Learning to Rank using Linear Regression

By Monica Vashu Kherajani

Person #50290424

mkheraja@buffalo.edu

Machine Learning | Project 1.2 | 10th October, 2018

Part I: Problem statement

For training a linear regression model on LeToR dataset using the following equation,

$$y(\mathrm{x}, \mathrm{w}) = w^T \phi(\mathrm{x}) \quad \text{.... (1)}$$

We obtain basis function ($\phi(\mathrm{x})$) from the input data($\mathbf{X}$) using Gaussian radial basis function and calculate weights (w^T) using two different solutions:

1. Using a closed-form solution:
   For closed-form solution, we use the Moore-Penrose pseudo-inverse for the matrix ϕ and calculate weight vector using:
   $$w^* = (\lambda I + \phi^T \phi)^{-1} \phi^T t \quad \text{.... (2)}$$

2. Using stochastic gradient descent (SGD):
   In Gradient descent solution, we iteratively update the weights by considering one sample at a time using:
   $$w^{\tau+1} = w^\tau + \Delta w^\tau \quad \text{.... (3)}$$

Part II: Conceptual and Technical understanding

For closed form solution, the steps followed are:

1. For finding the appropriate number of clusters to form from the data (input values) we use k-means algorithm. This is also the number of basis functions.
2. We introduce the non-linearity into the model w.r.t. data (input values) by using the Gaussian radial basis function.
3. For equation (2), we calculated basis functions using gaussian radial basis vector form.
4. And hence, calculated the weight vector.

The closed form derivation is as follows:

We have the root mean square error as follows:

$$E(w) = \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T \phi_n)^2$$

$$= \frac{1}{2} (t't - 2\, w'\phi' t + w'\phi'\phi w)$$

We differentiate this equation w.r.t. to weight(w) and get,

$$\phi'\phi w = \phi' T$$

On this equation, we use the Moore-Penrose pseudo-inverse and get equation (2).

For gradient descent solution, the steps followed are:

1. We update weights after every data (input value/ a row) is iterated.
2. The error function is differentiated w.r.t. weights and the update are performed using equation (3).
3. In this algorithm, we have the flexibility of choosing the number of data points we use for training. Thus, if we infer from the iterative updates that the accuracy of the model is not improved by training on more points, we can stop to iterate over rest of the data points.

The gradient descent solution is as follows:

$$\nabla \mathrm{E} = \nabla E_D + \lambda \nabla E_w$$

Where $\nabla E_D = -\left(t_n - w^{\tau^T} \phi(x_n)\right) \phi(x_n)$

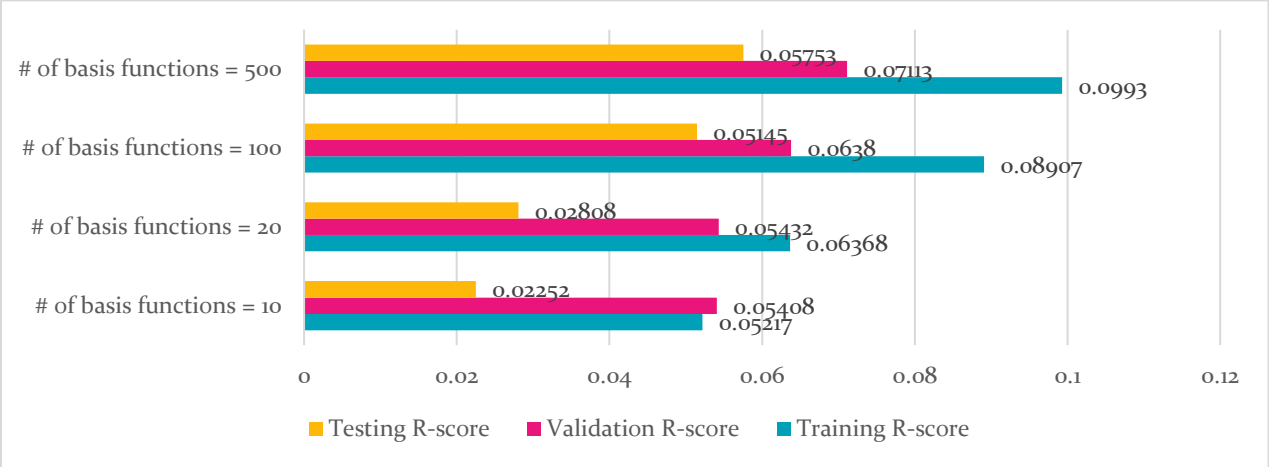
And $\nabla E_w = w\hat{}\tau$

Part III: Experiments and Results

Closed Form Solution:

