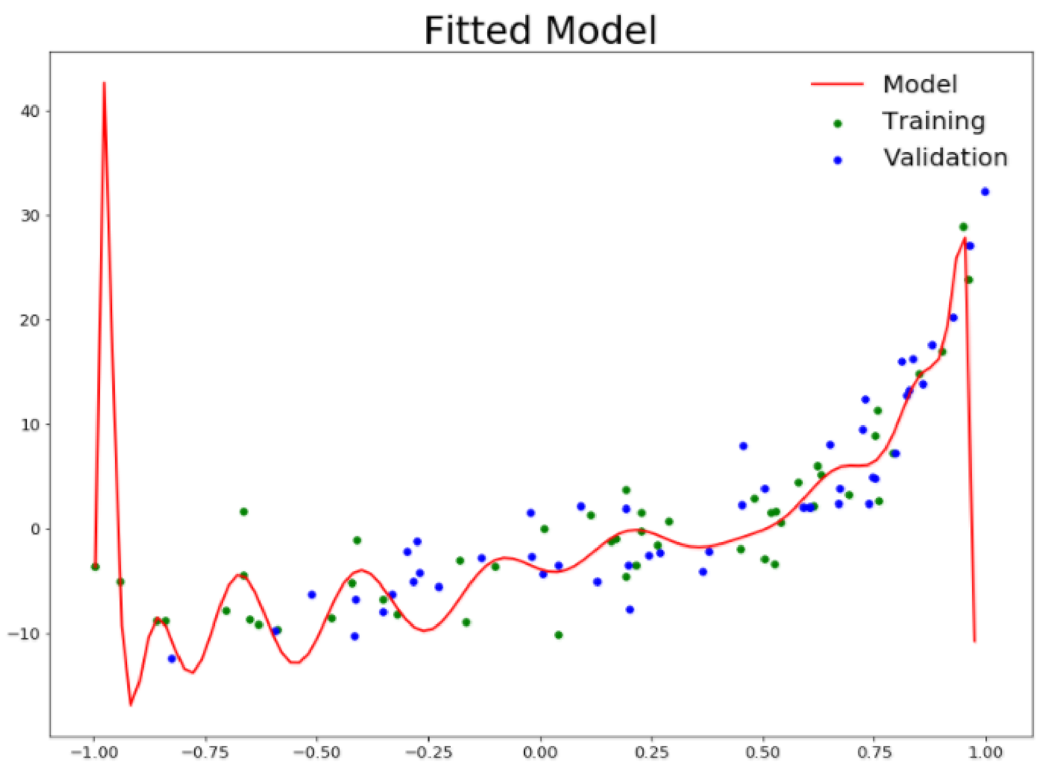
COMP 551 **Camilo Garcia La Rotta**
Assignment #1 **ID #260657037**

Q1.1

The model fitted to a 20-th degree polynomial is clearly over-fitting the data: boundary values are very badly predicted, learned parameters have extremely large weights and the model passes directly over many of the Training set point. This model is accurate in predicting known points but fails to model the underlying relationship of the data.

