Simulation Example on Designing and Assessing a Regression Function
By: Safiia Mohammed

1 Introduction

The purpose of this report is to simulate, design and assess a Regression Function. Firstly, the sample of 10 observations from Bi-normal distribution is generated, fitted by Regression model and assessed. The same data also have been fitted by calculating the conditional expectation function which denoted as the best regression function. In the second part, the same experiment is repeated with 1000 observation (represent the population). The third part of this report is performed using Monte-Carlo simulation considering different sizes of training set .

2 Generate Sample (10 observations) from Bi-normal distribution

Sample of 10 observations is generated using bi-normal distribution, we want to generate the pair (X,Y) which are the predictor and response, with mean [0,0] and covariance [[1.0,0.8],[0.8,1.0]]. Figure (1) shows the pair distribution. [1]

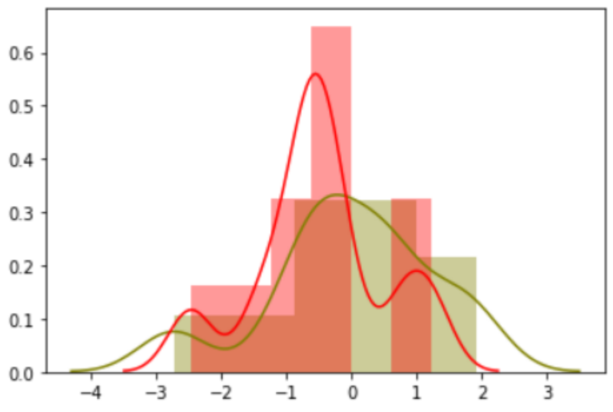

Figure 1: 10 observations from Binormal distribution

The data is fitted into regression model ,predicted and compared with actual data as shown in figure (2):

we can observe that, the model performs well in general. The MSE of this model is 0.226.
Now we try to calculate the conditional expectation function(CEF) for bivariate normal distribution given by: $\mathrm{E}[Y|X=x] = \mu_Y + \frac{\sigma_{YX}}{\sigma_X^2}(x-\mu_X)$

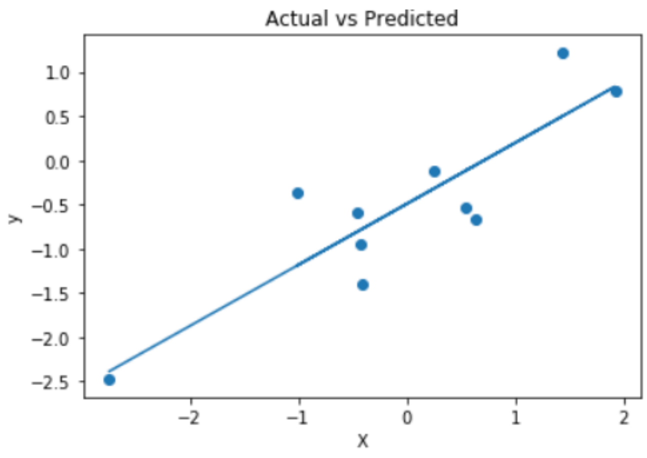

Figure 2: Actual and predicted data for 10 observations

Figure (3) illustrates the model given by this function. the MSE of CEF is 0.235, we can notice that from the models(figure 2 and 3) and from the MSE , these two models are very close to each other.

NOTE: in the first model we use the build-in model in sklearn library, the regression model can be written from scratch, however i preferred to use built-in model to compare it with CEF performance to measure to how extend CEF function can perform well, bear in mind that the sklearn usually chose the best algorithm / function to model the data, so we can conclude that the conditional expectation function(CEF) performs well!

