CMPT 726 ASSIGNMENT 1

1) PROBABILISTIC MODELLING

1.1. What are the parameters μ of this distribution? (See PRML Appendix B.)

Let us assume that the probabilities of P(NDP), P(Liberals), P(GreenParty) are the three parameters and taking μ as the parameters such as μ_{NDP} , $\mu_{Liberals}$, μ_{Green}.If we denote the probability of $x_k = 1$ by the parameter μ_k, where k {NDP, Liberals, Green} then the distribution of x is given

$$p(\mathbf{x}|\boldsymbol{\mu}) = \prod_{k=1}^{K} \mu_k^{x_k}$$

where $\mu = (\mu_1, \ldots, \mu_K)^T$, and the parameters μ_k are constrained to satisfy μ_k 0
and μk = 1, because they represent probabilities.

1.2. What would be the value of the parameters μ for an election where the outcome is an equal chance of each party winning?

Considering the outcome is an equal chance of each party mentioned above is winning,

$\mu_{NDP} = \mu_{Liberals} = \mu_{Green} = 1/3$

1.3. What would be the value of the parameters μ for an election that is completely "rigged"? E.g. the party currently in power is definitely going to win.

If suppose Green party has completely rigged election, the value of parameters
μNDP =0, μLiberals =0 , μGreen = 1.

P(Green, Liberals, NDP) = [1,0,0]

1.4. Specify a prior P(μ) that encodes a belief that one party has rigged the election, but there is an equal chance that it is any of the 3 parties.

The probability that a party wins is 1. Consider when one of the party has completely rigged the election, then using dirac delta function P(μ) = 1/3

1.5. Assume I see a set of polls where the NDP has the largest share of the vote in each poll. What would be my posterior probability on μ?

We know that Posterior probability ∝ Likelihood * Prior. So based on this, I perform the below:-

The distribution matching to the prior of multinomial distribution is dirichlet distribution, so using this we get $\alpha_1, \alpha_2, \alpha_3$ parameters for μ_1, μ_2, μ_3 (NDP, Liberals, Green Party)

$$P(\mu|D,\alpha) = \frac{\Gamma(\alpha_1+\alpha_2+\alpha_3)}{\Gamma\alpha_1 + \Gamma\alpha_2 + \Gamma\alpha_3} \quad \mu_1^{\alpha_1-1}, \mu_2^{\alpha_2-1}, \mu_3^{\alpha_3-1}$$

given $\boxed{\alpha > 0}$

$0 \leq \mu_K \leq 1$ and α denotes the parameters of the distribution

1.6. Suppose if party i is elected, they will set university tuition to be ti dollars. Write down an equation for the expected amount tuition will be, given a prior P(μ)

