Khizr Ali Pardhan | 301314376 | Feb 6th 2020

CMPT 419 – Assignment 1

1 Probabilistic Modeling and Bayes' Rule

1-1

a)

$$\mathrm{p(m)} = 0.01$$
$$\mathrm{p(m)} = 0.99$$
$$\mathrm{p(t|m)} = 0.95$$
$$\mathrm{p(t|{\sim}m)} = 0.05$$
$$\mathrm{p(t)} = \mathrm{p(t|m)} * \mathrm{p(m)} + \mathrm{p(t|{\sim}m)} * \mathrm{p({\sim}m)}$$
$$\mathrm{p(t)} = 0.95 * 0.01 + 0.05 * 0.99$$
$$\mathrm{P(t)} = 0.059$$

b)

$$\mathrm{p(m|t)} = \frac{\mathrm{p(t|m)} * p(m)}{p(t)}$$
$$\mathrm{p(m|t)} = \frac{0.95 * 0.01}{0.059}$$
$$\mathrm{p(m|t)} = 0.1610$$

1-2

equation from <u>*brilliant.org/wiki/bayes-theorem/*</u>

$$p(r_{t2}|r_{t1}) = \frac{p(r_{t2} \cap r_{t1})}{P(r_{t1})}$$

$$p(r_{t2}|r_{t1}) = \frac{0.25}{0.3} = 0.8\overline{33}$$

1-3

$$p(win) = p(one) + p(three) + p(five)$$

$$p(win) = 0.1 + 0.2 + 0.0 = 0.3$$

$$so, p(lose) = 0.7$$

A fair die is p(odd) = p(even) = 0.5

If you are betting on odd, you should use a fair die, you all be better off

2 Weighted squared Error

$$E\hat{D}(w) = \frac{1}{2}\sum an\{t_n - w^{t}\phi(x_n)\}^2$$

We stall refer to $\phi(x_n)$ $as\ \phi$

$$\frac{d}{dw} = \frac{1}{2}\sum 2a_n\{t_n - w^{t}\phi\}\phi^{t}$$

$$\frac{d}{dw} = \sum a_n\{t_n - w^{t}\phi\}\phi^{t}$$

Distribute matrixes

$$\frac{d}{dw} = \sum\{a_n t_n\phi^{t} - a_n w^{t}\phi\phi^{t}\}$$

Set the gradient to 0 and solve to w:

Please note that the left side of the equation is a column vector of all 0

$$0^t = \sum\{a_n t_n\phi^{t} - a_n w^{t}\phi\phi^{t}\}$$

$$0^t = \sum\{a_n t_n\phi^{t}\} - \sum\{a_n w^{t}\phi\phi^{t}\}$$

$$0^t = \sum\{a_n t_n\phi^{t}\} - w^{t}\sum\{a_n\phi\phi^{t}\}$$

$$w^{t} = \frac{\sum\{a_n t_n\phi^{t}\}}{\sum\{a_n\phi\phi^{t}\}}$$

$$w^{t} = \sum\{a_n\phi\phi^{t}\}^{-1} \times \sum\{a_n t_n\phi^{t}\}$$

Utilize the *Quadratic form*

$$w = (\phi^{t}A\phi)^{-1}\,\phi^{t}\,At$$

3 Training vs. Test Error

3- 1 Suppose we perform unregularized regression on a dataset. Is the **validation error** always higher than the **training error**?

False. Vast majority of the time validation error is higher than training error. However, in very particular cases such as by nature of the data, by chance, or by a sample of a small data set, validation error can indeed be less than the training error.

Suppose the set of validation samples has a large intersection with the set of training samples (reasonable chance in a small dataset) and If the validation set has relatively less variance, this might let the regression learn the validation samples (indirectly) very well, which would result in a validation error less than the training error.

Note: This latter case shows the importance to ensuring the validation set does not have significantly less variance that the training set

3- 2 Suppose we perform **unregularized** regression on a dataset. Is the **training error** with a degree 10 polynomial always lower than or equal to that using a degree 9 polynomial?

True, if we use the method we are taught, a 10th degree polynomial will always yield a lower training error because, 10th degree polynomials can form all 9th degree polynomials.

Note: Some implementations, as taught in Dr. Andrew NG's famous ML course, use a gradient decent type method to compute weights, if the learning rate is too high or a poor optimization algorithm is employed, it is <u>*possible*</u> *for a 9th degree polynomial to outperform a 10th degree polynomial, I would call this an 'edge case by user-negligence'*

3- 3 Suppose we perform both **regularized** and **unregularized** regression on a dataset. Is the testing error with a degree 20 polynomial always lower using **regularized** regression compared to **unregularized** regression?

