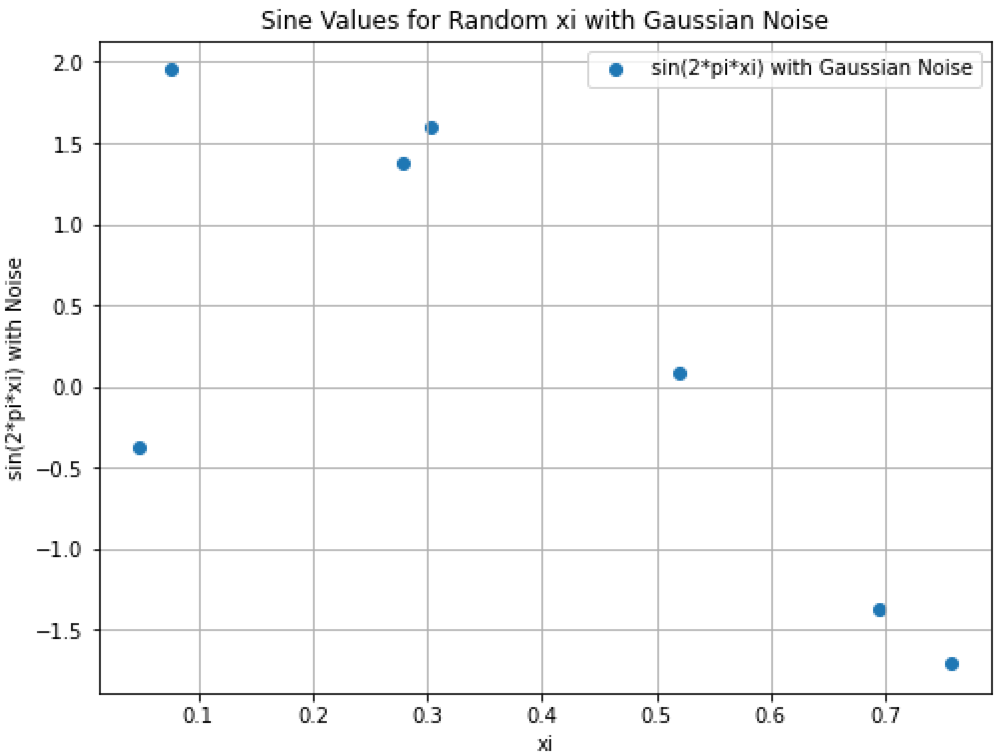
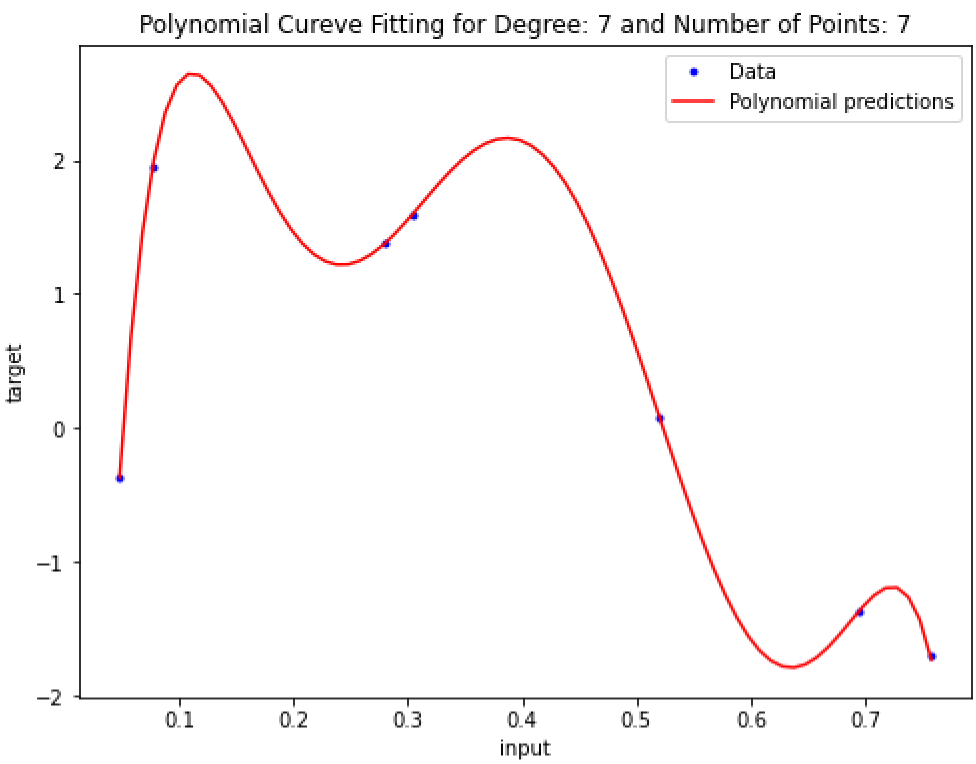
Polynomial Curve Fitting on Sine Curve with Gaussian Noise

Q1) A)
For Degree/M = 7:
For N = 7:

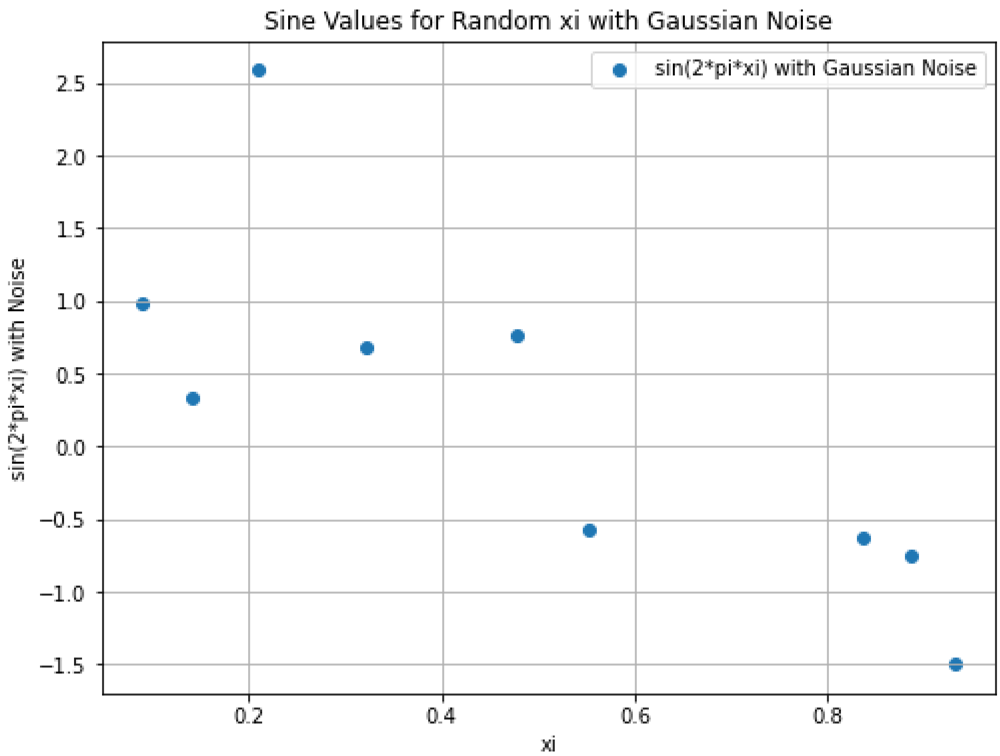


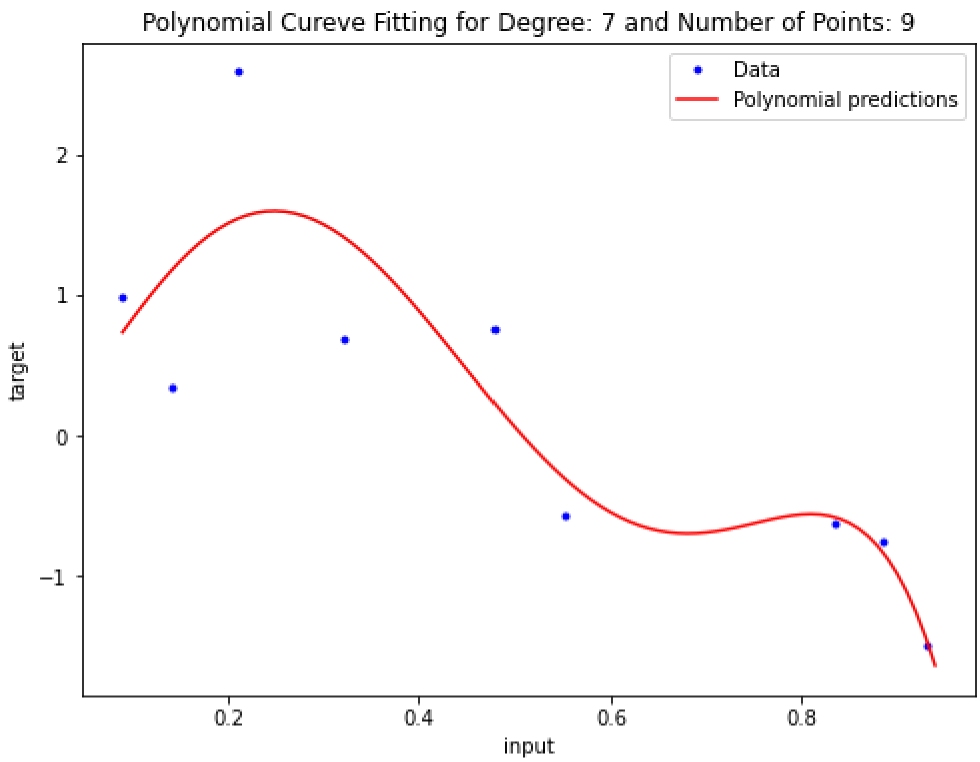


(Loss: 7.5e-15, R2 Score: 0.999)

The graph clearly overfits, as the number of parameters equals the number of data points.

For N = 9:

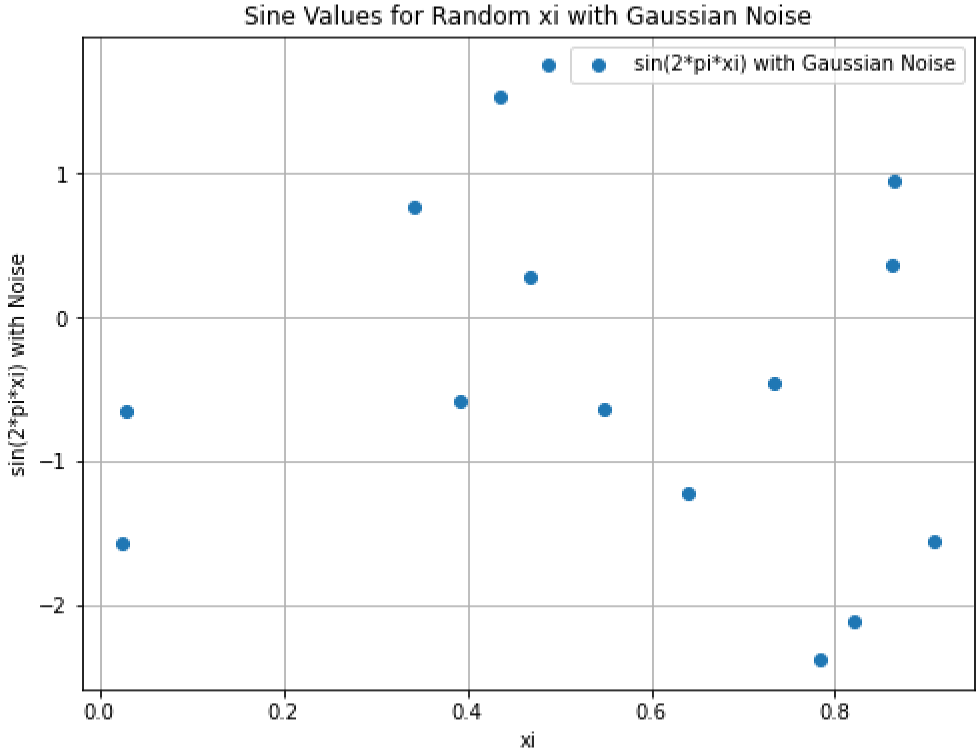


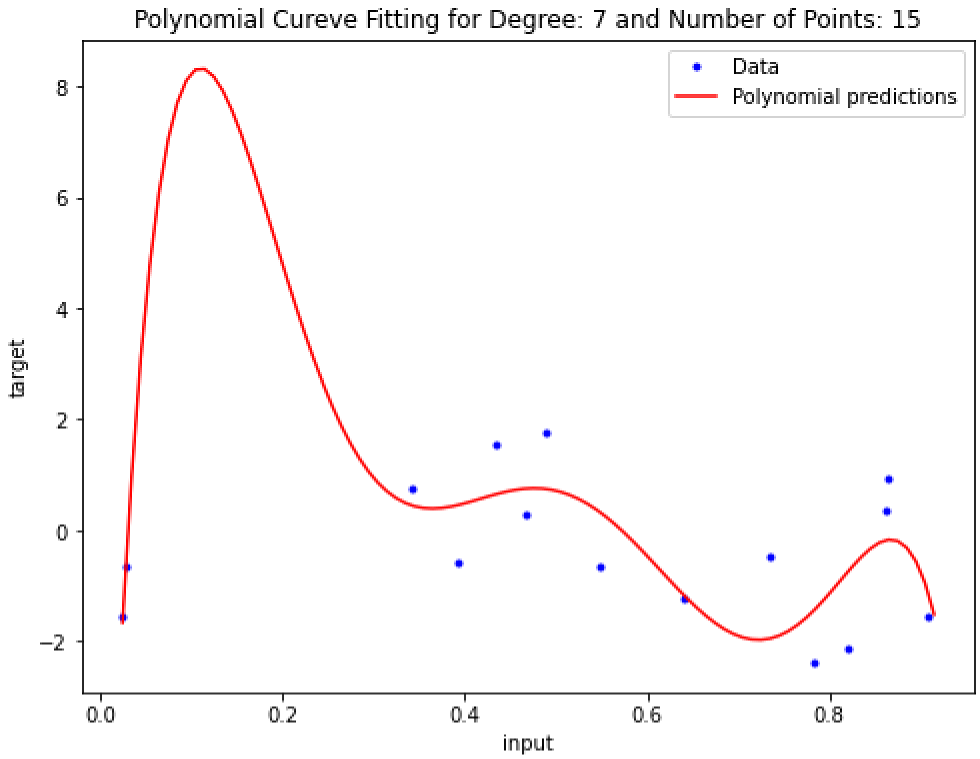


(Loss: 0.303, R2 Score: 0.770)

The graph is slightly overfitting as the difference between the number of parameters and the number of data points is less.

For N = 15:

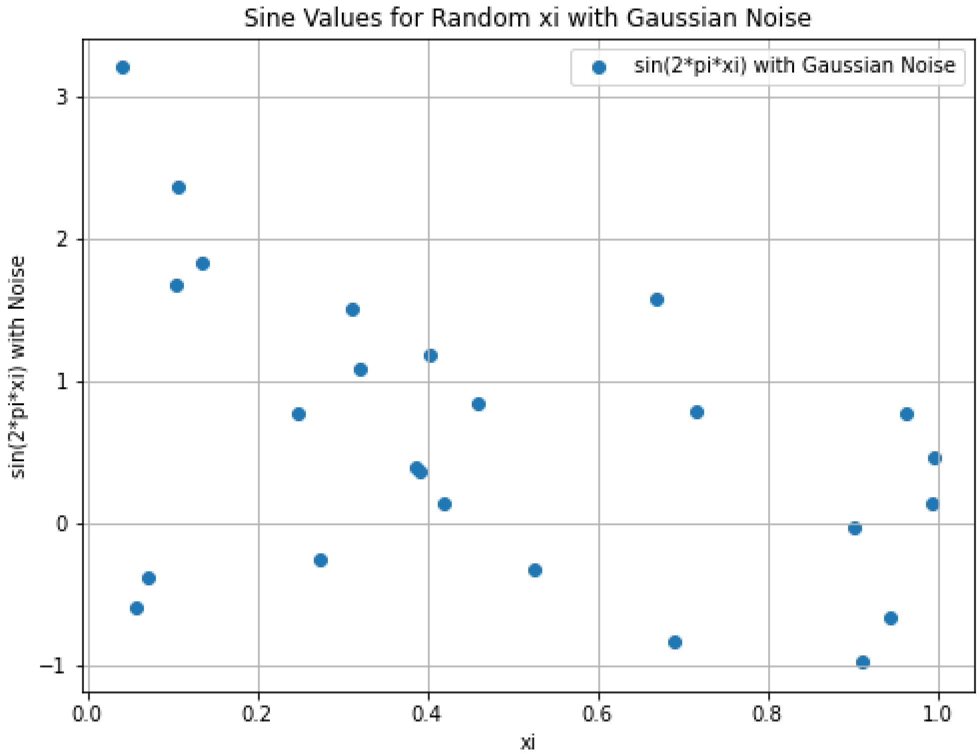


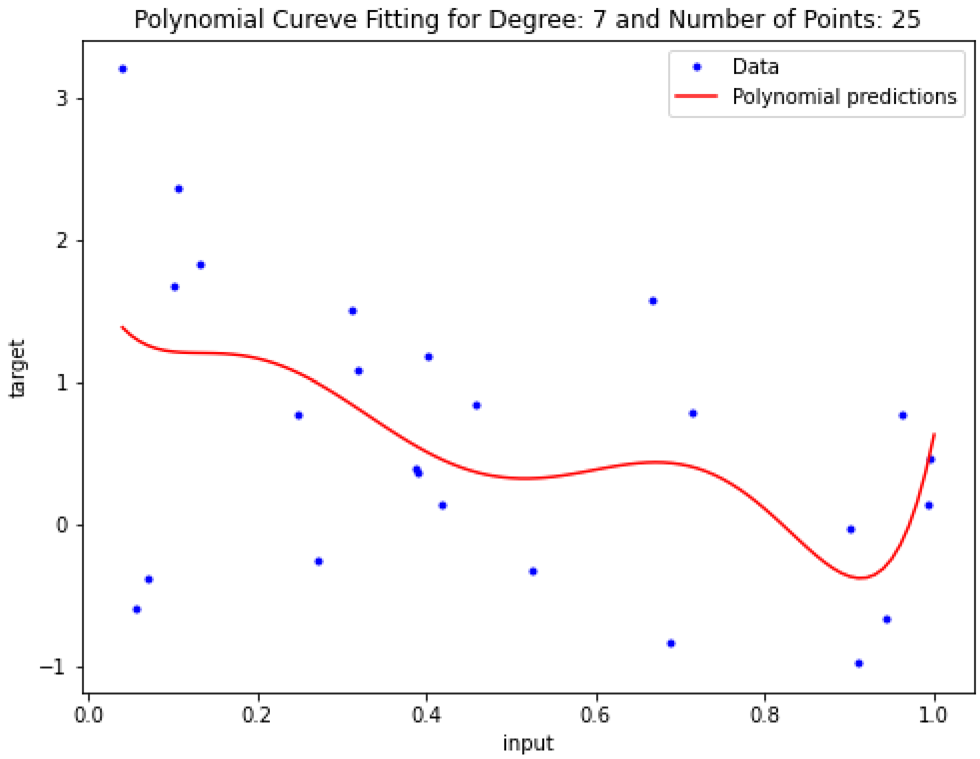


(Loss: 0.728, R2 Score: 0.525)

The graph is a good fit, as the difference between the number of parameters and the number of data points is more or less optimum for the complexity.

For N = 25

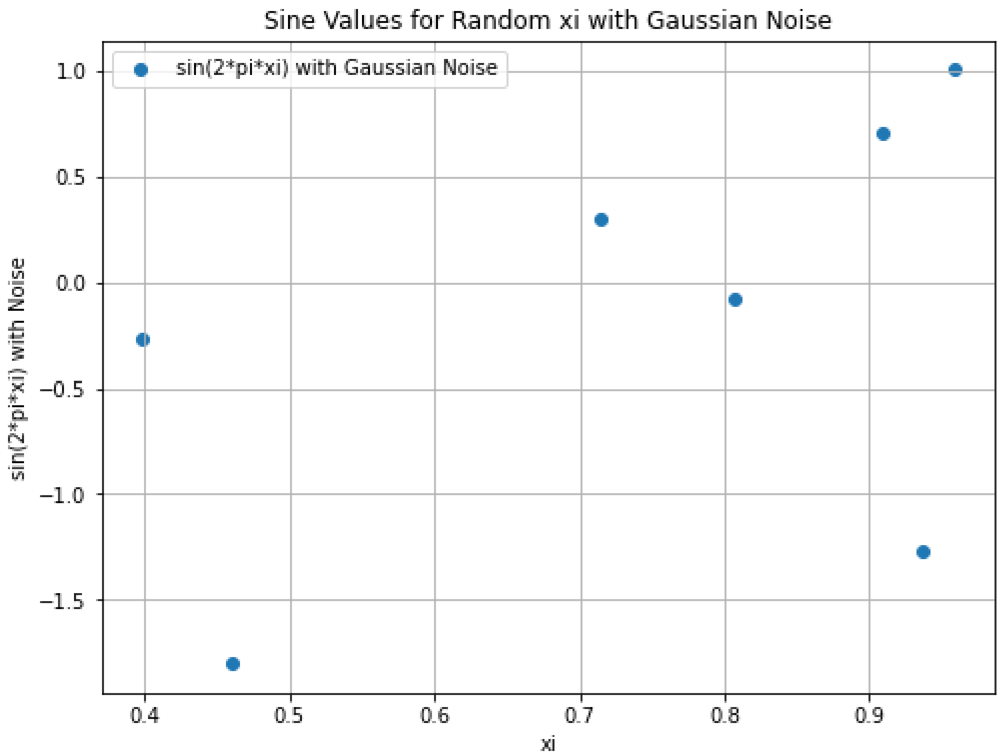


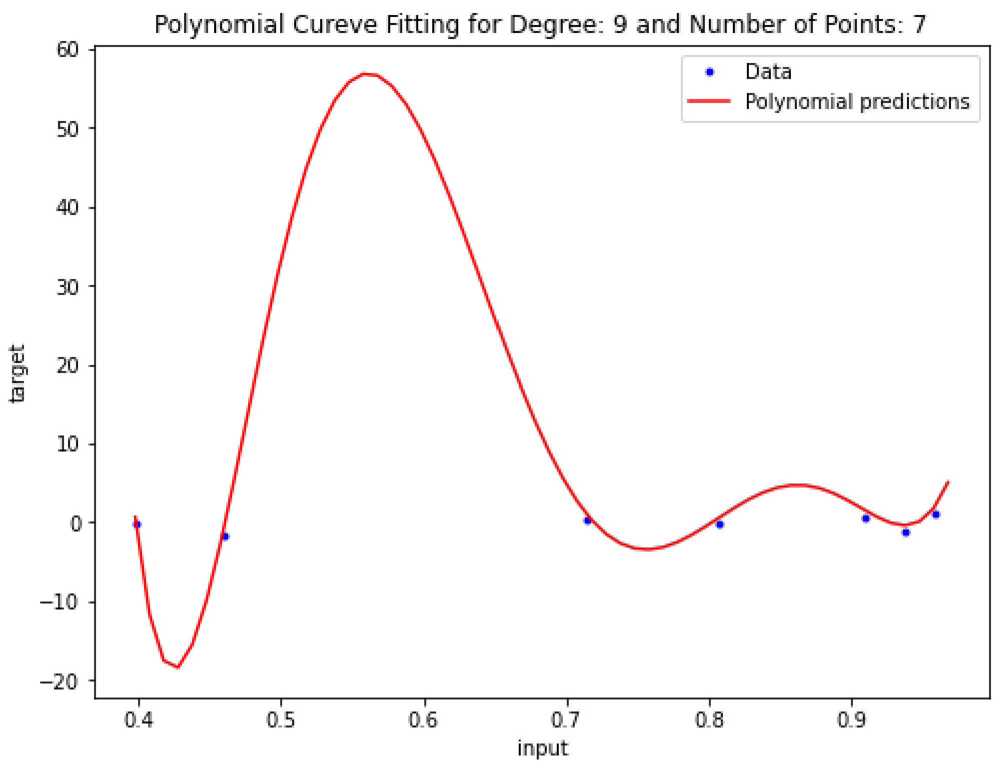


(Loss: 0.780, R2 Score: 0.254)

The graph tends to underfit as the difference between the number of parameters and data points has increased.