Expected tuition amount is

$$\sum_{i=1}^{3} \mu_i t_i$$

In other words, the expected amount is μ1t1 + μ2t2 + μ3t3

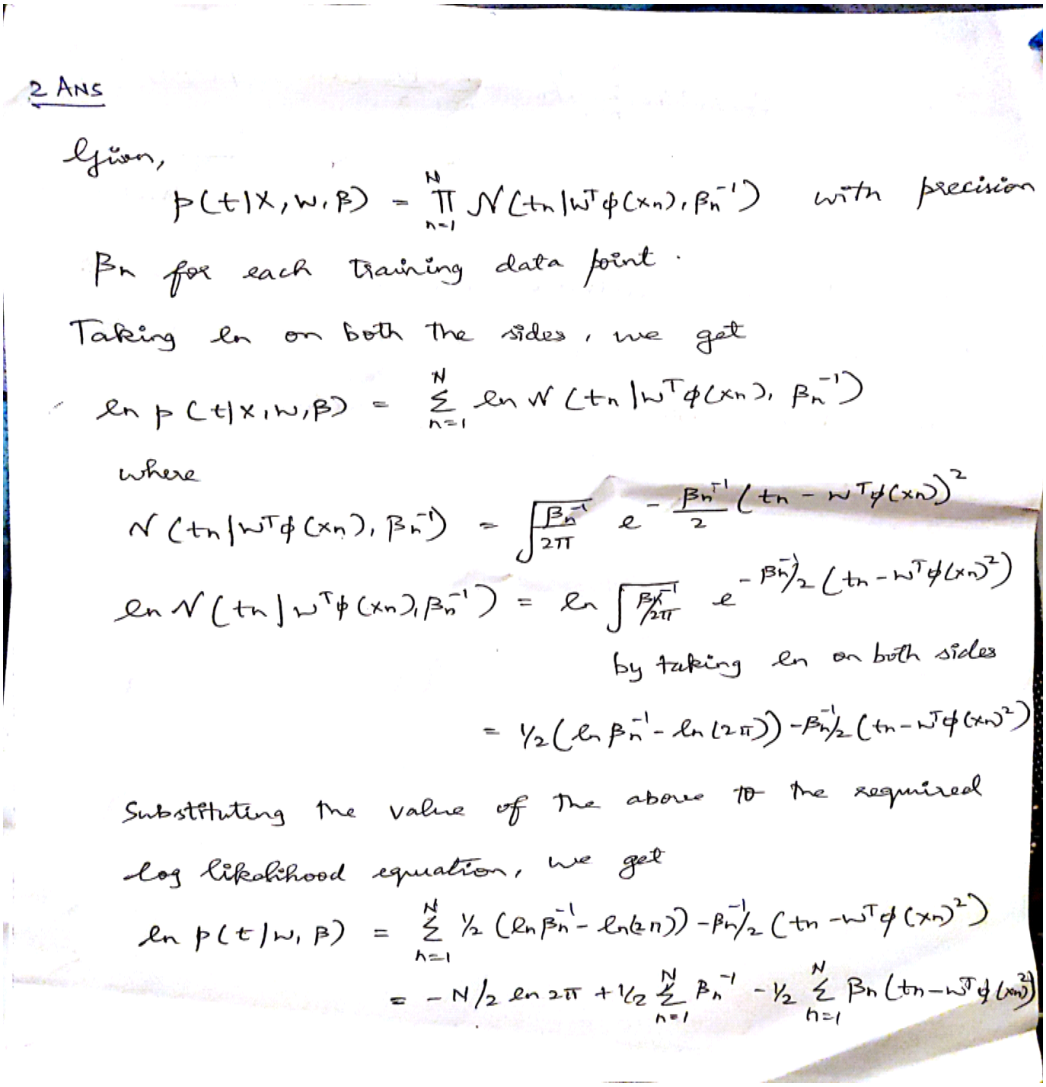
2) PRECISION PER DATAPOINT

The log likelihood for regression (Eqn. 3.10 in PRML) assumes an additive Gaussian noise with the same value of variance at every training data point. In some instances, we may wish to have a different value of noise variance at each training data point. This could arise if we have precision estimates at each training data point. Consider the likelihood function with different precision values corresponding to each data point:

$$\boldsymbol{p}(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathbf{T}}\phi(\mathbf{x_n}), \beta_n^{-1})$$

Derive the relation between the log likelihood function p(t|w, β) and the sum-of-squares error function in this scenario.

The derivations is as follows:-



2 Ans

Given,

$p(t|X, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|w^T\phi(x_n), \beta_n^{-1})$ with precision β_n for each training data point.

Taking ln on both the sides, we get

$$\ln p(t|X, w, \beta) = \sum_{n=1}^{N} \ln \mathcal{N}(t_n|w^T\phi(x_n), \beta_n^{-1})$$

where

$$\mathcal{N}(t_n|w^T\phi(x_n), \beta_n^{-1}) = \sqrt{\frac{\beta_n^{-1}}{2\pi}} e^{-\frac{\beta_n^{-1}}{2}(t_n - w^T\phi(x_n))^2}$$

$$\ln \mathcal{N}(t_n|w^T\phi(x_n), \beta_n^{-1}) = \ln \sqrt{\frac{\beta_n^{-1}}{2\pi}} e^{-\beta_n^{-1}/2 (t_n - w^T\phi(x_n)^2)}$$