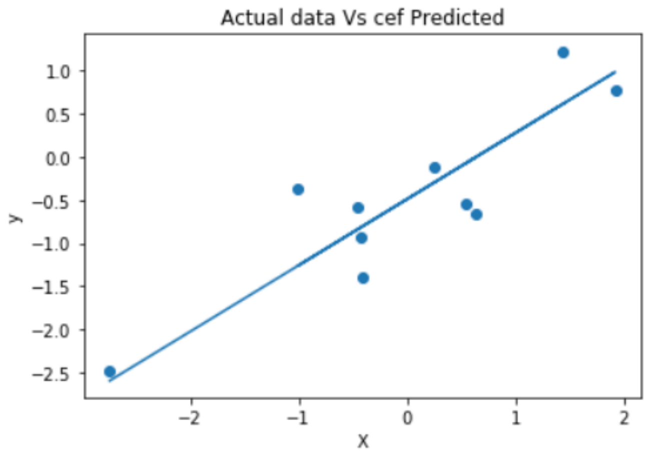

Figure 3: Actual and predicted data for 10 observations using CEF function

let us plot the two models in the same plot to compare between them clearly.

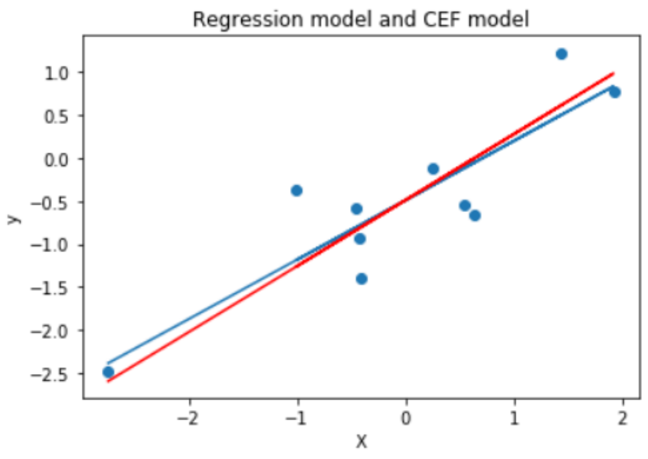

Figure 4: Regression model and CEF model

The performance of the two models are so closed (the red line is CEF function where as the blue is the regression model)

3 Represent the population with 1000 observations from Binormal distribution

In the second part, we want to simulate our population with 1000 observations generated from the same distribution, see figure 5:

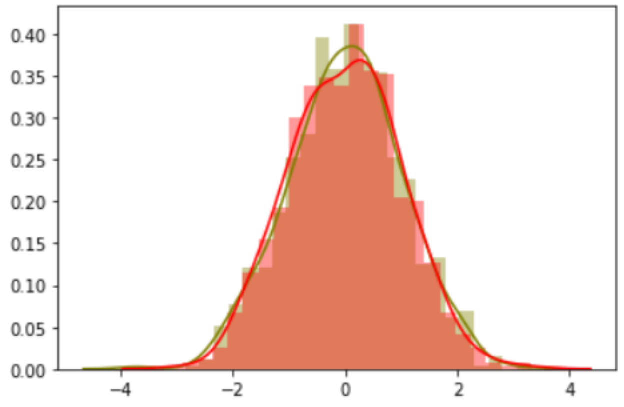

Figure 5: Distribution of 1000 observations

The data is fitted into regression model, predicted and scored 0.354 as MSE. Next step is to predict the target using conditional expectation function and

compare the two models.
By predicting using conditional expectation function, we get 0.354 as MSE, looks the same MSE! . Now figure (6) shows the plot of two models for 1000 observation. (the red line is CEF function where as the black is the regression model)

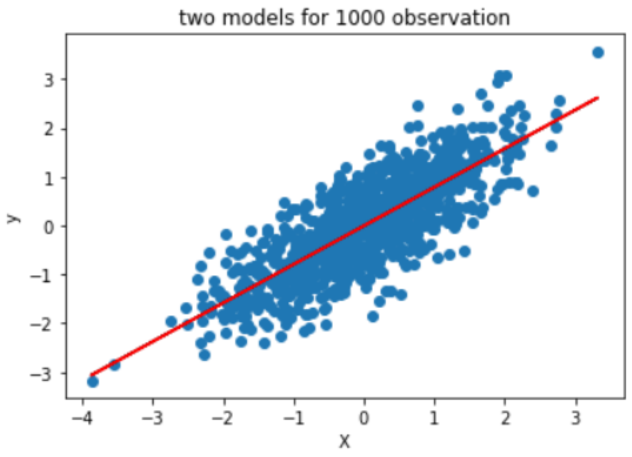

Figure 6: Regression and CEF models for 1000 observations

From figure (6) and the result of MS errors for the two considerable models, we can conclude that, the performance of the best regression function is increasing when the number of observations increased, to perform mostly the same as built-in regression models.

