Name: Shervin Halat

ID: 98131018

First assignment of:

“Machine Learning”

“First Section”

1)

Supervised Learning:

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output. Supervised learning problems are categorized into "regression" and "classification" problems.

Semi-Supervised Learning:

Semi-supervised learning is a class of machine learning tasks and techniques that also make use of unlabeled data for training – typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data).

Unsupervised Learning:

Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common unsupervised learning method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data. The clusters are modeled using a measure of similarity which is defined upon metrics such as Euclidean or probabilistic distance.

Reinforcement Learning:

Reinforcement learning (RL) is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning. It differs from supervised learning in that labelled input/output pairs need not be presented, and sub-optimal actions need not be explicitly corrected. Instead the focus is finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).

Regression:

In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships between a dependent variable (often called the 'outcome variable') and one or more independent variables (often called 'predictors', 'covariates', or 'features'). The two basic types of regression are linear regression and multiple linear regression, although there are non-linear regression methods for more complicated data and analysis. Linear regression uses one independent variable to explain or predict the outcome of the dependent variable Y, while multiple regression uses two or more independent variables to predict the outcome. The most common form of regression analysis is linear regression, in which a researcher finds the line (or a more complex linear function) that most closely fits the data according to a specific mathematical criterion.

Online Learning:

In computer science, online machine learning is a method of machine learning in which data becomes available in a sequential order and is used to update our best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once. Online learning is a common technique used in areas of machine learning where it is computationally infeasible to train over the entire dataset, requiring the need of out-of-core algorithms. It is also used in situations where it is necessary for the algorithm to dynamically adapt to new patterns in the data, or when the data itself is generated as a function of time, e.g., stock price prediction.

Active Learning:

Active learning is a special case of machine learning in which a learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points. In statistics literature it is sometimes also called optimal experimental design. There are situations in which unlabeled data is abundant but manually labeling is expensive. In such a scenario, learning algorithms can actively query the user/teacher for labels. This type of iterative supervised learning is called active learning. Since the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. With this approach, there is a risk that the algorithm is overwhelmed by uninformative examples.

Transfer Learning:

Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. Transfer learning is an optimization that allows rapid progress or improved performance when modeling the second task. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks.

Classification:

In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. Examples are assigning a given email to the "spam" or "non-spam" class, and assigning a diagnosis to a given patient based on observed characteristics of the patient (sex, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance.

Deductive Learning:

Deductive Learning is a method of learning which is based on getting general rules and extract data. Deductive learning is a subclass of machine learning that studies algorithms for learning provably correct knowledge. Typically such methods are used to speedup problem solvers by adding knowledge to them that is deductively entailed by existing knowledge, but that may result in faster solutions.

Inductive Learning:

Inductive Learning is a general method of learning which is based on learning from some given set of Training data or examples as observation in order to get to a conclusion. In fact inductive learning is the general theory behind supervised learning. From the perspective of inductive learning, we are given input samples (x) and output samples (f(x)) and the problem is to estimate the function (f). Specifically, the problem is to generalize from the samples and the mapping to be useful to estimate the output for new samples in the future.

Interpolation:

In the mathematical field of numerical analysis, interpolation is a type of estimation, a method of constructing new data points within the range of a discrete set of known data points. In engineering and science, one often has a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. It is often required to interpolate, i.e., estimate the value of that function for an intermediate value of the independent variable.

Extrapolation:

In mathematics, extrapolation is a type of estimation, beyond the original observation range, the value of a variable on the basis of its relationship with another variable. It is similar to interpolation, which produces estimates between known observations, but extrapolation is subject to greater uncertainty and a higher risk of producing meaningless results.

Over-fitting:

In statistics, overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably". An overfitted model is a statistical model that contains more parameters than can be justified by the data.

2)

In fact, the classification problem is just like the regression problem that both are defined as supervised learning problems, except that the values we want to predict in classification take on only a small number of discrete values as against Regression that the output values could be continues. In order to convert a Regression problem to a Classification one it is enough to label each group of outputs with specific features into a defined class and do this to all outputs. Therefore, for each input that leads to an certain instance of each class, the class label would be the final output that the general outputs would be discrete values or labels.

Reversely, in order to convert a Classification problem into Regression one, there should be too many classes to be defined closely in order to accordingly deal with an Regression problem.

3)

Features are said to be “correlated” or in other words “linearly dependent” when by increasing one feature the other feature increases too and vice versa. There are some upsides and downsides corresponding to this subject. On one hand, having correlated features may help us in feature selection and reduction also it helps us in the way of defining new features. On the other hand, having correlated features may higher the chance of facing noninevitability of term **(**$X^T X$) in normal equation.

