Assignment 1

Abhinav Tyagi - 2020A7PS2043H
Rishiraj Datta - 2020A7PS2075H
Ritvik - 2020A7PS1723H

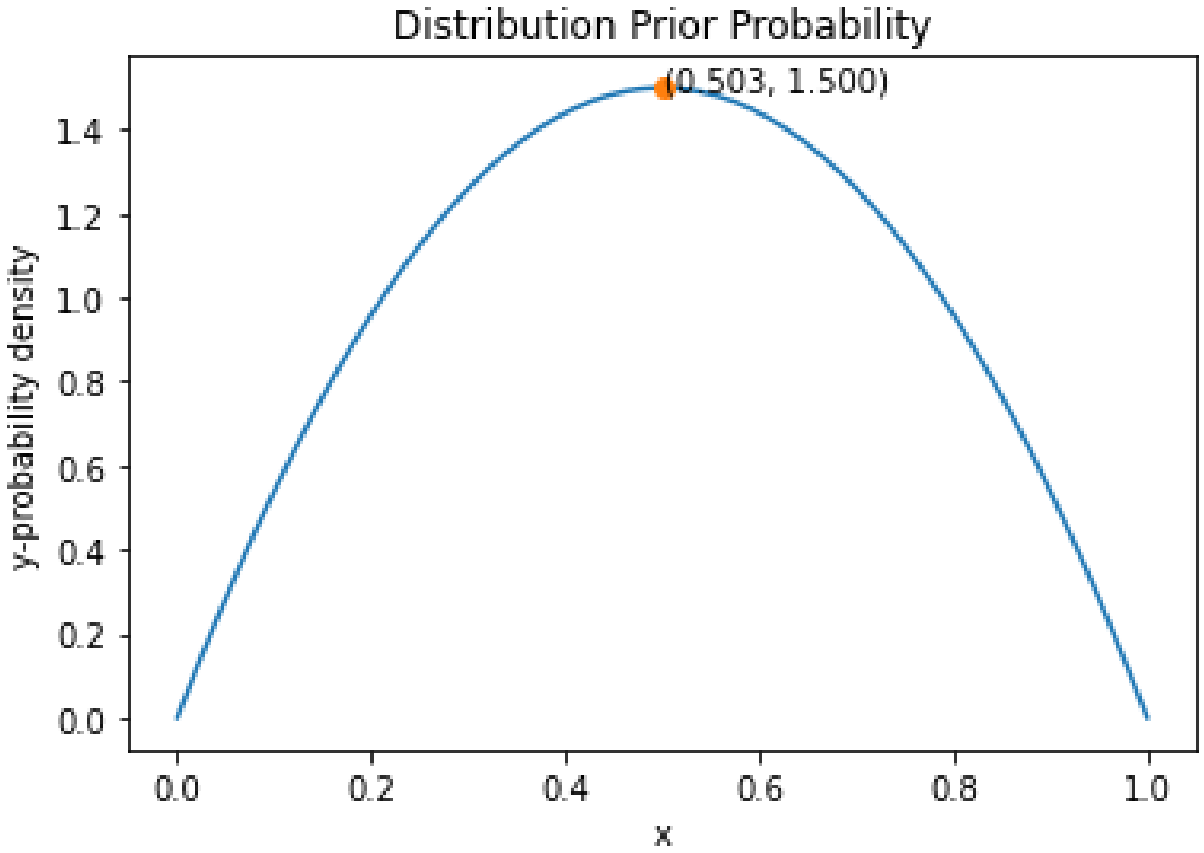
Assignment 1-A Prior and Posterior Distributions

Problem Statement

A survey of whether a customer likes the new update of the software or not was done by a company. Let s denote the probability of a customer liking the new update. Before the survey it was assumed that s follows a beta distribution with parameters $\alpha,\beta = (2,2)$. Out of the 50 customers surveyed, 40 of them liked the update. Plot the prior and posterior probability distribution of s. After few days another survey is conducted in which out of the 30 customers surveyed 17 of them disliked the update. Plot distribution of s after this survey. What is the prior distribution in this case, and justify it with appropriate reasoning. Again, a final survey was conducted in which 70 out of the 100 people surveyed liked the update. Plot distribution of s after the final survey.

Finding Posterior Probability and describing the likelihood of s:

Given in the question that the probability of customers liking the new update is s. Before the survey was conducted; it is assumed that the probability distribution of s follows a beta distribution with parameters a(= 2) and b(= 2).



To find the posterior probability:

$$\beta(\mu \mid a,\ b) \ = \ \frac{\Gamma(a+b)}{\Gamma(a) * \Gamma(b)} \ * \ \mu^{(a-1)}(1 \ - \ \mu)^{(b-1)}$$ over the parameter μ.

Likelihood of s:

Bernoulli distribution can be verified to be a normalized distribution; it would be safe to assume that our likelihood function follows Bernoulli distribution. Therefore,

s = 1 => Number of customers liking the new update
s = 0 => Number of customers disliking the new update

$$p(s \ = \ 1 \mid \mu) \ = \ \mu$$
$$p(s \ = \ 0 \mid \mu) \ = \ 1 \ - \ \mu$$
$$Bern(s \mid \mu) \ = \ \mu^{(s)}(1 \ - \ \mu)^{(1-s)}$$

Suppose there is a dataset D = {s1, s2, s3, ..., sN} of the observed value of s. It can construct the likelihood function, which is over the parameter μ, on the assumption that the observation are drawn independently from p(x|μ), so that.

$$p(D \mid \mu) = \Pi\, p(s \mid \mu) = \Pi\, \mu^{(s)}(1 - \mu)^{(1-s)}$$

Assuming the likelihood function is proportional to $\mu^{(s)}(1 - \mu)^{(1-s)}$.

It is known that posterior probability is proportional to the product of prior probability and the likelihood function; therefore on calculating, the posterior probability also comes out to be a beta distribution with new parameters α,β.

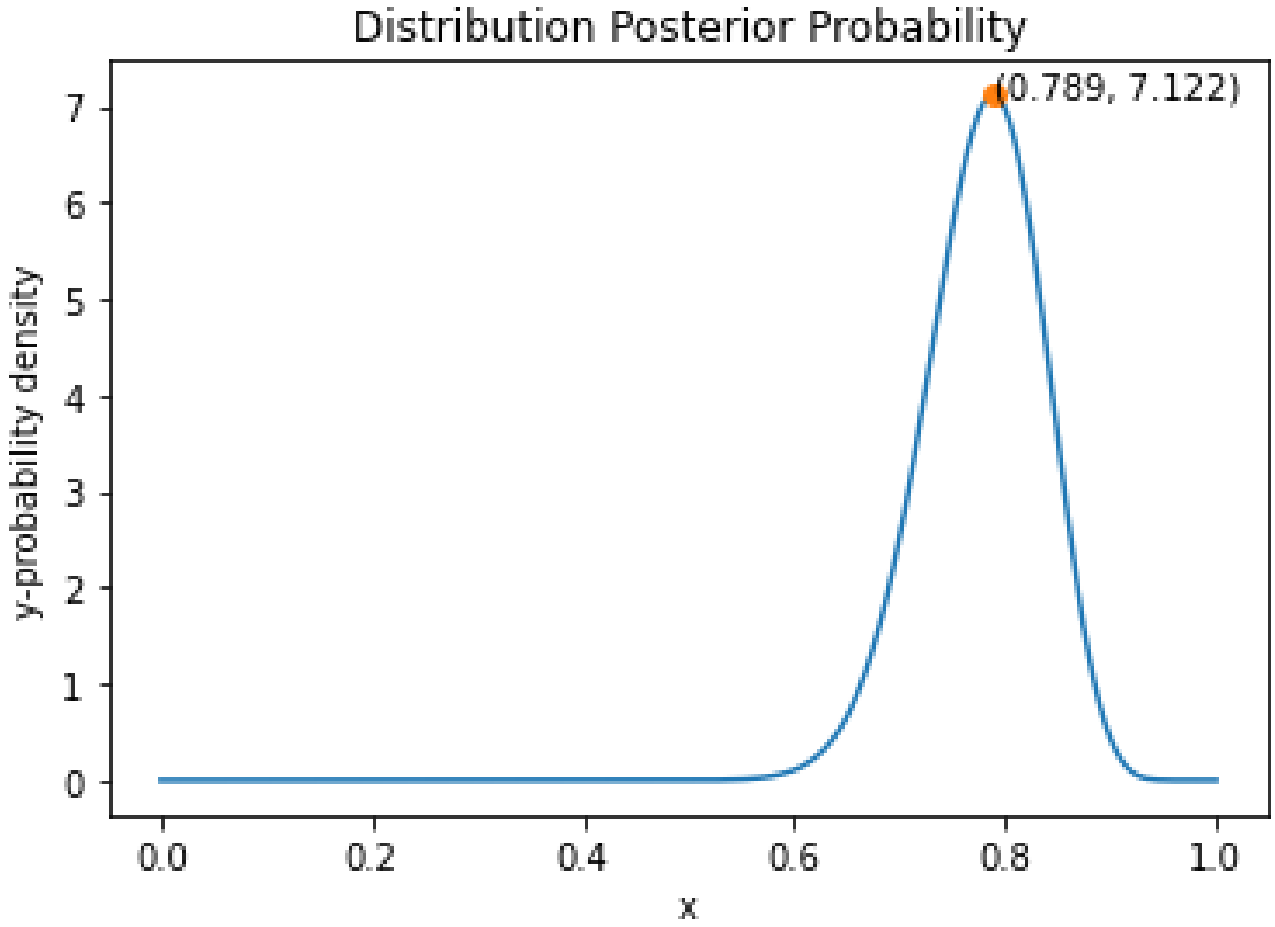
Now, if considering a dataset of m observations of s = 1, i.e customer liking the new update and l observations of s = 0, i.e customer disliking the new update, the parameters α,β in the posterior probability comes out to be:

α = a + m (where a is one of the parameter of the prior probability)
β = b + l (where b is one of the parameter of the prior probability)

Therefore, we can provide a simple interpretation for the hyperparameters a and b in the prior as an effective number of observations of s = 1 and s = 0, respectively.
Furthermore, the posterior distribution can act as the prior if we subsequently observe additional data.