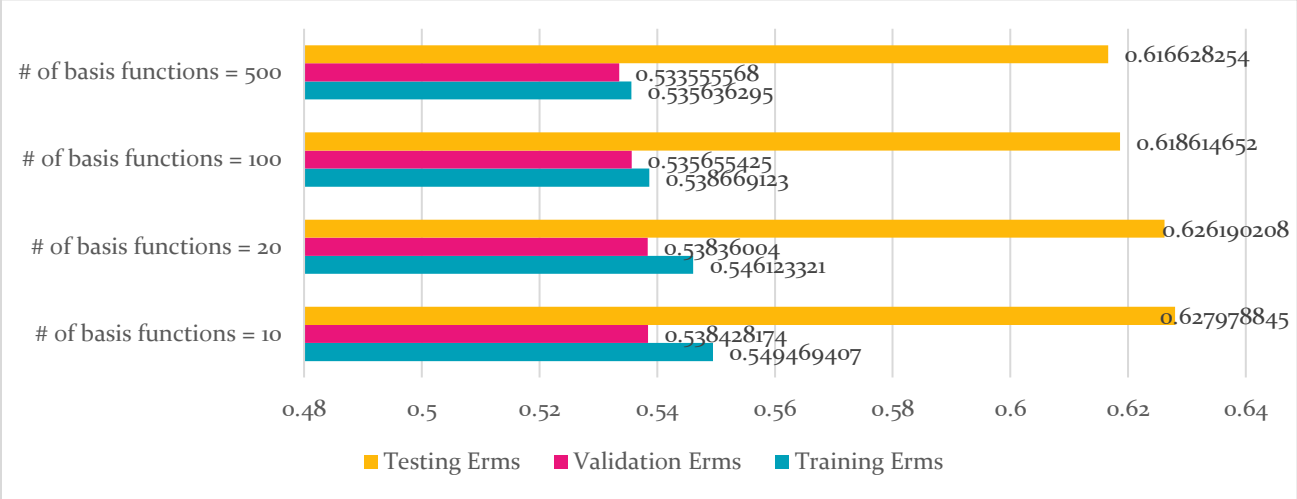
1. Observed R-Squared (the coefficient of determination) between the predicted and the actual value for different number of basis functions.

lambda = 0.03	# of basis functions = 10	# of basis functions = 20	# of basis functions = 100	# of basis functions = 500
Training R-squared	0.05217	0.06368	0.08907	0.0993
Validation R-squared	0.05408	0.05432	0.0638	0.07113
Testing R-squared	0.02252	0.02808	0.05145	0.05753



2. Observed Root mean square error between the predicted and the actual value for different number of basis functions.

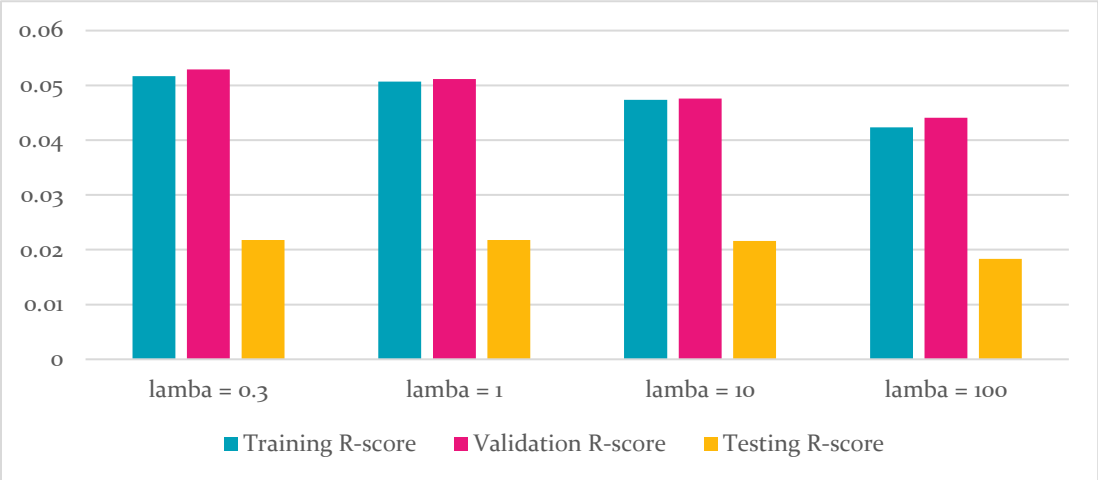
lambda = 0.03	# of basis functions = 10	# of basis functions = 20	# of basis functions = 100	# of basis functions = 500
Training Erms	0.549469407	0.546123321	0.538669123	0.535636295
Validation Erms	0.538428174	0.53836004	0.535655425	0.533555568
Testing Erms	0.627978845	0.626190208	0.618614652	0.616628254