In order to determine if a set of vector variables or features are linearly dependent or not it should be examined that if there exists a set of coefficients $a_{1,}$ $a_{2,}$ $a_{3,}$ ..., a_n such that there exists at least one different zero among the mentioned coefficients by which the equation below is satisfied.

$$“\ a_1 x_1 + a_2 x_2 + \dots\ a_n x_n = 0\ ”$$

4)

Similarity: Both Linear Regression and Locally Weighted Linear Regression are supervised learning algorithms with Training Sets and in both the main challenge is computing the term “Θ_j”.

Difference: One of the main differences is that Locally Weighted is a non-parametric algorithm as against linear regression which means there is no fixed set of parameters of “Θ”s but rather for each query point “X”, theta values are computed individually that a higher preference is given to the points in the training set lying in the vicinity of “X” than the points lying far away from “X”. Another important difference is that there exists no training phase in Locally Weighted and all the work is done during the testing phase while making predictions.

5)

In the following some pros cons of Normal Equation and Gradient Descent has been described and the condition of utilization of each is mentioned:

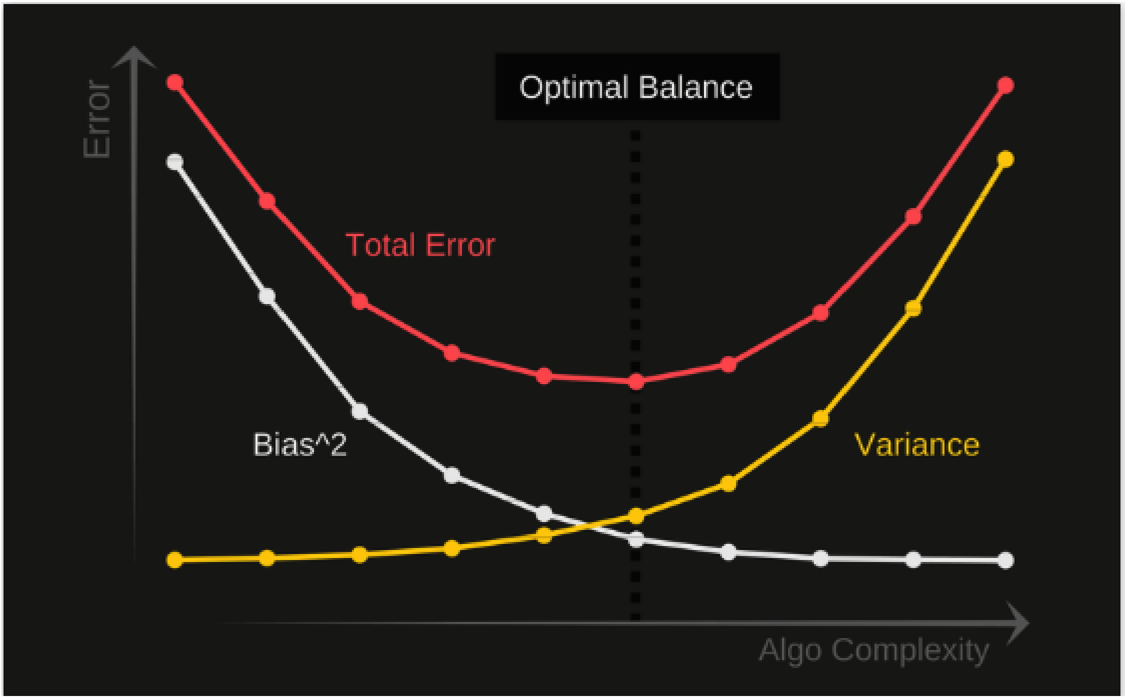
By Normal Equation there is no need to choose parameters like Learning Rate and also no need of large scale iteration as oppose to gradient descent which needs many iteration and choosing alpha.

On the contrary, Gradient Descent works faster when we have large data as against Normal equation since their computation time order are $O(n^2)$ and $O(n^3)$ respectively. What’s more Normal Equation needs calculation of X matrix which is not computable under certain circumstances. In general when we have large number of features(n) it is advisable to use Gradient Descent cause Normal Equation will be slow.

6)

a) **Bias:** Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data. Therefore, as the complexity of the model increases Bias term tends to decrease as shown below.

Variance: Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data. Hence, as the complexity of the model decreases Variance term tends to decrease as shown below.



Mathematical Explanation:

Let the variable we are trying to predict as Y and other covariates as X. We assume there is a relationship between the two such that:

Y=f(X) + e

Where e is the error term and it’s normally distributed with a mean of 0.

Assume f^(X) as linear regression using f(X). Hence, the expected squared error at a point x is

$$Err(x) = E\left[(Y - \hat{f}(x))^2\right]$$

The Err(x) can be further decomposed as:

$$Err(x) = \left(E[\hat{f}(x)] - f(x)\right)^2 + E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right] + \sigma_e^2$$

$$Err(x) = \mathrm{Bias}^2 + \mathrm{Variance} + \mathrm{Irreducible\ Error}$$

Err(x) is the sum of Bias^2, variance and the irreducible error and the Irreducible error is the error that can’t be reduced by creating good models.