Therefore,
After the first survey, out of 50, 40 liked the update; 10 disliked the update; updated parameters for the posterior probability will be:

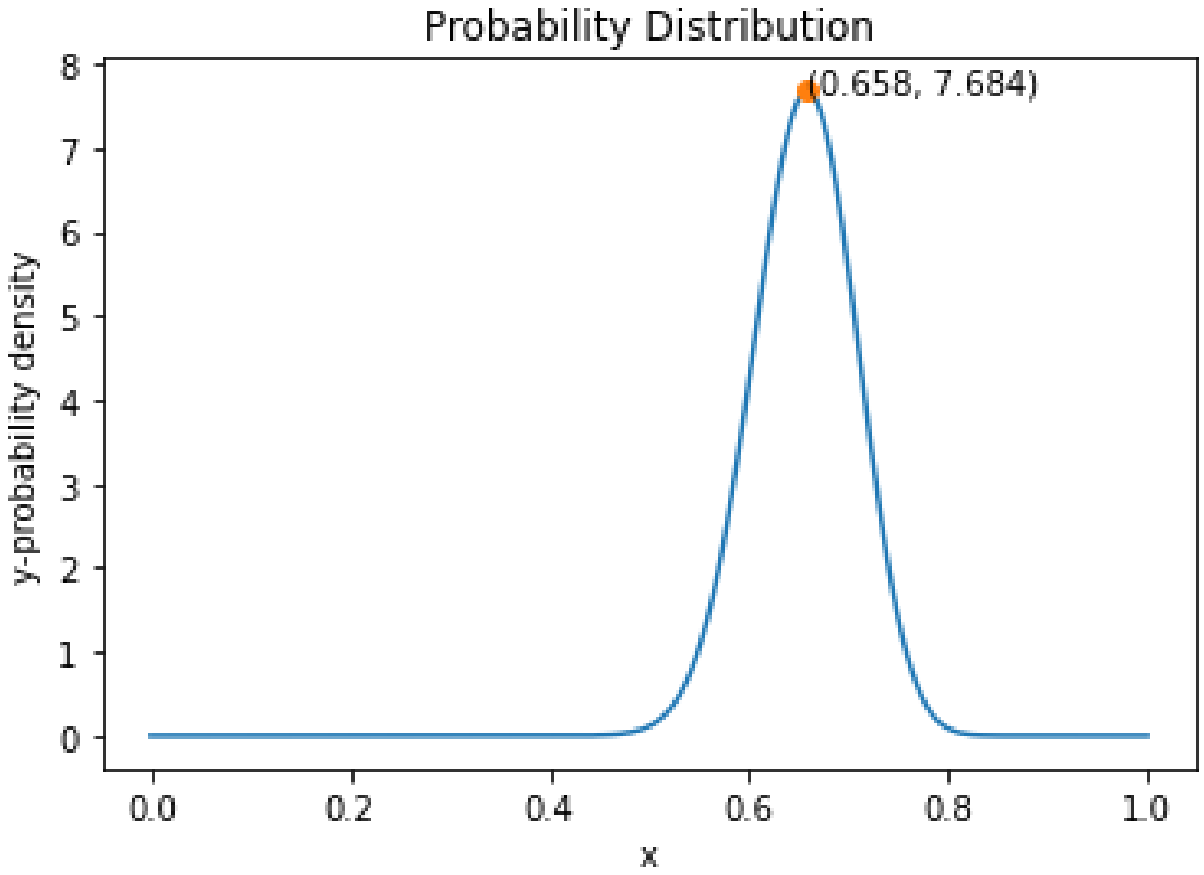
α = 42 (where a is one of the parameter of the prior probability)
β = 12 (where b is one of the parameter of the prior probability)



After the second survey out of 30, 17 disliked the update; 13 liked the update; updated parameters for the posterior probability will be:

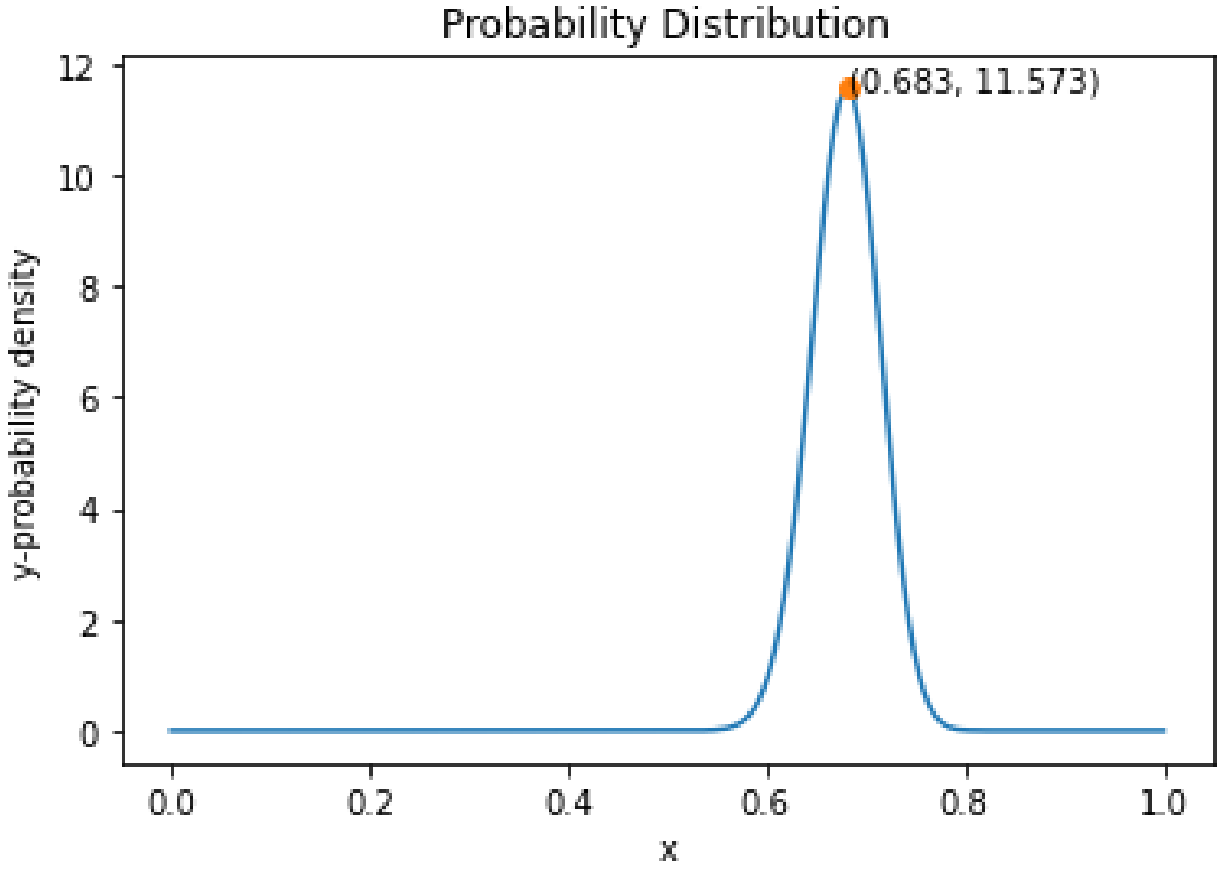
As we know the posterior distribution can act as the prior if we subsequently observe additional data which we are doing in this case. Therefore,

- $\alpha = 55$ (where a is one of the parameter of the prior probability)
- $\beta = 29$ (where b is one of the parameter of the prior probability)



After the third survey, out of 100, 70 liked the update; 30 disliked the update; updated parameters for the posterior probability will be:

α = 125 (where a is one of the parameter of the prior probability)
β = 59 (where b is one of the parameter of the prior probability)



Assignment 1-B Polynomial Regression and Regularization

Problem Statement

• Aquatic toxicity caused due to manufactured chemicals and other anthropogenic and natural materials severely affects aquatic organisms at various levels of organization. The dataset consists of 2 molecular descriptors: MLOGP and GATS1i which affect the LC50 value (quantitative experimental response).

• Dataset: Link: https://drive.google.com/file/d/1nfA1Qet7qOR46tWCGnFR2oHwzWSV0nO/view?usp=sharing

Models:

1. Polynomial regression model (Degree: 0 to 9)
    a. Using Gradient Descent
    b. Using Stochastic Gradient Descent
2. Polynomial regression of degree 1 with regularization
    For q = 0.5, 1, 2, 4 at different values of λ.

Algorithms:

The given dataset was split into training and testing data using a random 80-20 split. Then the data points were normalised to a range of 0 to 1. As the data consisted of outliers and without normalisation, the value obtained for the polynomial coefficients exceeded a certain threshold, giving values as nan.

The training and testing data were normalised based on the given formula:

$$x_{normalised} = \frac{x - min(x_{training})}{(max(x_{training}) - min(x_{training})}$$

After normalising the data, the normalised training data was used to train the above specified model.

The learning rate used in training all the models is 0.001

1. Polynomial regression model (Degree: 0 to 9)
    a. Using Gradient Descent

        Equations Used:

        $$w^{(k+1)} = w^{(k)} - \eta\left(\frac{\partial(E)}{\partial w}\right)_{w^{(k)}}$$

        Here $w = \left[w_0\, w_1\, w_2 \ldots\, w_m\right]^T$ after k iterations in gradient descent

        $$E = \frac{1}{2N}\sum_{i=0}^{N}\left(y_i - w^T \cdot x_i\right)^2$$ where y_i is the given value for a given required combination of x_1 and x_2.

        The above equations were used, and 500000 iterations were performed for each degree of the polynomial (from 0 to 9) in order to find the minima of the error function and hence the required coefficients.
        The number of iterations was decided based on the learning rate, experimentation with different numbers of iterations, the time required to train the model (increase the accuracy) and the computational power of the system on which the model was trained.

    b. Using Stochastic Gradient Descent

The models for each degree of the polynomial (from 0 to 9) were trained based on the same formula as above but here, instead of taking the complete dataset, for each iteration, a random value from the training data was selected. This reduced the number of operations and provided with the flexibility to increase the number of iterations.
The number of iterations taken for stochastic gradient descent is 1000000.