False, in practice, due to low test set size, or not randomizing (or perhaps even pure chance), it is possible the testing data is 'closer' to the regularized regression better.

Notes: Error metrics are typically distance based, so I deliberately use the work closer

In theory, unregularized regression would yield over fitting, which means a higher testing error than regularized regression, in other words, regularized regression resists overfitting, which is bad for training error.

4 Regression

1. Which country had the lowest child mortality rate in 1990? What was the rate?

Iceland had the lowest child mortality rate in 1990, with a rate of: 6.3

2. Which country had the lowest child mortality rate in 2011? What was the rate?

San Marino had the lowest child mortality rate in 2011, with a rate of: 11.9

3. Some countries are missing some features (see original .xlsx/.csv spreadsheet). How is this handled in the function `assignment1.load_unicef_data()`?

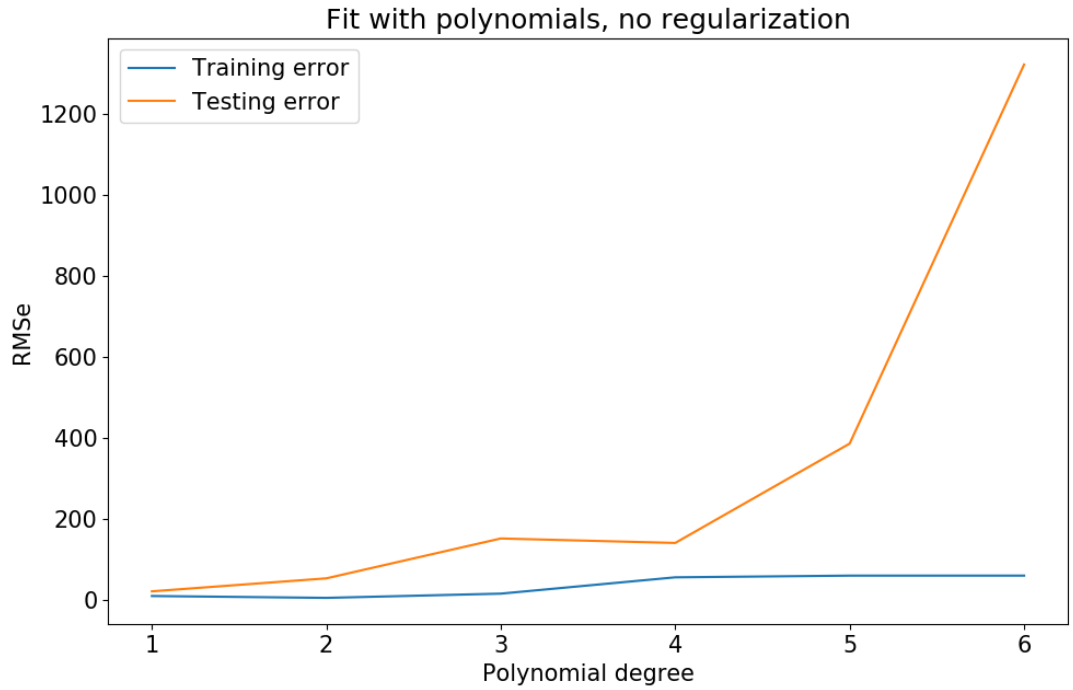
We set the N/A *or* NaN (depending on our vocabulary) values to the NaN_Mean, which is the arithmetic mean ignoring NaNs.