For Degree/M = 9:
For N = 7:

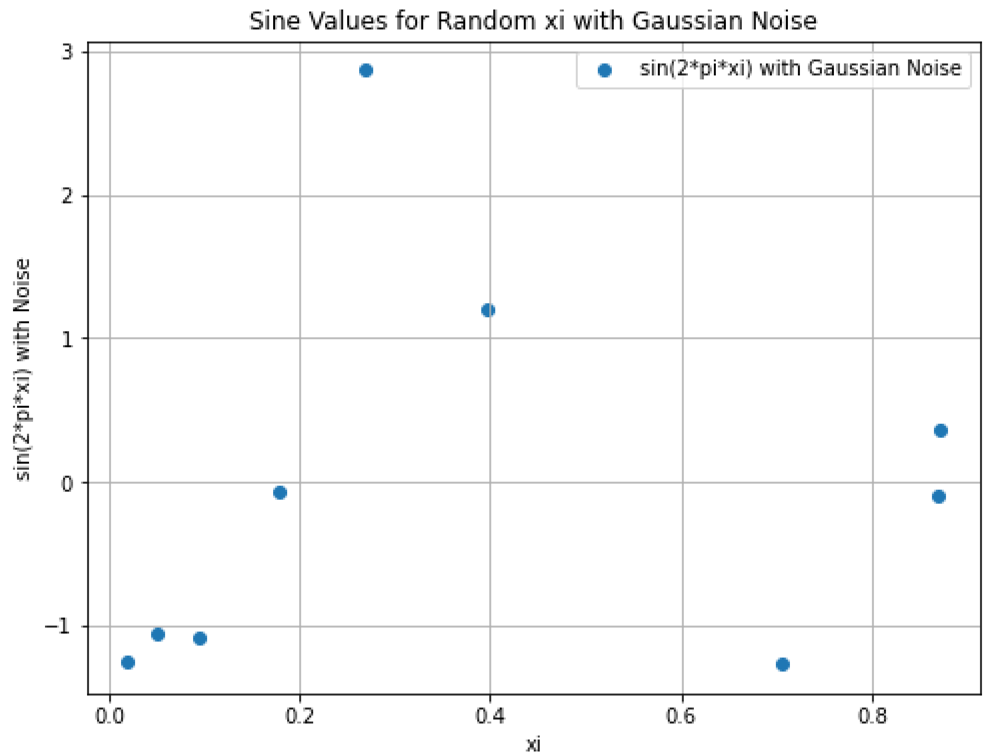


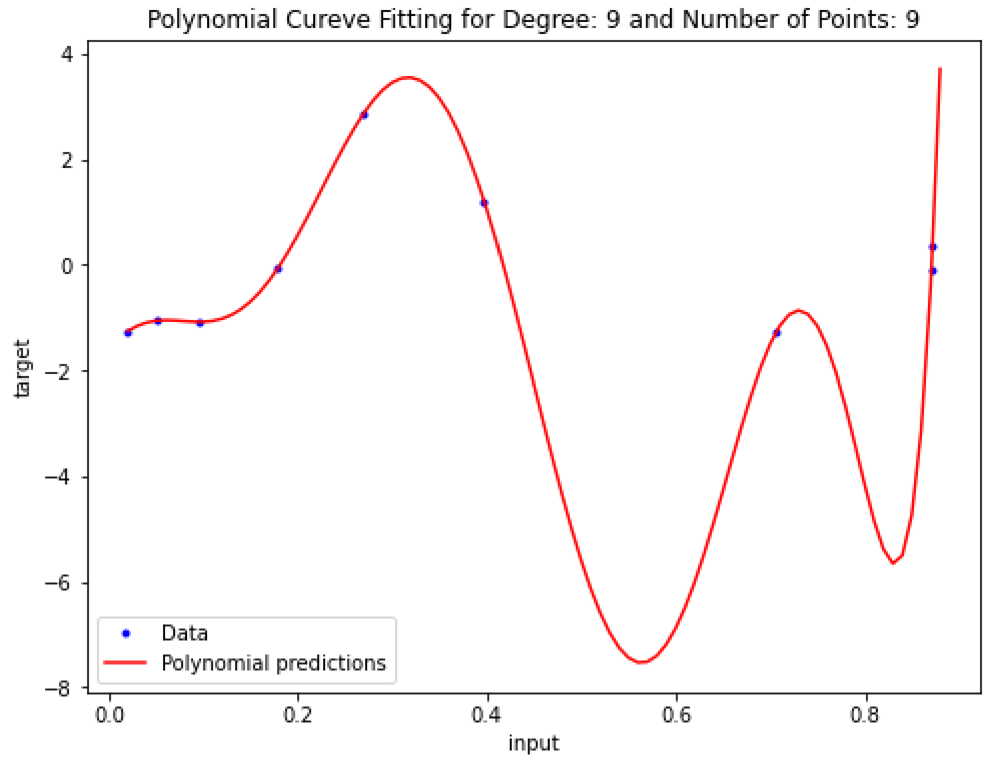


(Loss: 0.717, R2 Score: 0.193)

The graph clearly overfits, as the number of parameters exceeds the number of data points.

For N = 9:

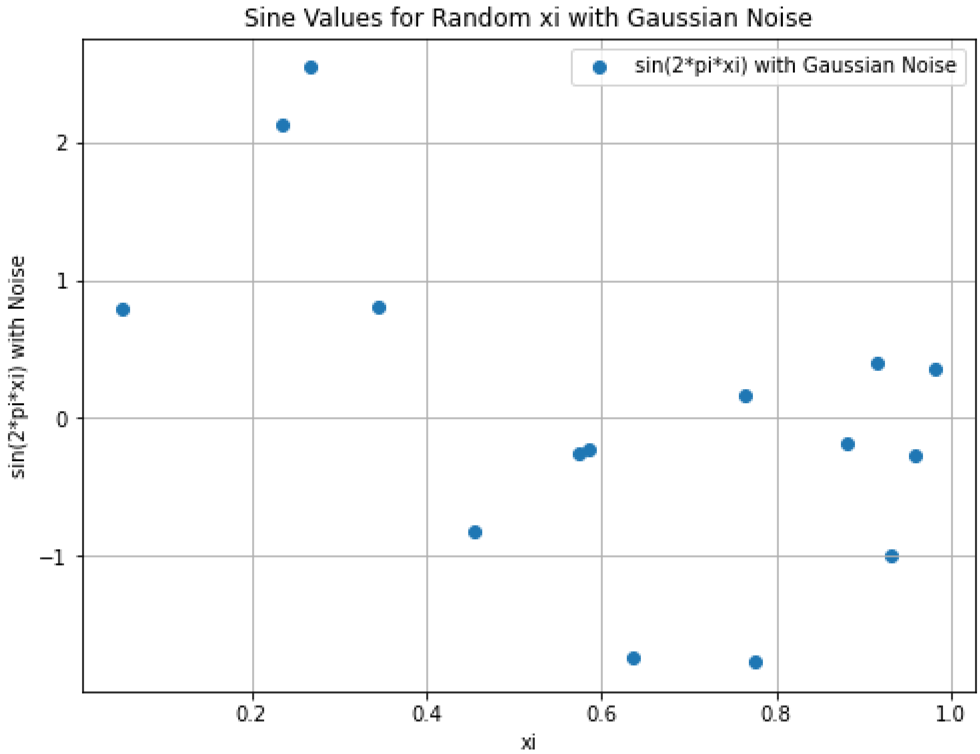


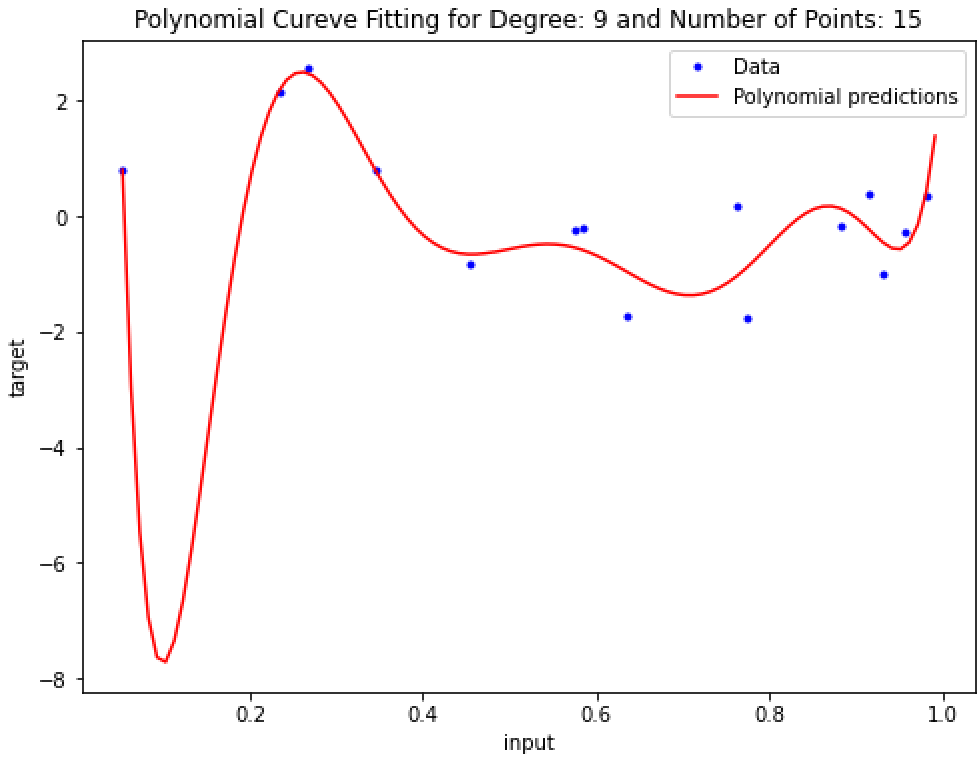


(Loss: 1.8e-10, R2 Score: 0.999)

The graph clearly overfits, as the number of parameters equals the number of data points.