4 Monte-Carlo Simulation

In this part we want to run many scenarios with different random inputs - instead of generate only one sample - and summarizing the distribution of the results by getting the mean and the variance -. Regarding to that, we want to consider more samples to better represent the population as can as possible.[2][3]
We test 500 samples of size 10 (in gray) and the best regression function(in black) as illustrated by figure (7):

We can notice that, the best regression function is seem to be like an average (located at the middle of the plot) for these simulations, in other words it can represent the different simulations of the distribution,on the other hand, the mean and variance, the models scored 0.435 for the mean and 0.01 variance . From this we can see that, the variances between the models is small that means these samples can represent the distribution very well.

In the last part of the report, we consider different sizes for the training set: [20, 40, 80, 100, 200, 300, 400, 500, 700, 1000]. The purpose is to observe the performance of these different sizes. the next plot shows the expectation of the error and variance for each training set, as well as the performance of the best regression function.

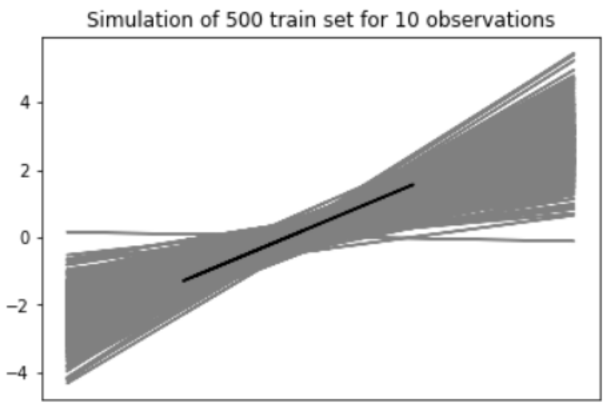

Figure 7: Monte-Carlo Simulation

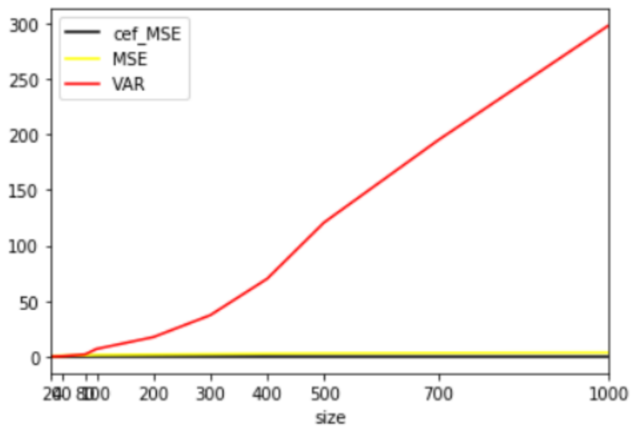

Figure 8: MSE, Variance and MSE of CEF

From above figure, the performance of the best regression function is very close to the performance of the model with different training set sizes, by performance here we mean the MS error(MSE), which is small for all different sizes and mostly the same. The noticeable difference comes in the variance, different sizes have different variances, and the variance increased dramatically when we increase the sample size as figure (8) shown.

5 Conclusion

In conclusion, we observed that the performance of the best regression function which is the the conditional expectation of a bi-variate normal is very close to what we use in the models library like sklearn. On the other hand, Monte-carlo simulation is good in representing different samples (with the same or different size) because it can give a summery of the distribution by considering more than one sample of observations.

6 References

References

[1] ttps://het.as.utexas.edu/HET/Software/Numpy/reference/generated/numpy.random.multivariate_normal.html ,
Access: 20, May , 2020

[2] https://www.datacamp.com/communitys/tutorials/tutorial-monte-carlo ,
Access: 21, May , 2020

[3] https://towardsdatascience.com/monte-carlo-simulations-with-python-part-1-f5627b7d60b0 ,
Access: 23, May , 2020