We replace NaN values with the column wise mean, computed while ignoring NaNs

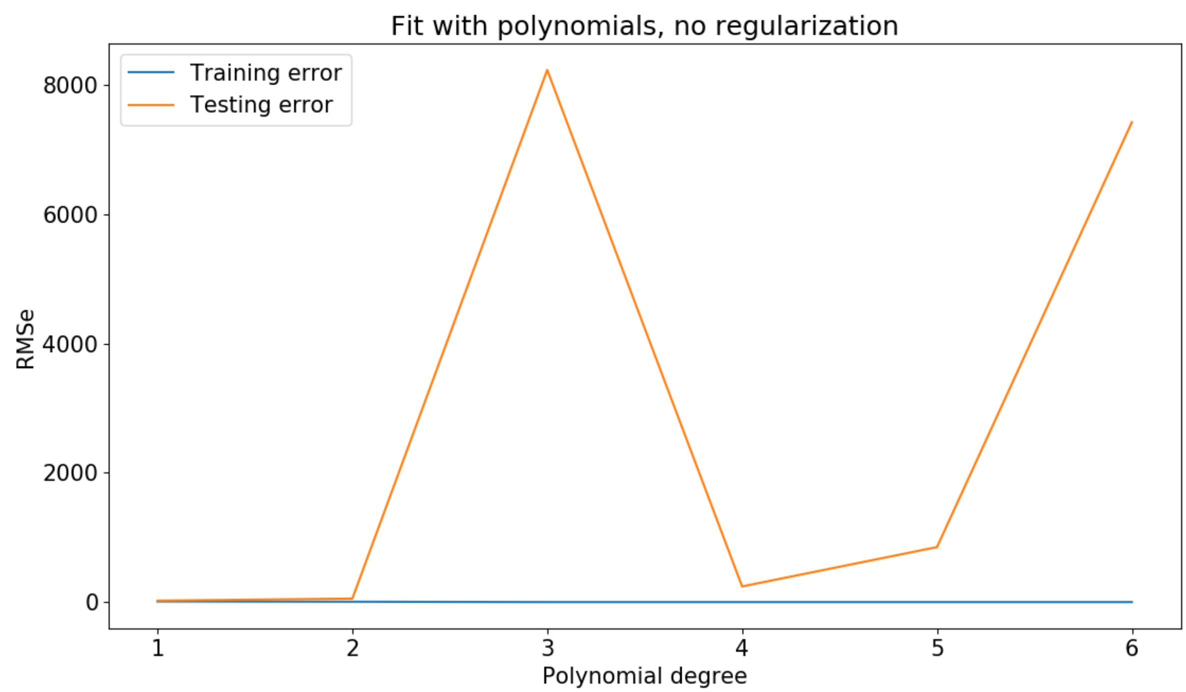
4.2 - 1 Polynomial Regression

Plot of training error and test error (in RMS error) versus polynomial degree.

The testing error is very high with a 6th degree polynomial, indicating overfitting. This can was remedied with regularization



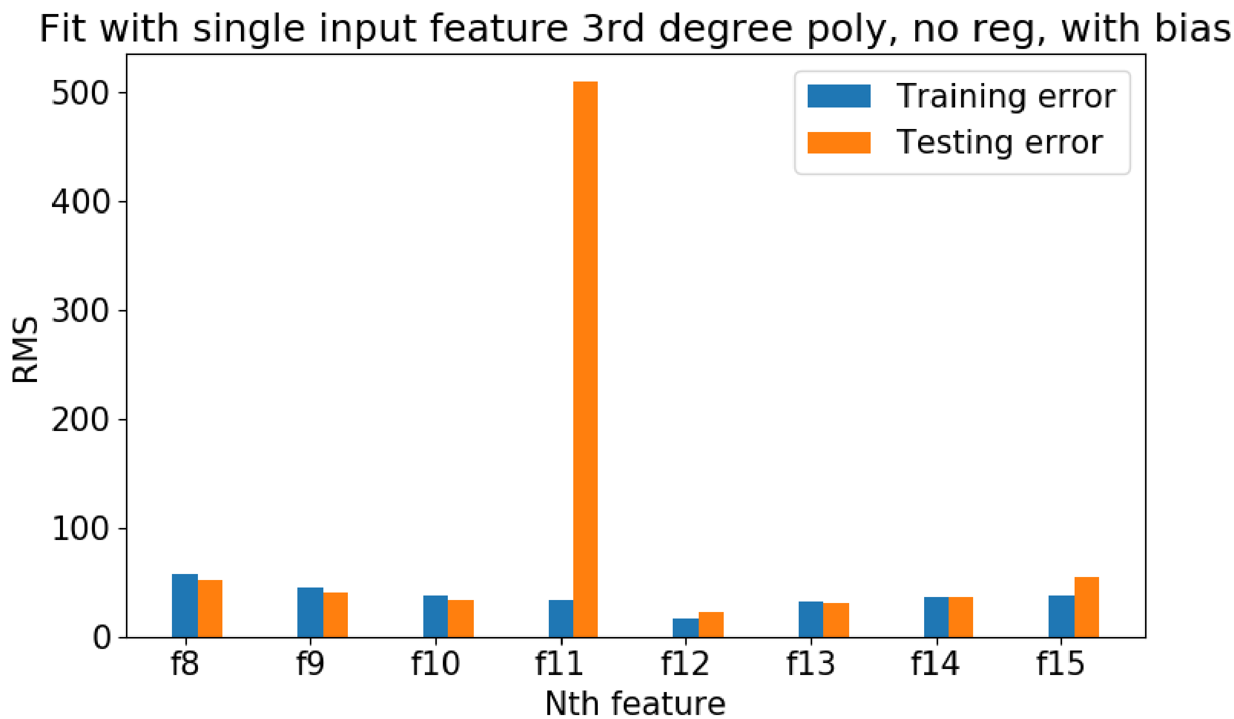
Now if I normalize the input features with `assignment1.normalize data()` the plot looks as follows