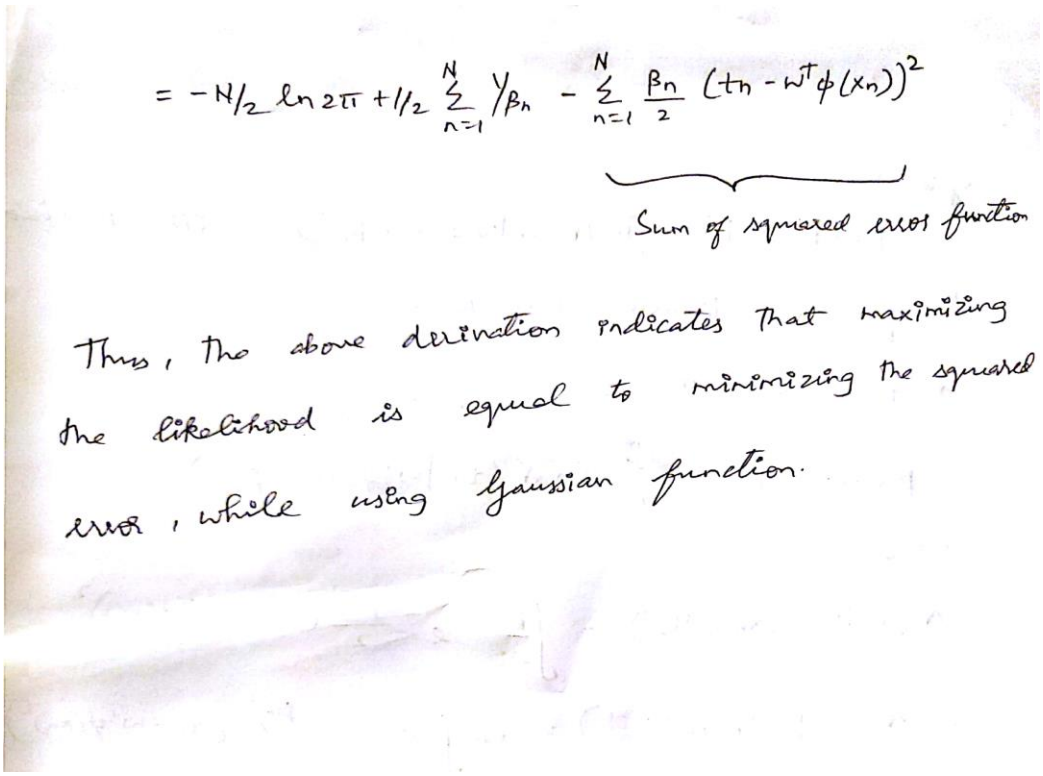
by taking ln on both sides

$$= 1/2(\ln \beta_n^{-1} - \ln(2\pi)) - \beta_n^{-1}/2 (t_n - w^T\phi(x_n)^2)$$

Substituting the value of the above to the required log likelihood equation, we get

$$\ln p(t|w, \beta) = \sum_{n=1}^{N} 1/2 (\ln \beta_n^{-1} - \ln(2\pi)) - \beta_n^{-1}/2 (t_n - w^T\phi(x_n)^2)$$

$$= -N/2 \ln 2\pi + 1/2 \sum_{n=1}^{N} \beta_n^{-1} - 1/2 \sum_{n=1}^{N} \beta_n (t_n - w^T\phi(x_n))^2$$



3) TRAINING VS TEST ERROR

For the questions below, assume that error means RMS (root mean squared error).
1. Suppose we perform unregularized regression on a dataset. Is the validation error always higher than the training error? Explain.

$$E_{RMS} = \sqrt{\frac{2E(w)}{N}} \quad \text{(RMS formula)}$$

During the unregularized regression, there are no penalties and the above The above Erms error function is used. Assuming the datasets are divided into validation and training errors, it is not always true that validation error is higher than training error. There are some exceptional cases too. While it is usually true that training error is lower than validation error, the case in which our validation is better than the training when the dataset is specific to validation data that will give a result that is validation error lesser than training error. **So the answer is No, the validation error need not be always higher than the training error**

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? Explain.

$$E_{RMS} = \sqrt{\frac{2\,E(w)}{N}} \quad \text{(RMS formula)}$$

For the unregularized regression, we use the above error function. By using this error function, which is root mean square where the errors are squared before taking the mean, so the RMSE gives a relatively high weight to large errors. A 10^{th} degree polynomial also holds the 9 degree polynomials..At the least, if our 10^{th} degree coefficient is a zero , then training error of 10^{th} degree polynomial is the same as that of the test error using a 9^{th} degree polynomial .**So the answer is yes, the training error with a degree 10 polynomial always lower than or equal to that using a degree 9 polynomial**

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? Explain.