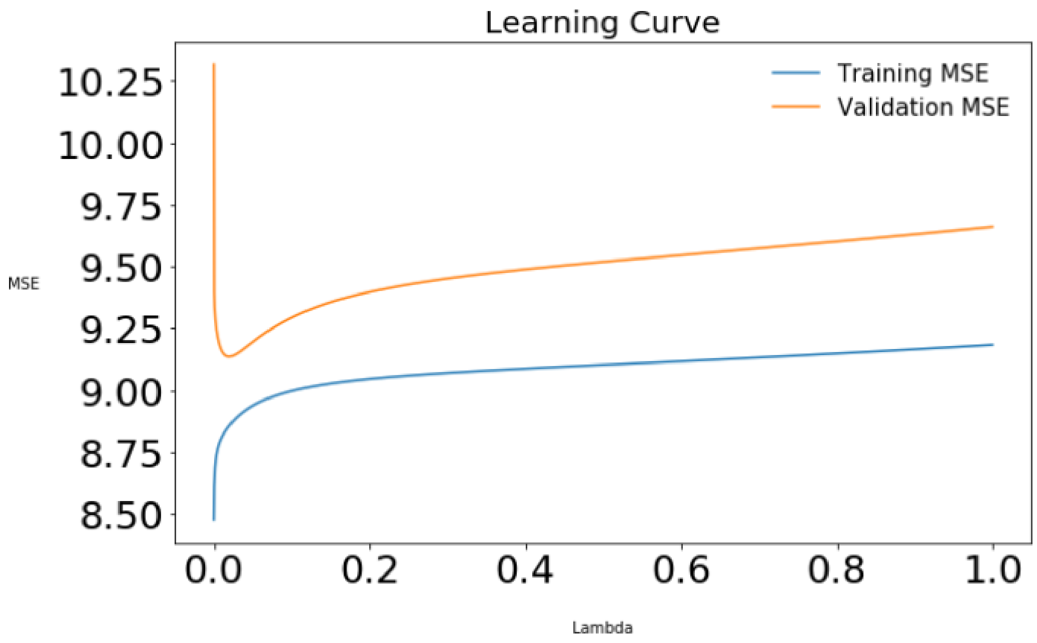
Learned weights:

```
[ -3.91227730e+00 -1.57623166e+01 1.57191679e+02 1.92680243e+03
-7.81623267e+03 -3.60693130e+04 1.28355407e+05 2.92987449e+05
-1.01101380e+06 -1.26812404e+06 4.42929738e+06 3.19914306e+06
-1.15648529e+07 -4.85129424e+06 1.84134772e+07 4.34854405e+06
-1.75279391e+07 -2.11795029e+06 9.16361760e+06 4.30713926e+05
-2.02344972e+06]
```



Q1.2

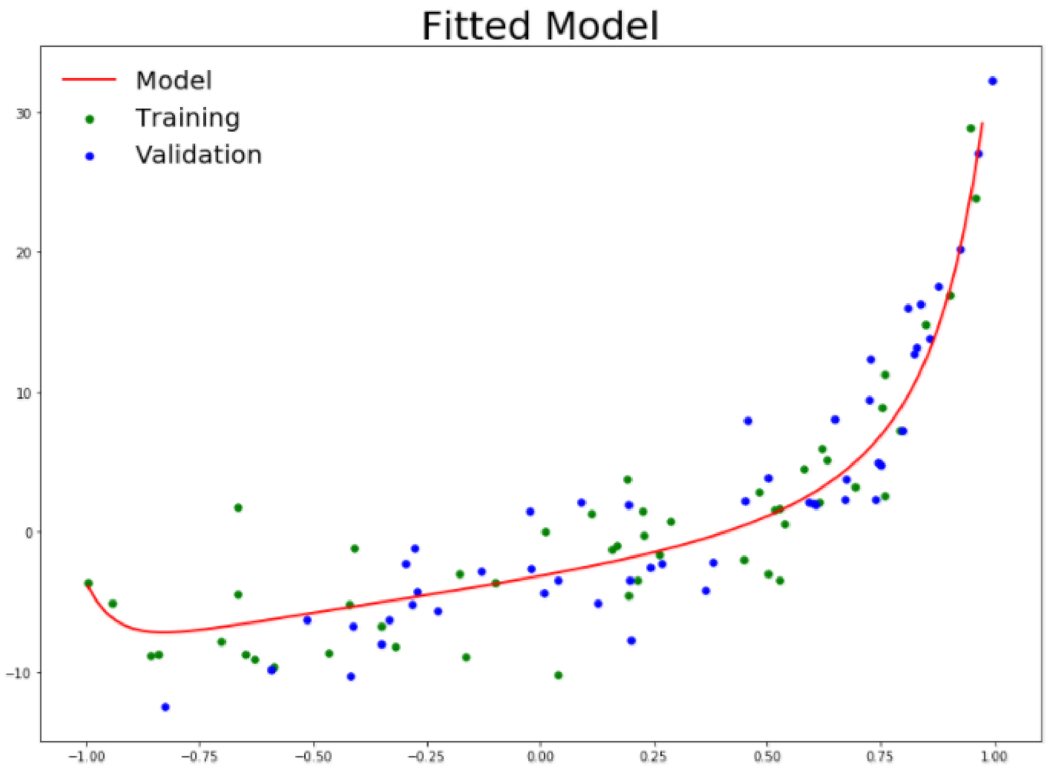
The plotted learning curve demonstrates the impact of regularization in reducing over-fitting and as such reducing the MSE. The optimal value for the regularization coefficient sits very close to zero, where training and validation MSE are closest. $\lambda = 0.0196$ and Test MSE: 9.6606



