You have just joined a new "full stack" real estate company in Ames, Iowa. The strategy of the firm is two-fold:

* Own the entire process from the purchase of the land all the way to sale of the house, and anything in between.
* Use statistical analysis to optimize investment and maximize return.

The company is still small, and though investment is substantial the short-term goals of the company are more oriented towards purchasing existing houses and flipping them as opposed to constructing entirely new houses. That being said, the company has access to a large construction workforce operating at rock-bottom prices.

This project uses the [Ames housing data recently made available on kaggle](https://www.kaggle.com/c/house-prices-advanced-regression-techniques).

The objective is to predict Sale Prices based on fixed features (i.e.



**1  Estimating the value of homes from fixed characteristics.**

Your superiors have outlined this year's strategy for the company:

1. Develop an algorithm to reliably estimate the value of residential houses based on *fixed* characteristics.
2. Identify characteristics of houses that the company can cost-effectively change/renovate with their construction team.
3. Evaluate the mean dollar value of different renovations.

Then we can use that to buy houses that are likely to sell for more than the cost of the purchase plus renovations.

Your first job is to tackle #1. You have a dataset of housing sale data with a huge amount of features identifying different aspects of the house. The full description of the data features can be found in a separate file:

housing.csv

data\_description.txt

You need to build a reliable estimator for the price of the house given characteristics of the house that cannot be renovated. Some examples include:

* The neighborhood
* Square feet
* Bedrooms, bathrooms
* Basement and garage space

and many more.

Some examples of things that **ARE renovateable:**

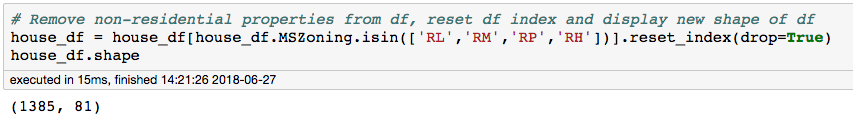
* Roof and exterior features
* "Quality" metrics, such as kitchen quality
* "Condition" metrics, such as condition of garage
* Heating and electrical components

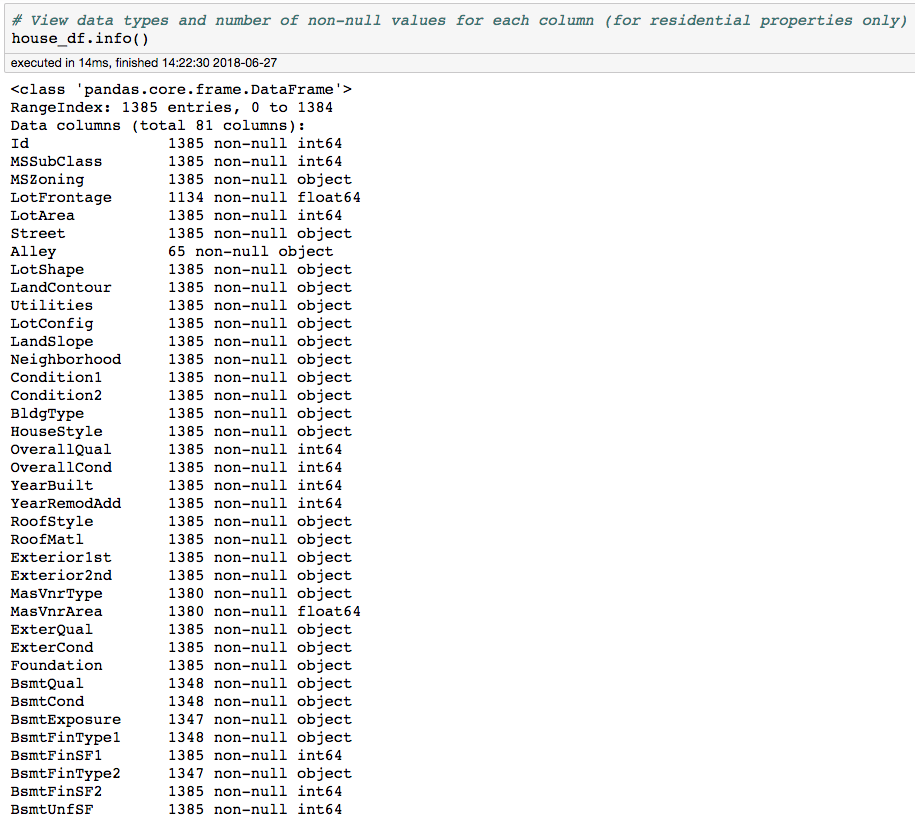
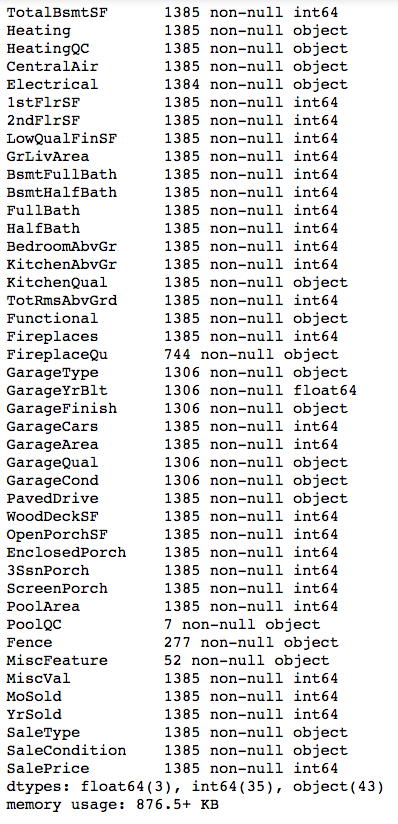
and generally anything you deem can be modified without having to undergo major construction on the house.