For N = 15:

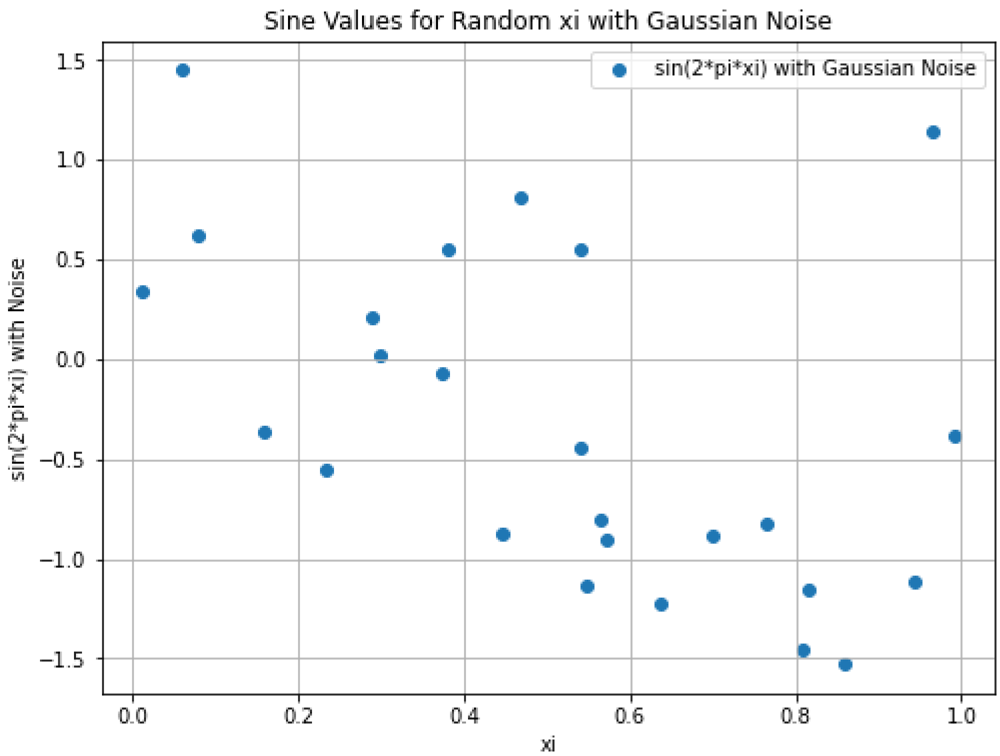


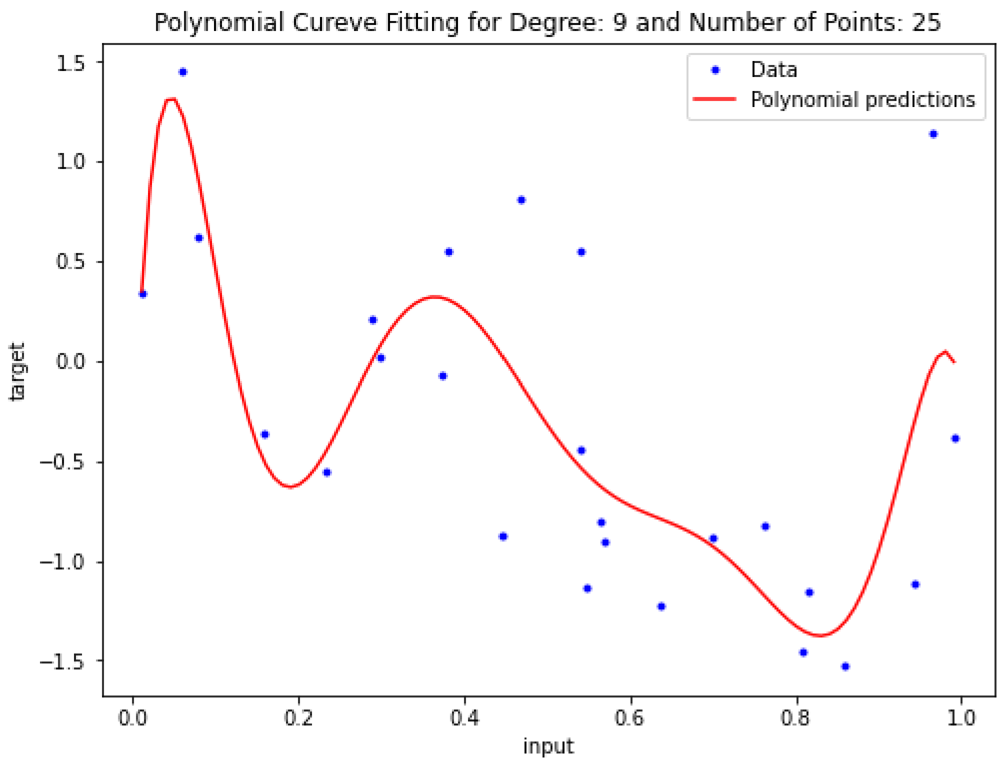


(Loss: 0.261, R2 Score: 0.809)

The graph is a good fit, as the difference between the number of parameters and the number of data points is more or less optimum for the complexity.

For N = 25:





(Loss: 0.250, R2 Score: 0.629)

This graph also fits well, as the difference between the number of parameters and the number of data points is more or less optimum for the complexity.

So for M = 7, overfitting occurs for N = 7 and N = 9
And for M = 9, overfitting occurs for N = 7 and N = 9
With Extremely clear overfitting at (M = 7 & N = 7) && (M = 9 & N = 9)

The reason for this is the lesser difference between the number of parameters and the number of data points (which is 0 for the extreme cases).

Q1) B) Yes, we can solve the problem of overfitting by introducing a regularization parameter lambda and corresponding Loss. For our case, we use L2 Reg. Loss, i.e., lambda / 2 * (||w||)^2 and add it to the traditional MSE Loss.

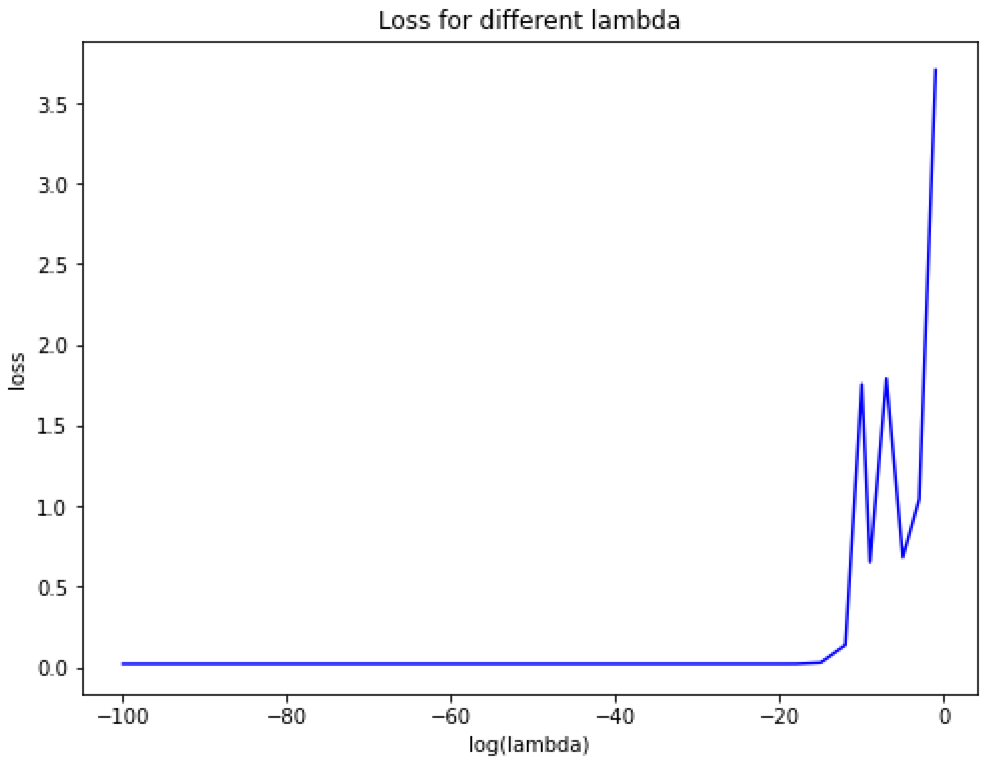
Final Loss = MSE Loss + L2 Reg Loss
= 1 / 2 * [(|| y - y_pred ||) ** 2] + 1 / 2 * lambda * [(|| w ||) ** 2]

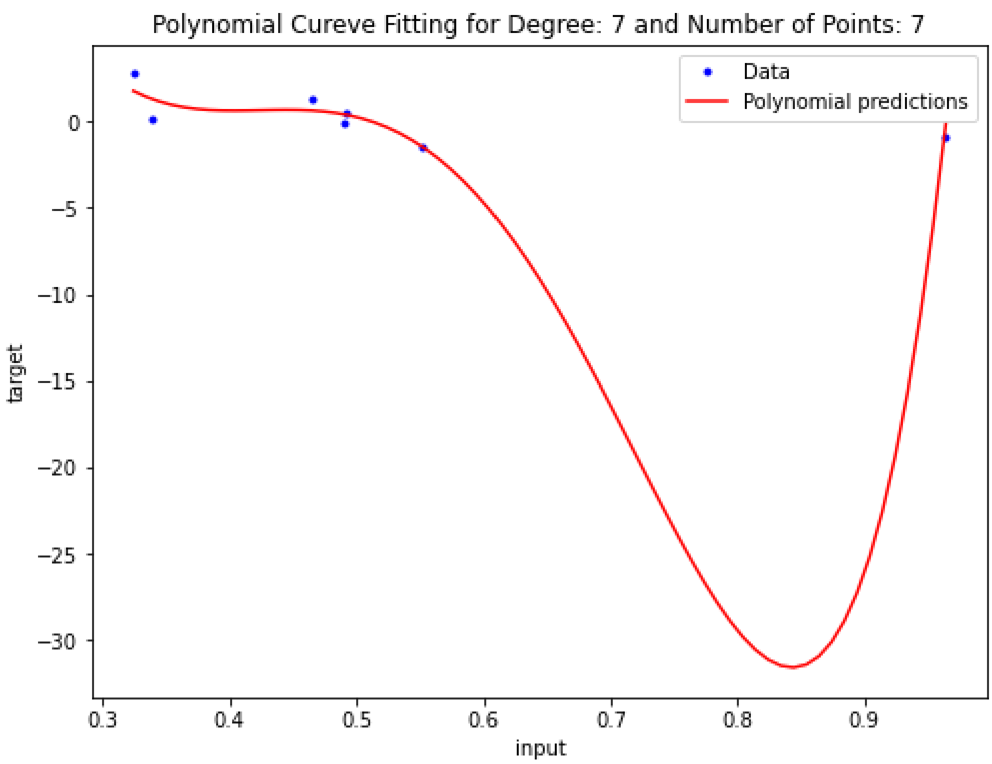
Using this, we can control our loss using the regularization term and try to reduce the problem of overfitting.

Solving this for W, we get the Linear Least Squares Solution:
W = (phi.transpose * phi + lambda * identity matrix) * phi.transpose * target

Regularization Using Linear Least Squares Parameter: Lambda

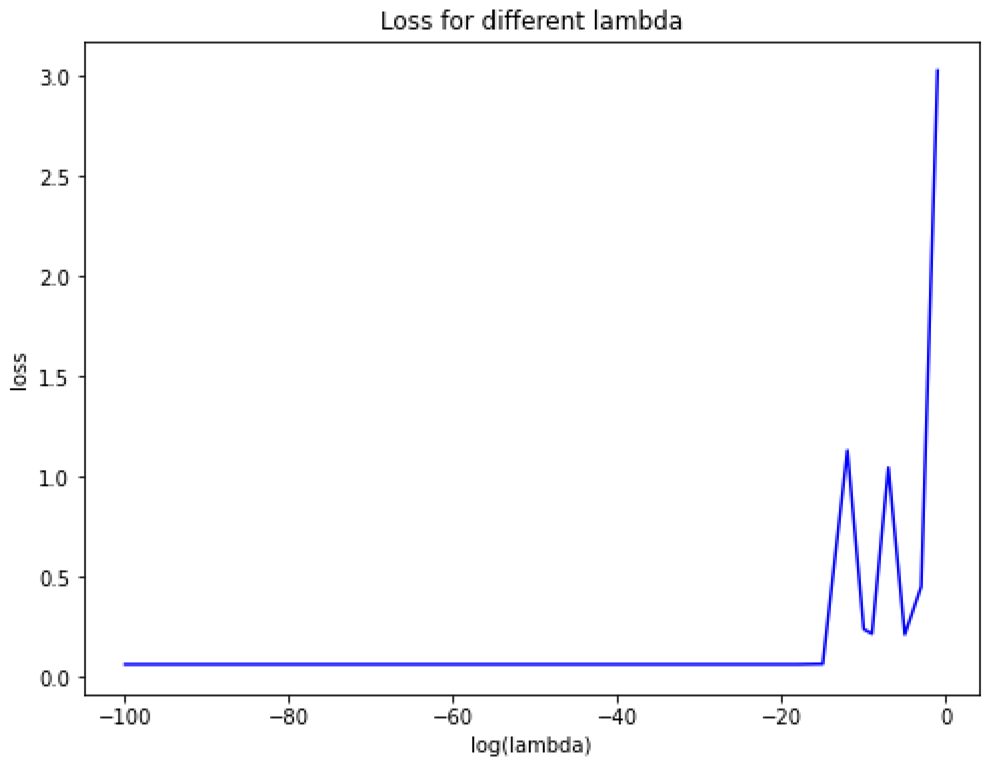
For Degree/M = 7:
For N = 7:

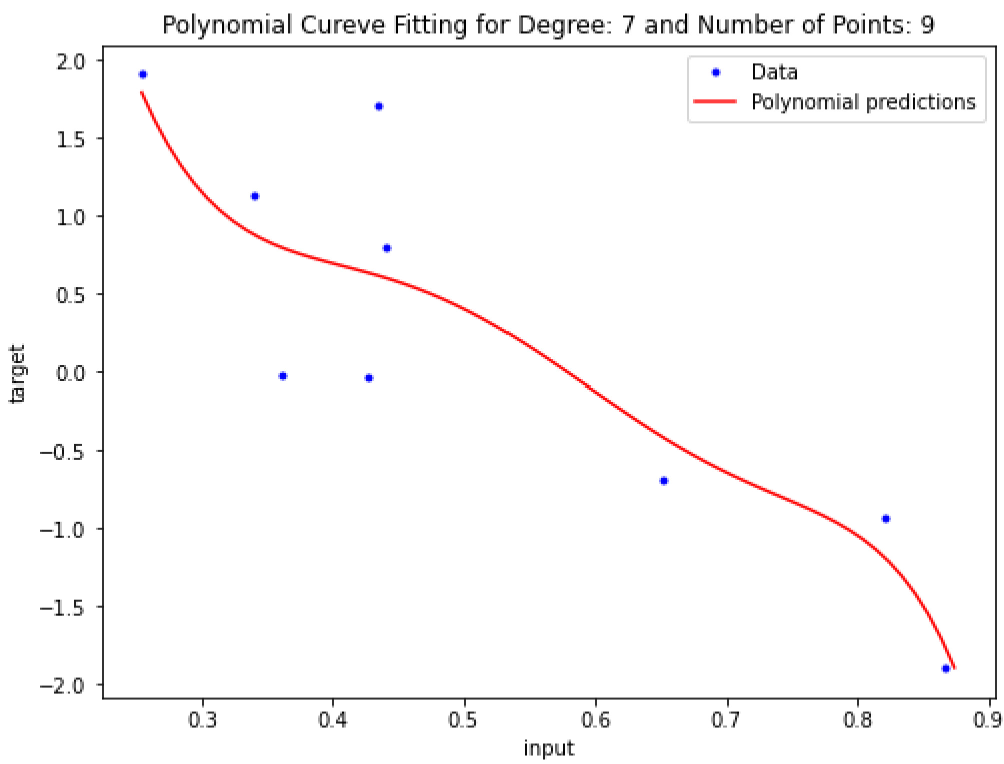




The best fit is achieved for Lambda: 1e-07
(Loss: 1.790, R2 Score: 0.732)

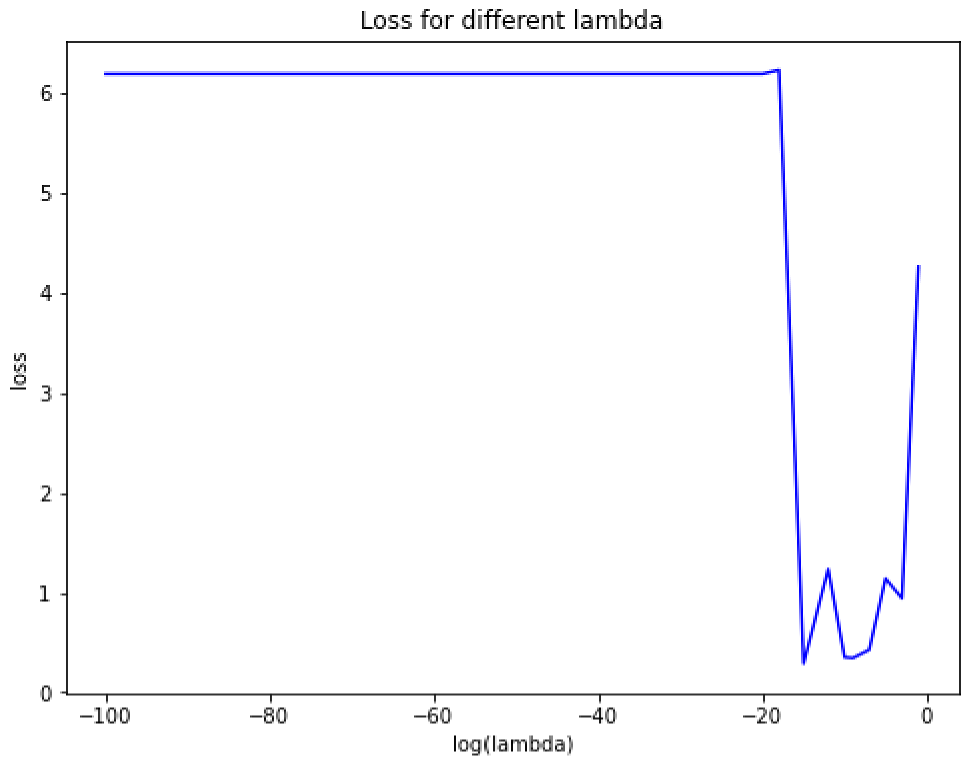
For N = 9:

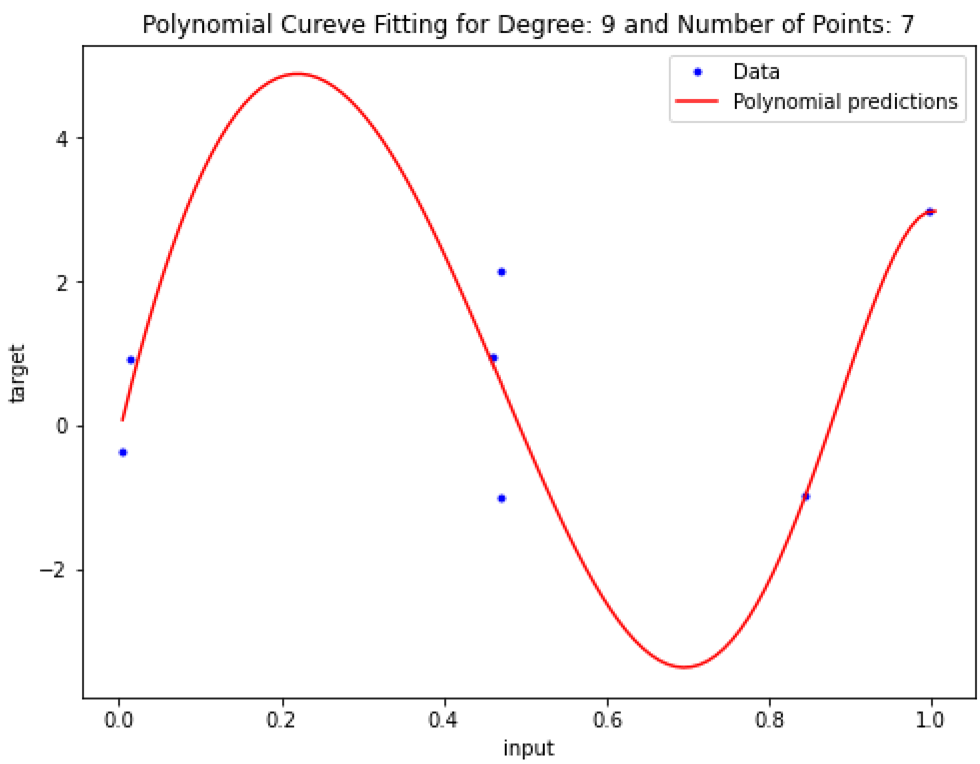




The best fit is achieved for Lambda: 1e-07
(Loss: 1.044, R2 Score: 0.799)

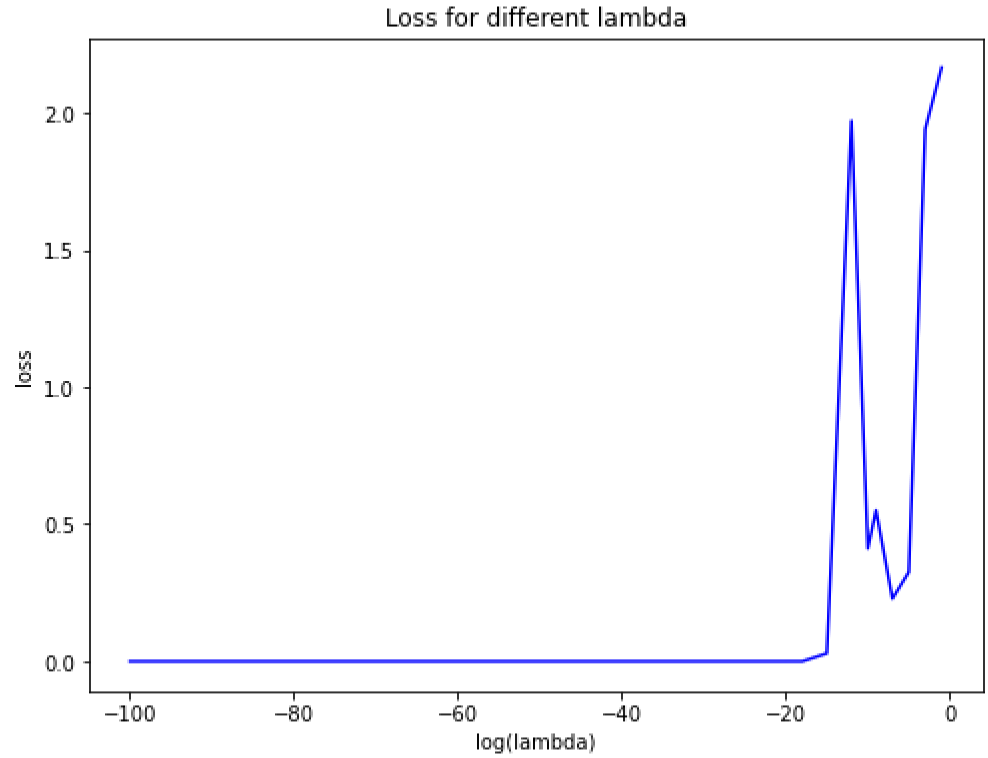
For Degree/M = 9:
For N = 7:

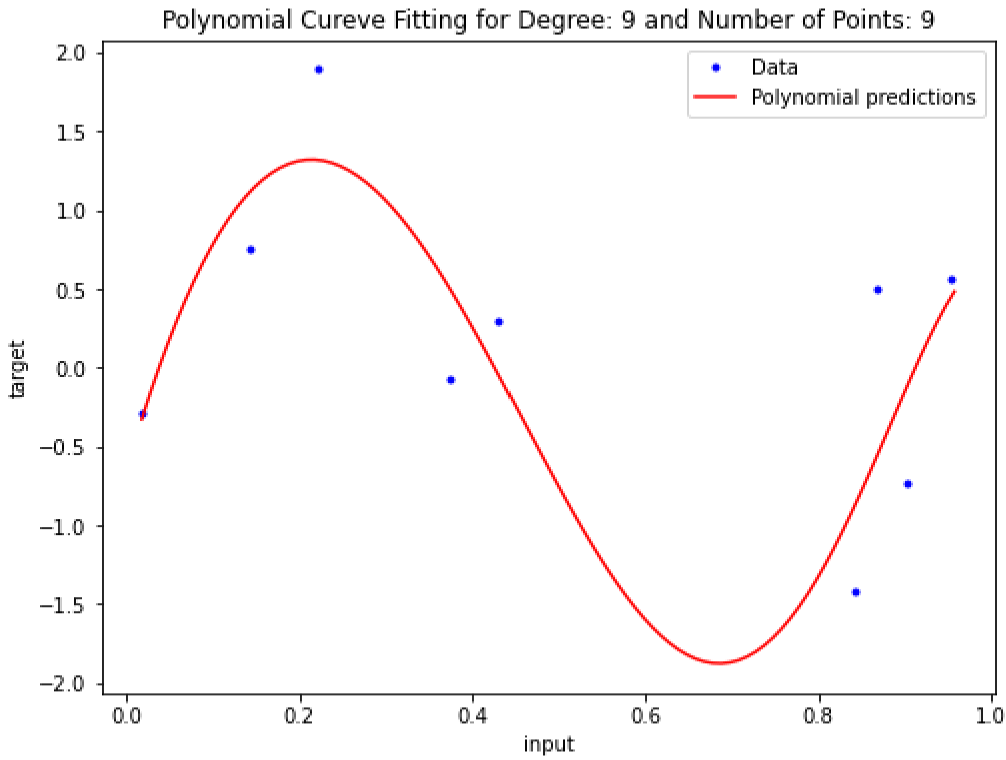




The best fit is achieved for Lambda: 1e-05
(Loss: 1.138, R2 Score: 0.627)