After fitting the model with the optimal regularization coefficient, it captures very well the relationship of the data. This is shown by the smaller weights given to each parameter, the good boundary-value prediction and the understanding of the general relationship of the samples.

Learned weights:

[-3.12653857 5.89052927 2.32327641 3.22195356 2.73630051 2.67814137
2.51872545 2.16048191 2.23895096 1.70375425 2.00180943 1.32710174
1.80435456 1.02420542 1.63335947 0.78301251 1.48055689 0.59155196
1.34196556 0.43944464 1.21578348]

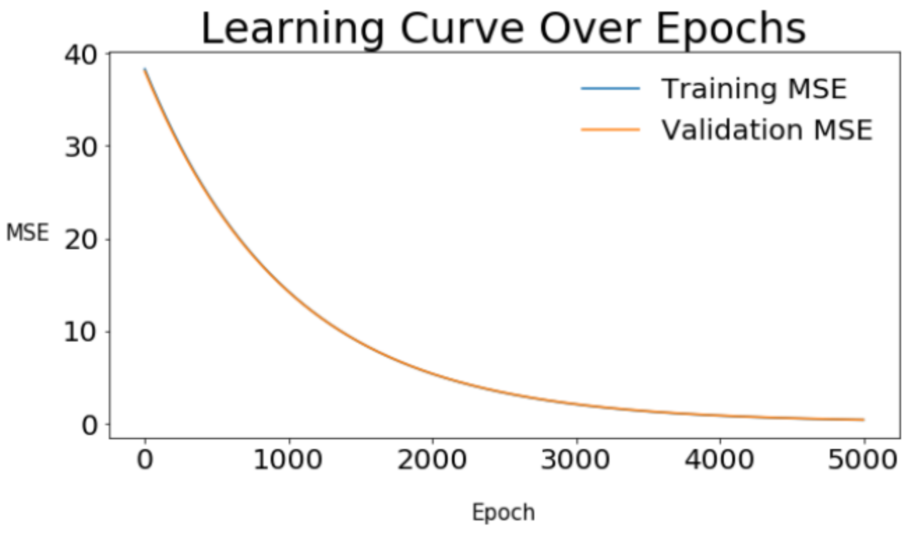


Q1.3

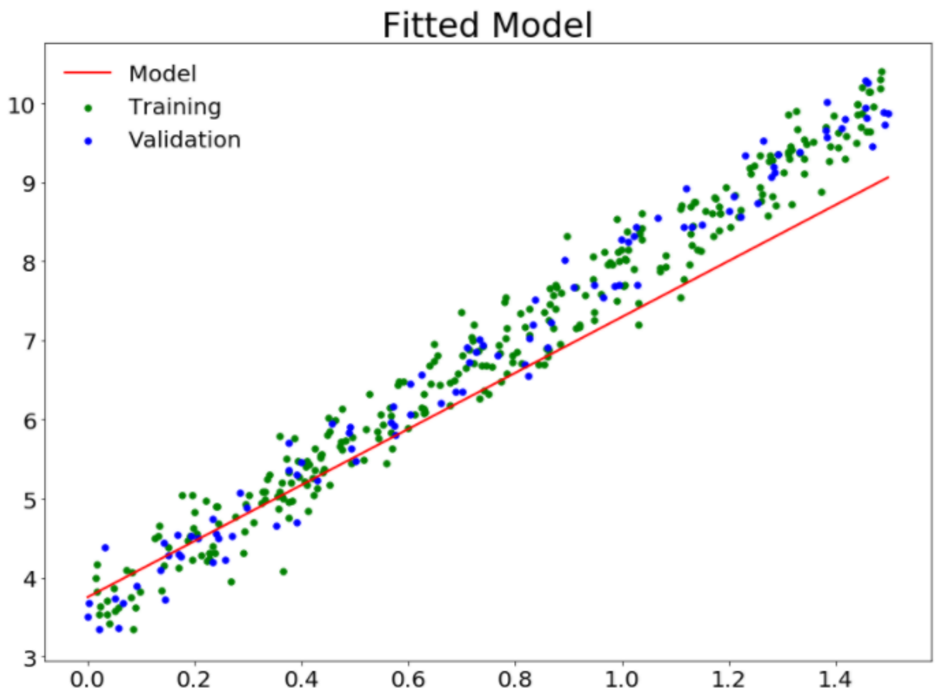
As a lower bound anything below 4-th degree would fail to represent the generally linear middle section of the sample space with the exponential-like growth of the right edge. Counting the number