**Your goals:**

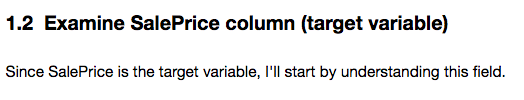
1. Perform any cleaning, feature engineering, and EDA you deem necessary.
2. Be sure to remove any houses that are not residential from the dataset.
3. Identify **fixed** features that can predict price.
4. Train a model on pre-2010 data and evaluate its performance on the 2010 houses.
5. Characterize your model. How well does it perform? What are the best estimates of price?

**Part 1: Cleaning, feature engineering and EDA**

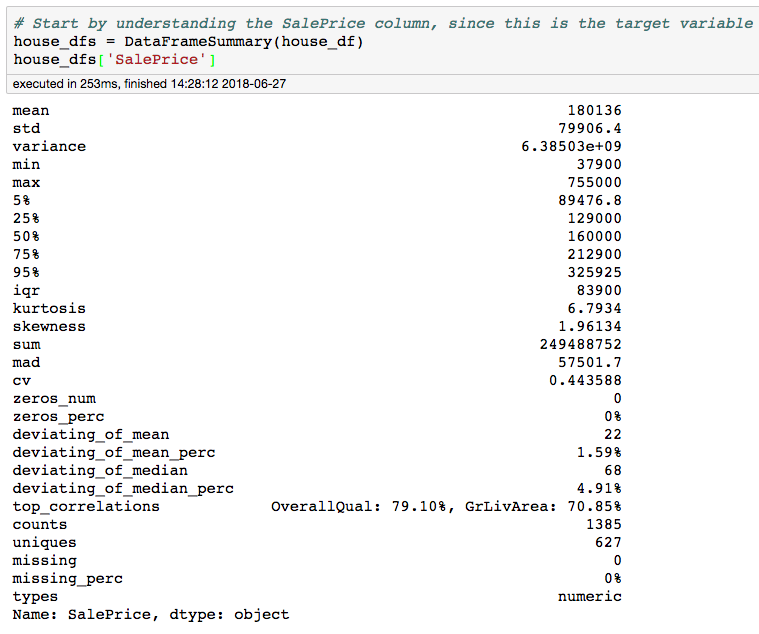
* Remove 10 commercial and 65 floating village properties from data  
  
* Starting point of data:

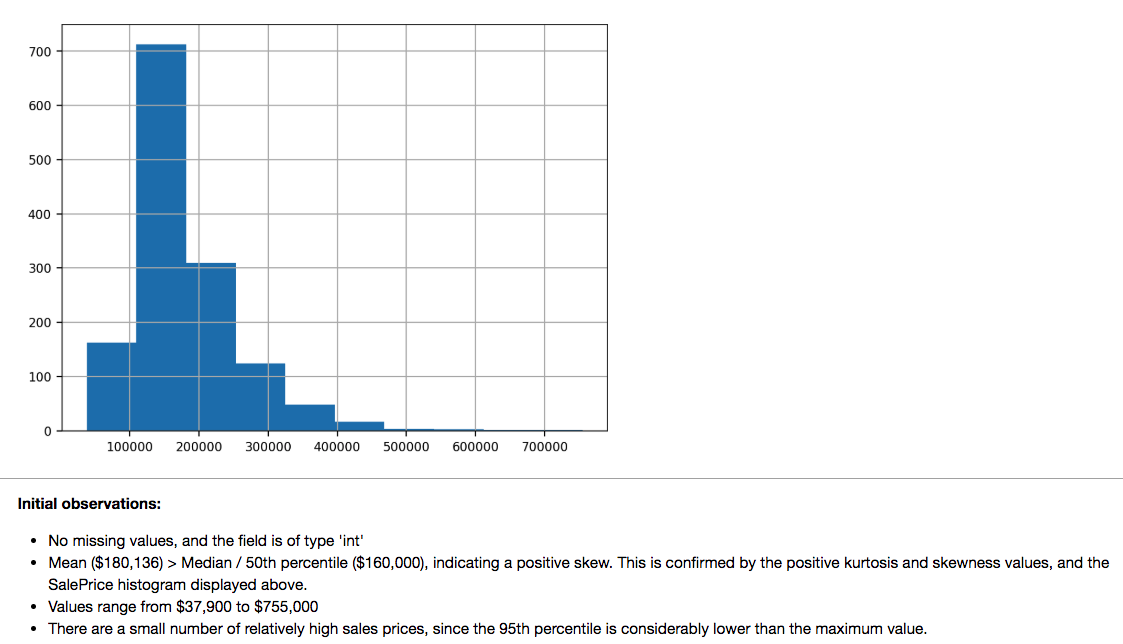


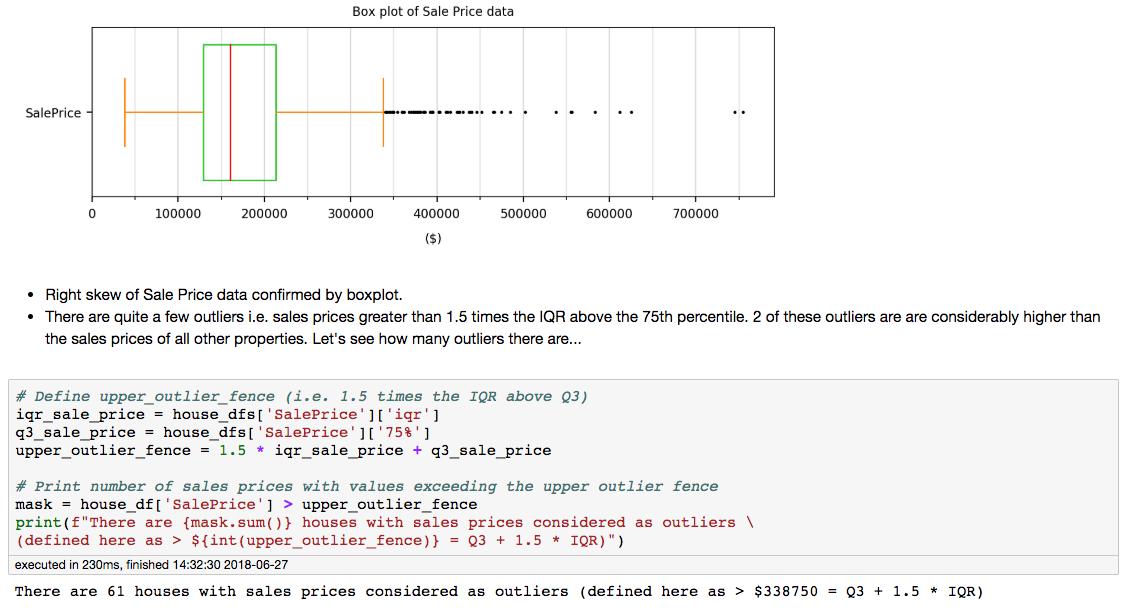
Several columns have null values. These will need to be managed. 43 fields are type ‘object’ - I'll need to confirm that this is correct as I use the data.

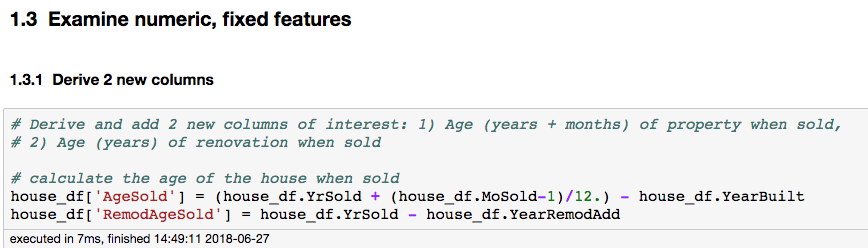


Use DataFrameSummary from pandas\_summary to display more detailed descriptive statistics of the SalePrice field.









