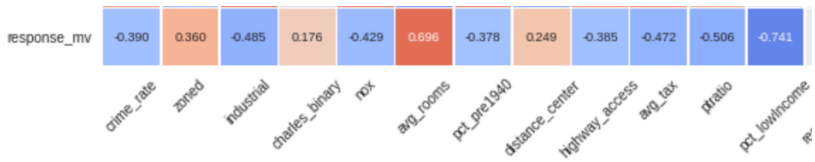
Introduction: Summary and problem definition for management

Assignment 3 is an exercise using SciKit Learn Machine Learning methods to determine the best fit for a linear model. The exercise uses the Boston housing data from a book about regression diagnostics by Belsley, Kuh, and Welsch.[1] The machine learning methods used are a reflection of the linear regression chapters in our course textbook.[2] The link to the Google CoLab Python Notebook for this analysis is posted at the end of this report. Within it, you will see a series of steps that explore the data set, transform it for linear regression, and then a series of model exploration steps that try to determine the best modeling method in order to avoid over-fitting the training set, and to take into account outlier data.

Discussion of the research design, measurement & statistical methods

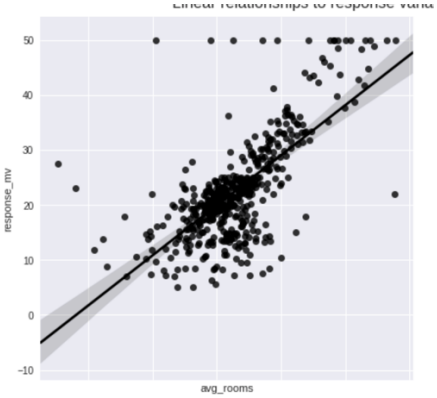
The data set provided is a csv file with 506 rows of data that reflects the estimated value of a home at the time that the research was completed in the 1970's. There are 13 variables in the original data set that give an overview of the neighborhood, home size (in # of rooms), and qualitative aspects of the neighborhood such as air quality, the type of zone the home is in, how close it is to the city center, and other considerations. Some turn out to be more influential than others, such as the # of rooms in the home. There is one categorical variables which is the neighborhood that the home is located in, which will need to be taken into consideration for the linear analysis. At first glance, it also appears that the scale of the variables will be an issue. Some variables are presented as percentages while others are a much different scale, such as the average taxes paid on the home. Within the modeling exploration, we take a look at these factors to determine if they influence the model.

Correlation matrix to response variable 'mv':



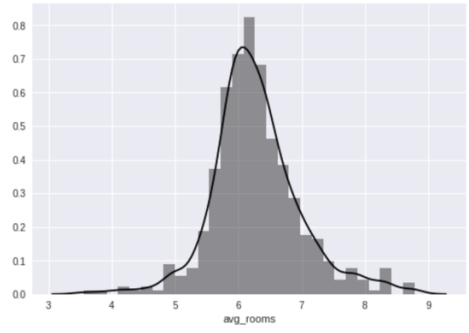
Prior to any data transformation, the response variable ('median' value of the home) seems to be influenced most by its relationship with the variable 'avg_rooms'. This may be a factor of the neighborhoods that have bigger homes also seem to have the highest home value, so there may be some inter-correlation once we split the categorical variable of 'neighborhood. But at first glance, avg_rooms is the most normally distributed variable with the highest correlation.

Linear plot – avg_rooms to response_mv

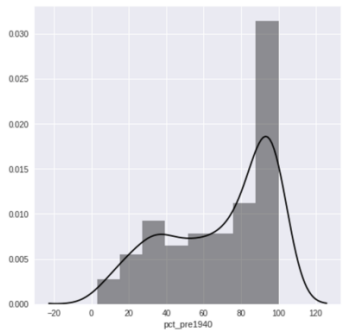


Below are some initial distribution views of some of the more influential variables. As noted above, very few of the variables are normally distributed, which becomes a factor when we consider different scaling options.

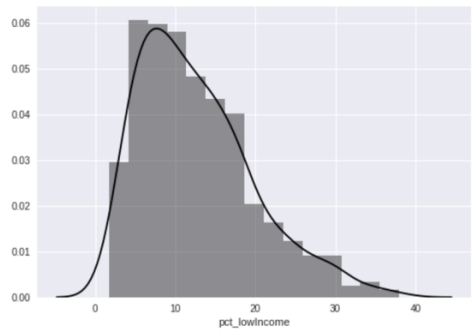
'avg_rooms'
Average # of rooms in the home



'pct_pre1940'
% of homes built before 1940



'pct_lowIncome'
% of population with lower income



[1] David A. Belsley, Edwin Kuh, and Roy E.Welsch. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. Wiley, New York, 1980.