Therefore, the Bias term would be:

$$E[\hat{f}(x)] - f(x)$$

And the Variance term would be:

$$E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right]$$

b) The adequate values of both Bias and Variance in each model, as it is shown in figure above, is in a model complexity where the Cross-Validation error (shown by red line in figure above) reaches its low(minimum).

c) As it is fully mentioned and described in part “a”:

The Bias term would be:

$$E[\hat{f}(x)] - f(x)$$

And the Variance term would be:

$$E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right]$$

7) Regularization is a way of regularize our model in order to overcome both overfitting and underfitting. Regularization is used mostly when we have a lot of slightly useful features and want to keep all of them but also reduce the magnitude of parameters Θ_j. Therefore, we add term ($\lambda * \Theta_j^2$) to the cost function equation. And as we increase the value of λ(Regularization Parameter) the Θ_j with higher value of corresponding feature tends to decrease more.

8)

Training Set: Generally, in learning algorithms we prefer proposing a model with best fit which results in problems like over fitting the initial data. Therefore, it is suggested to divide the initial data into three categories namely, Training, Validation and Test sets to tackle over fitting issues. These sets give us different errors based on our suggested model by which we are able to choose the best generalized model. For example in order to find the best Dimension of our polynomial model we use our sets as the following:

Training Set: We choose 60% of the data by which we find the values of Theta for different numbers of dimensions ("D").

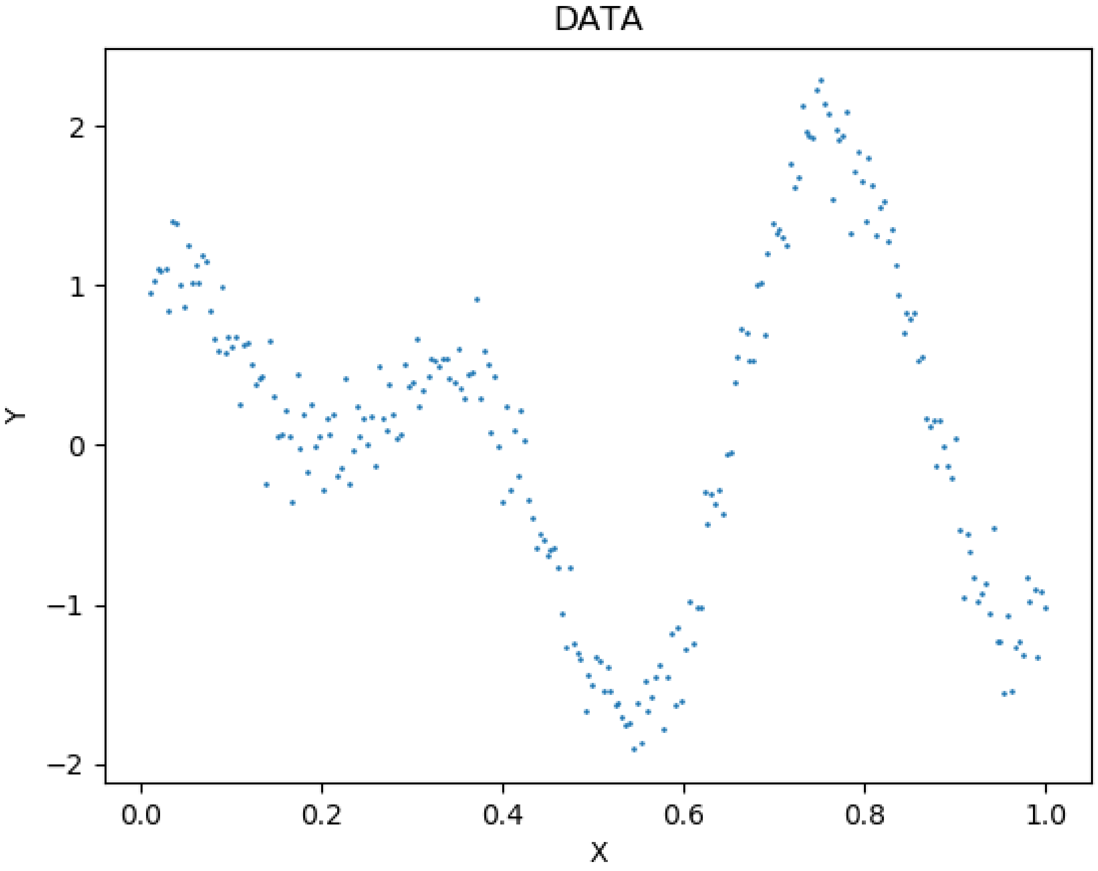
Validation Set: By this set (20% of our data) we calculate the error ($JCV(\Theta(d))$) corresponding to each dimensions of our polynomial models and choose the dimension with least error value using Validation Set.