of inflection points in the over-fitted graph a rough estimate of 6-th degree could be hypothesized. To prove such statement it would suffice to run the regression model on different degree polynomials without regularization. The one with the smallest Testing MSE would have high probability of being the underlying function of this problem.

Q2.1

I chose to iterate SDG for 5K epochs or while the difference between the difference between the validation errors of 2 consecutive epochs was less than 10^{-4} (in other words the precision).

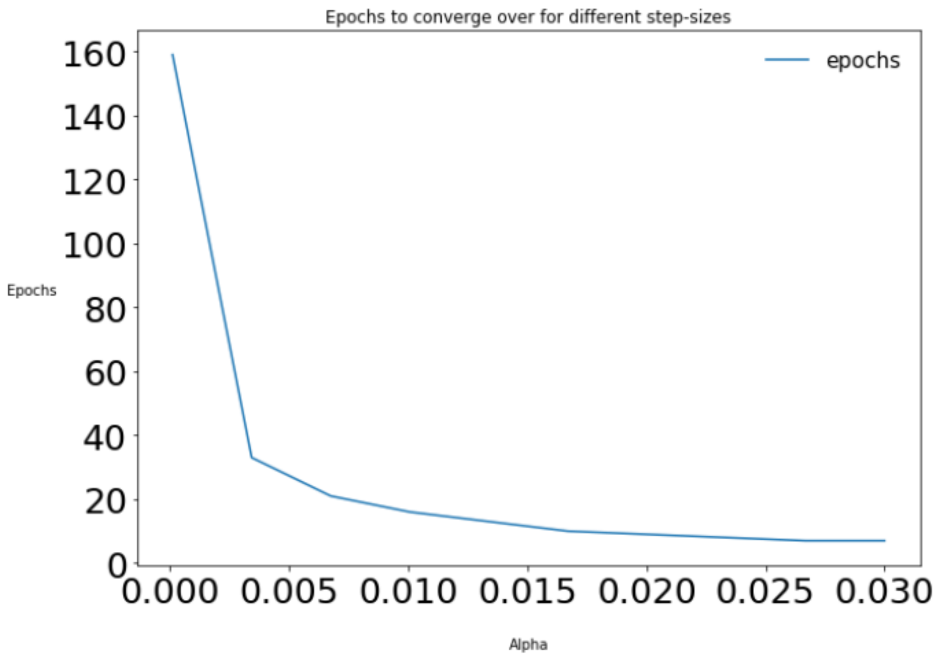


Because of the very small step-size, the fitted model was relatively off from the linear function describing the samples. A higher epoch limit or a smaller precision would be necessary to allow online-SDG to fit the model better.