2. Polynomial regression of degree 1 with regularization

   Equation:

   $$E = \frac{1}{2}\sum_{i=1}^{N}\left(t_n - w^T \cdot \phi(x_i)\right)^2 + \frac{\lambda}{2}\left(\left(|w_1|\right)^q + \left(|w_2|\right)^q\right)$$

   To get the value of w, for minimum error, the above equation was differentiated with respect to the coefficients and gradient descent was used to find the minimum error. The number of iterations taken was 100000
   The values for λ were taken to be 0.05, 1 and 2.5. A λ value of 0 was also taken to show the difference regularization causes to the value of w as compared to the classical regression model.
   For each of these penalty values, the model with q = 0.5, 1, 2 and 4 was trained based on the same normalised training data.

Training and Testing Error for models of degree 0 to 9:

Formula for finding the error:

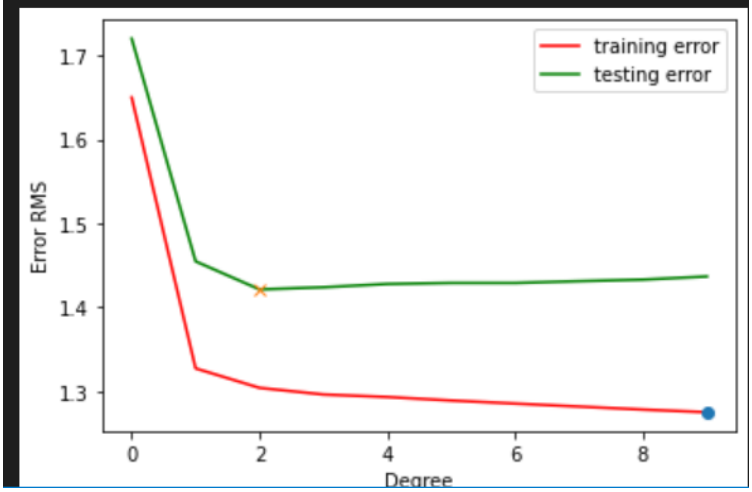
$E_{RMS} = \sqrt{\frac{(t - y_{pred})^2}{N}}$ where t is the given value and y_{pred} is the predicted value based on the given model and N is the number of datapoints.

The training and testing error calculated and mentioned are RMS error values.

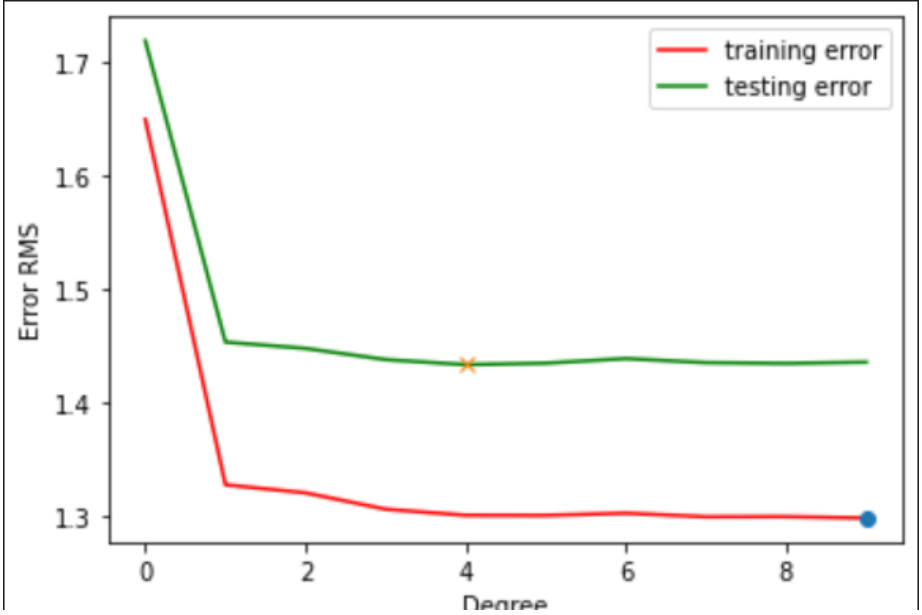
The Errors in Gradient Descent:

These are the RMS Errors on the training and testing data.
It can be observed that the value of the training error decreases with an increase in the degree of the polynomial.
But the testing error first decreases with an increase in the degree of the polynomial but then starts to increase.
This observation indicates that the curve for the higher degrees of polynomials has started to overfit the data. This results in decreasing training error but increasing testing error.

Degree	Training Error	Testing Error
0	1.649978	1.720058
1	1.326912	1.454544
2	1.303711	1.421062
3	1.295793	1.423662
4	1.292796	1.427575
5	1.288618	1.428798
6	1.284971	1.428794
7	1.281408	1.430900
8	1.277834	1.432763
9	1.274656	1.436490



Degree	Training Error	Testing Error
0	1.649985	1.719483
1	1.327324	1.453332
2	1.320176	1.447735
3	1.305744	1.437832
4	1.300378	1.433366
5	1.300263	1.434705
6	1.302204	1.438738
7	1.299072	1.435159
8	1.299178	1.434508
9	1.297933	1.435625



The Errors in Stochastic Gradient Descent:

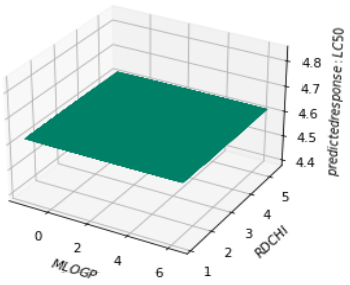
The errors in the stochastic gradient descent also follow a similar trend when compared to gradient descent.
The training errors decrease with an increase in the degree of polynomials but in the case of testing error, it can be observed that the error first decreases and then increases.

This is an indication that due to the flexibility in higher degree polynomials the model overfits the training data and hence gives higher error for testing data.

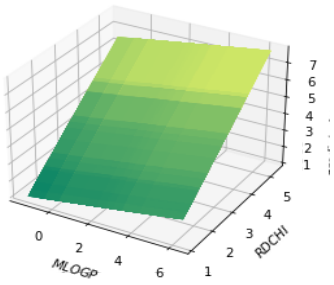
Surface Plots:

Polynomial Regression (Degree: 0 to 9) - Gradient Descent

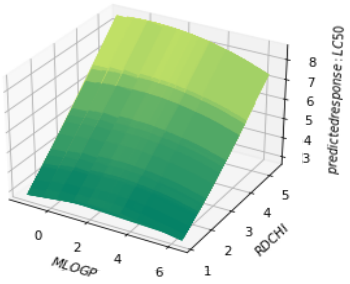
Degree: 0



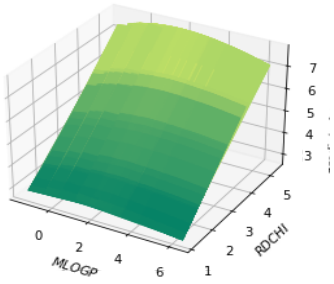
Degree: 1



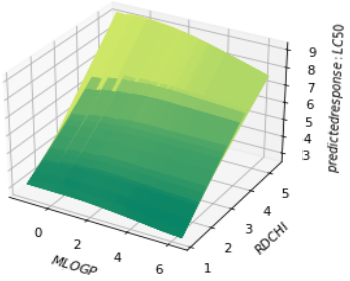
Degree: 2



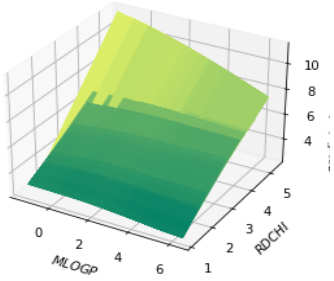
Degree: 3



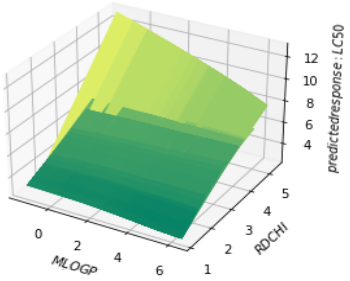
Degree: 4



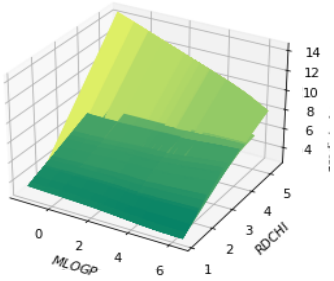
Degree: 5



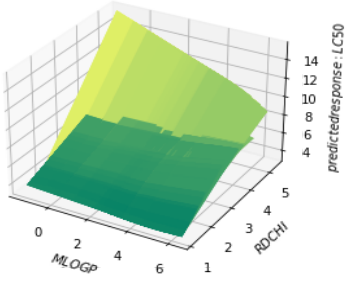
Degree: 6



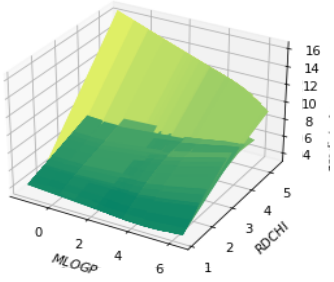
Degree: 7



Degree: 8

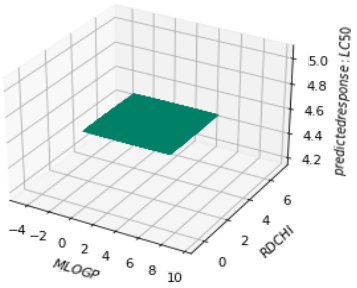


Degree: 9

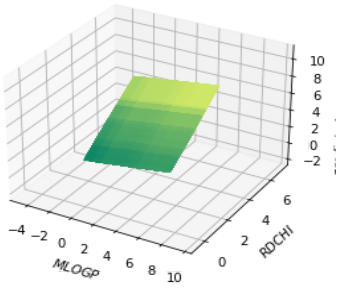


Polynomial Regression (Degree: 0 to 9) - Stochastic Gradient Descent

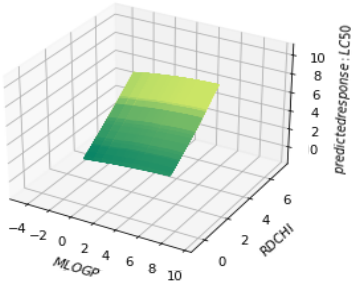
Degree: 0



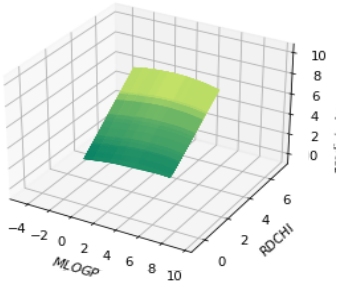
Degree: 1



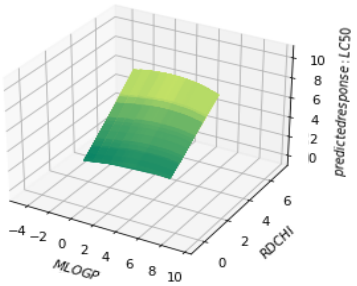
Degree: 2



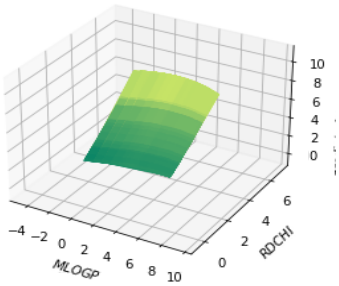
Degree: 3



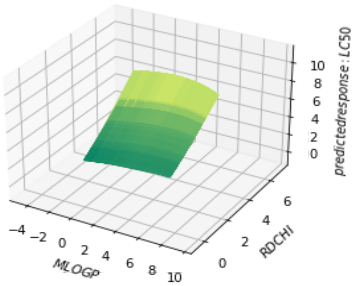
Degree: 4



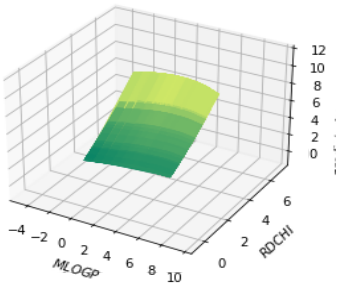
Degree: 5



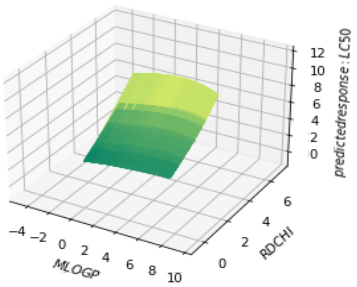
Degree: 6



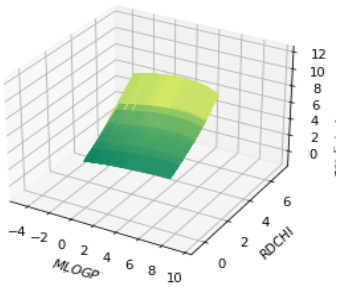
Degree: 7



Degree: 8

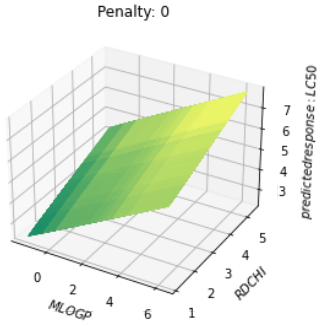


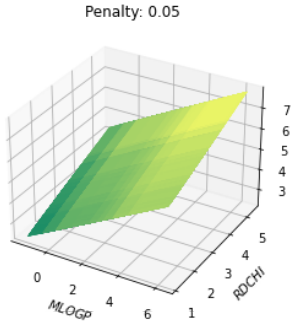
Degree: 9

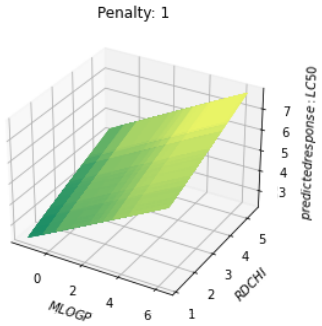


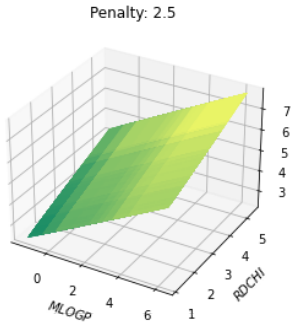
Polynomial Regression (Degree: 1) - Regularized