Test Set: After that the best dimension chose by Validation Set, the generalization error is estimated using the Test Set.

As described above each Training and Validation Sets are used several times regarding number of dimension and after all Test Set is used once to estimate generalization error.

“Second Section”

1)



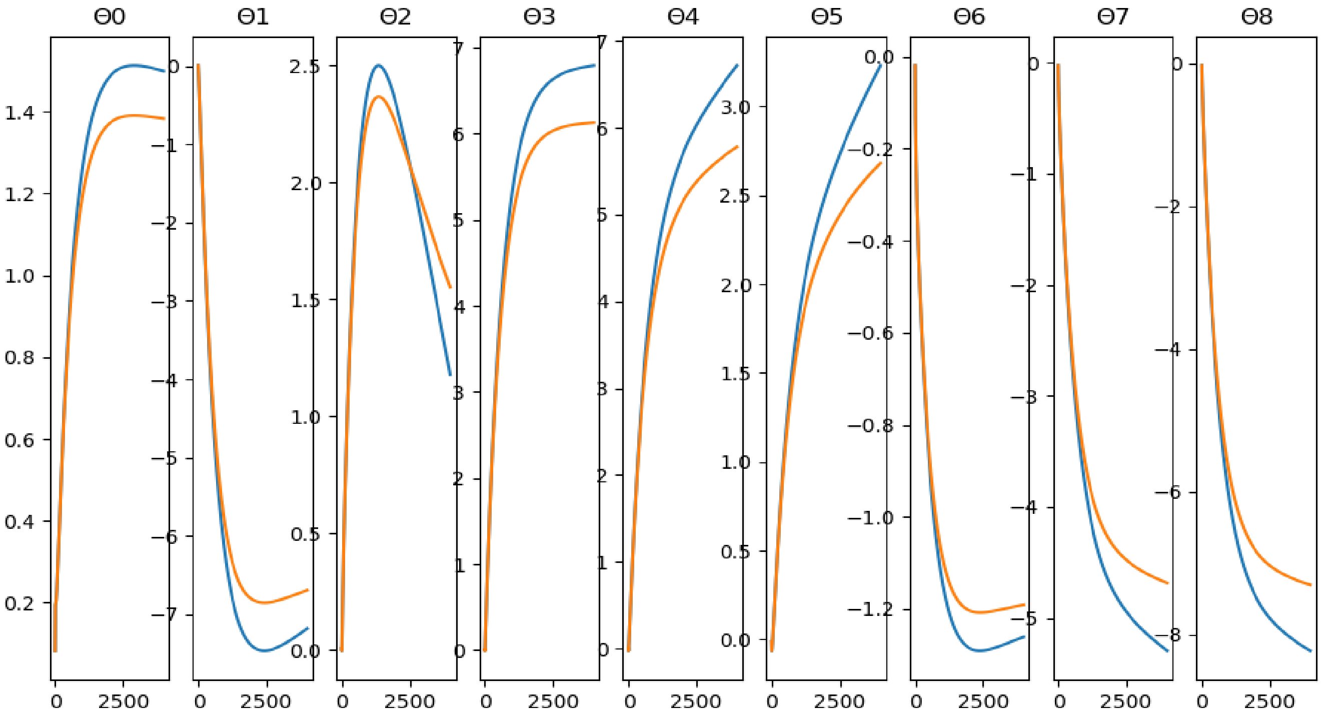
2)

Approximately, appropriate values for parameters of Iteration, Dimension, Learning Rate and Regularization Parameter are 3000, 8 (but 18 would be perfectly fitted!), 0.8 and 0.05 respectively.

a) In the following figure from left to right there are ***plots of Θ_0 to Θ_8*** for both gradient descent with and without regularization. The **Regularization Parameter is 0.05** for these plots and other values are as follows: (Iteration: 3000, Dimension: 8, Learning Rate: 0.8, Theta_Initial_Value: 0)