We can't generalize or be so sure regarding this because performing regularization will yield less training error than otherwise. When applying the regularization using the below formula we reach the below conclusion.

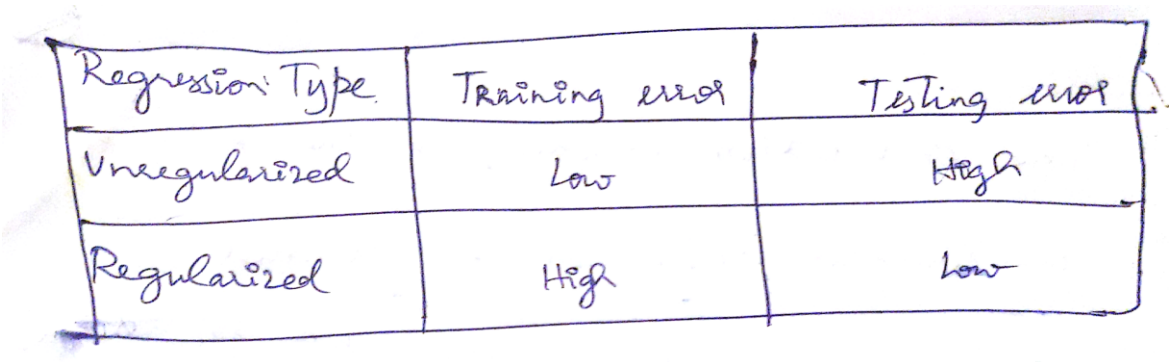
$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2}\|w\|^2$$

where $\|w\|^2 = w_0^2 + w_1^2 + w_2^2 + w_3^2 + \cdots + w_N^2$

For a degree 20 polynomial,

$$y(x, w) = w_0 + w_1 x^1 + \cdots + w_{20} x^{20}$$

Regularization will help overfitting & improve the model. So, the regularized & unregularized regression takes as



Regression Type	Training error	Testing error
Unregularized	Low	High
Regularized	High	Low

So, the answer is no, the testing error with a degree 20 polynomial need not be always lower using regularized regression compared to unregularized regression

4 BASIS FUNCTION DEPENDENT REGULARIZATION

In lecture we saw a regularization technique applied to linear regression where all weights in the regression model are regularized in the same fashion (like L1, or L2), and with a common value for λ. Consider the case where for each weight wn, we have a different tradeoff parameter λn, and a choice from among one of L1 or L2 regularizer. Derive the formula of the gradient for the regularized squared error loss function in this scenario.b

$$\nabla E(w) = ?$$

(Hint: Let J1 be the set of indices of basis functions whose weights have L1 regularization, and J2 be the set of indices of basis functions whose weights have L2 regularization.)

Using both the Lasso regularization(L1) and Ridge regularization (L2) by means of standard regularization pattern (derived below) derives the below equation

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n,w) - t_n\}^2 + \frac{1}{2}\sum_{n\in J_1}\lambda_n|w_n| + \frac{1}{2}\sum_{n\in J_2}\lambda_n w_n^2$$

Sum of squared error function

Regularization L1

L2 Regularization

$$E(w) = \frac{1}{2}\sum_{n=1}^{N}(w^T\phi(x_n) - t_n)\phi(x_n) + \frac{1}{2}\sum_{n\in J_1}\lambda_n\frac{w_n}{|w_n|} + \frac{1}{2}\sum_{n\in J_2}\lambda_n w_n .$$

because $y(x_n,w) = w^T\phi(x_n)$

Therefore, by further solving the above duration, we get

$$\nabla E(w) = \sum_{n=1}^{N}(w^T\phi(x_n) - t_n)\phi(x_n) + \frac{1}{2}\sum_{n\in J_1}\lambda_n\frac{w_n}{|w_n|} + \sum_{n\in J_2}\lambda_n w_n$$

5) REGRESSION

5.1 Run the provided script polynomial regression.py to load the dataset and names of countries / features. Answer the following questions about the data. Include these answers in your report.

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

The country with the highest child mortality rate in 1990 - **NIGER** and Rate is **313.7** using the max() and the where function in the dataset given.