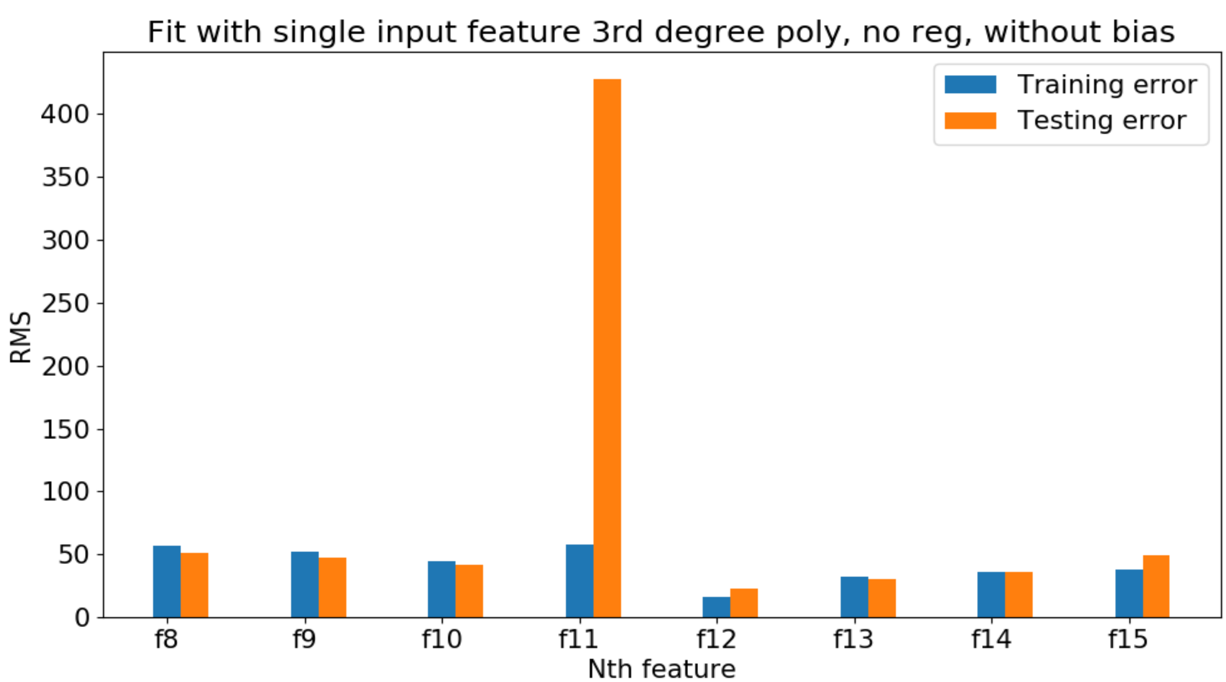
This may be better, but is still far from ideal or 'professional'

4.2 - 2 Polynomial Regression

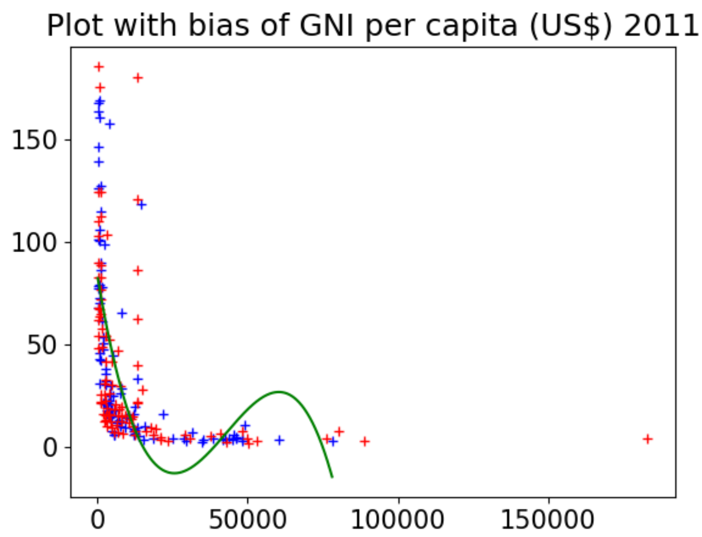
bar chart of 3rd degree polynomial of features 8-15, without normalization, with bias term