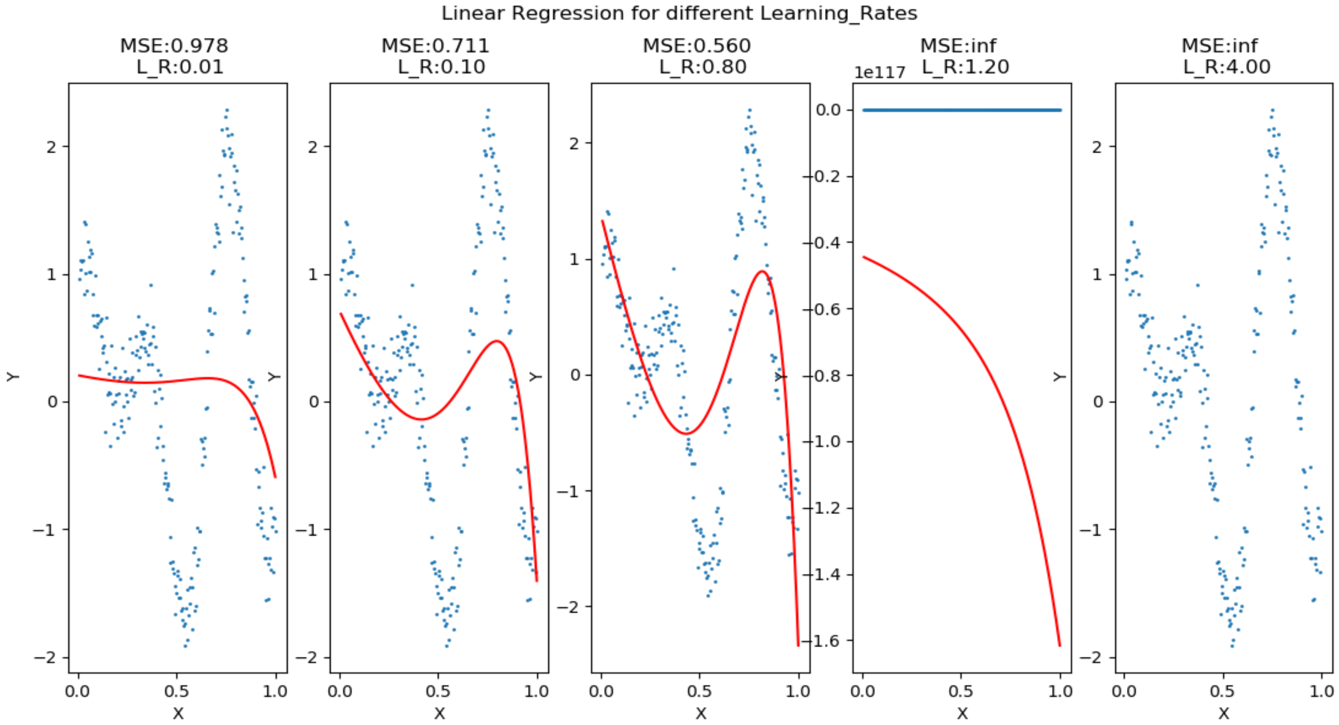
Yellow Lines: Regularization_Param = 0.05
Blue Lines: Regularization_Param = 0

"Theta vs each Iteration"

