Kunal Chhabria
Student no.: 301392150 Assignment 1 CMPT 726 Machine Learning

Q1. Probabilistic Modeling

In lecture we went over an example of modeling coin tossing – estimating a parameter μ, the probability the coin comes up heads.
Consider instead the problem of modeling a 6-sided die.

1. What is the parameter that explains the behavior of the die in this case (in analogy to the μ for the coin)?

Ans. μ is the parameter that describes the probability of a certain outcome. Hence, μ may be a vector μ =[μ1, μ2, μ3, μ4, μ5, μ6]. So, μi = probability of i being rolled.

2. What is the value of the parameter for a fair die (equal probability of rolling any number)?

Ans. The value of μ for a fair die is μ=[1/6, 1/6, 1/6, 1/6, 1/6, 1/6]. Since, we have a fair die the probability of any number being rolled is equal. Also ∑μ =1 so μi=1/6.

3. What is the value of the parameter for a die that always rolls a 2?

Ans. The value of the parameter for a die that always rolls a 2 is μ=[0,1,0,0,0,0].

4. Specify the domain of the parameter – which settings of the parameter are valid.

Ans. The domain of μ is as follows:

For μi in μ: 0 <= μi <= 1.

The parameter is valid if

For i=1 to 6: ∑μi =1.

Q2. Weighted Squared Error

KUNAL CHHABRIA
Student no. 301392150 Assignment-1 CMPT-726 Machine Learning

Q-2

The weighted sum of squares error function:

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N} \alpha_n (t_n - w^T \phi(x_n))^2$$

Here α_n is the weight for n^{th} data point

To find the optimal weights we take derivative of $E(w)$ and set it to zero

$$\nabla E(w) = -\sum_{n=1}^{N} \alpha (t_n - w^T \phi(x_n))\, \phi(x_n)^T$$

$$= -\sum_{n=1}^{N} \alpha t_n \phi(x_n)^T + \sum_{n=1}^{N} \alpha w^T \phi(x_n) \phi(x_n)^T$$

∇E must ~~be these~~ be set to 0.

Hence $$\sum_{n=1}^{N} \phi(x_n) \alpha t_n^T = \sum_{n=1}^{N} w^T (\phi(x_n)^T \alpha \phi(x_n))$$

$$\Phi \alpha t^T = w^T \Phi \alpha \Phi^T$$

where -

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_n(x_n) \\ \phi_0(x_2) & \cdots & & \vdots \\ \vdots & & & \\ \phi_0(x_n) & \cdots & & \phi_n(x_n) \end{pmatrix}$$

Taking Transpose both sides.

$$\Phi^T \alpha t = w \Phi^T \alpha \Phi$$

$$\Rightarrow \boxed{w = (\Phi^T \alpha \Phi)^{-1} \Phi \alpha t}$$

$$\alpha = \begin{pmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & & \\ \vdots & & \ddots & \\ 0 & & & \alpha_n \end{pmatrix}$$

Q3. Training vs. Test Error

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

Ans. No, it is not guaranteed that validation error will always be higher than training error. Although the validation error is generally higher than training error, there can be some cases where this is not true. Some possible reasons are as follow:

- It is possible that due to a specific train/validation split the train data may end up having outliers which can significantly increase error.
- The validation set might get chosen in such a way that the data points happen to be very close to the learned curve.

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? Explain.

Ans. Yes, the training error for a degree 10 polynomial will always be lower than or equal to a degree 9 polynomial because a degree 10 polynomial has more degrees of freedom and it also contains all the degree 9 polynomials as special cases. (reference.: PRML page 8)

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? Explain.

Ans. No, the testing error with a degree 20 polynomial is not guaranteed to be lower using regularized regression as compared to unregularized regression. While, it is often the case in real world data but there can be cases when it is not true. For example, it is possible to have data with many steep curves. In such a case a high degree polynomial would provide a good fit without regularization and test error would actually increase if regularization was applied because regularization penalizes larger coefficients.