q = 0.5

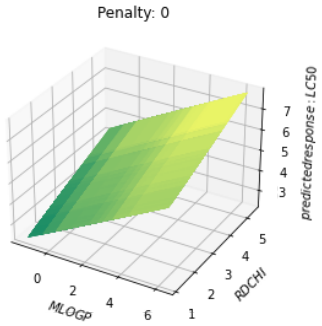


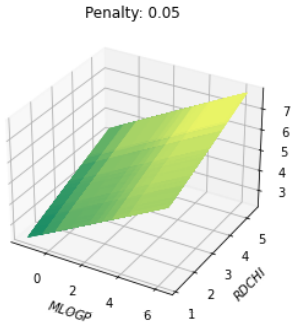


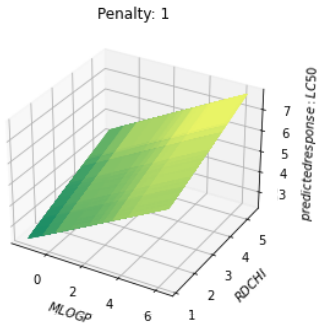


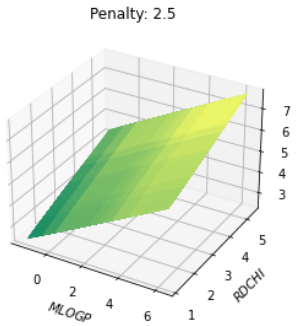


q = 1



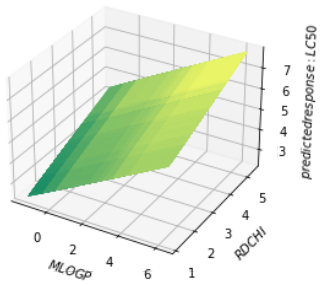




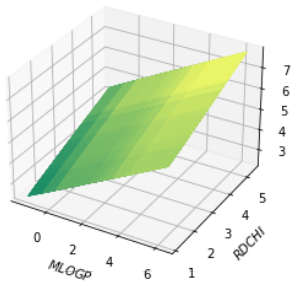


q = 2

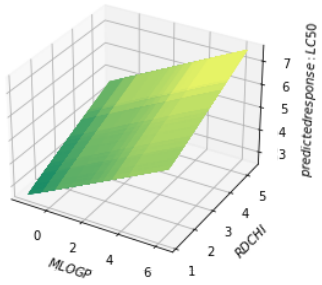
Penalty: 0



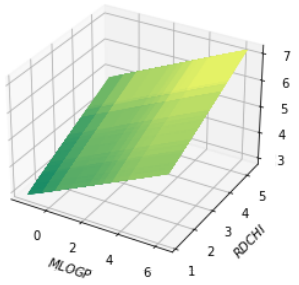
Penalty: 0.05



Penalty: 1

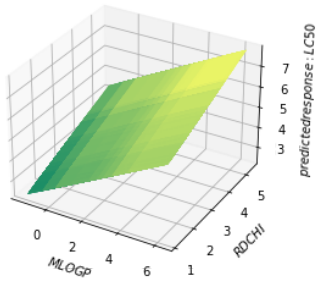


Penalty: 2.5

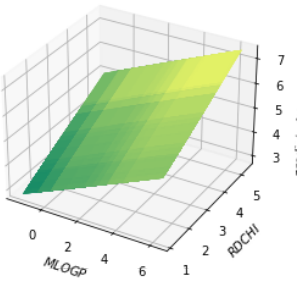


q = 4

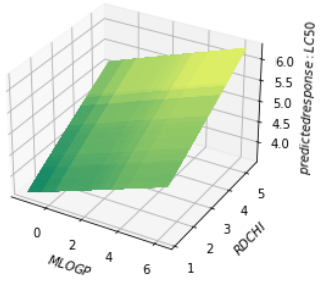
Penalty: 0



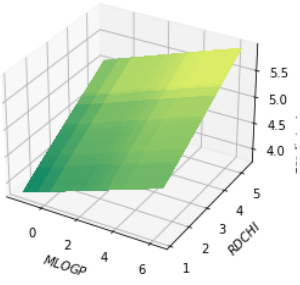
Penalty: 0.05



Penalty: 1



Penalty: 2.5

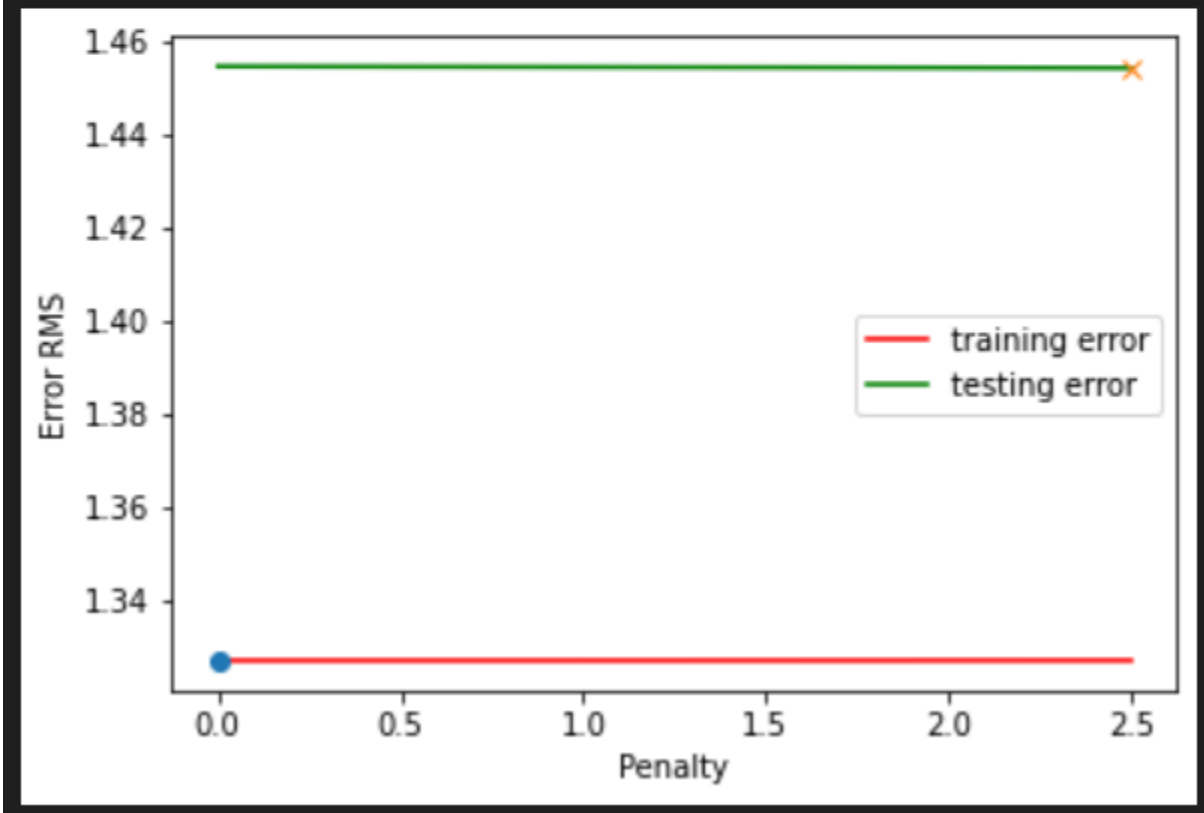


Comparative analysis study of the four optimal regularized regression models and best-fit classic polynomial regression model

For q = 0.5

```
q: 0.5
```

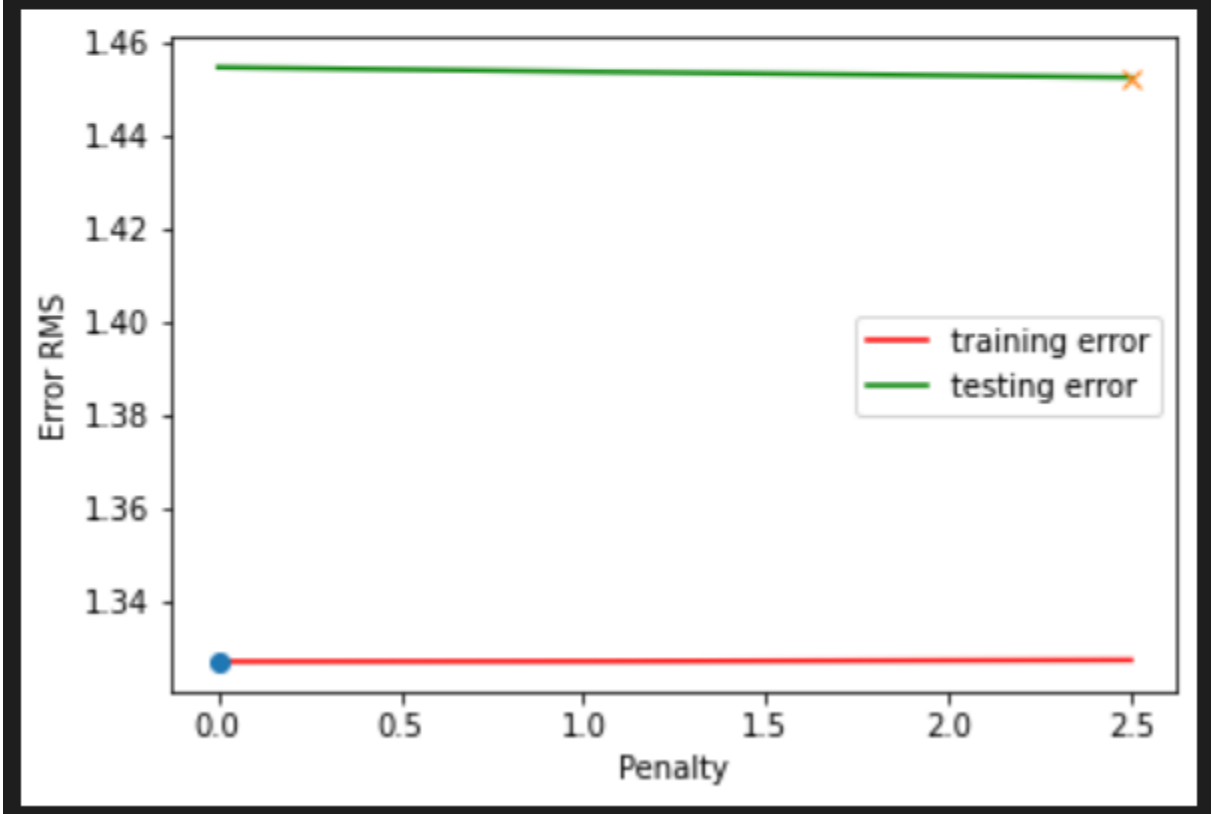
Penalty	Training Error	Testing Error
0.00	1.326912	1.454544
0.05	1.326912	1.454536
1.00	1.326915	1.454391
2.50	1.326930	1.454172



For q = 1.0

```
q: 1
```

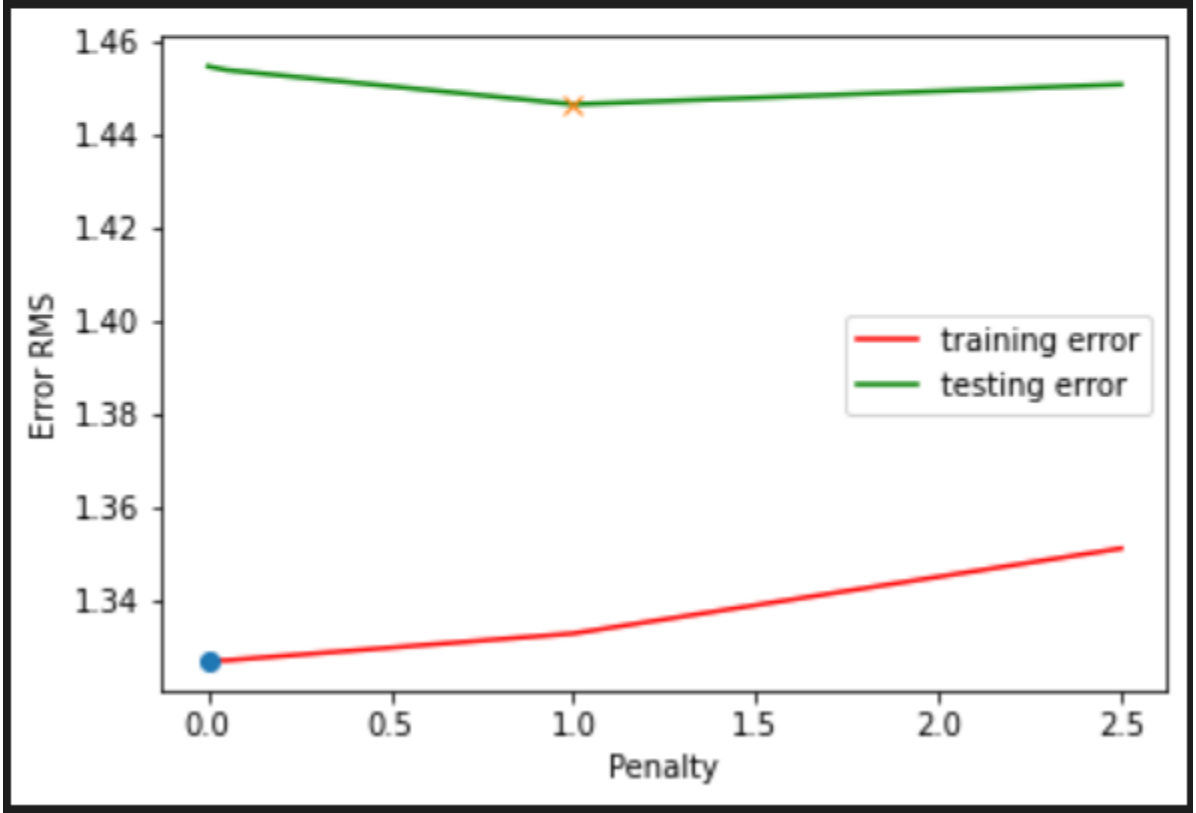
Penalty	Training Error	Testing Error
0.00	1.326912	1.454544
0.05	1.326912	1.454495
1.00	1.326961	1.453600
2.50	1.327218	1.452378