3. Observed R-Squared (the coefficient of determination) between the predicted and the actual value for different values of lambda.

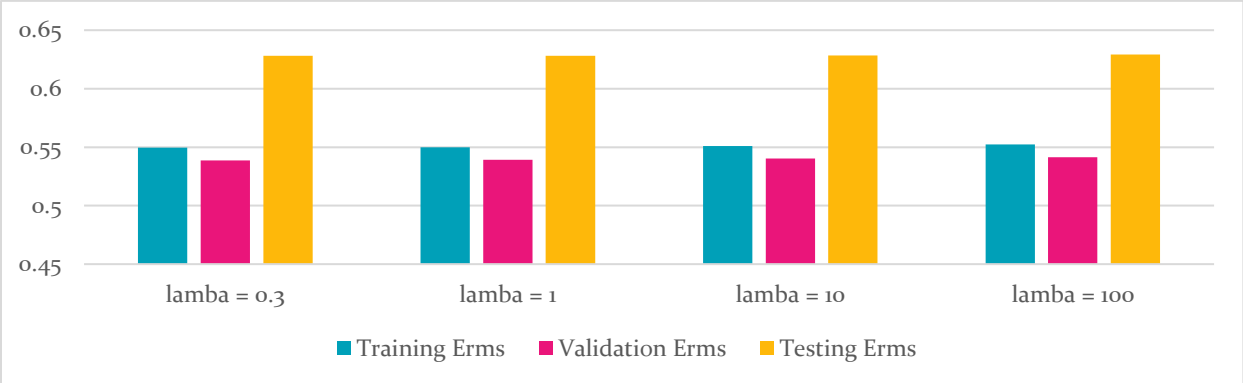
# of basis functions = 10	lambda = 0.3	lambda = 1	lambda = 10	lambda = 100

Training R-squared	0.05166	0.0507	0.04736	0.04232
Validation R-squared	0.05291	0.05117	0.04759	0.04408
Testing R-squared	0.02179	0.02175	0.02163	0.01831



4. Observed Root mean square error between the predicted and the actual value for different values of lambda.

# of basis functions = 10	lambda = 0.3	lambda = 1	lambda = 10	lambda = 100
Training Erms	0.549618403	0.549896478	0.550863418	0.552316907
Validation Erms	0.538763563	0.539256063	0.540273066	0.541267205
Testing Erms	0.628213475	0.628224734	0.628264931	0.629329195