Q4. Basis Function Dependent Regularization

KUNAL CHHABRIA CMPT - 726
Student no- 301392150 Assignment- 1 Machine Learning

Q-4

L_1 error function $E = \frac{1}{2}\sum_{n=1}^{N} (t_n - w^T\phi(x_n))^2 + \frac{\lambda}{2}|w|$

L_2 error function, $E = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2 + \frac{\lambda}{2}|w|^2$

In the given problem we have a different λ_n tradeoff parameter λ_n for each w_n.
J_1 is the set of indices of basis functions that have L_1 regularization and J_2 is the set of indices that have L_2 regularization

Hence

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2 + \frac{1}{2}\sum_{i \in J_1}\lambda_i|w_i| + \frac{1}{2}\sum_{j \in J_2}\lambda_j|w_j|^2$$

$$\nabla E(w) = \sum_{n=1}^{N}(w^T\phi(x_n) - t_n)\phi(x_n)^T + \frac{1}{2}\sum_{i \in J_1}\lambda_i + \cancel{\frac{2}{2}}\sum_{j \in J_2}\lambda_j w_j$$

Ans:

$$\boxed{\Delta E(w) = \sum_{n=1}^{N}(w^T\phi(x_n) - t_n)\phi(x_n)^T + \frac{1}{2}\sum_{i \in J_1}\lambda_i + \sum_{j \in J_2}\lambda_j w_j}$$

Q5. Regression

Q5.1 Getting started

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

Ans: The country which had the highest child mortality rate in 1990 was Niger and the rate was 313.7.

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

Ans. The country which had the highest child mortality rate in 2011 was Sierra Leone and the rate was 185.3

3. Some countries are missing some features (see original .xlsx/.csv spreadsheet). How is this

handled in the function assignment1.load_unicef_data()?

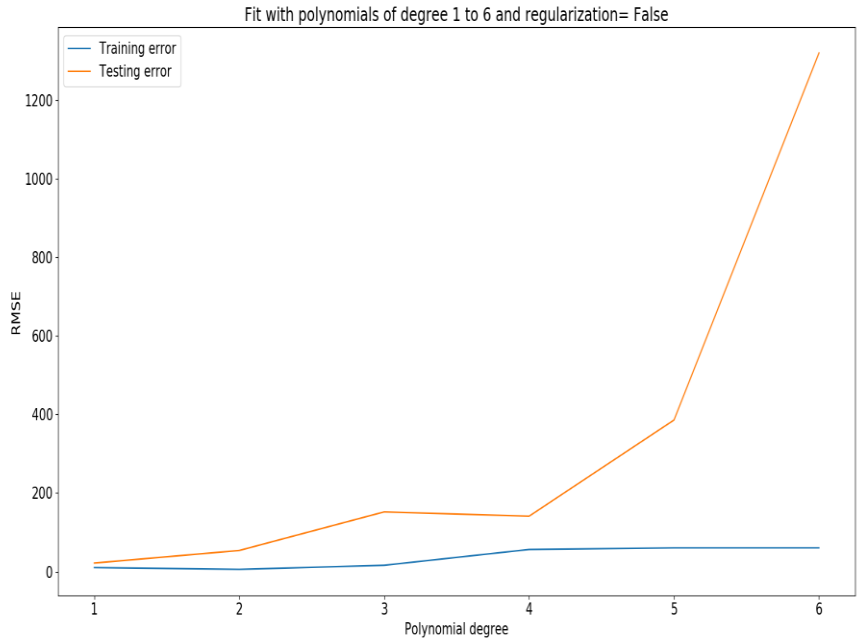
Ans. In the function load_unicef_data() we are replacing the missing values with mean of their respective column.