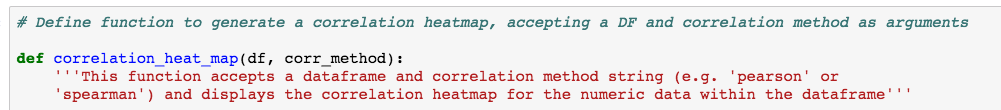


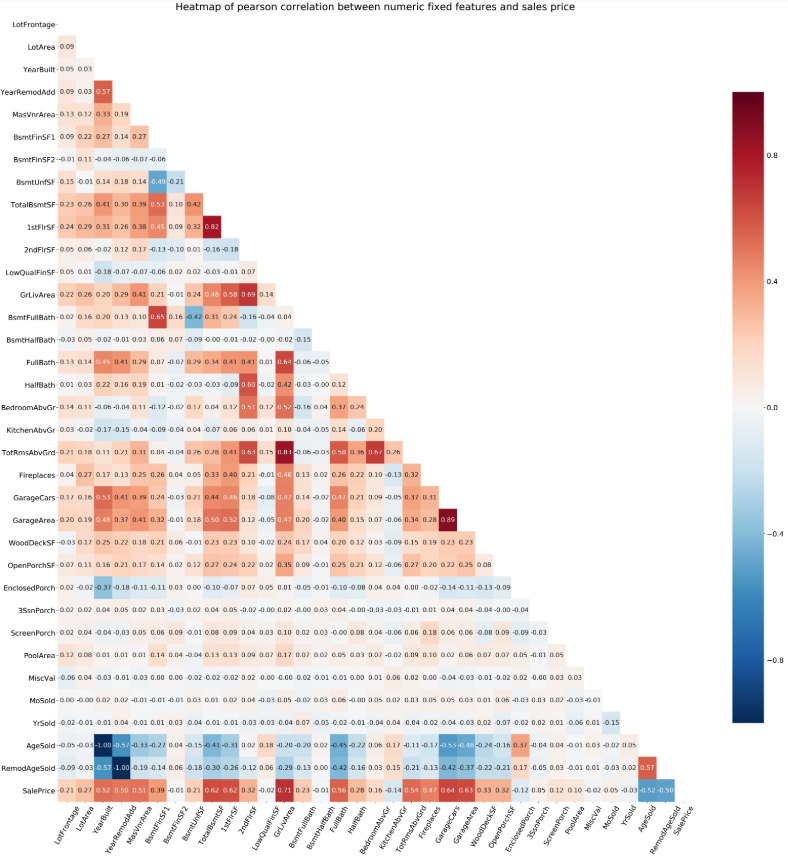




Investigated null values and decided to set to zero, based on analysis



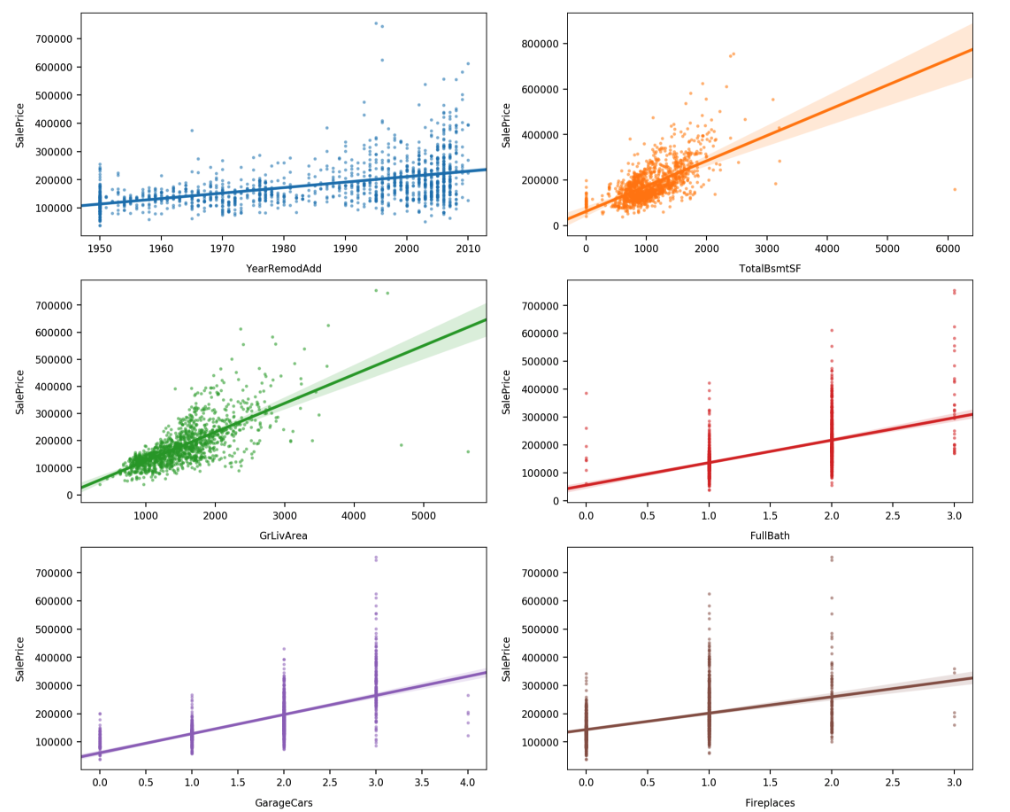


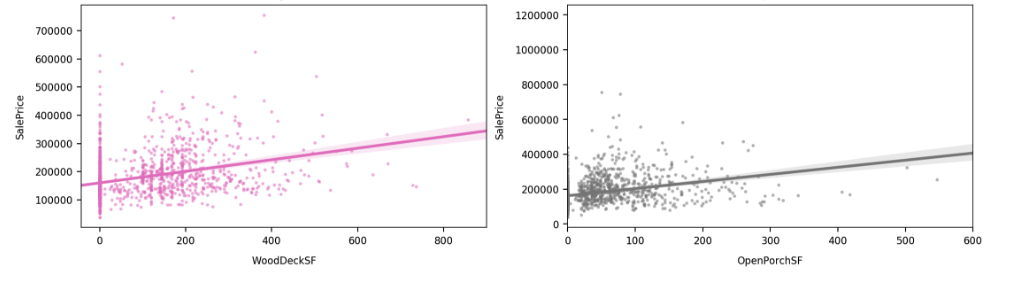
­

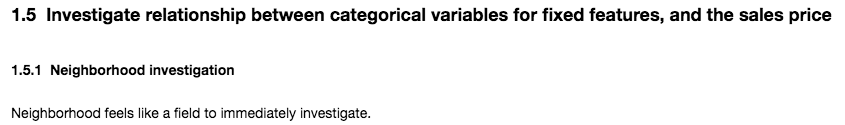
selected features based on correlation with Sales Price, and excluded features that exhibited a high correlation with each other (e.g. YearBuilt/AgeSold and GrLivArea / TotRmsAbvGrd)

A lot of the variables proposed for inclusion in my predictors above are related to space, and I'd like to include some non-space related predictors in my model to reflect specific features that may