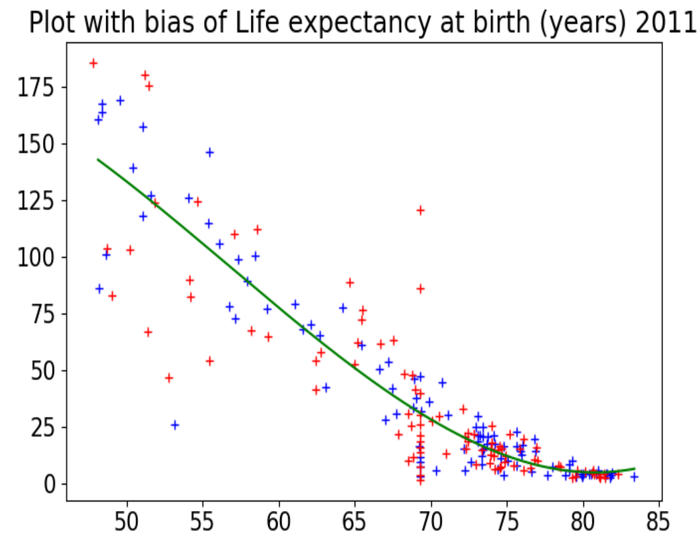


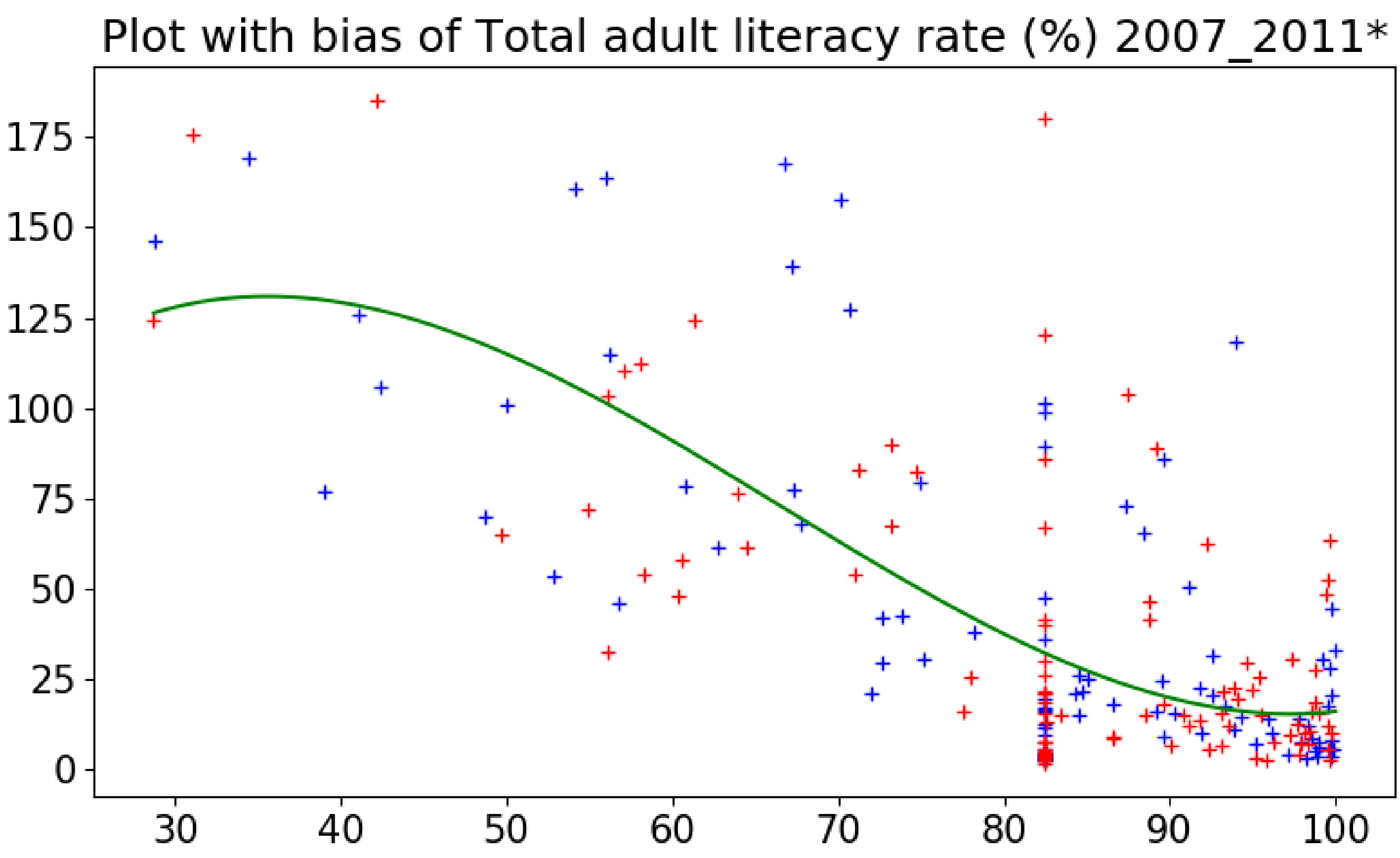
bar chart of 3rd degree polynomial of features 8-15, without normalization, without bias term



