge or Lasso. Their main benefit is the automatic feature selection, that was able to get rid of unwanted variables. This is reflected in the time that was needed to run this methods. Ridge seems to be nearly as fast as OLS, Lasso was around 100% slower, while ElasticNet was nearly 5 times slower.