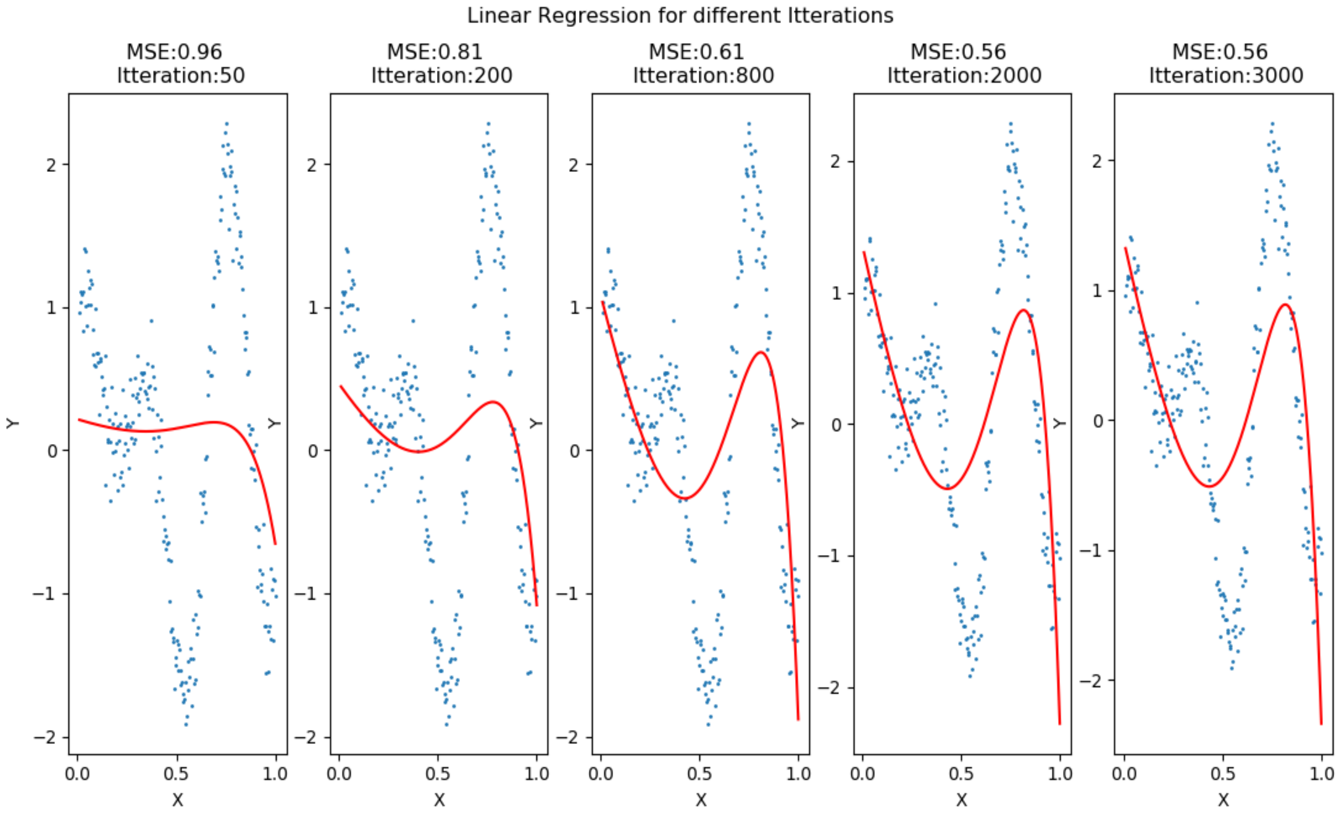


As it is obvious from the graph, the values of some of the thetas (***Θ_1, Θ_2, Θ_6, Θ_7, Θ_8***) has increased and others(***Θ_0, Θ_3, Θ_4, Θ_5***) have decreased after implementing Regularization.

b) The following graph shows gradient descent linear regression for 5 different values of Learning Rate represented by L_R notation on top of each plot with their related MSEs' as well. As we can see the best fit as well as lowest MSE corresponds to a L_R value which is not too high or too low but something intermediate. The L_R of 0.8 seems to be **acceptable** as against values of 0.01, 1.2 or 4. The value of 0.1 also has resulted in an acceptable plot but 0.8 has lower MSE. It also should be noted that values greater than 1 seems to diverge which has led to infinite MSEs! (other values are as follows: Iteration: 3000, Dimension: 8, Reg_Par:0.05, Theta_Initial_Value: 0)