For q = 2.0

```
q: 2
```

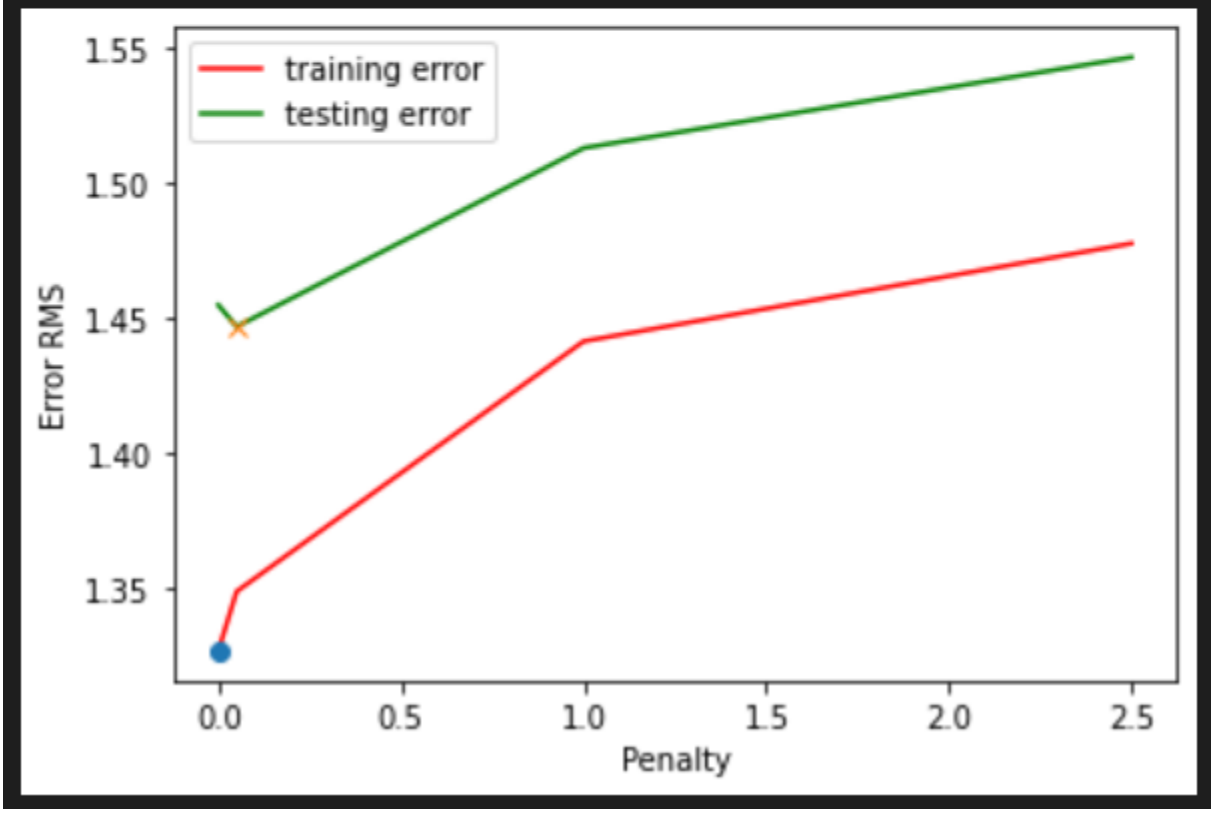
Penalty	Training Error	Testing Error
0.00	1.326912	1.454544
0.05	1.326932	1.453750
1.00	1.332696	1.446368
2.50	1.350971	1.450646



For q = 4.0

```
q: 4
```

Penalty	Training Error	Testing Error
0.00	1.326912	1.454544
0.05	1.348767	1.446572
1.00	1.441158	1.512540
2.50	1.477301	1.546085



The best-fit classic regression model with the least testing error is the model with degree 2.
The training error for this model is 1.303711
The testing error for this model is 1.421062

The optimal regularized regression models for q = 0.5 is at penalty 2.5
The training error for this model is 1.32693
The testing error for this model is 1.454172

The optimal regularized regression models for q = 1 is at penalty 2.5
The training error for this model is 1.327218
The testing error for this model is 1.452378

The optimal regularized regression models for q = 2 is at penalty 1.0
The training error for this model is 1.332696
The testing error for this model is 1.446368

The optimal regularized regression models for q = 4 is at penalty 0.05
The training error for this model is 1.348767
The testing error for this model is 1.446572

Tabulating the above mentioned data:

Model	Training Error (RMS)	Testing Error (RMS)
Degree 2	1.303711	1.421062
Regularized (q = 0.5, penalty = 2.5)	1.32693	1.454172
Regularized (q = 1, penalty = 2.5)	1.327218	1.452378
Regularized (q = 2, penalty = 1.0)	1.332696	1.446368
Regularized (q = 4, penalty = 0.05)	1.348767	1.446572

Observing the values of the testing error, it can be deduced that the best-fit classic regression model of degree 2 is more accurate than the regularized models.

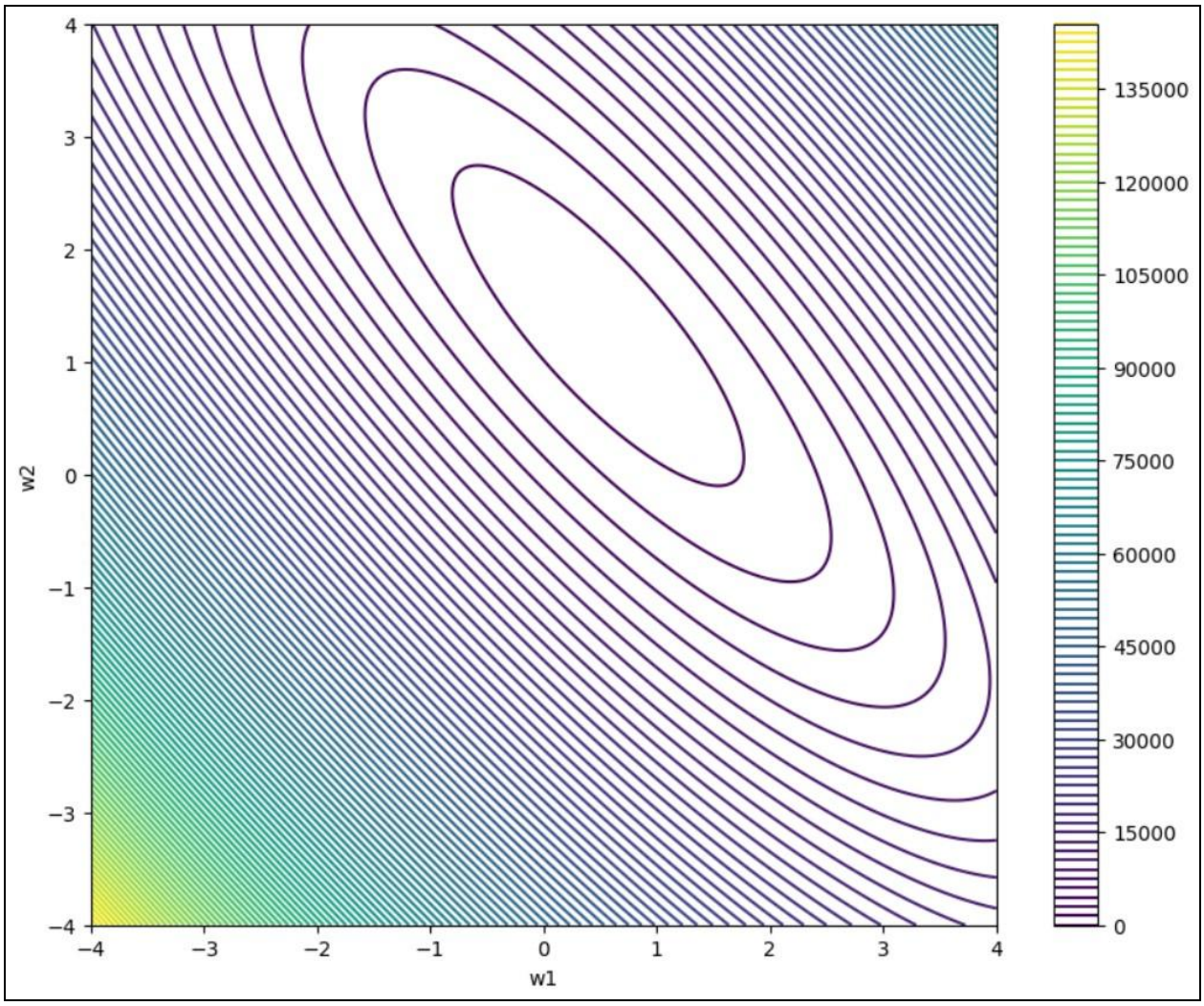
This can be due to the fact that degree 2 is the most accurate in classical regression which corresponds to the fact that degree 1 is underfitting the data and hence regularizing it constraints the value of w1 and w2 even further leading to an even more underfitted model.

Assignment 1-C Visualizing Regularization

Problem Statement

Plot the contours of the error function (unregularized) and the constraint regions for q = 0.5, 1, 2 and 4 (Refer to Fig 3.4 of textbook) and η = 1.4, 0.1, 0.035 and 0.052 respectively. Make a plot of error function contours. Also make plots of the constraint regions and error function contours, showing the tangential contour where the minima occurs. Indicate the values of $w1$, $w2$ at the point of intersection of the tangential contour and the constraint region for which the global minima will be obtained.

(i) Error Contour Plot

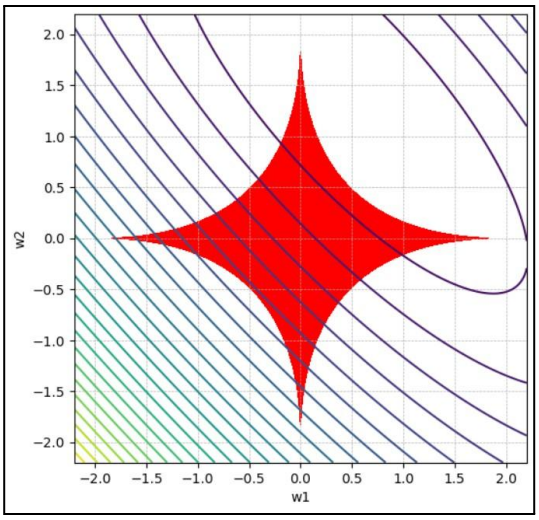


The above graph represents the error contour plot for the following unregularised error function:

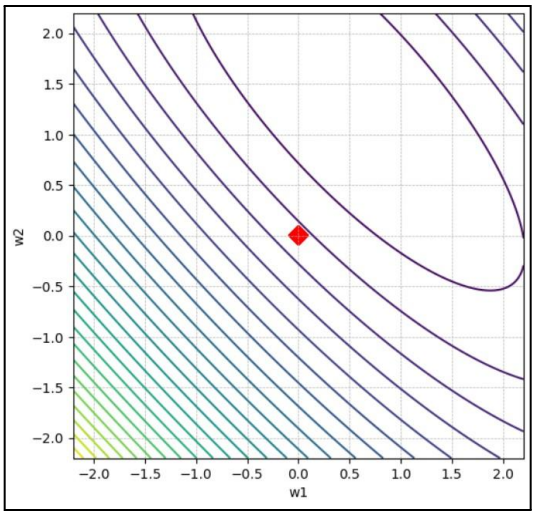
$$E(w) = \frac{1}{2} \sum (y \quad - t_n)^2$$

Here in order to plot the contour, 1000 equally spaced values are taken from -4 to 4 and stored in arrays *w1* and *w2*, and for each pair of these values, the error is calculated. The error is stored in a 2-D array and passed as parameter in the *matplotlib.pyplot.contour()* function, along with the meshgird formed by arrays *w1* and *w2*. This function plots the contour for the error function on a 2-D plane, with *w1* along the x-axis and *w2* along the y-axis. The colorbar alongside the graph provides values of the points that are present on that level of the 3-D plot of the error function.