influence the sales price. Based on the correlation heatmap above, I'm going to include Fireplaces (correlation 0.47), WoodDeckSF (correlation 0.33) and OpenPorchSF (correlation 0.32). These predictors reflect specific features and have minimal correlation with other predictors.

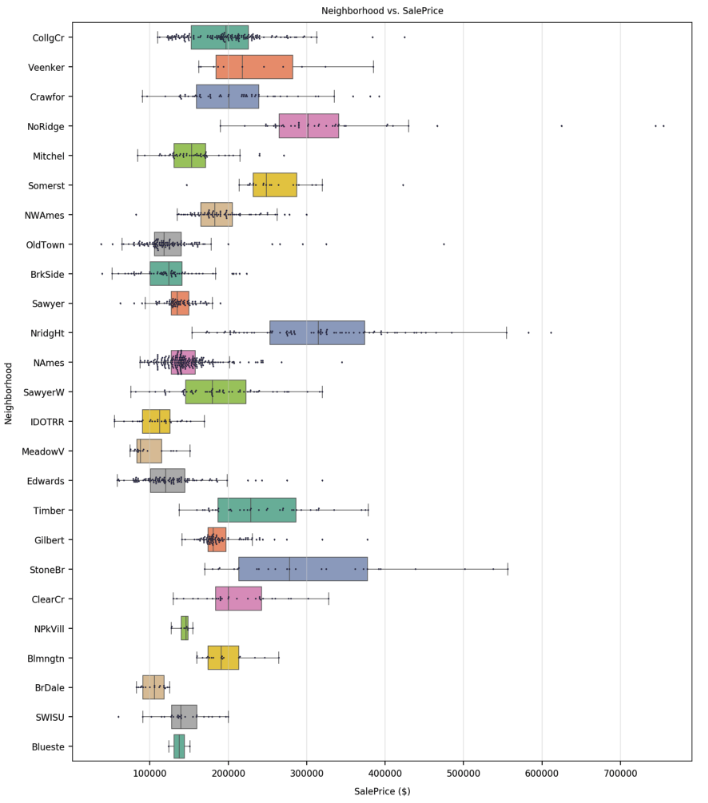
Plotted relationship between selected numerical predictors and sales price to confirm linear relationship visually



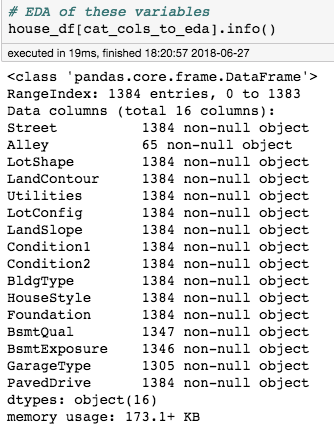




Plot Sales Price across 25 neighbourhoods

­ 

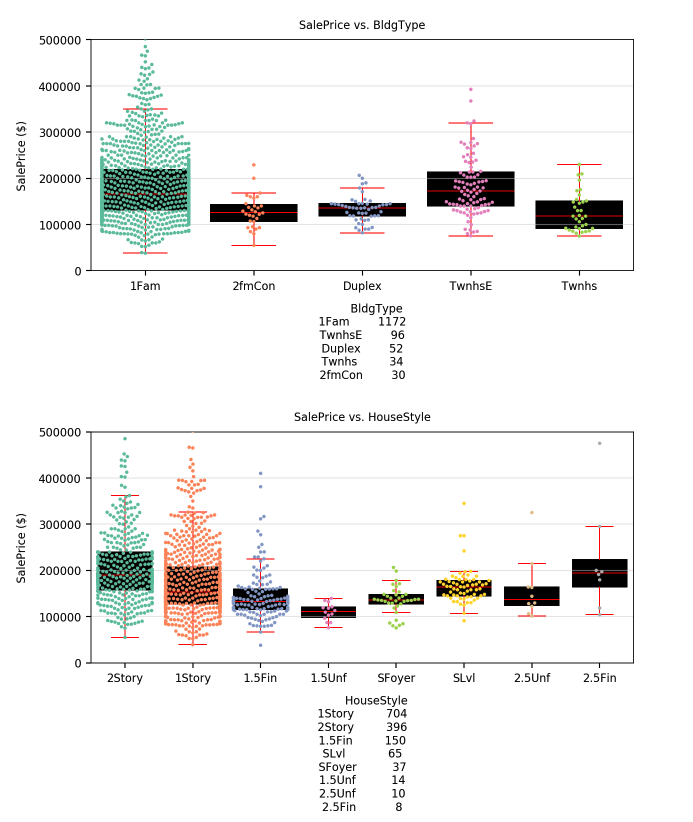
* There are several other features of interest with null values. These will need to be managed.