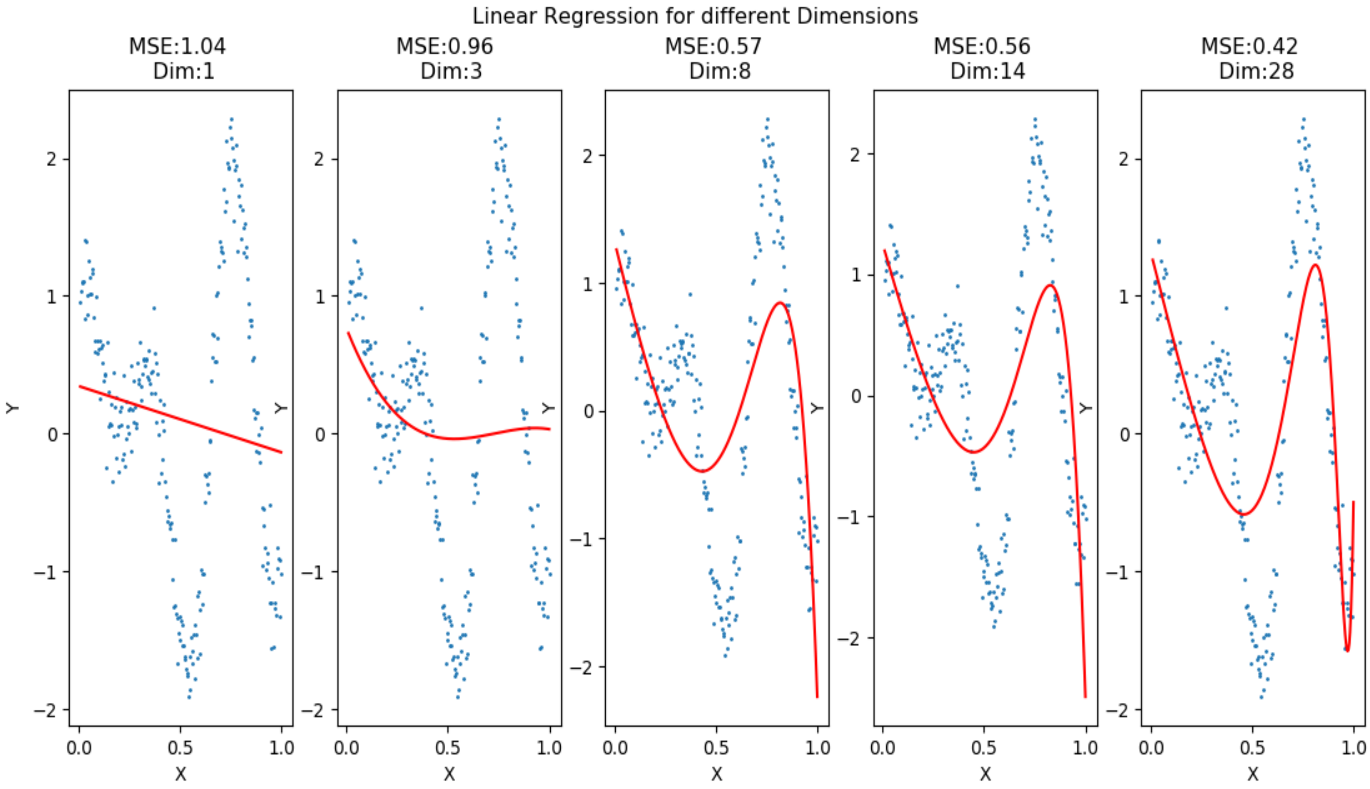


c) As we can see in the following graph, gradient descent linear regression has been plotted for different values of Iteration(50, 200, 800, 2000 and 3000) Other constant values are as follows: (Learning Rate: 0.8, Dimension: 8, Reg_Par:0.05, Theta_Initial_Value: 0)



As it is obvious from the plots' variation as we increase the iteration number the linear regression fits better and MSE decreases to a specific value from which increasing the iteration number further hasn't any noticeable effect on our results. As an instance in the graph above we can't find any sensible variation based on changing iteration number from 2000 to 3000.

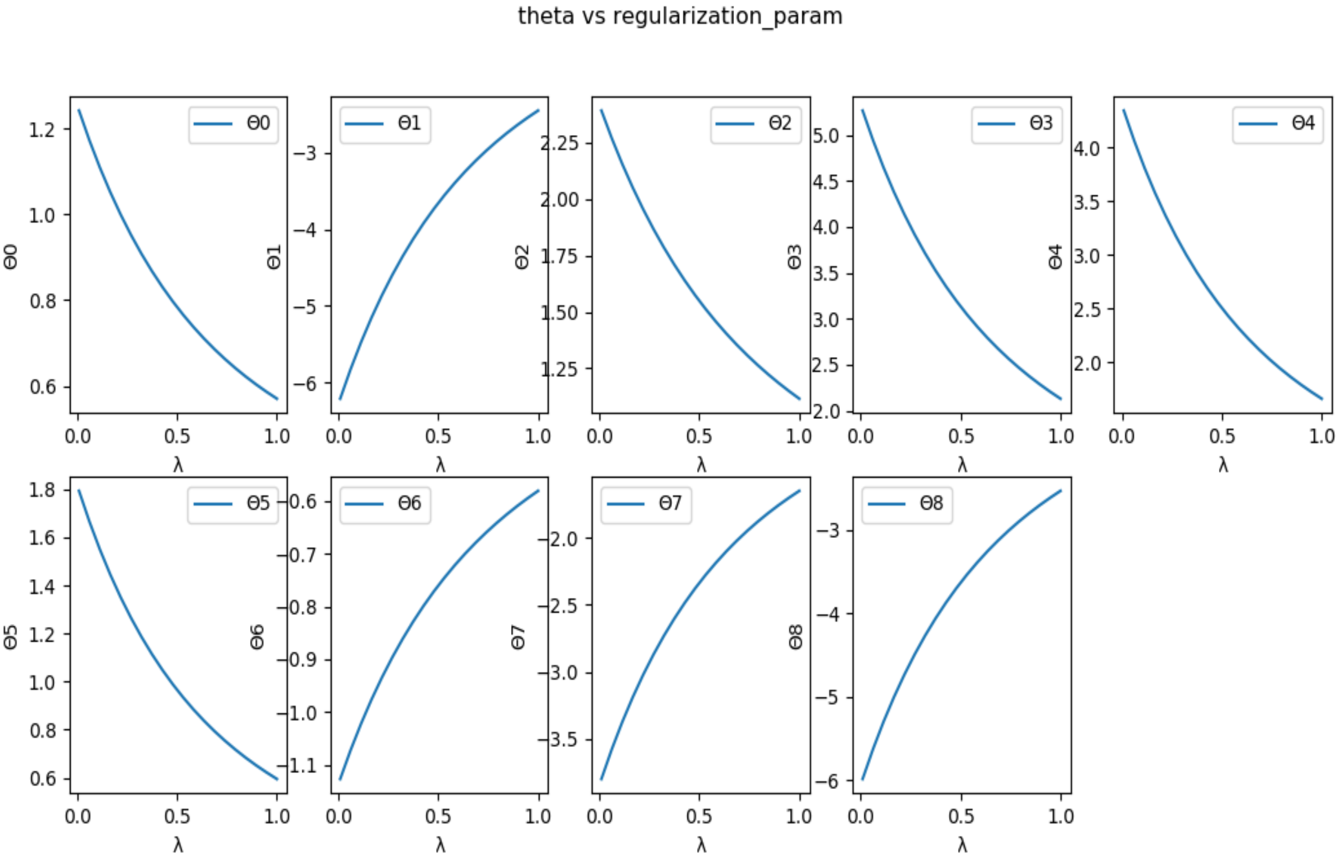
d) For this problem 5-dimension values have been tested such as (1, 3, 8, 14 and 28) which are mentioned in the following graph using notation of "Dim" over each plot. As we can see rising Dimension's value results in lower MSE and better fit to the training set but it should be noticed that increasing Dimension more than something logical it can result in higher risk of overfit and Variance of the model which will lose predicting newly given data. Therefore, choosing a logical and intermediate value is highly recommended. Among the following plots dimension 8 seems to be the best.