Plots of the fits for degree 3 polynomials for features 11 (GNI), 12 (Life expectancy), 13 (literacy) with bias,

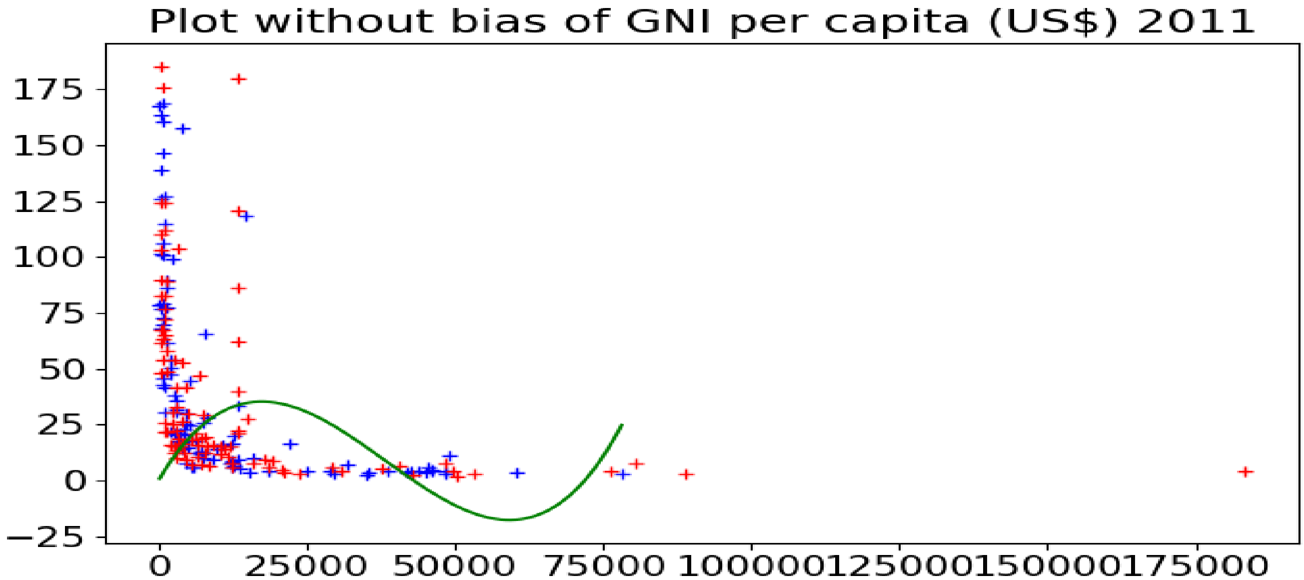


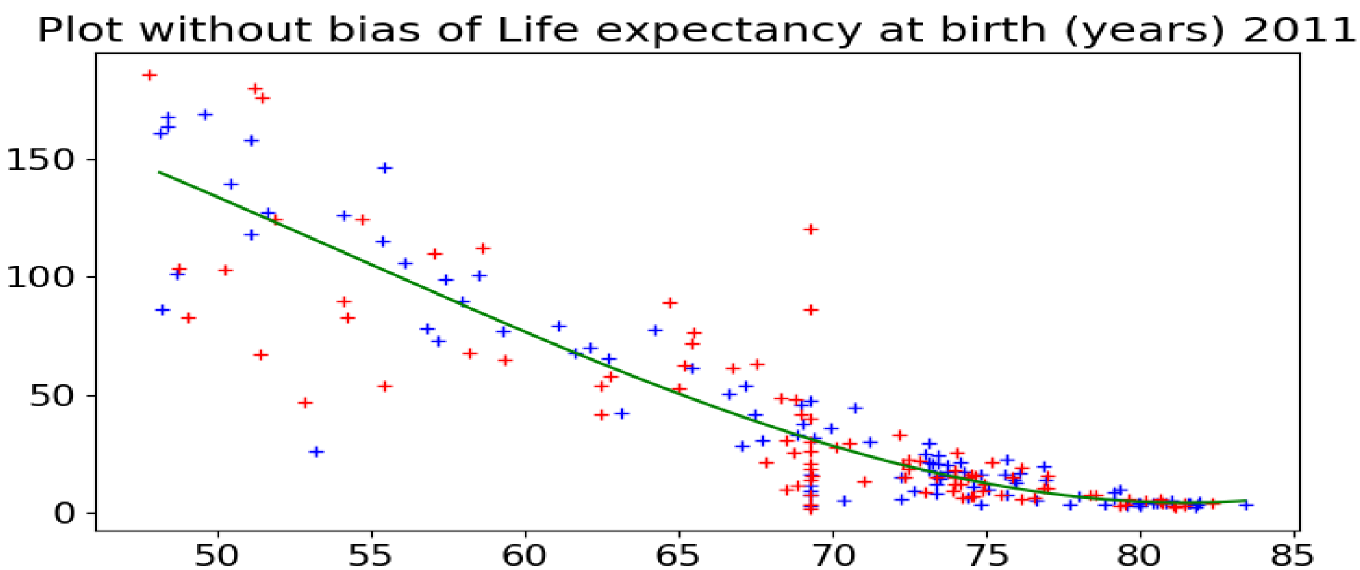