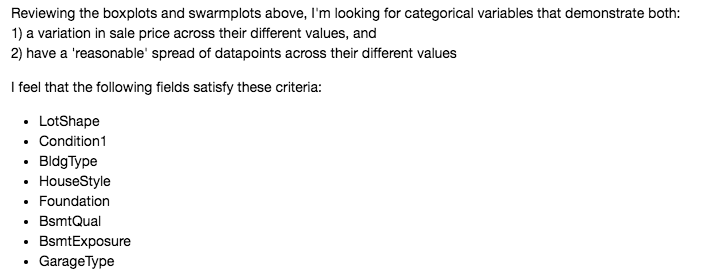


**Filled null values in** Alley, BsmtQual, BsmtExposure and GarageType fields, as per data dictionary.

Continue EDA.

Plotted boxand swarm plots of each categorical feature of interest, along with value counts in x-axis to understand relationship with Sale Price across different values of the feature. 2 features below, for example:

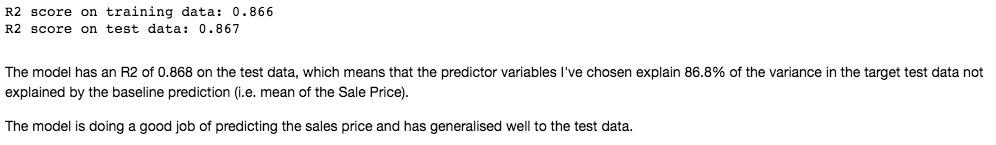




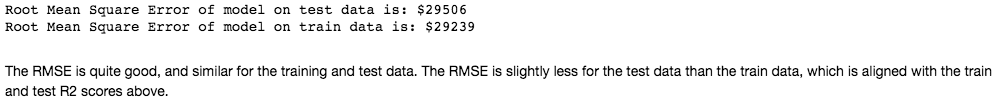


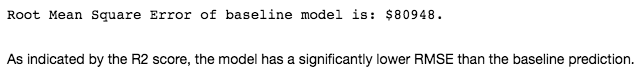


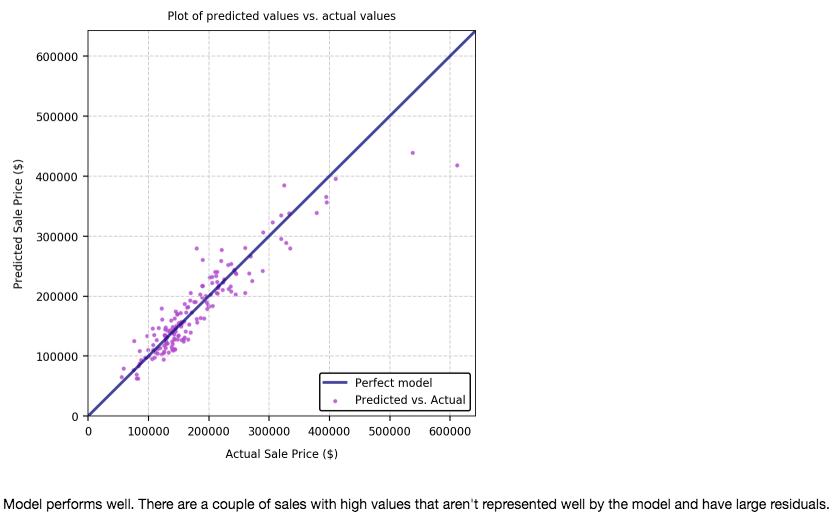
Dumified categorical features, split data into train and test, standardised train and test feature data with mean and standard deviation of training data and fit a Linear Regression model, without any cross validation or regularisation.