3)

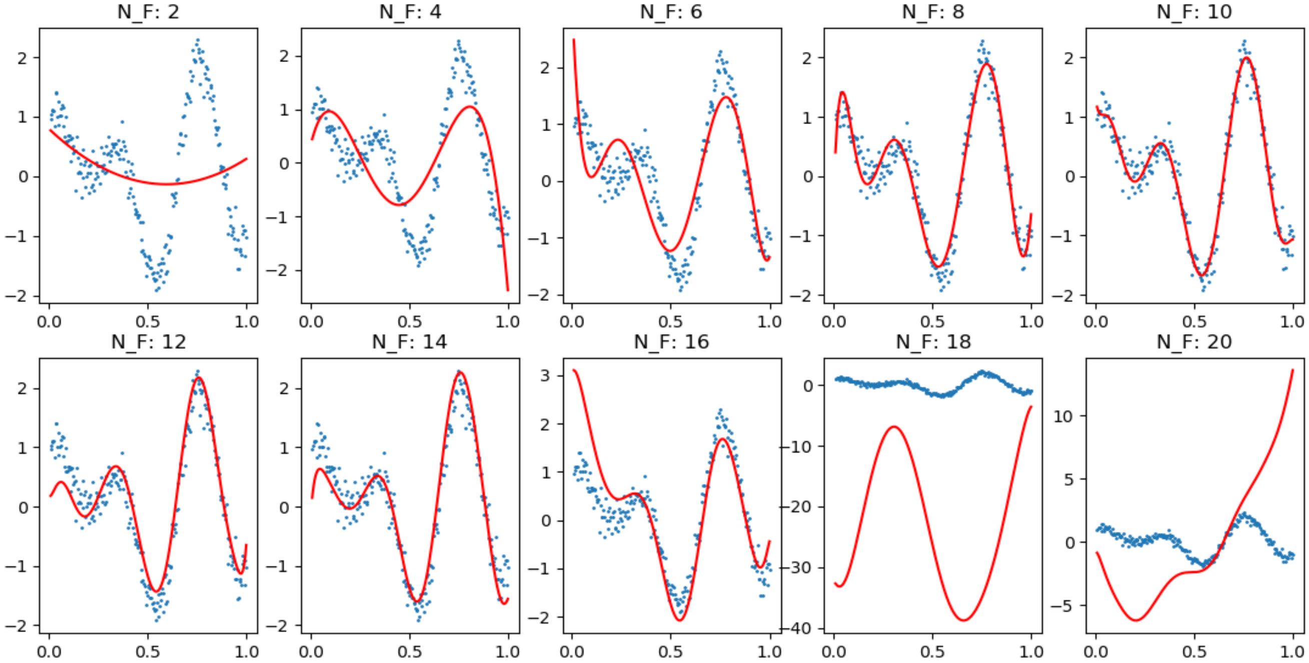
In the following graph, results of variations of each theta ("Θ") is plotted based on variation of Regularization Parameter ("λ") with respect to the following constants: (L_R: 0.8, Iteration: 1000, theta_initial: 0, Dimensio:8)

*It should be noted that variation of "λ" has been generated using ("Input = np.linspace(1 , 0.01 , N) ") command that "N" is user's input and also the list "Input" is a list of defined values of "λ".



4)

In the following graph 10 figures are plotted based on number of features concerning Normal Equation. Each plot's number of features is noted over it by the "N_F" notation. As it can be figured out, obviously, number of features of 8 and 10 has the best fit. However, in some degree we can tell that they may overfitted and 6-feature-number may be a better choice for future predictions. Any value over or under 6 to 10 is either underfitted or has remarkable prediction error.


N_F: 2
N_F: 4
N_F: 6
N_F: 8
N_F: 10
N_F: 12
N_F: 14
N_F: 16
N_F: 18
N_F: 20
0.0
0.5
1.0
−2
−1
0
1
2
3
−10
−20
−30
−40
−5
5
10