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

Country with highest mortality rate for children in 2011**: Sierra Leone**

Rate for highest mortality among children in 2011: **185.3**

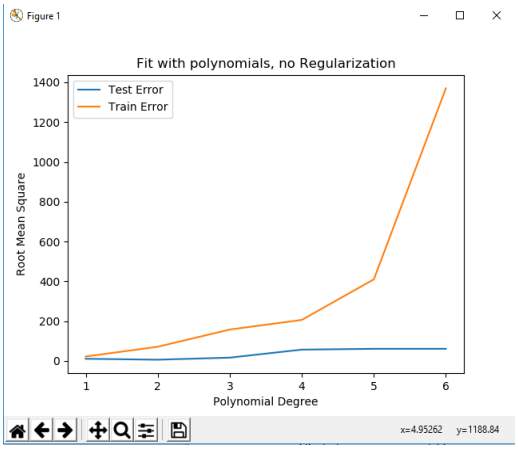
5..3. Some countries are missing some features (see original .xlsx/.csv spreadsheet). How is this handled in the function assignment1.load unicef data()?

The data's are loaded using the data=pd.read_csv(fname, na_values = ' ') method. The na_values argument identifies the missing values. The missing values are replaced with the mean of all the values in the provided column. This is done via the function np.nanmean which finds the mean for the expected missing values. Then the mean are replaced in the expected places using the function

```
inds= np.where (np.isnan(values))
values[inds]=np.take[mean_vals,inds[1])
```

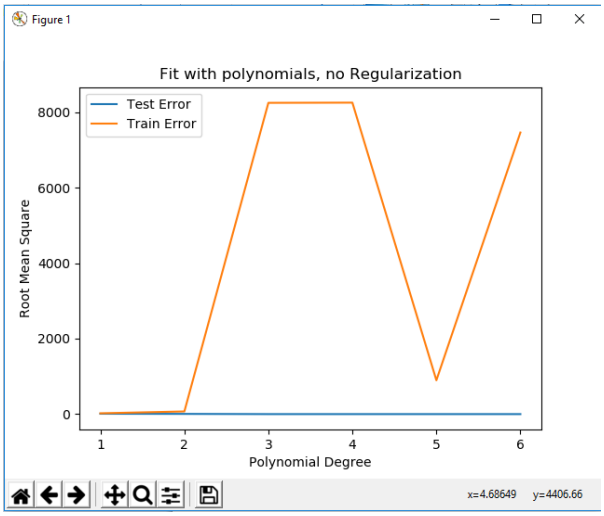
5.2 POLYNOMIAL REGRESSION

Unregularized Polynomial Regression (Degree 1 – 6) - without Normalization



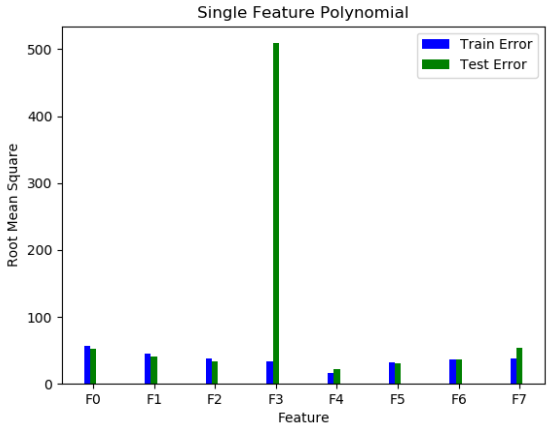
The RMS value of the training data should decrease as the degree of the polynomial increases since the higher polynomial will overfit the data, but in our scenario the training error increases and the testing error remains stable.