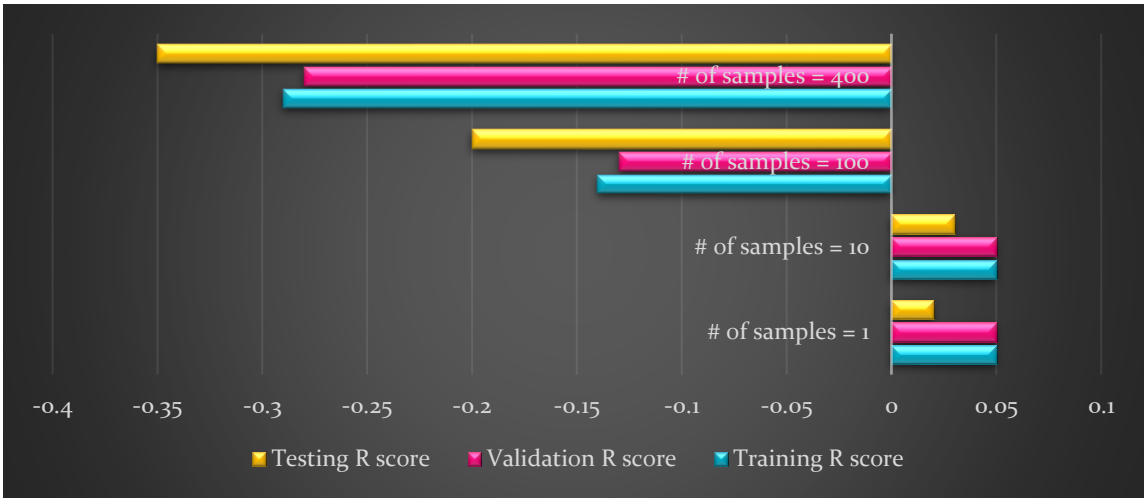


Gradient Descent Solution

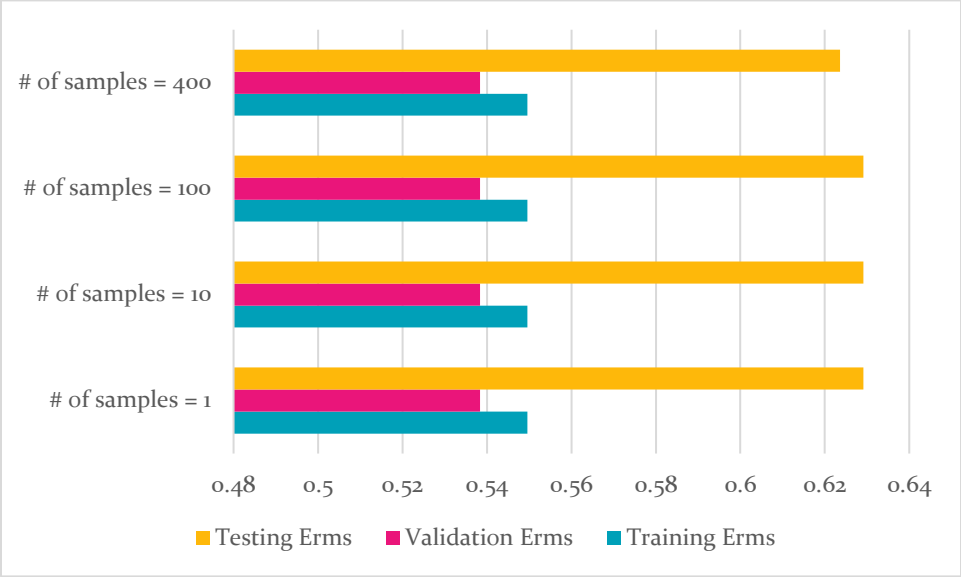
1. By initializing weight to the vector obtained from the closed form solution, we check for different number of samples how the value of R-squared (the coefficient of determination) and Root mean square error changes.

Other parameters
W
M = 10
Lambda = 2
learningRate(eta) 0.01

	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training R squared	0.05	0.05	-0.14	-0.29
Validation R squared	0.05	0.05	-0.13	-0.28
Testing R squared	0.02	0.03	-0.2	-0.35



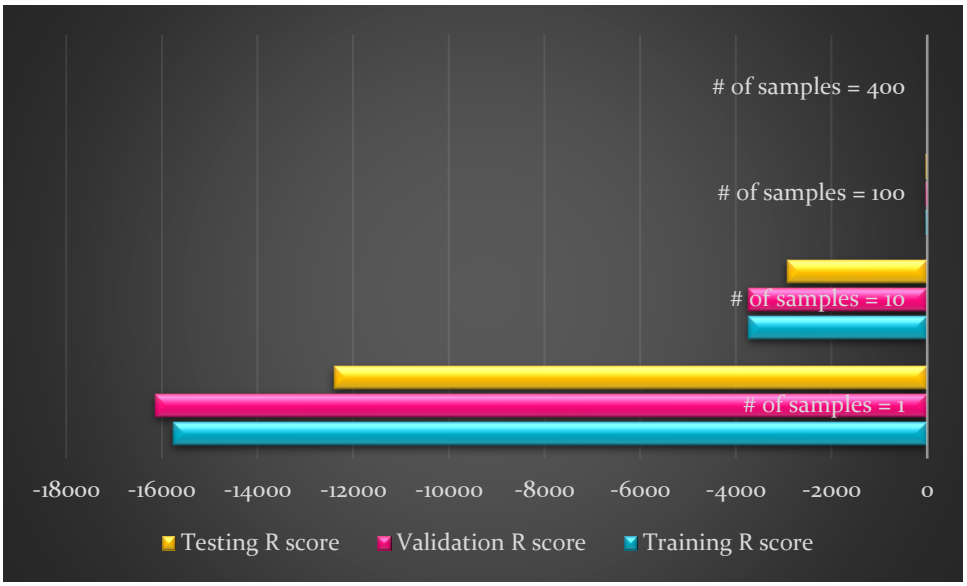
	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training Erms	0.54956	0.54956	0.54956	0.54956
Validation Erms	0.53837	0.53837	0.53837	0.53837
Testing Erms	0.62914	0.62914	0.62914	0.62364