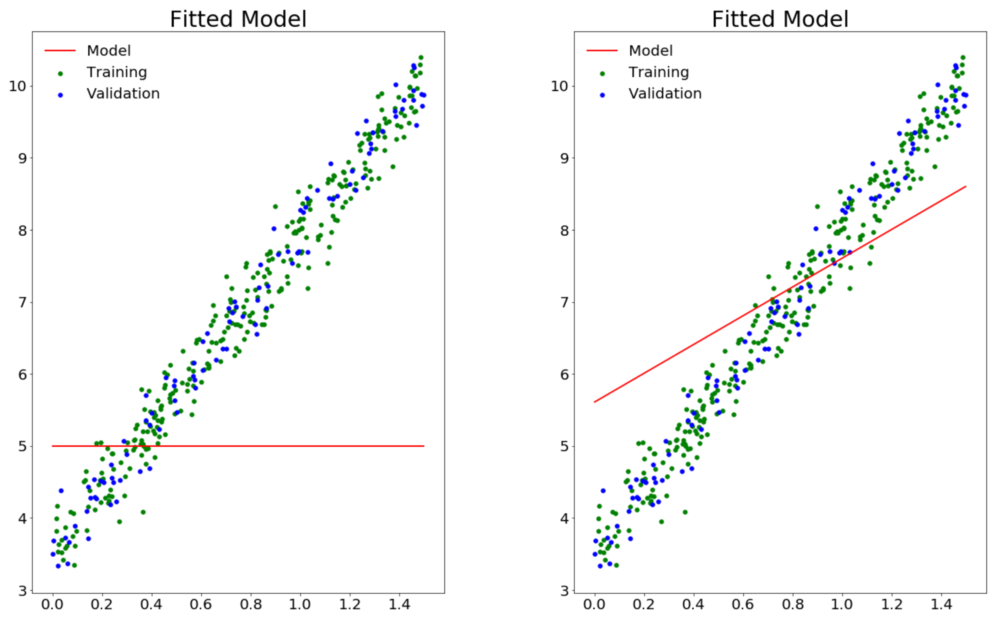
Q2.2

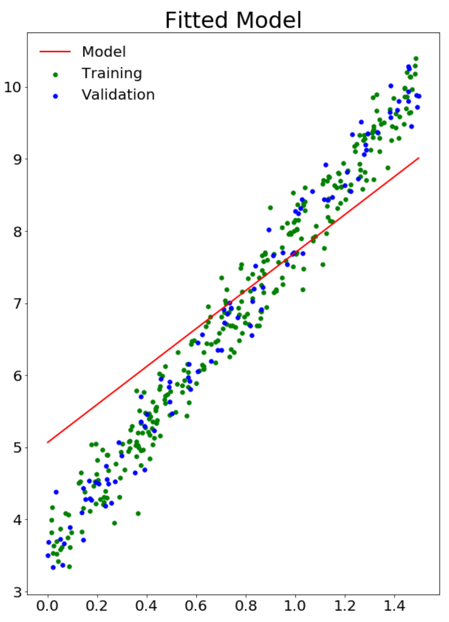
Fitting the model with step-sizes: $\alpha \in [0{,}0.03]$ and plotting how many epochs it took to converge for each one yields the following graph. Interesting to note the "elbow": a very distinct boundary of α after which the epochs to converge stabilizes. I chose $\alpha = 0.03$ for which it took 7 epochs to converge with a Test MSE of 0.08103. Much better than the 5k epochs for $\alpha = 10^{-6}$.

