**King County House Price Prediction**

**Abstract**

In this assignment, I am trying to build a model that can generate predictions for future housing prices for Kings County CA. In order to select a prediction method, various regression methods were explored and compared. Gradient Boost regression was chosen as model due to its flexible and probabilistic approach to learning and model selection.

**Introduction:**

The goal of this project is to use machine learning to predict the selling prices of houses based on a number of factors. We are using data from a dataset of housing prices in King County, USA, from kaggle.com. We partitioned the dataset into a training set (19,451 data points, 90% of total dataset), and testing set (2,162 data points, 10% of total dataset)

Raw input data:

For learning, our input is the following 21 features in the which make up each data point

id - Unique ID for each home sold

date - Date of the home sale price –

Price-Price of each home sold

bedrooms - Number of bedrooms

bathrooms - Number of bathrooms, where .5 accounts for a room with a toilet but no shower

sqft\_living - Square footage of the apartments interior living space

sqft\_lot - Square footage of the land space

floors - Number of floors

waterfront - A dummy variable for whether the apartment was overlooking the waterfront or not

view - An index from 0 to 4 of how good the view of the property was

condition - An index from 1 to 5 on the condition of the apartment,

grade - An index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design.

sqft\_above - The square footage of the interior housing space that is above ground level

sqft\_basement - The square footage of the interior housing space that is below ground level

yr\_built - The year the house was initially

built yr\_renovated - The year of the house’s last renovation

zipcode - What zipcode area the house is in

lat - Lattitude

long - Longitude

sqft\_living15 - The square footage of interior housing living space for the nearest 15 neighbors

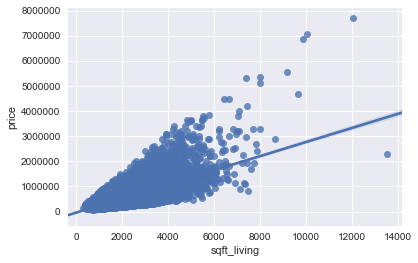
sqft\_lot15 - The square footage of the land lots of the nearest 15 neighbors

It’s been found that not all of these features are meaningful. For example, we immediately removed ‘id’ and ‘date’ from the input features. The unique ID for each home sold is useless information that intuitively does not contribute to the selling price of a home. We found that since the date at which each home was sold varied in our dataset only between two years, and under closer examination we found that this did not sufficiently alter the selling price of the 3 home and therefore was not a feature that should contribute to our learning.

**Code with Documentation**

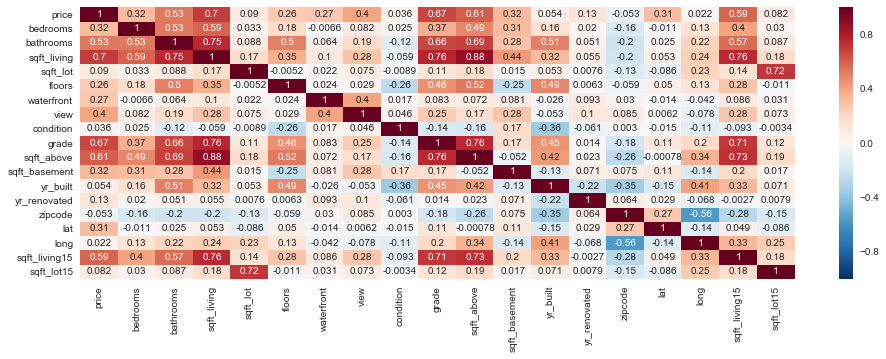
**Linear Regression:**

Linear Regression A predictor that incorporates Linear Regression seems to be the best way to model this dataset. This is because the data follow a highly linear relationship - all we have to do is select features that represent that linear relationship best.

****

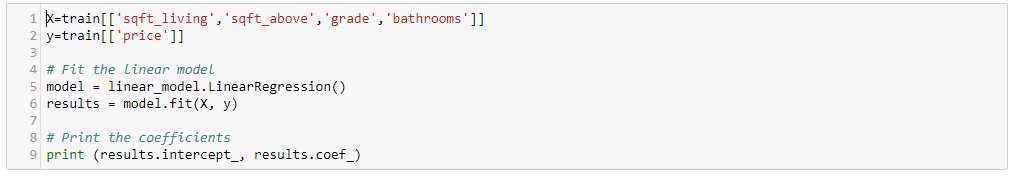
**Figure 1**

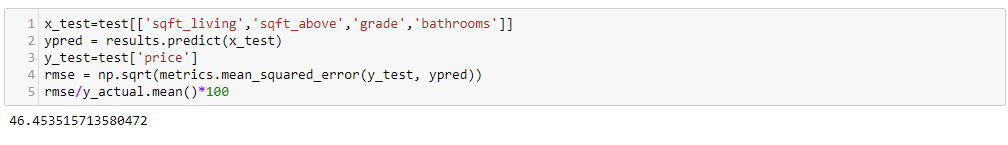
The following figure shows the correlation matrix heatmap with Python Seaborn library. We can see that there are a few variables have quite high correlated between each other. The correlation between sqft\_above and sqft\_living is 0.88. The correlation between sqft\_livng and grade is 0.76. In general, the correlation for sqft\_living associated features are higher than others. High correlation between features might have multicollinearity problem.

