**Assignment 2 Report**

For this assignment I have used the below to approach

1) Training the linear regression model from scratch

2) Training the regression model to fit the polynomial function

**Description of File:**

Akash\_Assignment2\_Approach1.py

Training the linear regression model from scratch

Akash\_Assignment2\_Approach2.py

Training the regression model to fit the polynomial function

**Approach 1**

Training the linear regression model.

**Concept:**

Our model is based on Y=βTX

The input data is considered as **Independent Variable and the out variable or target variable is** called **Dependent Variable**. The input variable in our case is the four features of the iris data set and output or target variable is label as 0 1, and 2

Data split: used **train\_test\_split** method to divide the iris data in to train and test data.

I have used 2/3 of the data the train the model and 1/3 of the data to test the model. Since we required more data to train the model properly as well as enough data to have a reliable prediction so I have divided the iris data into train and test in ratio 2:1 respectively.

We can define this linear relationship as follows:

Y=β0+β1X1+B2X2+B3X3+B4X4

We created a regression model to train our beta coefficients which were initialized to zero.

Used gradient descent algorithm to get the new values of beta and hence minimize cost.

And the cost is,

J(β)=12m∑i=1m(hβ(x(i))−y(i))2J(β)=12m∑i=1m(hβ(x(i))−y(i))2

**Steps/Flow:**

* The program execution begins with the invocation of train\_regression\_model.
* In this function we load the data from iris dataset from sckit a split it into train and test.
* Initialed the beta confidents as 0 and convert them into matrix using numpy.
* calculated the cost at each step and minimized it using gradient descent algorithm for this purpose I have called optimization\_function.
* Calculated the output target variable and with the new beta values we predicted the output variable in test and ploted it using scatter plot.

Please find the attached screenshot with the zip file attached in canvas

**Approach 2:**

2) Training the regression model to fit the polynomial function

In this approach I followed the below steps:

Imported the required libraries in python, created a classification class which is used to classify the test data into three output categories and plot it using scatter plot.

**Train and test split**

Data split: used **train\_test\_split** method to divide the iris data in to train and test data.

I have used 2/3 of the data the train the model and 1/3 of the data to test the model. Since we required more data to train the model properly as well as enough data to have a reliable prediction so I have divided the iris data into train and test in ratio 2:1 respectively.

**Steps/Flow:**

* Initialized the input and output variable using a constructor call.
* Invoked regression train model used
* I have used NumPy library to fit the training data in a polynomial function. Used polyfit and polval function to train the train the coefficients of polynomial and to test it with test data
* With the new beta values we predict the output variable in test and plot it using scatter plot.

Please find the attached screenshot with the zip file attached in canvas

**References:**

<http://scikit-learn.org>

https://en.wikipedia.org/wiki/Linear\_regression