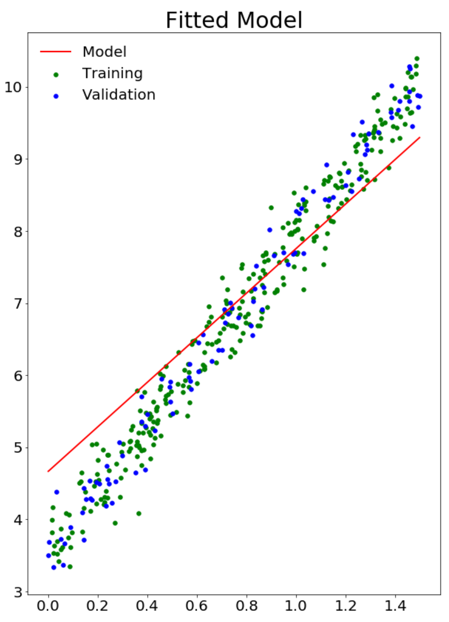


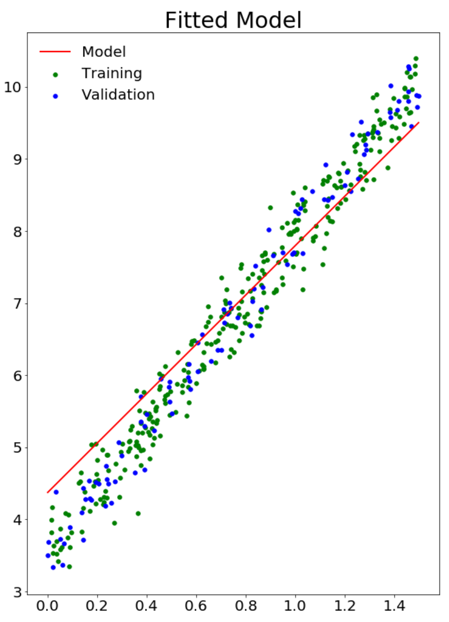
Q2.3

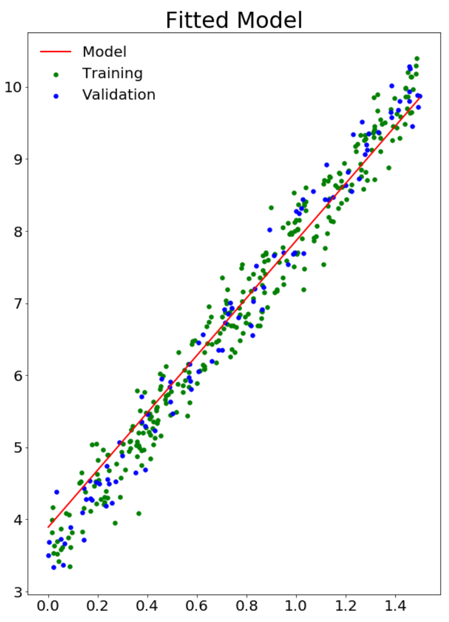
I chose the starting weights [5,0] so as to show how the slope and bias would slowly converge towards the ideal fit. Note that at a greater MSE, the gradient is much larger, hence the magnitude of the change is larger. As the MSE decrements, the magnitude of the steps tend to zero.











Q3.1

Using the mean to fill in missing values has 2 fundamental problems:

- Imagine a all the samples have a value in the range of [0,1] for feature x_1, except for 1 outlier sample with $x_1 = 1000$. the mean will be a value far from the expected values for that feature.

- Another situation in which the mean fails to represent missing values is when the feature represents a set of specific numbers: taking the mean does not guarantee the value obtained is part of that set.

For real valued features I will choose the median of each feature as its default value. The clean data-set will be called **communities.csv**

The learned weights for each fold can be found in **./Datasets/Q3_2_k_fold_weights.txt**

MST for test set 0: 0.020111646207106326
MST for test set 1: 0.02460433001449902
MST for test set 2: 0.0171805553697446
MST for test set 3: 0.016273530588166253
MST for test set 4: 41.27244809777993
AVG MSE for the 5 splits: 8.27012363199189

Q3.2

The learned weights for each fold and for each lambda can be found in
./Datasets/Q3_3_k_fold_weights.txt

Best regularization factor lambda: 0.5

AVG MSE for the lambdas:

0.5 - 0.01667801671794706,
1.0 – 0.016693225965552345,
1.5 – 0.01669263400877757,
3.0 – 0.01678969538225828,
5.0 – 0.016814945390268076

Q3.3

Following the property of Lasso regression by which parameter with very low impact on the prediction are given weights which tend to zero, we could choose the features for which the absolute fitted weight is highest.

In the Jupiter notebook you'll see I chose as a cut-point 2 standard deviations away from the mean for the weight of the features to be dropped. We reduced the columns space from 122 to 116 features and got a negligible improvement of 0.5767%.

If we wanted a more efficient feature selection, other approaches such as: Class separability, Consistency-based or Correlation-based feature selection.