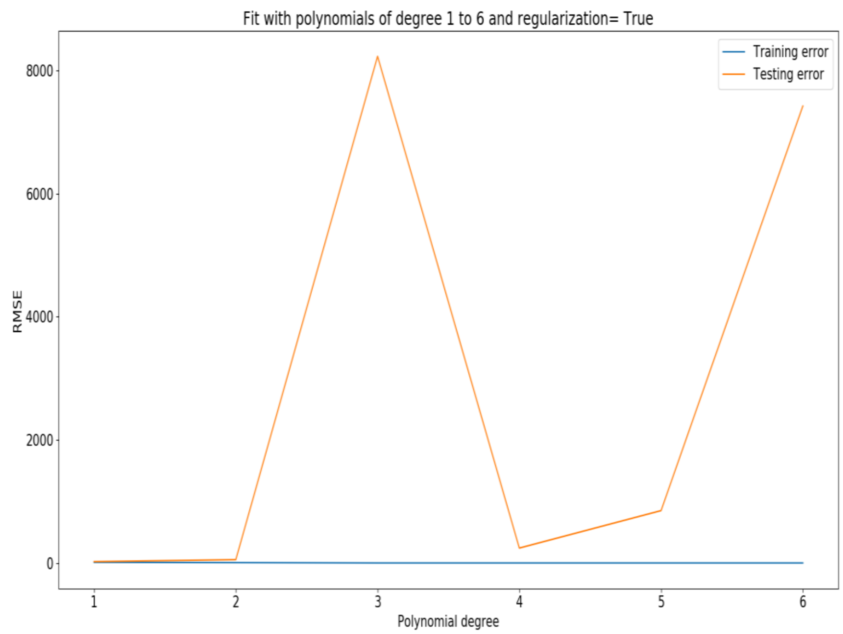
(Q 5.2 on next page)

Q5.2 Polynomial Regression

1. The plots of polynomial regression with degree 1 to 6 before and after regularization are as follows.

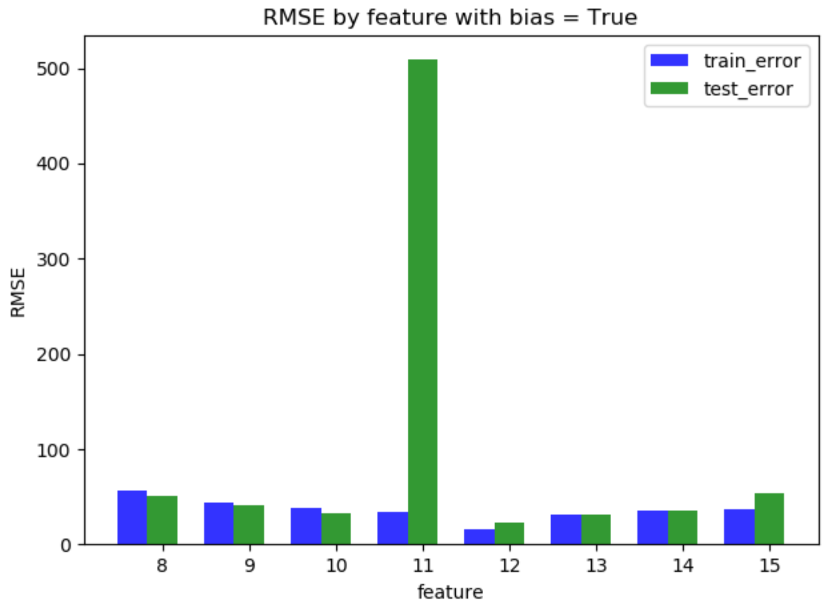
It can be seen from the plots that as the degree of polynomial increases, we tend to overfit the data, which can be seen by sharp increase in testing error. In case of unregularized data both training and testing error increase because the data will have large values and it won't be stable without normalization. However, after normalization the training error approaches 0 but the testing error becomes worse as degree increases.