[2] Géron, A. 2017. *Hands-On Machine Learning with Scikit-Learn & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.* Sebastopol, Calif.: O'Reilly. Source code available at https://github.com/ageron/handson-ml

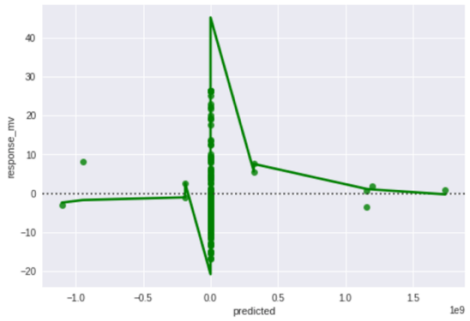
Overview of the programming work

In this analysis, I ran three core modelling methods: standard linear analysis, standard Ridge and Lasso 'regularized' linear models, and finally I experimented with scaling the data and removing outliers. For each model, I used the SciKit Learn function train_test_split' to randomly sort the data and split it according to the recommended sizing after reviewing the learning curve - the RMSE was shown to be minimum in the test data/maximum in the training data at ~70/30 split.

Model 1:	**Model 2:**	**Model 3:**
Standard Linear Regression **results in over fit of training data*	Ridge and Lasso Regression **non-scaled data*	Ridge and Lasso Regression *using MinMax and Robust scaling methods

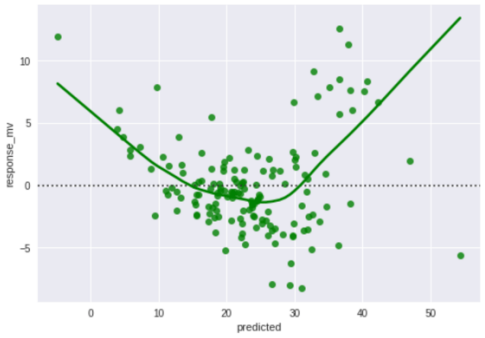
The **first model** was a standard linear regression using sklearn.linear_model. First I tested the model without converting the categorical variable 'neighborhoods' to dummy variables, but it was clear from testing that the dummy conversion improved the model despite adding ~90 new variables.

The training set score in the new model jumped to about .89 from .75, but the test set score dropped very low, which is a classic indication of over-fitting the training data. As shown in the diagram to the right, the residual plot in the predicted response vs the measured response shows a sharp spike along the zero X axis. Although, we will not recommend this model, it was positive indication that the new dummy variables were adding context. The next step would be to investigate 'regularized' linear models using the Ridge and Lasso methods.



Test model residual plot – example of overfit

```
Training set score: 0.89
Test set score: 0.85
<matplotlib.axes._subplots.AxesSubplot at 0x7f36d0fcc7b8>
```



Model 2 – Lasso - residual plot (.85 test score)

In the **second model**, I apply the practice of regularization to avoid over-fitting. In this method, I experimented with both the Ridge and Lasso model techniques. Both turned out to have a good balance between training and test set scores (.89 to .85), so my instinct tells me that the Lasso method is slightly best as it uses fewer variables for the model. With each technique I experimented with the alpha value in order to limit or relax the degrees of freedom within the regularized model. I found that the model performed best when I loosely controlled the number of variables considered (α = .001).

In the **third model**, I explored the effects of scaling on the data set. I used two scaling formats, MinMax and Robust. In the end, the step did not seem to improve model performance. I even experimented with removing a few extreme outliers in the data set, but in the end the performance did not go beyond model 2 the scores in model 2. It was a useful exercise, however to increase my confidence in using the Lasso method without scaling or outlier removal.

Conclusion: based on my analysis, my recommendation would be to use a Lasso regularized model with low alpha, which provides some restriction, but also enough degrees of freedom to achieve a good balance between the training and test data sets.

Note:

A shared version of the python notebook used to create this analysis can be found at Google CoLab:
https://colab.research.google.com/drive/1Mb_7uanUZq5IN5mE43EKgHlLTmHOHmV1