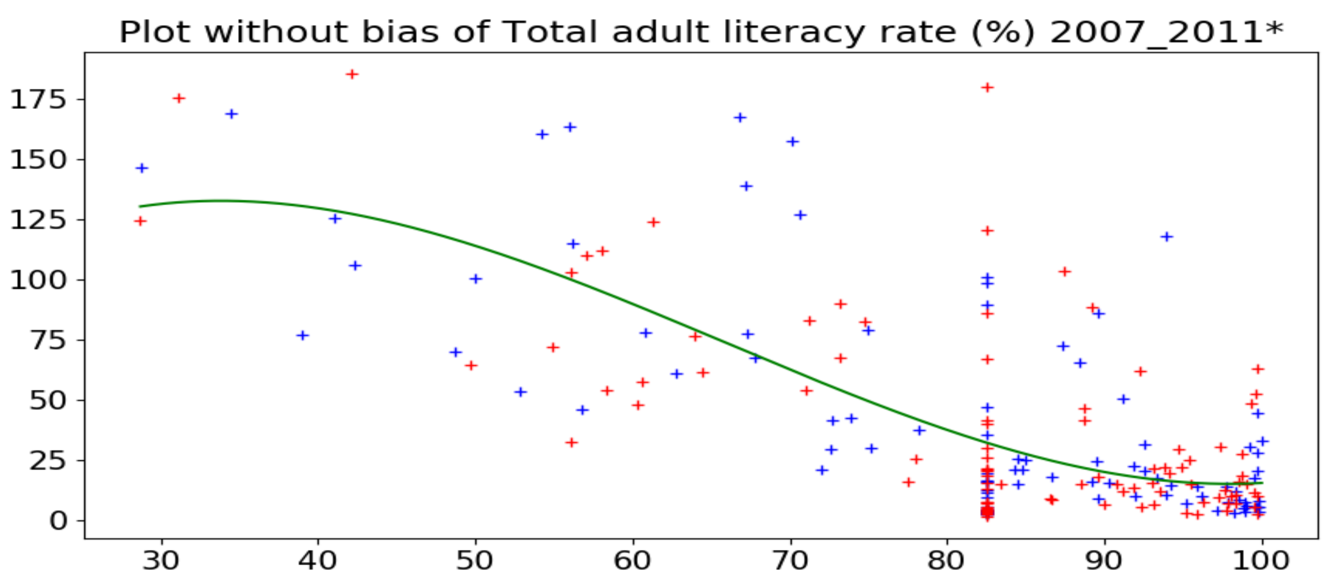


We can see the feature 13 has high variance, this is a attribute of the data

Plots of the fits for degree 3 polynomials for features 11 (GNI), 12 (Life expectancy), 13 (literacy) without bias,