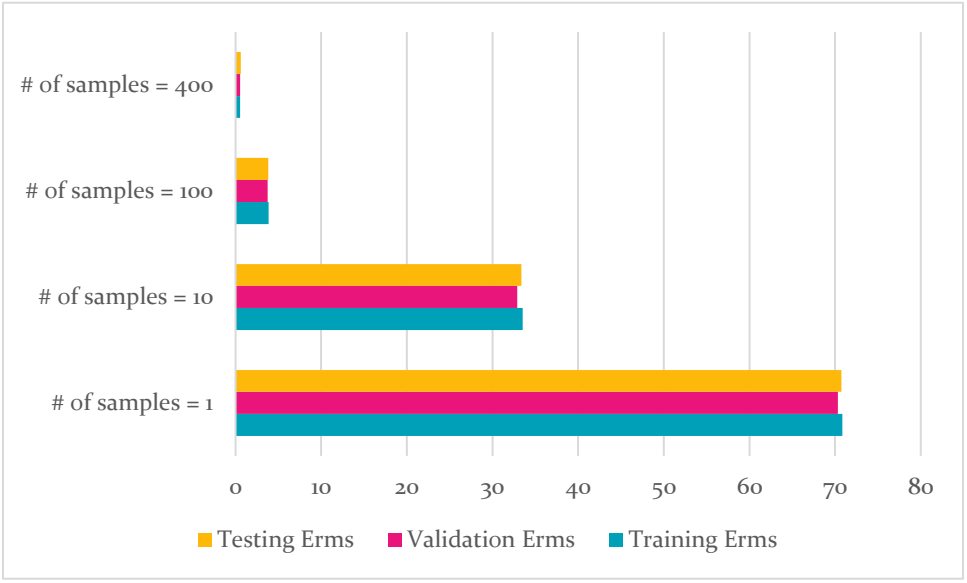
2. By initializing weight to a multiple of the vector obtained from the closed form solution, we check for different number of samples how the value of R-squared (the coefficient of determination) and Root mean square error changes.

Other parameters
W*220
M = 10
Lambda = 2
learningRate(eta) 0.01

	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training R squared	-15756.25	-3733.69	-45.51	-0.27
Validation R squared	-16132.14	-3741.27	-44.35	-0.26
Testing R squared	-12397.05	-2921.6	-35.11	-0.34



	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training Erms	70.84666	33.53403	3.84903	0.54964
Validation Erms	70.31711	32.87577	3.72792	0.53846
Testing Erms	70.724	33.37299	3.81687	0.62372