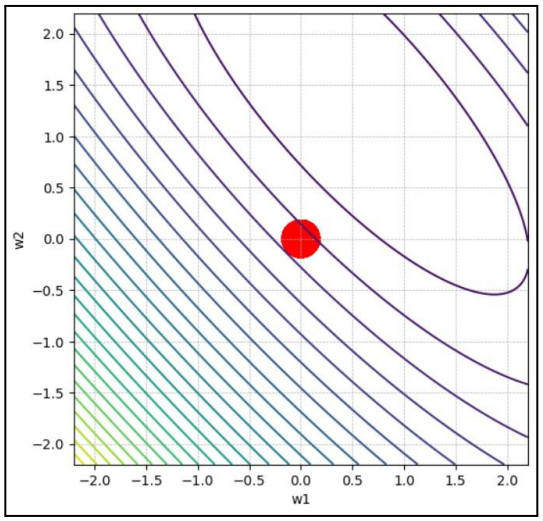
(ii) Constraint Regions and Error Function Contours



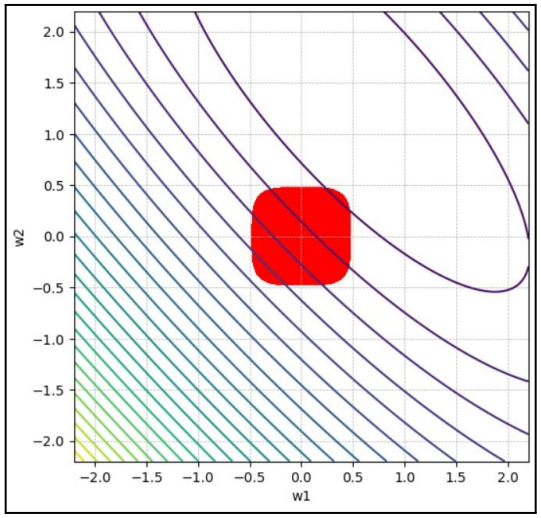
(a)



(b)



(c)



(d)

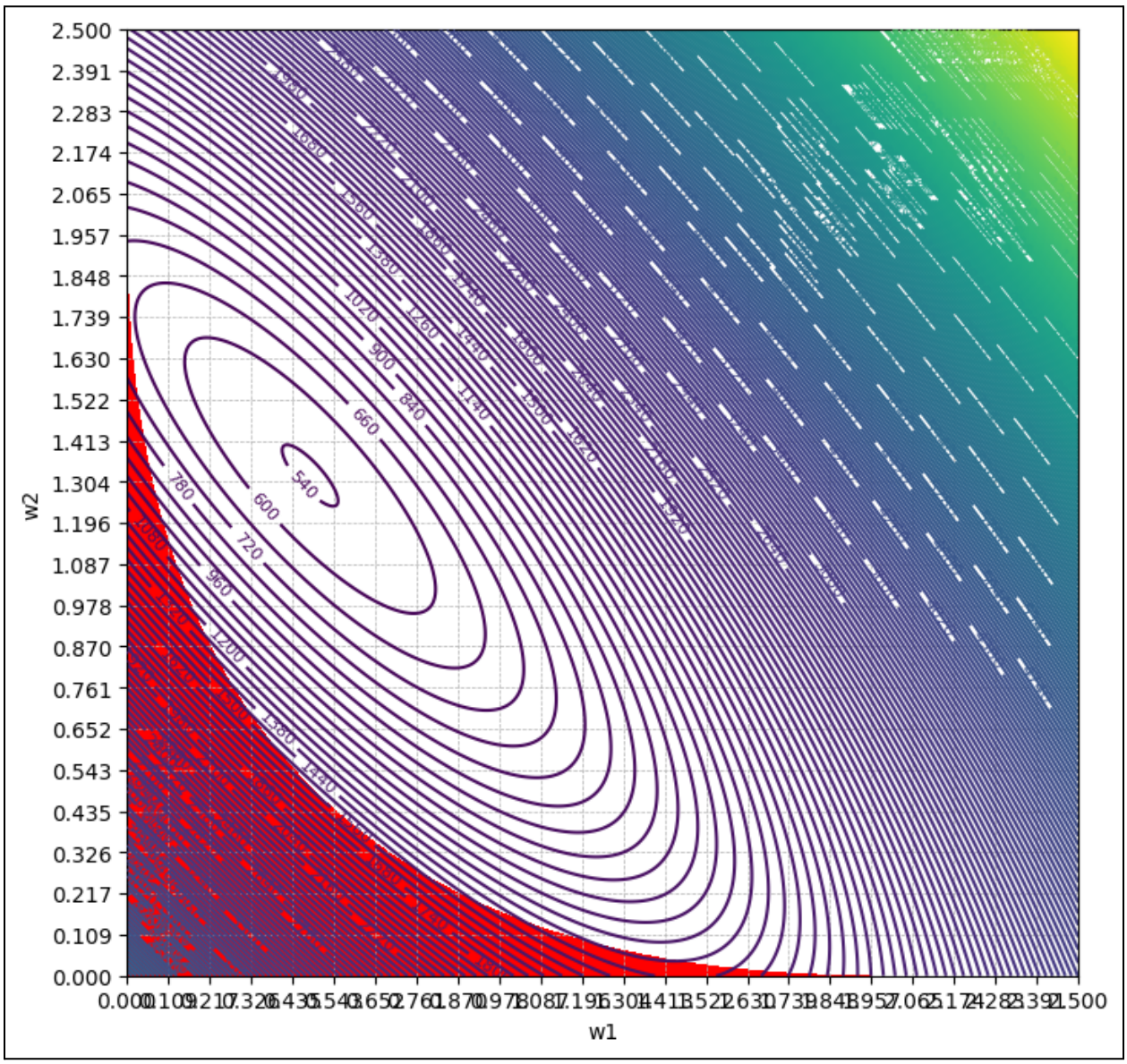
The above graphs represent constraint regions, along with the error function contours, for the following values of q and η:

(a) $q = 0.5$ and $\eta = 1.4$
(b) $q = 1$ and $\eta = 0.1$
(c) $q = 2$ and $\eta = 0.035$
(d) $q = 4$ and $\eta = 0.052$

Upon observation of these plots, we can see that the corresponding tangential contours for all the different constraint regions will be present in the first quadrant. So, for further analysis, it is sufficient to plot only the first quadrant of these functions.

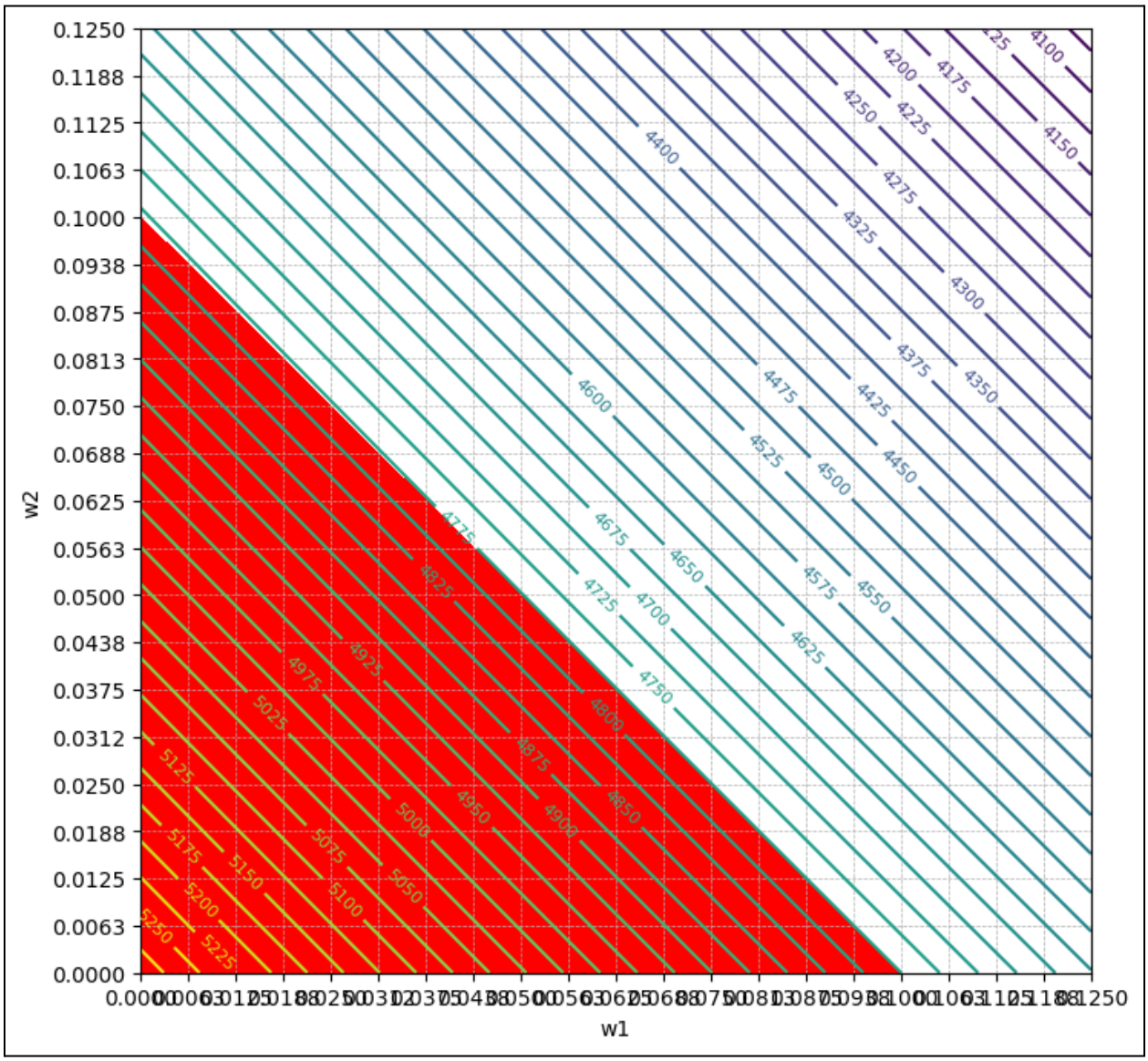
In order to obtain approximate values of *w1* and *w2* of the point of intersection of the tangential contour and the constraint region for which global minima is obtained, we experiment with the different values of level that we use to plot the error contour function. This will help us to visualise which is the approximate tangential contour and the point at which it touches the constraint region.

