Information Engineering and Technology Faculty
German University in Cairo

Assignment 1

NETW 1013: Machine Learning

Submitted by:

Samar Aly Saad 40-20130

25^{th} April 2021.

First, the values of the dataset were displayed from the (CSV) file using the read_csv function from pandas. Then, we dropped the rows where at least one element is missing (NaN) while keeping the data frame with valid entries in the same variable using df.dropna (inplace=True). Finally, we removed the first row (the header row), the first column (id) and the second column (date) and converted the values of the dataset to float.

Feature Normalization:

In gradient descent algorithm, the results of calculating the cost after every gradient descent step will also vary a lot as the input values differ by order of magnitude. In this function, we subtract the average value (mu) from the input variable and then divide it by standard deviation (sigma) of input variable to avoid the variation of results by changing the range of our input variables. The standard deviation is a way of measuring how much variation there is in the range of values of a particular feature.

Gradient Descent:

This function is important to fit the linear regression parameters θ to our dataset using gradient descent and to minimize the cost function where the hypothesis is given by the linear model.

- m = no of training examples (no of rows of feature matrix)
- x's = input variables / independent variables / features
- y's = output variables / dependent variables / target
- J = the values of the cost function after each iteration.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

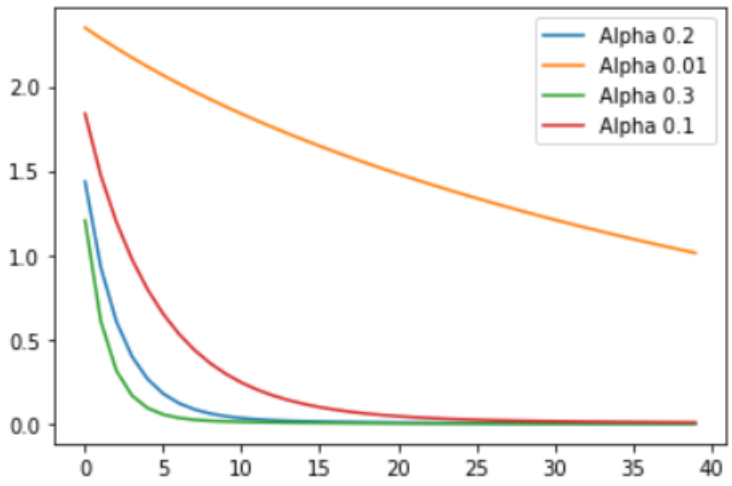
$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

This function returns theta θ and the cost J in every iteration.

Cost Function:

This function is helpful to monitor the convergence by computing the cost of a particular choice of theta.

We used the implementation of Gradient Descent function for about 40 iterations and a list of random values of alpha [0.2, 0.01, 0.3, 0.1] to get the best value of alpha then we plotted the values of cost against no of iterations to visualize the performance of the Gradient Descent Algorithm. As shown in the figure, With a small learning rate (alpha = 0.01), gradient descent takes a very long time to converge to the optimal value. As we increase the alpha value, slope becomes sharp and gradient descent will take less time to converge. the value of learning rate(alpha = 0.1) is too large then gradient descent may not decrease on every iteration, may even diverge. So , we used the learning rate (alpha = 0.1) to get the values of theta θ.



Once we have found θ, we can use it to get the price of the house using random values of x from the dataset and mu/sigma to predict the price

h(x) = theta_0 + theta_1 * x_1 + theta_2 * x_2 .....theat_n * x_n
where x_1, x_2...x_n are multiple input values

As we normalized the data to use the gradient descent algorithm. we have to also normalize the given input data before any prediction. So we have to add column of ones to input data the way we have added to input feature X using the hstack function

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

K Fold and Cross validation:

Cross validation is a process in which the original dataset is divided into two parts only- the ‘training dataset’ and the ‘testing dataset’. Training

dataset is used as a parameter to the given machine learning algorithm, to be trained upon.
Validation dataset is used to evaluate the model, Later the result is checked, and if the results are not appropriate, the hyperparameter value can be changed and it can be tested on the validation set.
The current training dataset would now be divided into ‘k’ parts (k is 5, 10 or any number which is less than the dataset’s length) out of which one dataset is left out and the remaining ‘k-1’ datasets are used to train the model and this is done multiple number of times. The one that was kept out of the training is used as a ‘validation dataset’. Then we apply the linear regression techniques on the train and test sets. We can estimate the test error from the polynomial regression.