For N = 9:



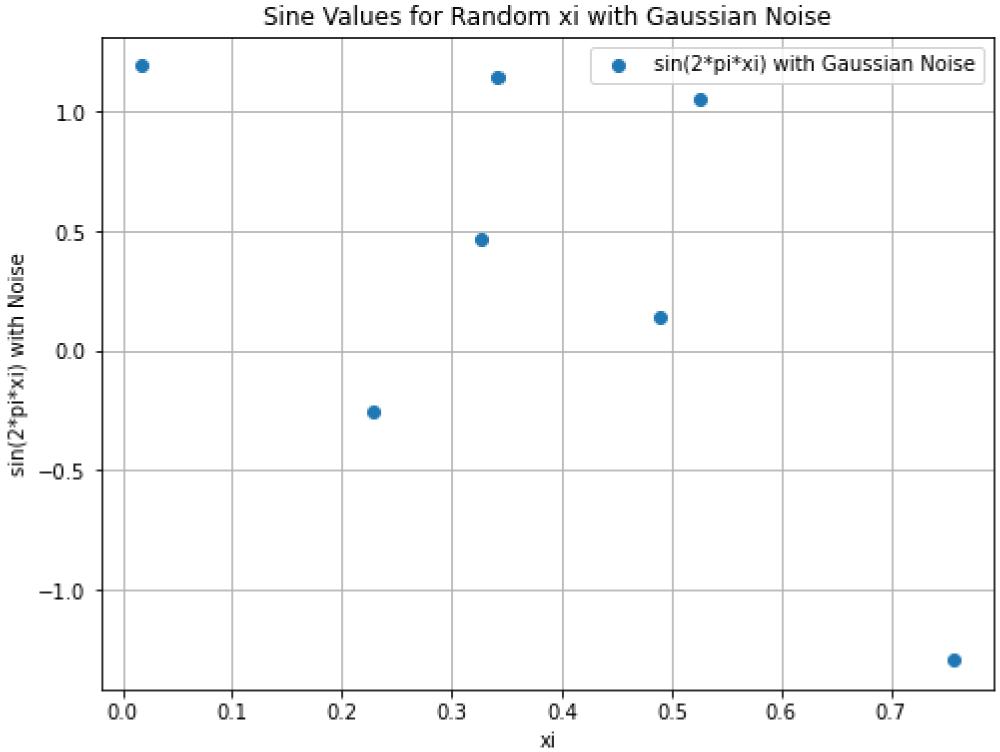


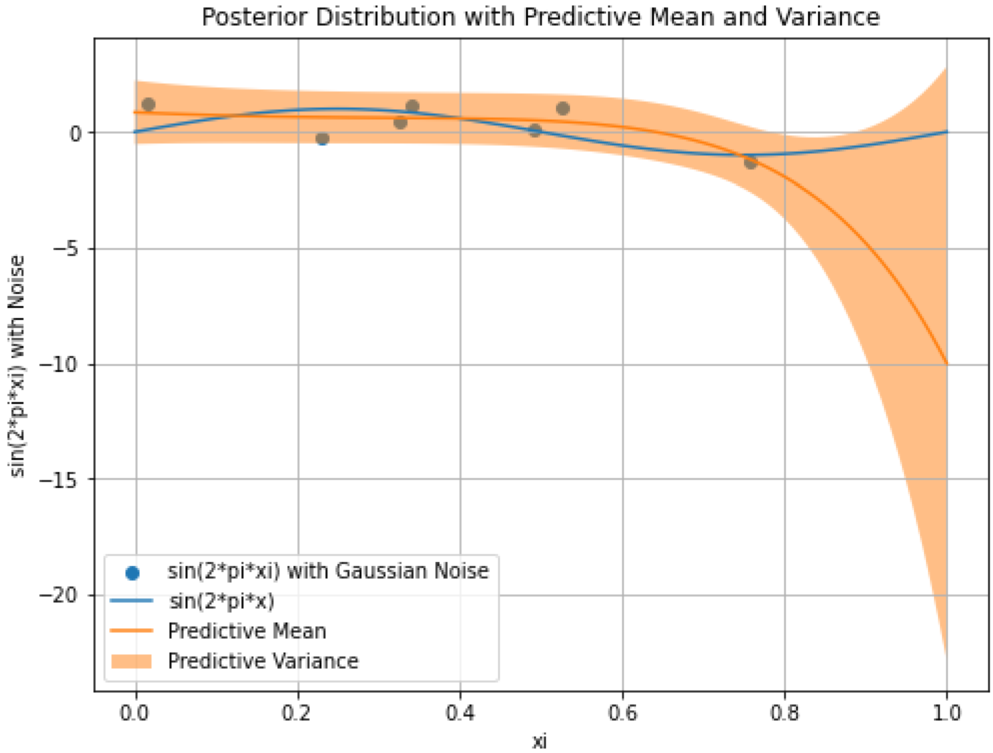
The best fit is achieved for Lambda: 1e-05
(Loss: 0.325, R2 Score: 0.623)

Predictive Sampling From Posterior Distribution

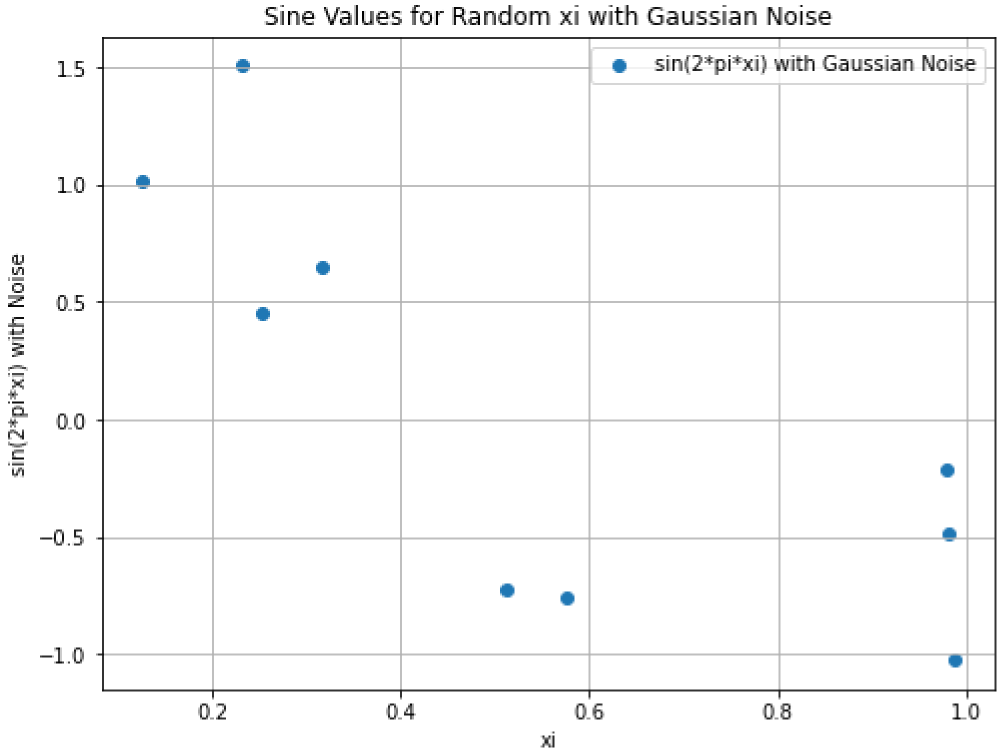
A) Plotting Confidence Region with Std Dev = 1

Q2) A)
For N = 7:

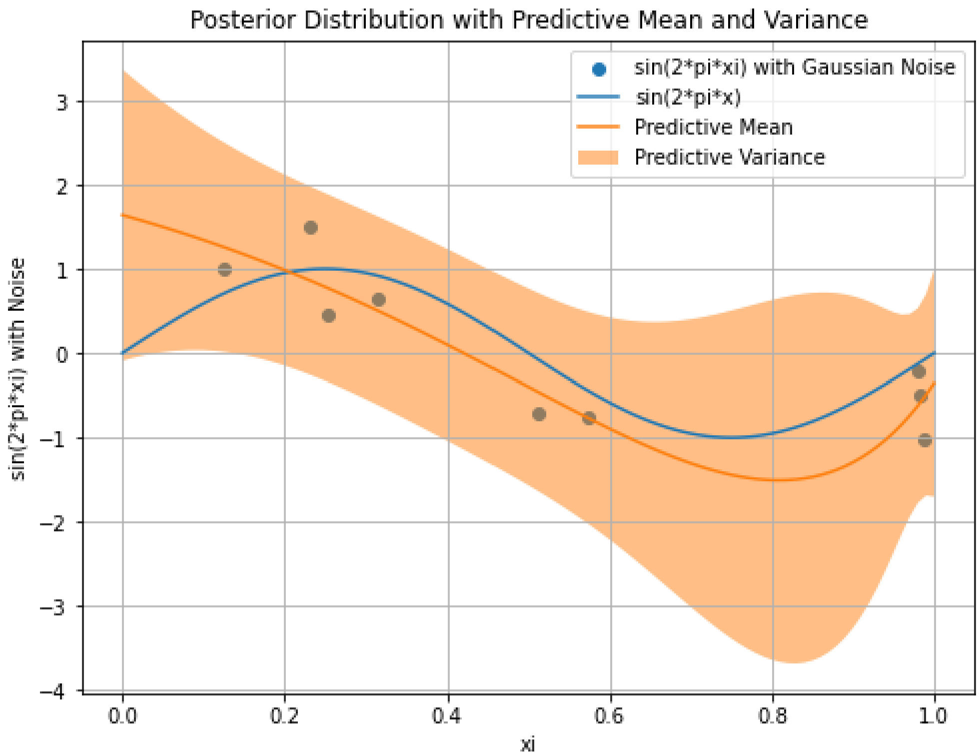




For N = 9:

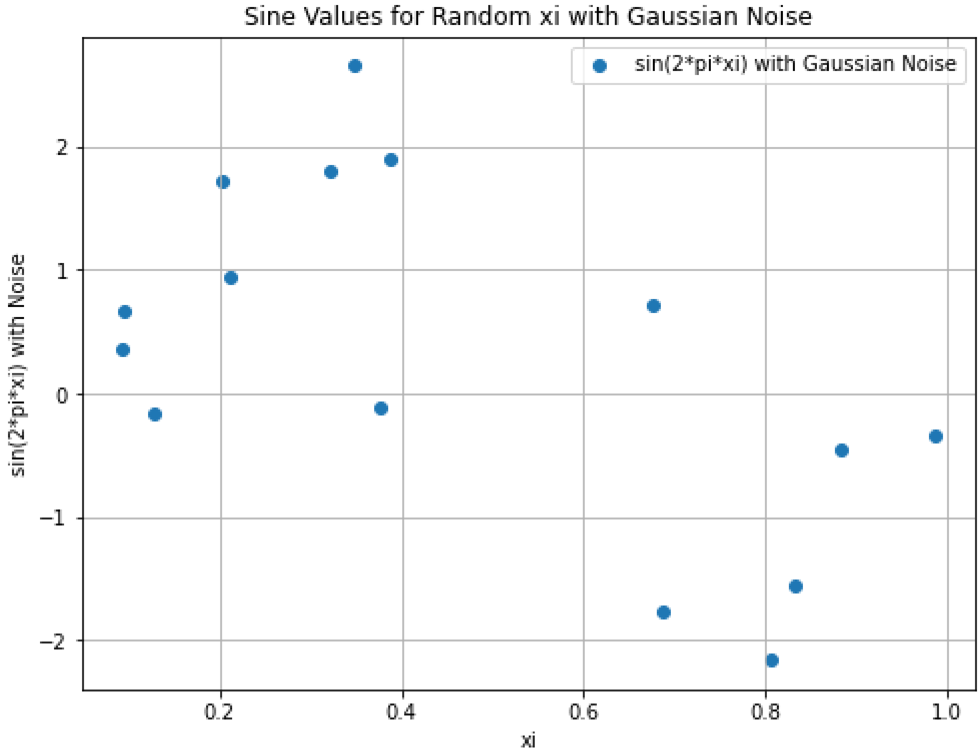


Sine Values for Random xi with Gaussian Noise
sin(2*pi*xi) with Gaussian Noise
1.5
1.0
0.5
0.0
−0.5
−1.0
sin(2*pi*xi) with Noise
0.2
0.4
0.6
0.8
1.0
xi

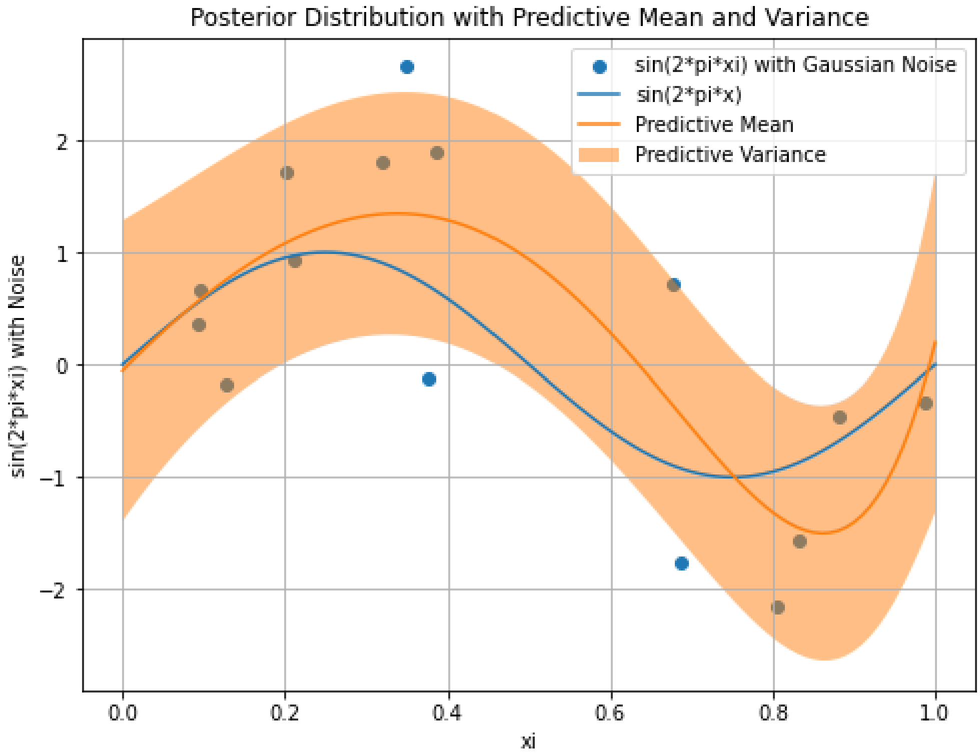


Posterior Distribution with Predictive Mean and Variance
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*x)
Predictive Mean
Predictive Variance
3
2
1
0
−1
−2
−3
−4
sin(2*pi*xi) with Noise
0.0
0.2
0.4
0.6
0.8
1.0
xi

For N = 15:

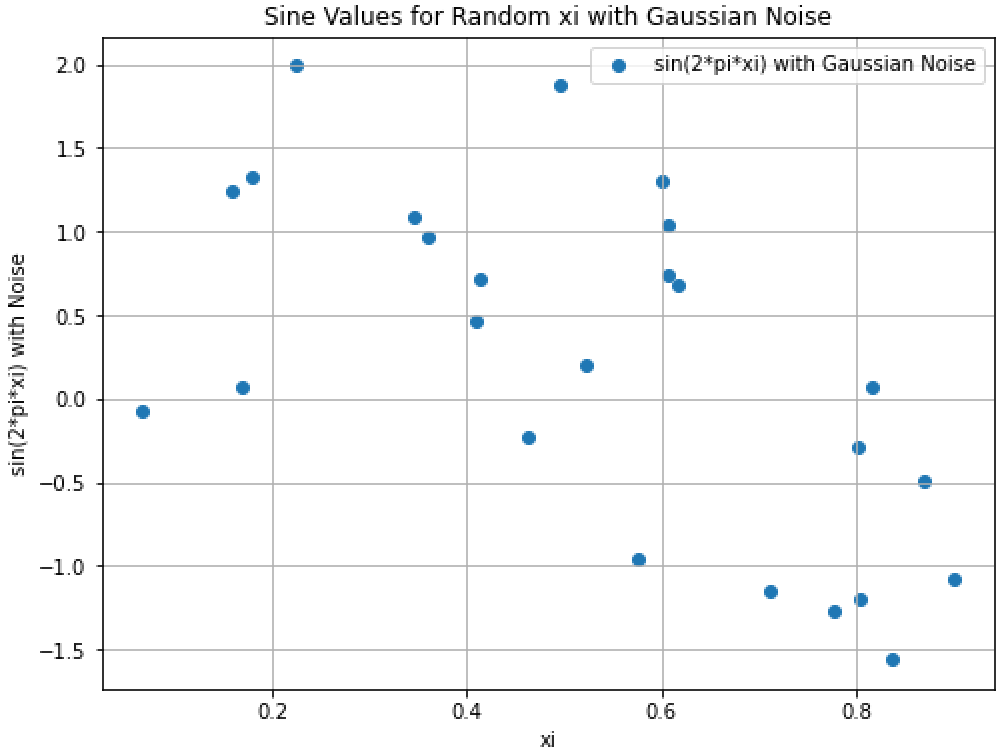


Sine Values for Random xi with Gaussian Noise
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*xi) with Noise
xi

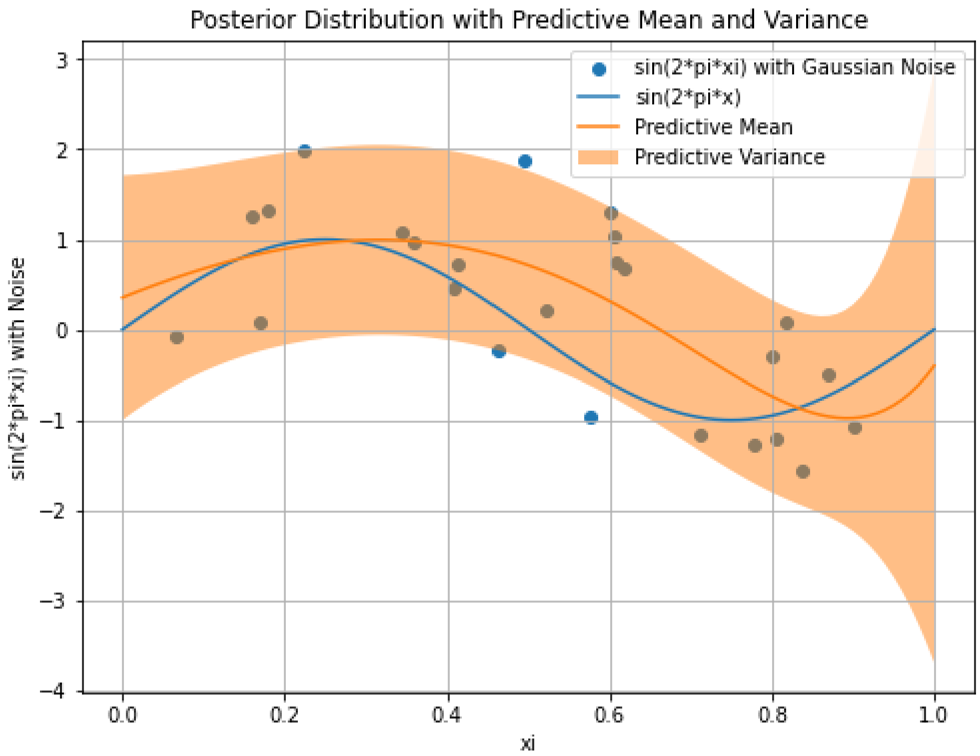


Posterior Distribution with Predictive Mean and Variance
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*x)
Predictive Mean
Predictive Variance
sin(2*pi*xi) with Noise
xi

For N = 25:



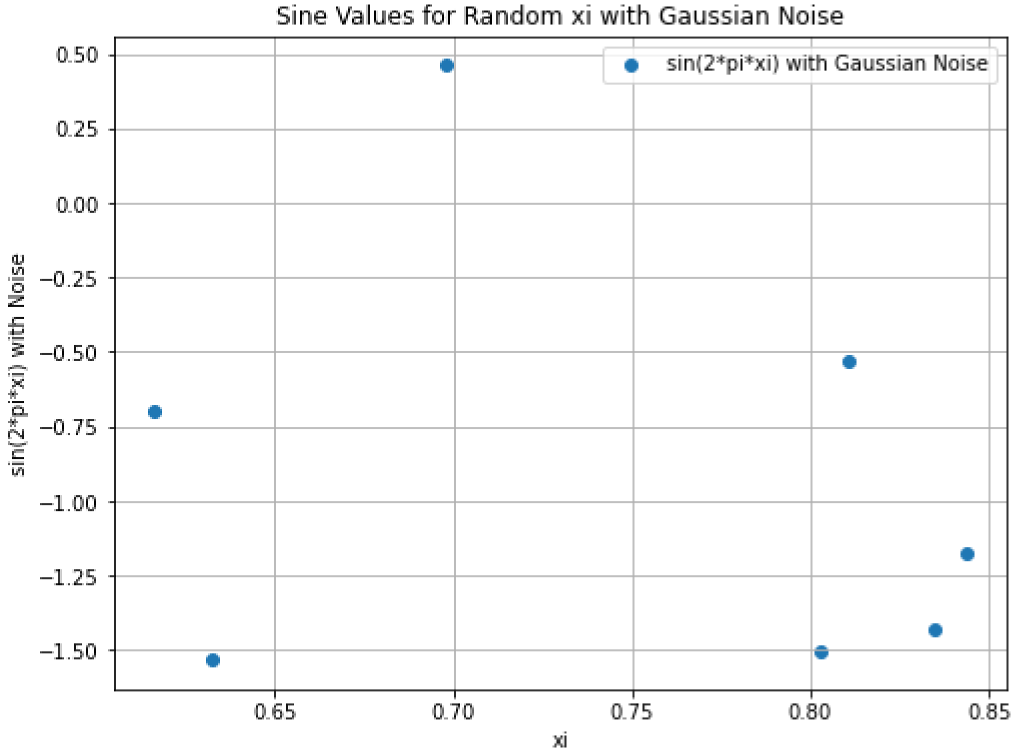
Sine Values for Random xi with Gaussian Noise
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*xi) with Noise
xi