****

**Figure 2**

By taking best feature in account which were highly co related to the dependent variable we are calculating the root mean square error to measure the performance of the model:





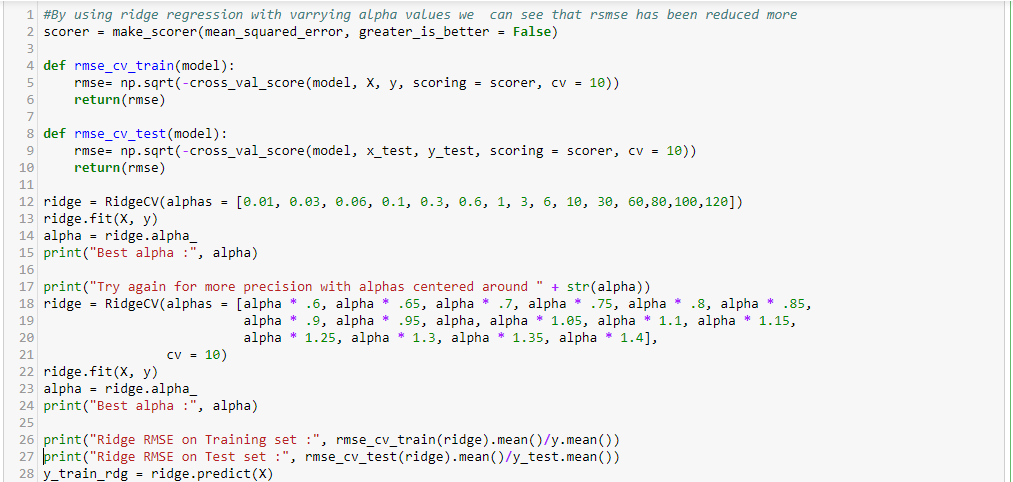
**Ridge Regression:**

Ridge Regression is a remedial measure taken to alleviate multicollinearity amongst regression predictor variables in a model. Often predictor variables used in a regression are highly correlated, which is being the case in our dataset. Ridge regression adds a small bias factor to the variables in order to alleviate this problem.

Ridge regression with fixed alpha factor of 0.5:

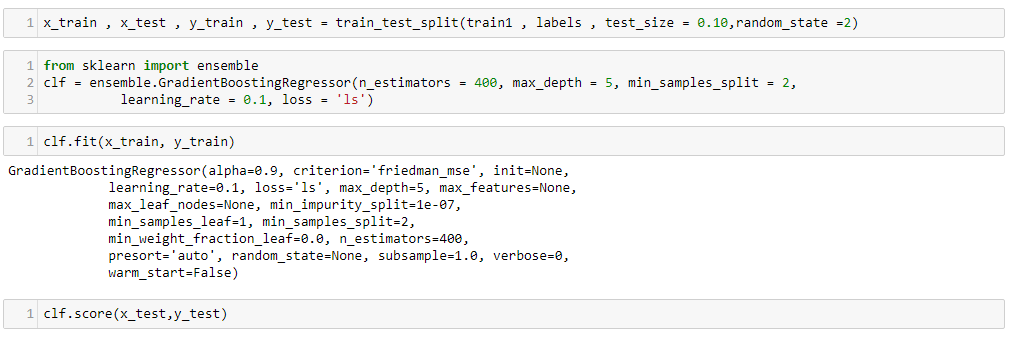
****

Ridge regression with cross validation and random alpha values:



**Gradient boosting regression:**

It is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models



**Results**

To evaluate which model is doing better, we will need some way to score the results. We choose root mean square error here as the score, for the reason that the result of root mean square error is more intuitive: it presents the percentage of difference in dollars between the prediction and the actual price.

|  |  |
| --- | --- |
| Model | RMSE (Root Mean Square Error) Value(Percentage) |
| Linear Regression | 46.45 |
| Ridge Regression | 43.89 |
| Ridge Regression with Cross Validation | 43.65 |
| Gradient Boost Regression | 20.41 |

Looking at the error from each of these four estimators Linear, Ridge and RidgeCV appear to be roughly the same. Still results are slightly better than with ridge regression. Gradient Boost Regression produced the best results with minimum error.

**Discussion**

Our analysis shows that we can successfully predict price using an array of features extracted from listings. Using different regression as comparison, the gradient boosting regression shows great accuracy and ability in avoiding over fitting. All the regression technique being used in the experiment shows somewhat fair performance, which suggests the features we use, are reasonable.

**References**

[**https://onlinecourses.science.psu.edu/stat857/node/155**](https://onlinecourses.science.psu.edu/stat857/node/155)

[**https://towardsdatascience.com/create-a-model-to-predict-house-prices-using-python-d34fe8fad88f**](https://towardsdatascience.com/create-a-model-to-predict-house-prices-using-python-d34fe8fad88f)

[**https://www.kaggle.com/**](https://www.kaggle.com/)