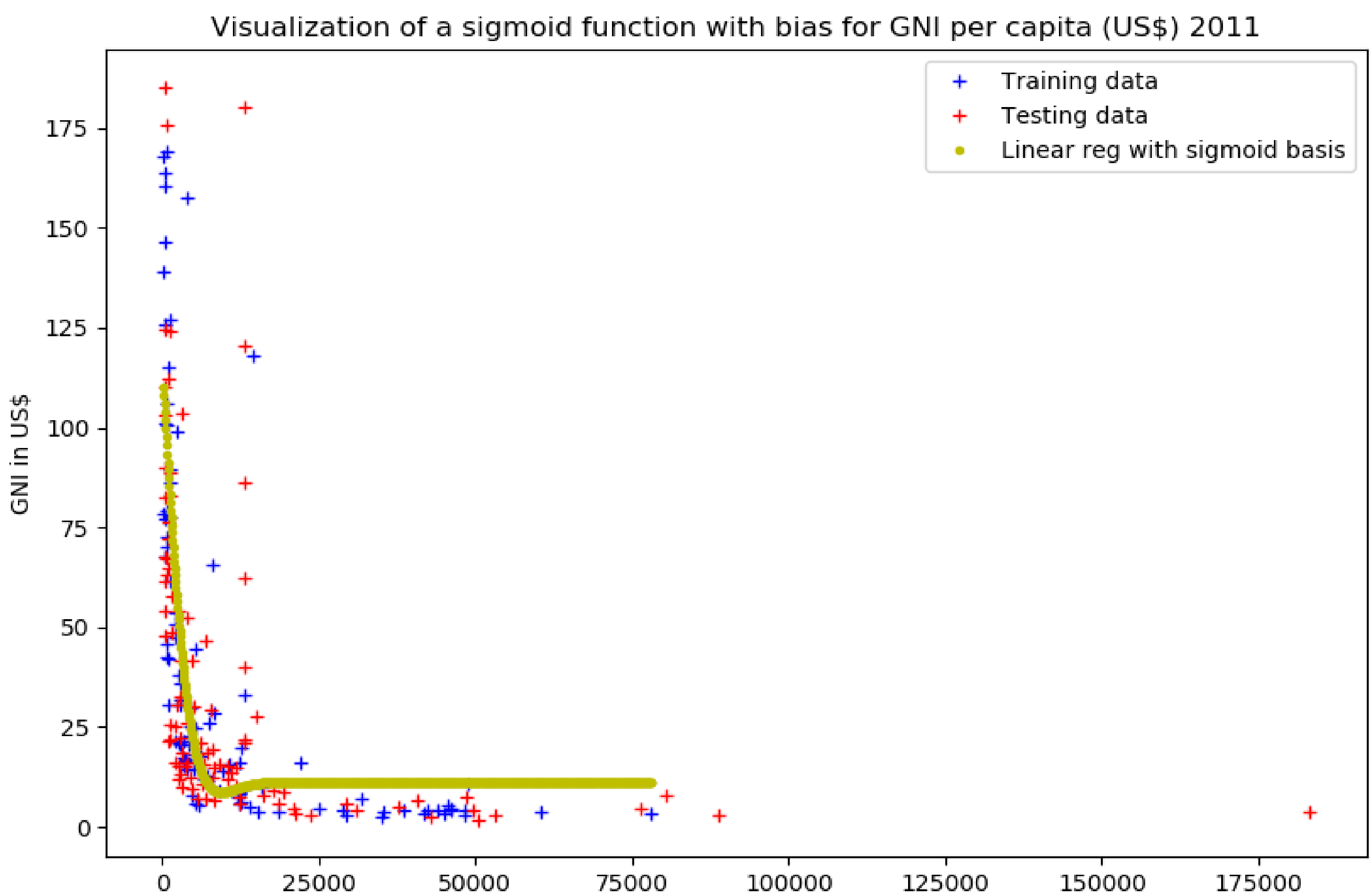




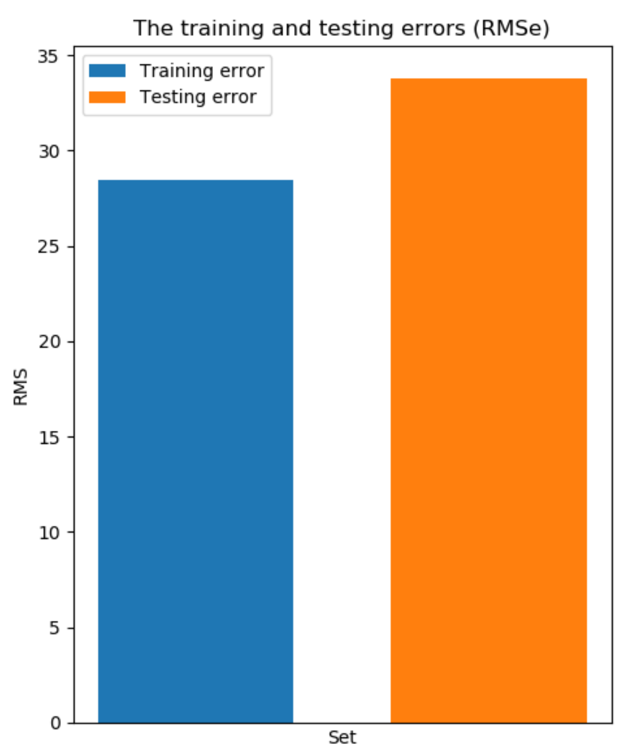


4.3 Sigmoid Basis Functions

Plot of the fit of feature 11, and the training and testing errors



Below is the training (28.46) and testing errors (33.80) for the above regression, shown as a bar graph.



4.4 Regularized Polynomial Regression

Fit of a degree 2 polynomial with L_2-regularized regression, using $\lambda=\{0,.01,.1,1,10,10^2,10^3,10^4\}$

Lambda = 100, givens an average/expected validation error of 21.31