(a)



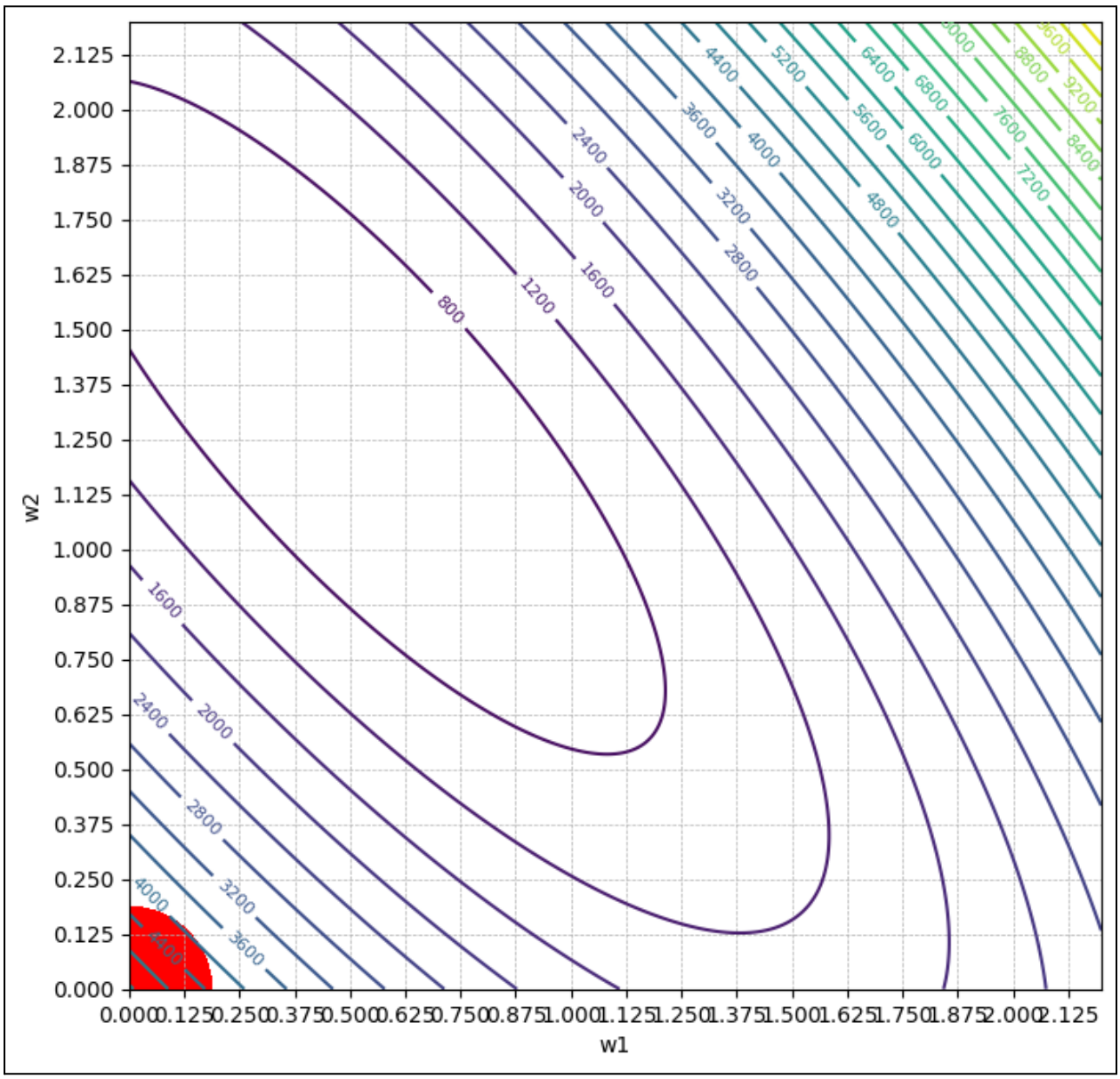
For q = 0.5 and η = 1.4, on taking 300 levels for the contour plot, we can observe that the tangential contour with value 660 touches the constraint region. The value of *w2* is approximately 1.739, and corresponding value of *w1* is 0.006.

(b)



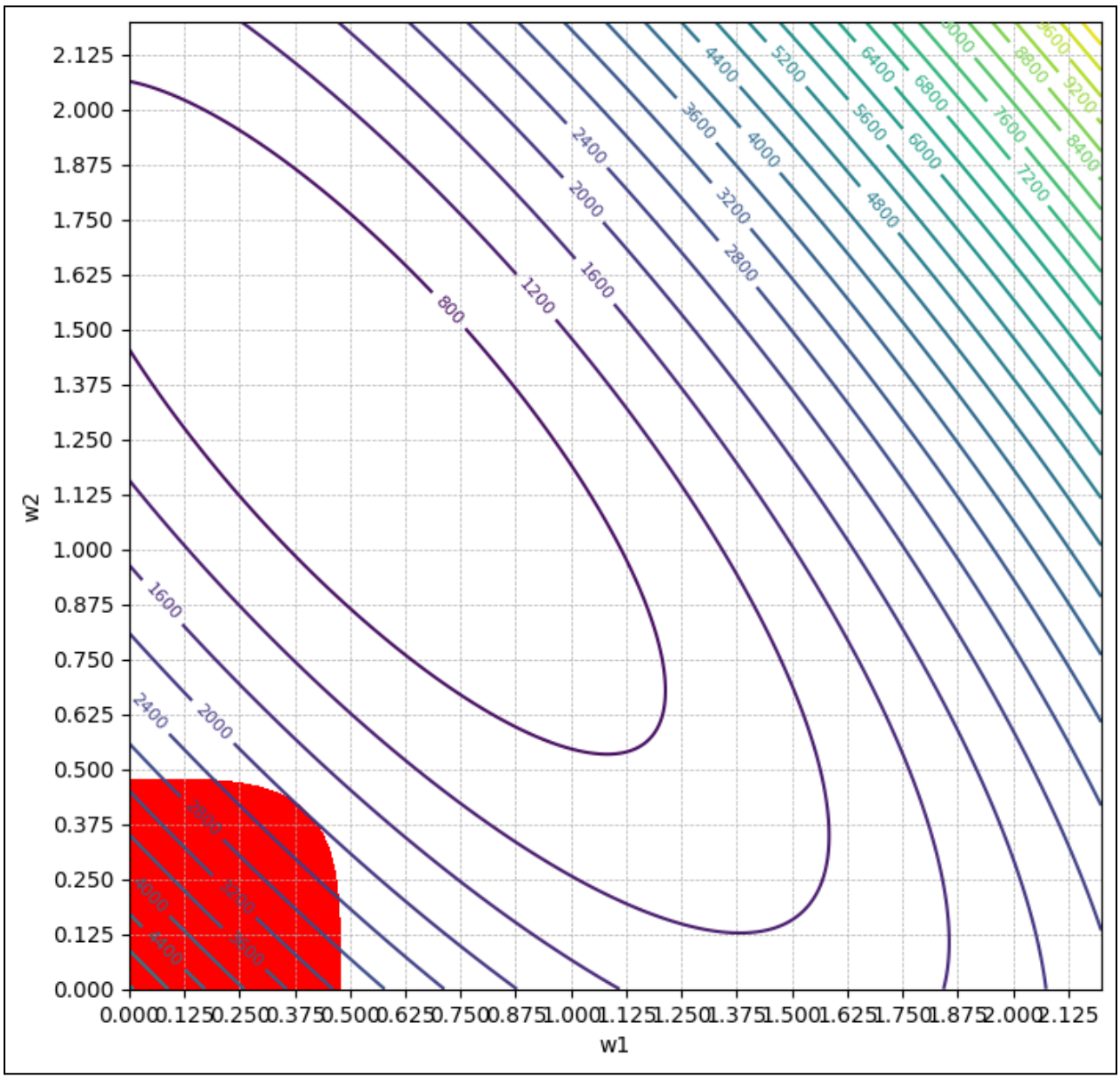
For q = 1 and η = 0.1, on taking 50 levels for the contour plot, we can observe that the tangential contour with value 4775 touches the constraint region. The value of *w2* is approximately 0.0063, and corresponding value of *w1* is 0.0938.

(c)



For q = 2 and η = 0.035, on taking 28 levels for the contour plot, we can observe that the tangential contour with value 4000 touches the constraint region. The value of *w2* is approximately 0.1323, and corresponding value of *w1* is 0.1323.

(d)



For q = 4 and η = 0.052, on taking 28 levels for the contour plot, we can observe that the tangential contour with value 2000 touches the constraint region. The value of *w2* is approximately 0.4063, and the corresponding value of *w1* is 0.3966.

(iii) Mean-Squared Error

$$E(w) = (1/n) \sum(y - t_n)^2$$

Using the above formula for mean-squared error, and the obtained values of *w1* and *w2* we can calculate the following values for error:

The values of w1 and w2 obtained from the above graphs are:

No.	q	eta	w1	w2
1	0.5	1.4	0.0066	1.7391
2	1	0.1	0.0938	0.0063
3	2	0.035	0.1323	0.1323
4	4	0.052	0.3966	0.4063

The values of Mean Square Error obtained from training data are:

No.	w1	w2	Mean Square Error
1	0.0066	1.7391	3.05851887
2	0.0938	0.0063	21.85280389
3	0.1323	0.1323	18.28542177
4	0.3966	0.4063	9.1959121

The values of Mean Square Error obtained from testing data are:

No.	w1	w2	Mean Square Error
1	0.0066	1.7391	2.8034597
2	0.0938	0.0063	23.01359021
3	0.1323	0.1323	19.26195
4	0.3966	0.4063	9.65198418