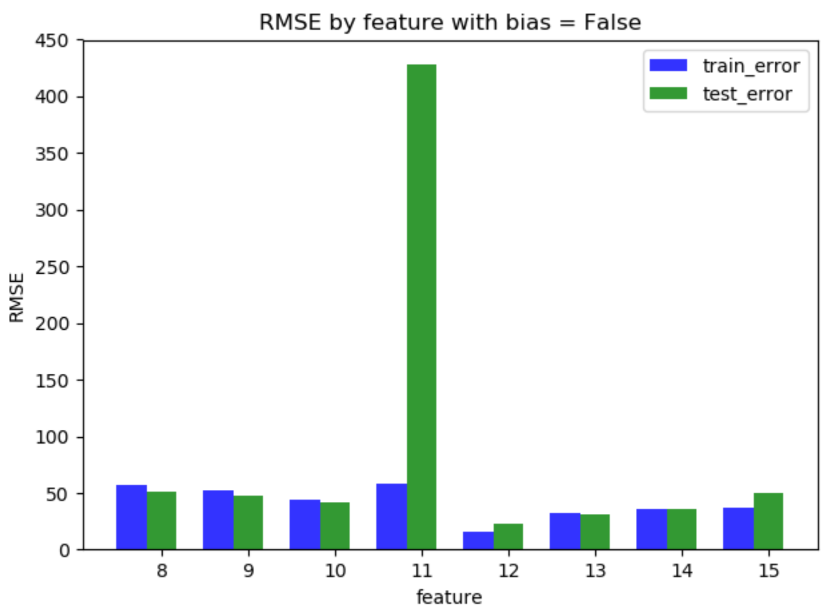




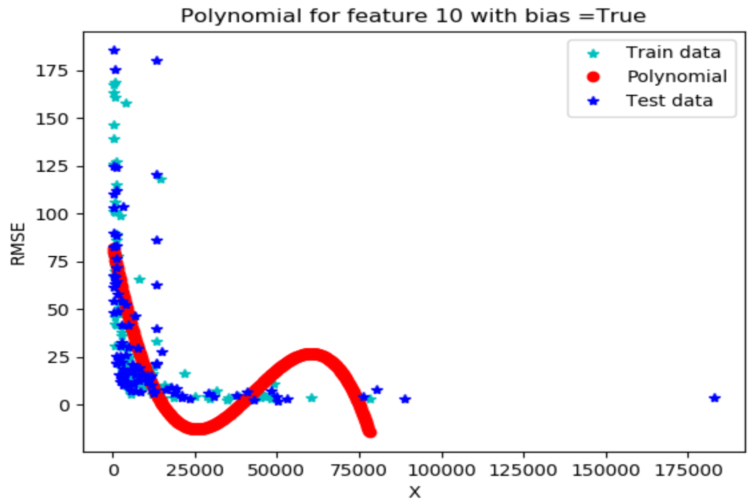
2. Output for polynomial_regression_1d.py

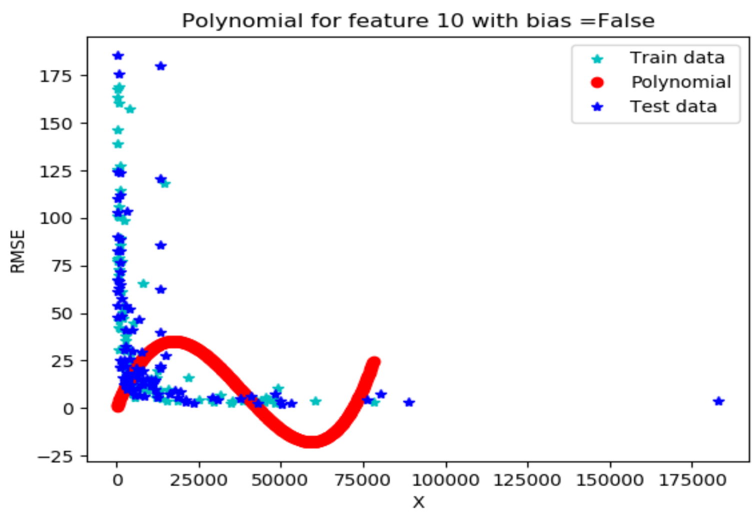
The plot of training error and test error (in RMS error) for each of the 8 features with and without a bias term is as follows:

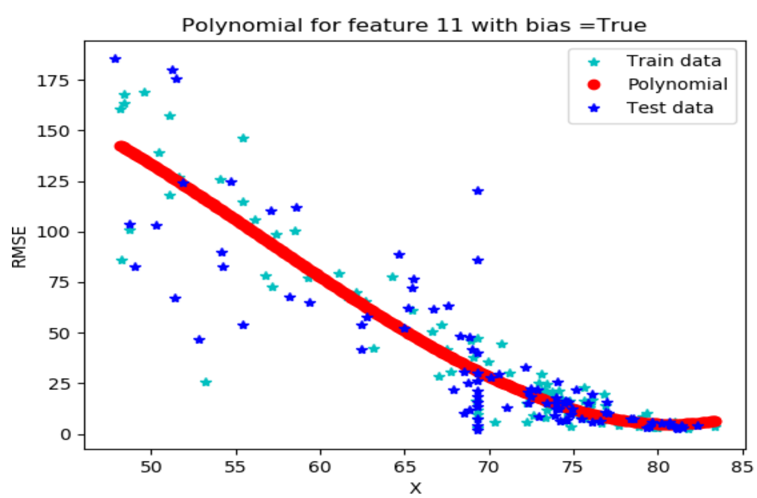


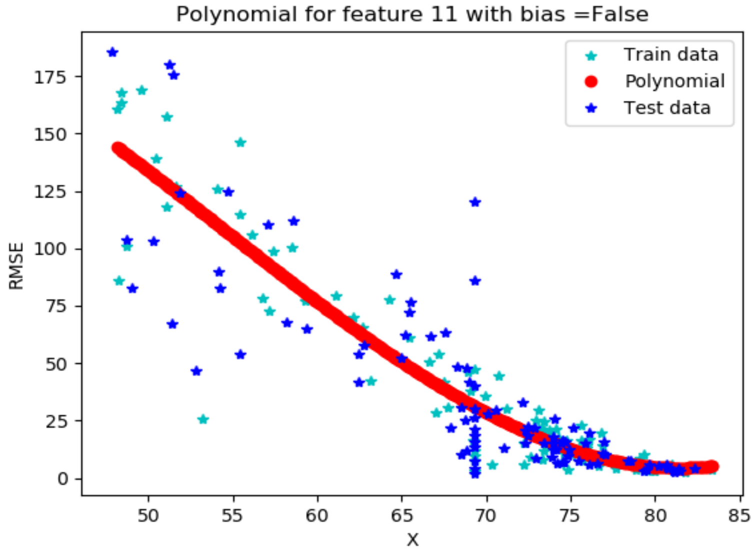


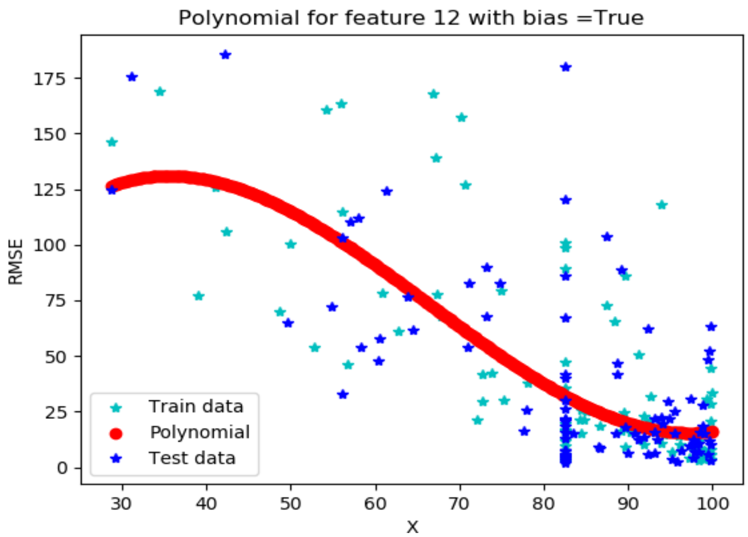
The plots of the fits for degree 3 polynomials for features 11 (GNI), 12 (Life expectancy), 13 (literacy) are as follows:

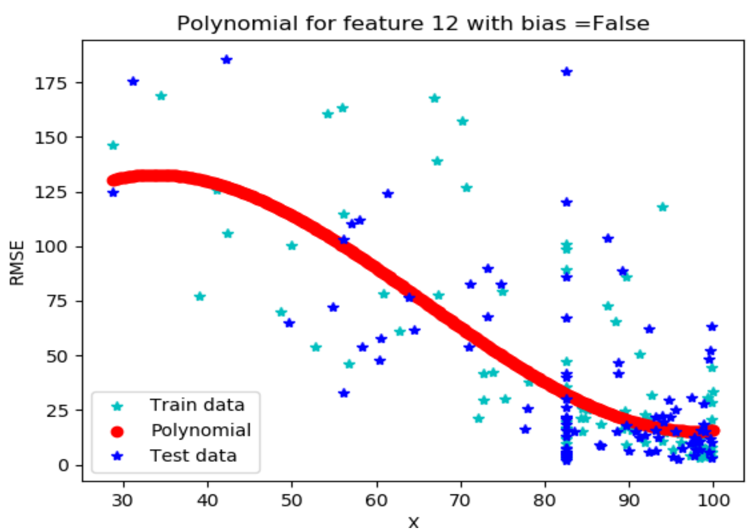






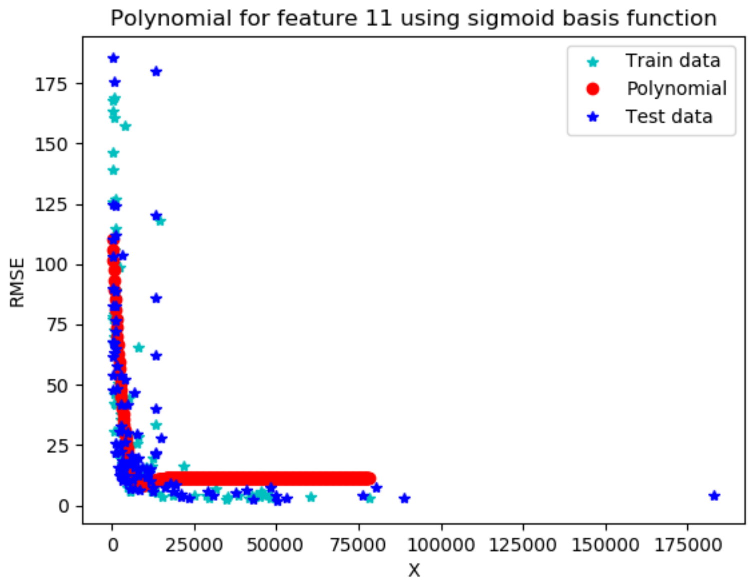

Polynomial for feature 11 with bias =False
Train data
Polynomial
Test data
RMSE
X
0
25
50
75
100
125
150
175
50
55
60
65
70
75
80
85


Polynomial for feature 12 with bias =True
Train data
Polynomial
Test data
RMSE
X
0
25
50
75
100
125
150
175
30
40
50
60
70
80
90
100


Polynomial for feature 12 with bias =False
Train data
Polynomial
Test data
RMSE
X
0
25
50
75
100
125
150
175
30
40
50
60
70
80
90
100

Q5.3 Sigmoid Basis Function

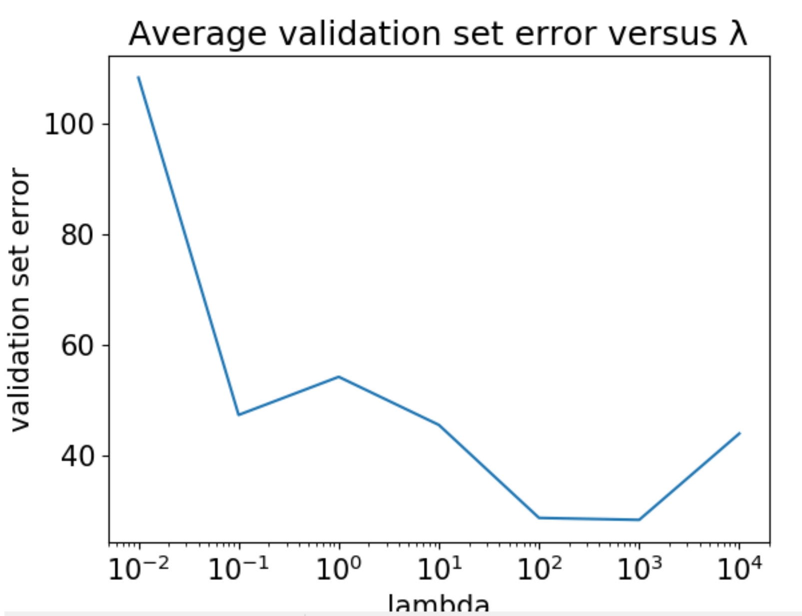
The plot of the fit for feature 11 (GNI) using sigmoid basis function is as follows.



Training and testing error using feature 11 for this regression model is as follows:
Training error = 28.458
Testing error = 33.807

Q5.4 Regularized Polynomial Regression

The plot of lambda vs Validation set error using L2 regularized regression is as follows,



Among the given lambda values, I would choose Lambda = 1000 as it has lowest validation error of 28.47. The Validation error for Lambda= 0 is 134.08.