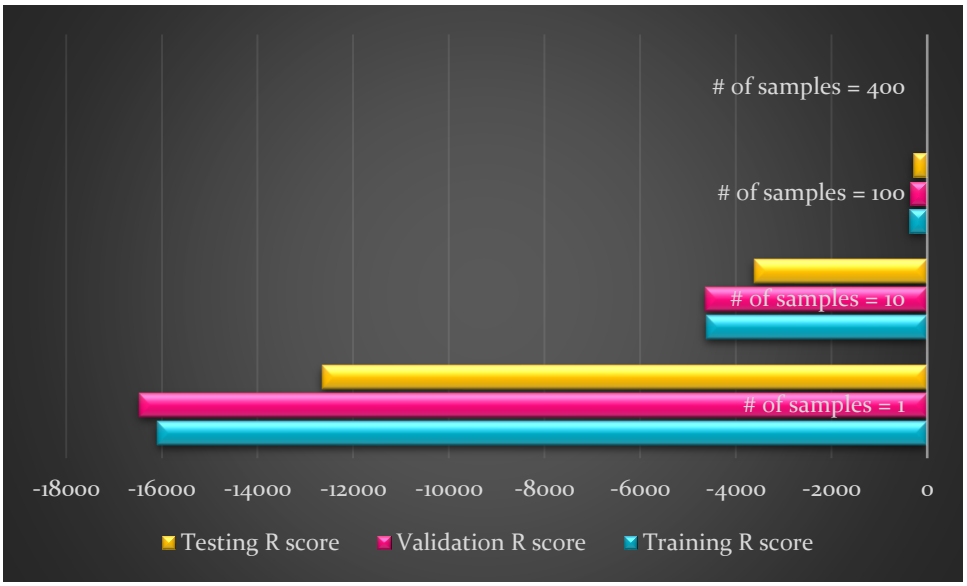


3. By initializing lambda to 1, we check for different number of samples how the value of R-squared (the coefficient of determination) and Root mean square error changes.

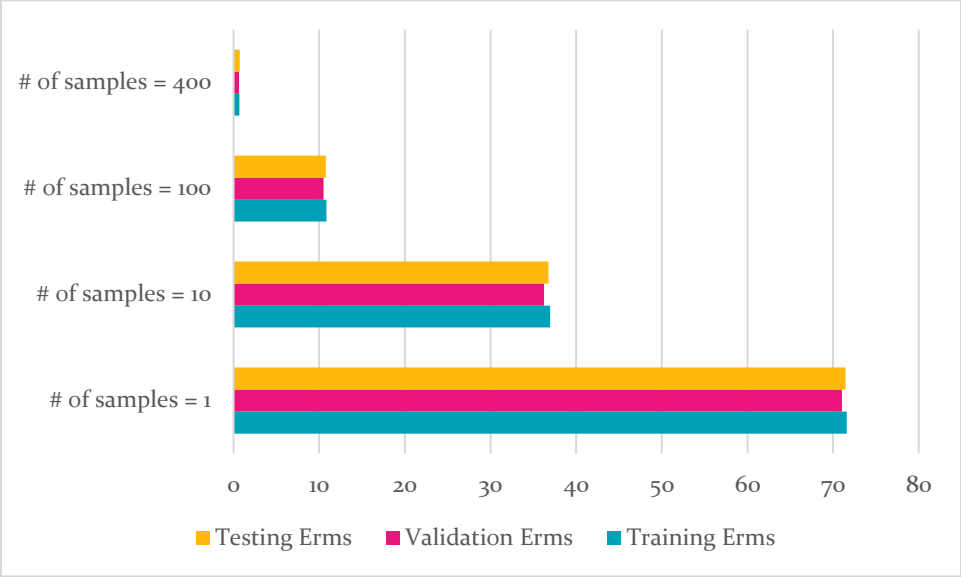
Other parameters
W*220
M = 10

Lambda = 1
learningRate(eta) = 0.01

	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training R squared	-16084.06	-4620.87	-369.2	-0.39
Validation R squared	-16467.78	-4631.42	-360.16	-0.37
Testing R squared	-12655.21	-3617.09	-286.14	-0.33



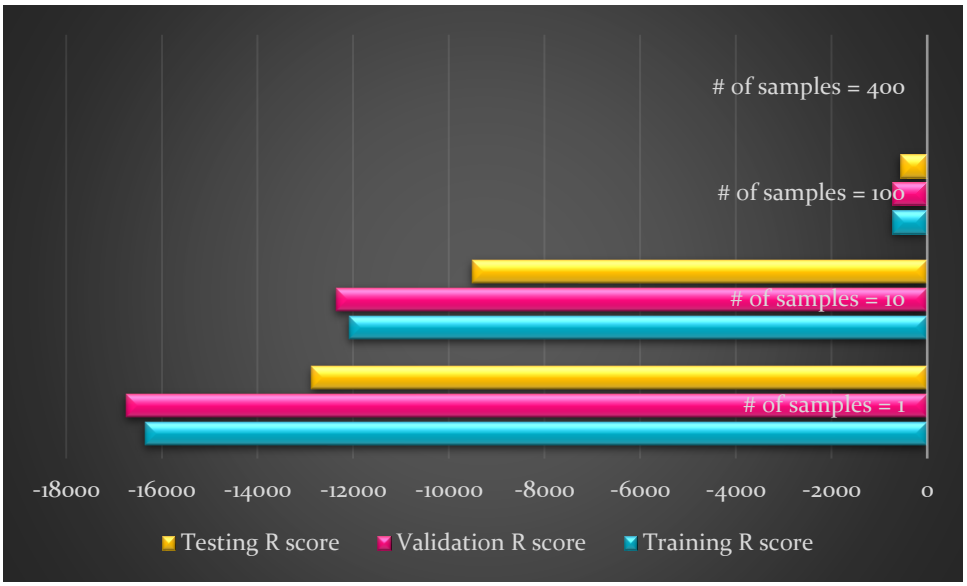
	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training Erms	71.5798	36.95731	10.85923	0.66502
Validation Erms	71.0448	36.23786	10.52085	0.64697
Testing Erms	71.45655	36.78587	10.76303	0.73303