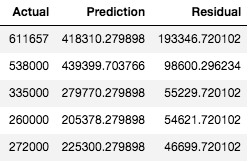




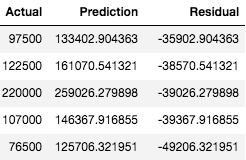




5 Largest underestimations by model:



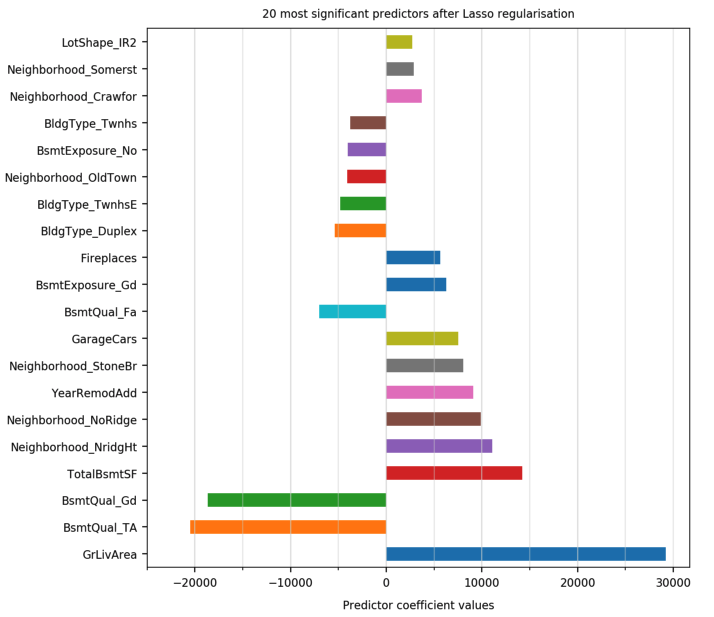
5 largest over estimations by model:

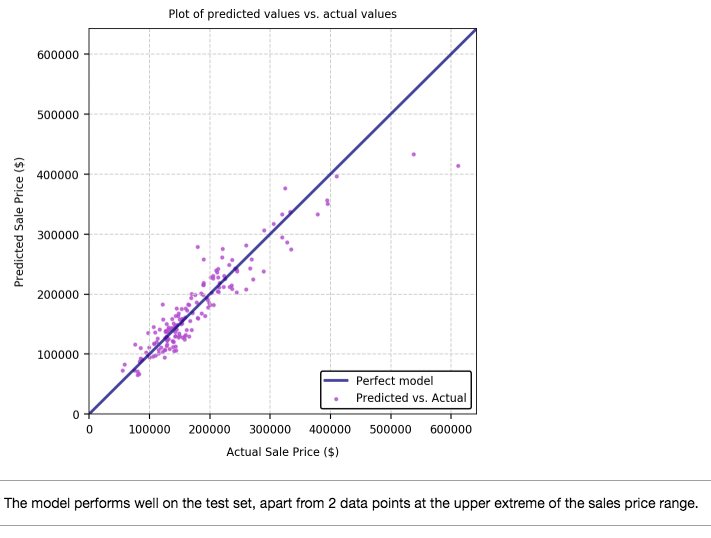


No improvement in model performance with GridSearch on Ridge regularisation hyperparameters.

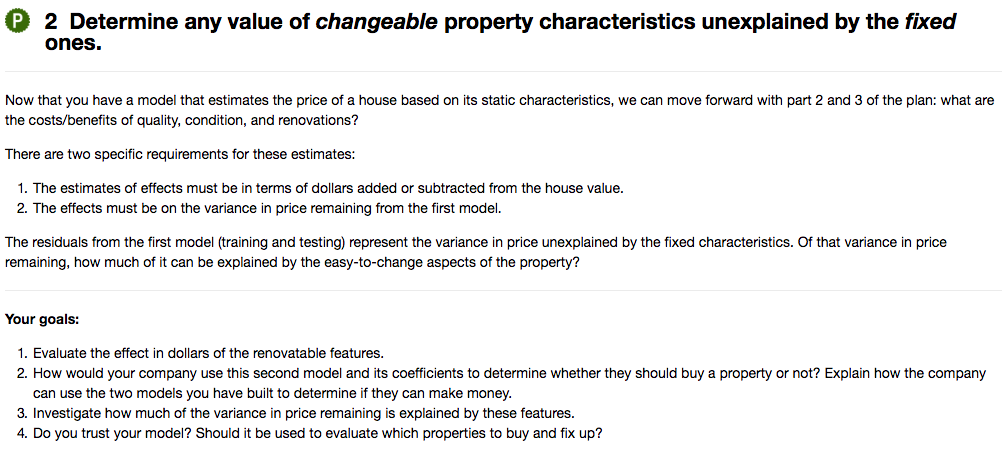
R2 score on test data for optimum Lasso model is 0.865.

Lasso regularisation zeroed out 22% of coefficients. The 20 most significant coefficients are displayed below.