Unregularized Polynomial Regression (Degree 1 – 6) - with Normalization



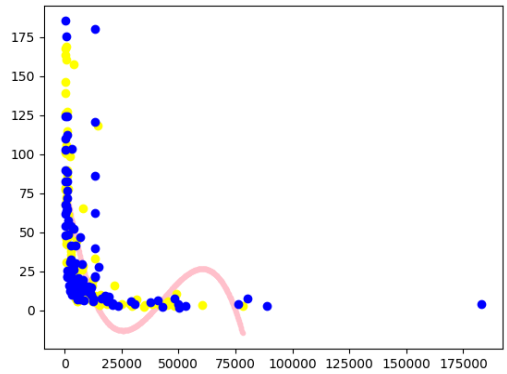
Polynomial Regression – 1d

Features 8-15 (Total population - Low birthweight).
After fitting each (un-normalized) feature with a degree 3 polynomial (unregularized).
The training and test for each of the 8 features is plotted below

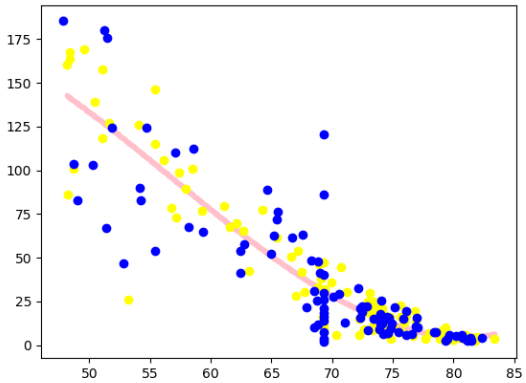


Plots of Degree – 3 polynomial fit for features 11(GNI), 12 (Life expectancy), 13 (literacy)

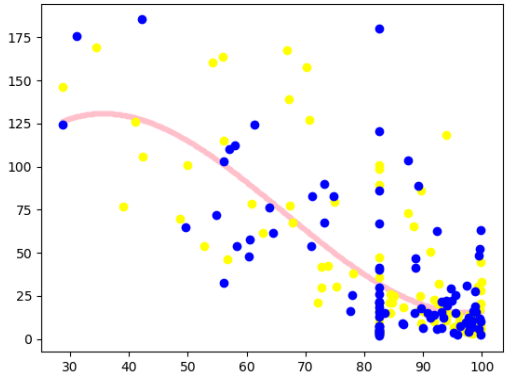
i. Feature 11 - GNI



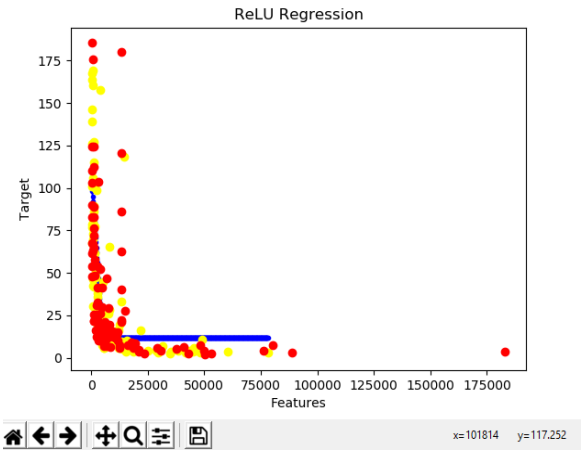
ii. Feature 12 – Life Expectancy

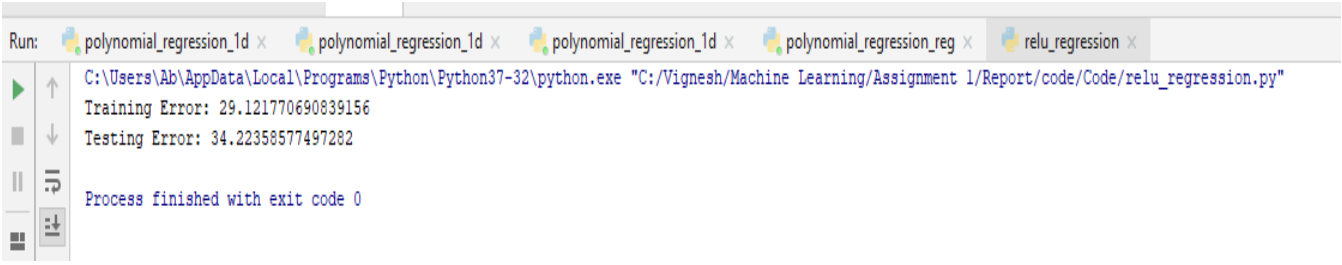


iii. Feature 13 – Literacy



5.3 ReLU Basis Function





5.4 Regularized Polynomial Regression

Upon viewing the plot, the lambda value can be chosen as **1000** since it has the least error value of **28.469223279**