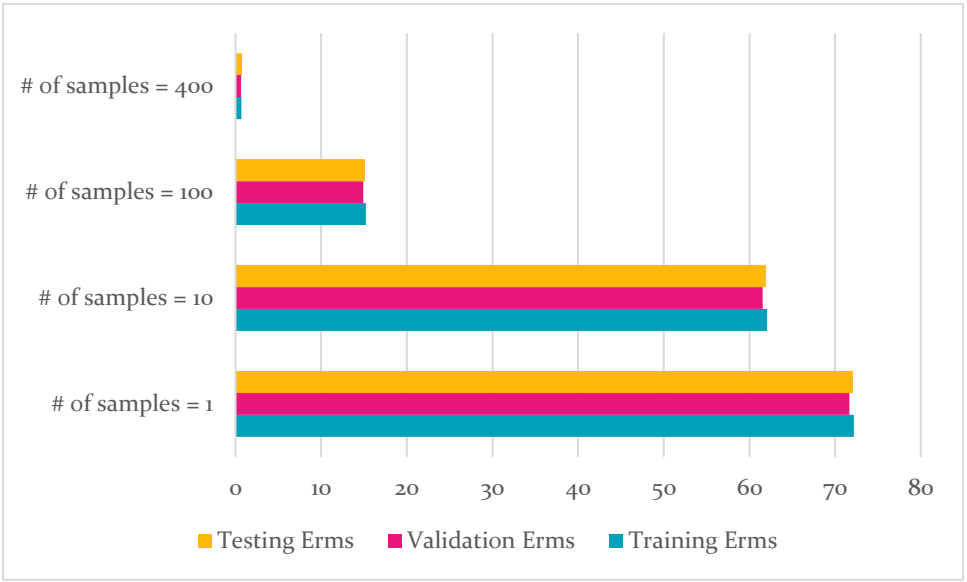
4. By initializing lambda to 1 and learning rate to 0.001, we check for different number of samples how the value of R-squared (the coefficient of determination) and Root mean square error changes.

Other parameters
W*220
M = 10
Lambda = 10
learningRate(eta) = 0.001

	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training R squared	-16357.52	-12087.61	-724.4	-0.46
Validation R squared	-16751.27	-12359.71	-725.55	-0.43
Testing R squared	-12870.88	-9506.43	-564.74	-0.43



	# of samples = 1	# of samples = 10	# of samples = 100	# of samples = 400
Training Erms	72.18569	62.05364	15.20086	0.6811
Validation Erms	71.65366	61.54932	14.92229	0.66204
Testing Erms	72.0628	61.93296	15.10767	0.75891



Part IV: Inferences and Conclusions

Closed form solution:

- The root mean square error does not decrease significantly by increasing the number of basis functions after a certain limit.
- However, increasing the number of basis functions increases the coefficient of determination (R squared / Regression score) between the input values a bit.
- Similar trend is observed with the regularization term, increasing the regularization parameter after a certain value does not decrease the root mean square error.
- And a very small increase is observed in the coefficient of determination (R squared / Regression score) between the input values even after increasing the regularization term significantly.

Gradient Descent solution:

- Significant decrease in root mean square error is observed if we increase the number of data points until 400.
- However, the coefficient of determination (R squared / Regression score) between the input values is still observed to be negative even after iterating over 400 data points. The negative R-score is the indicator that the model is not ideal for the given data.

Part V: Terms

R-squared: The coefficient of determination is the proportion of the variance in the dependent variable that is predictable from the independent variable. It is regression score function.

Part VI: References

https://en.wikipedia.org/wiki/Coefficient_of_determination