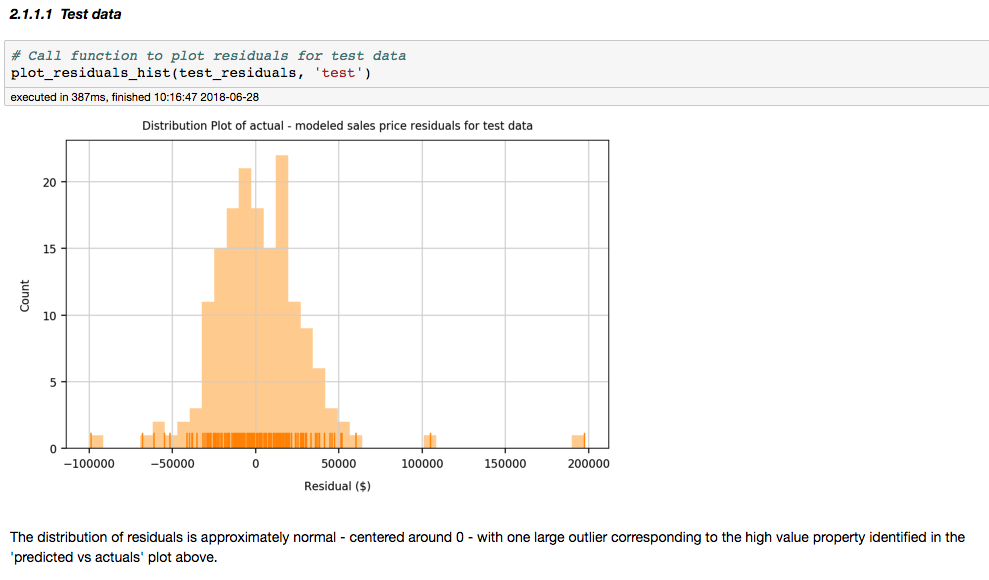


Part 2:

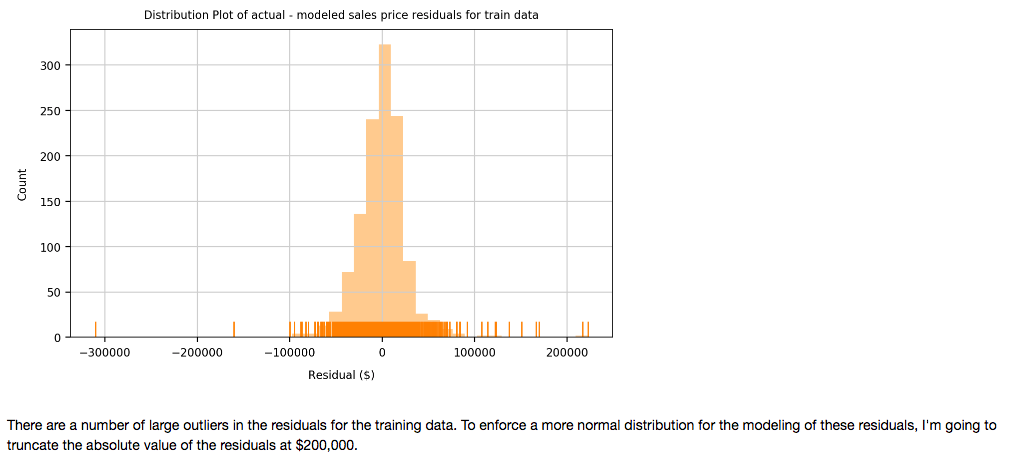


Started by looking at residuals for test and training data:

Test data:



Train data

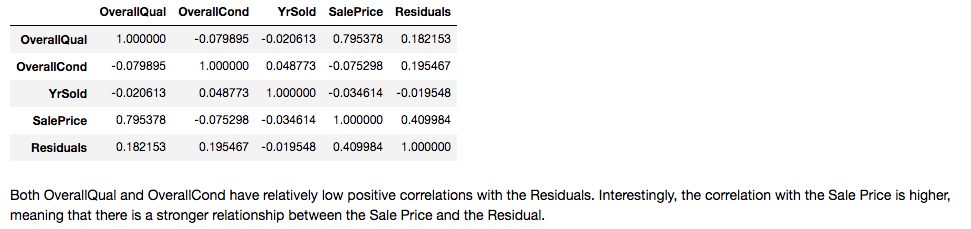


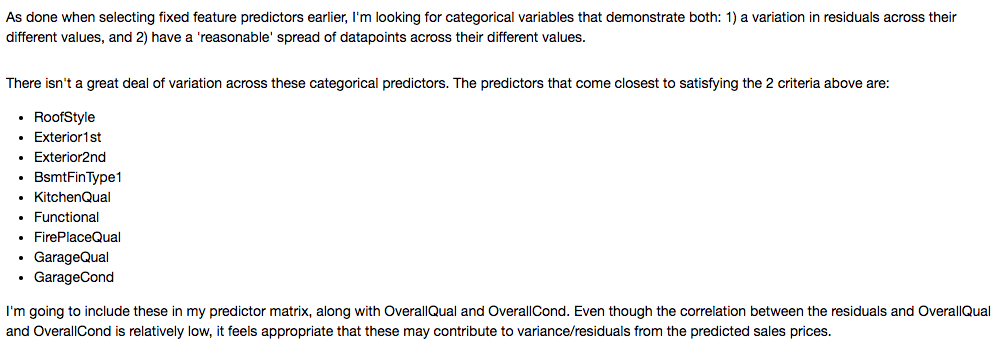


Subsetted data on changeable features only

Filled null values







Trained and evaluated a Lasso regularisation model with GridSearch