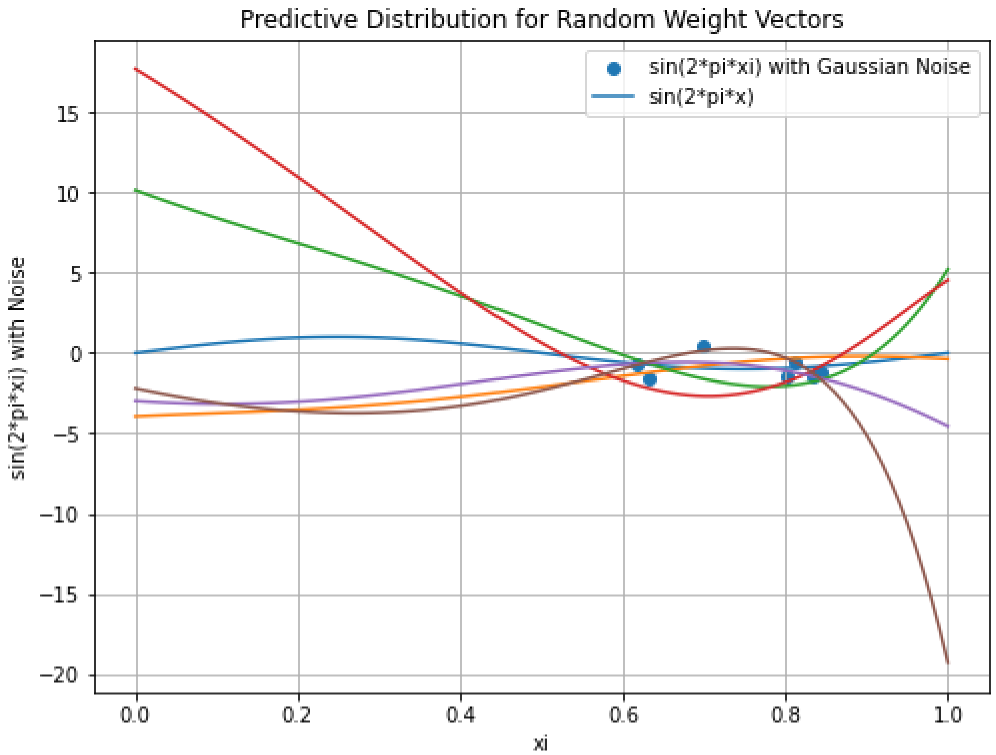


Posterior Distribution with Predictive Mean and Variance
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*x)
Predictive Mean
Predictive Variance
sin(2*pi*xi) with Noise
xi

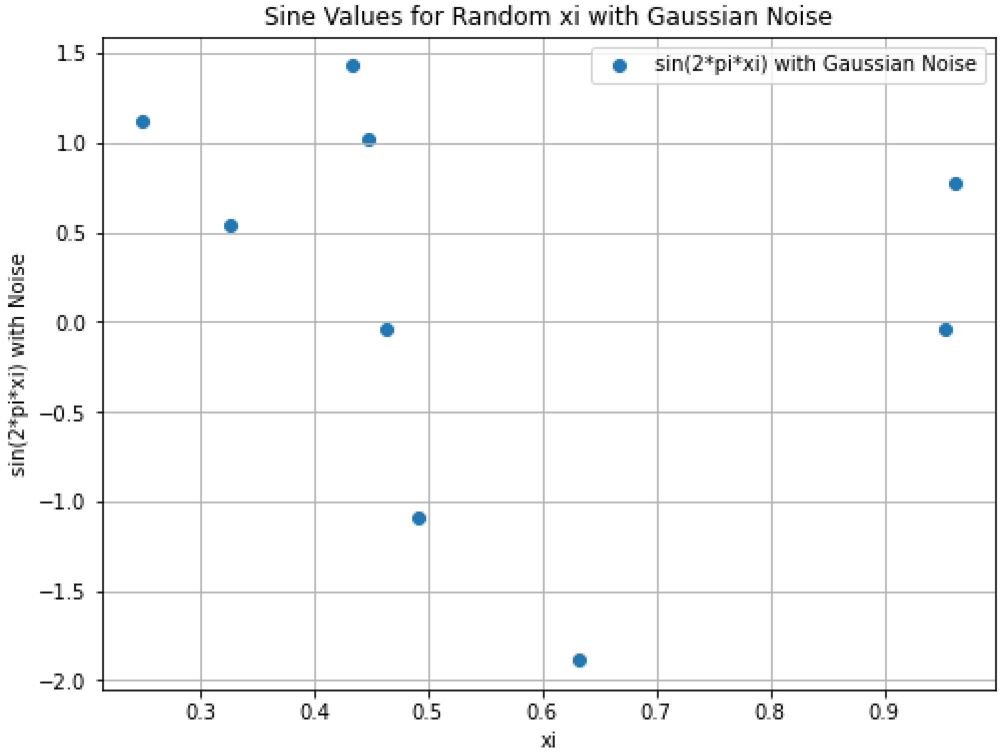
B) Randomly Sampling 5 Posterior Distributions

Q2) B)
For N = 7:

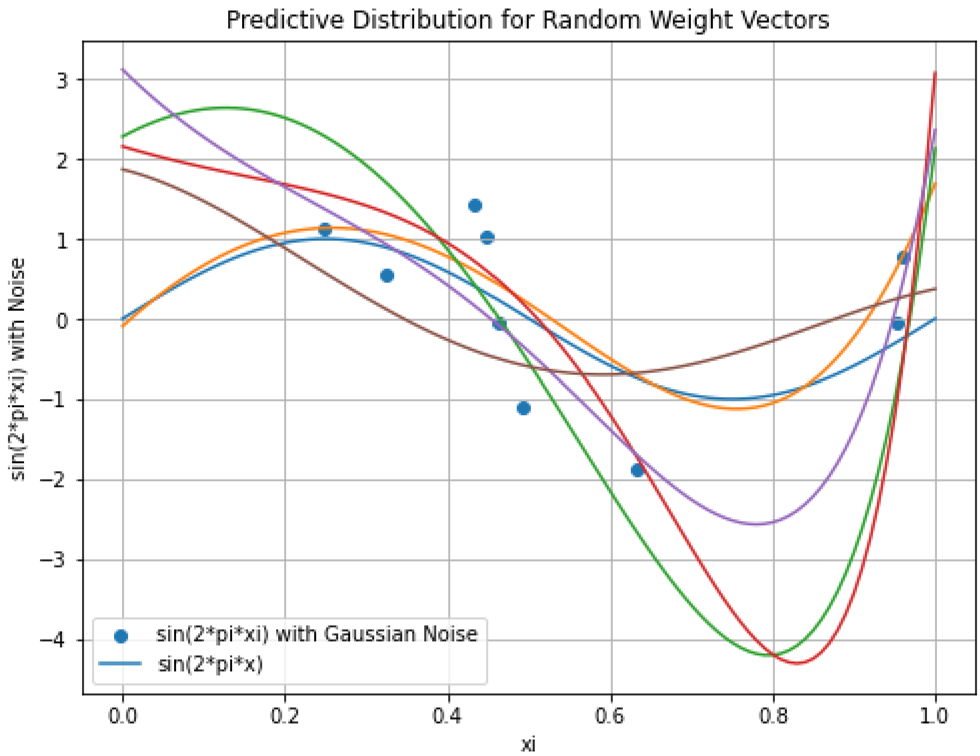




For N = 9:

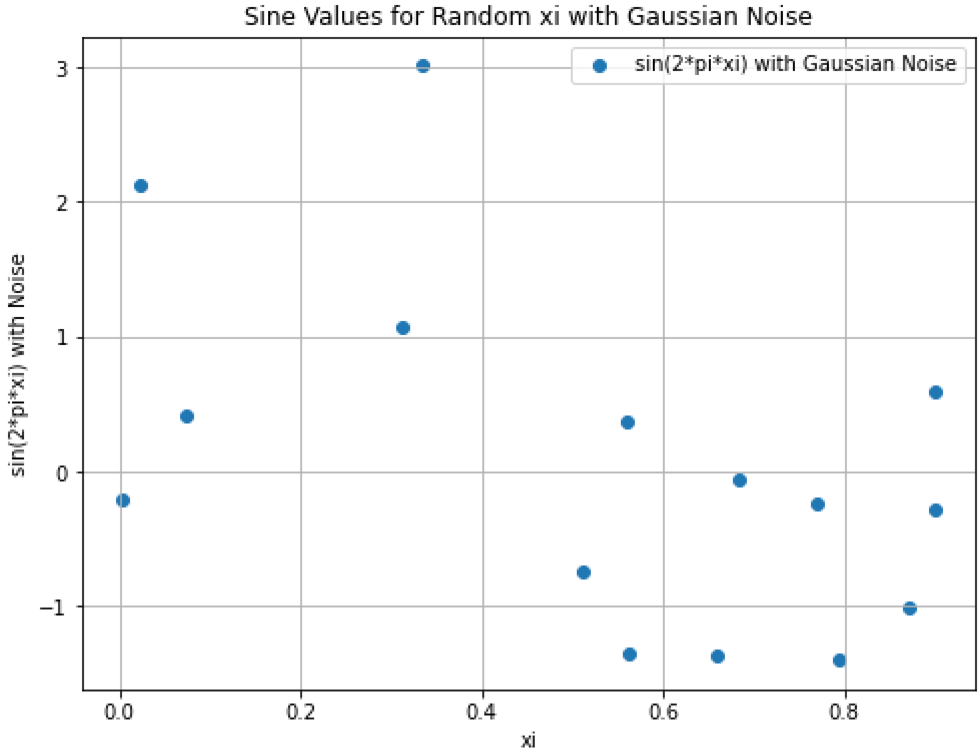


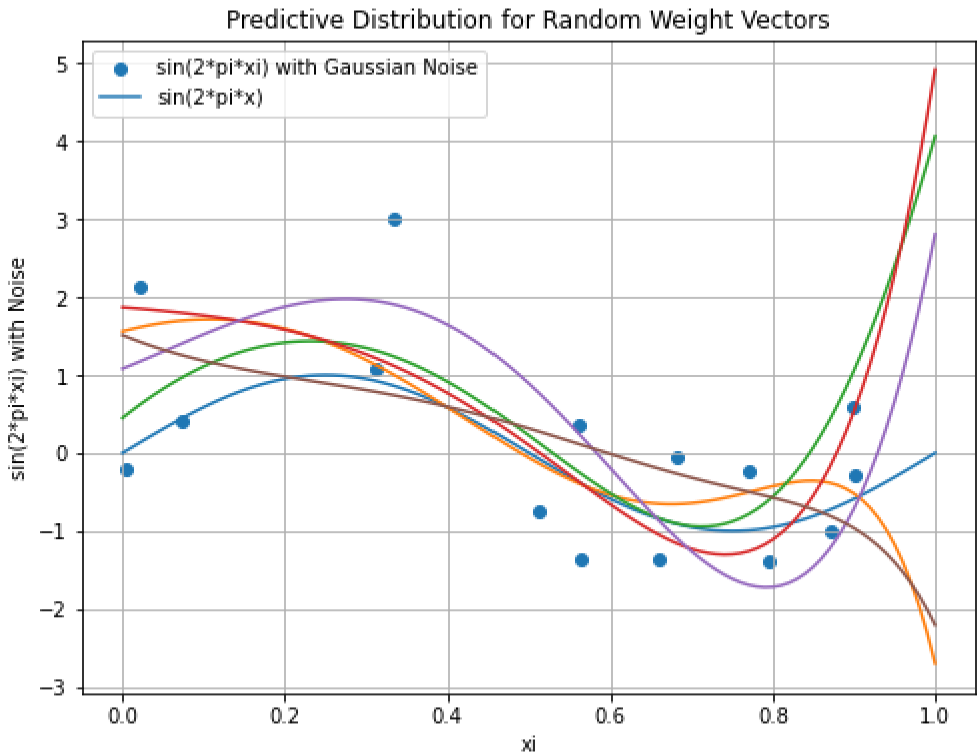
Sine Values for Random xi with Gaussian Noise
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*xi) with Noise
xi



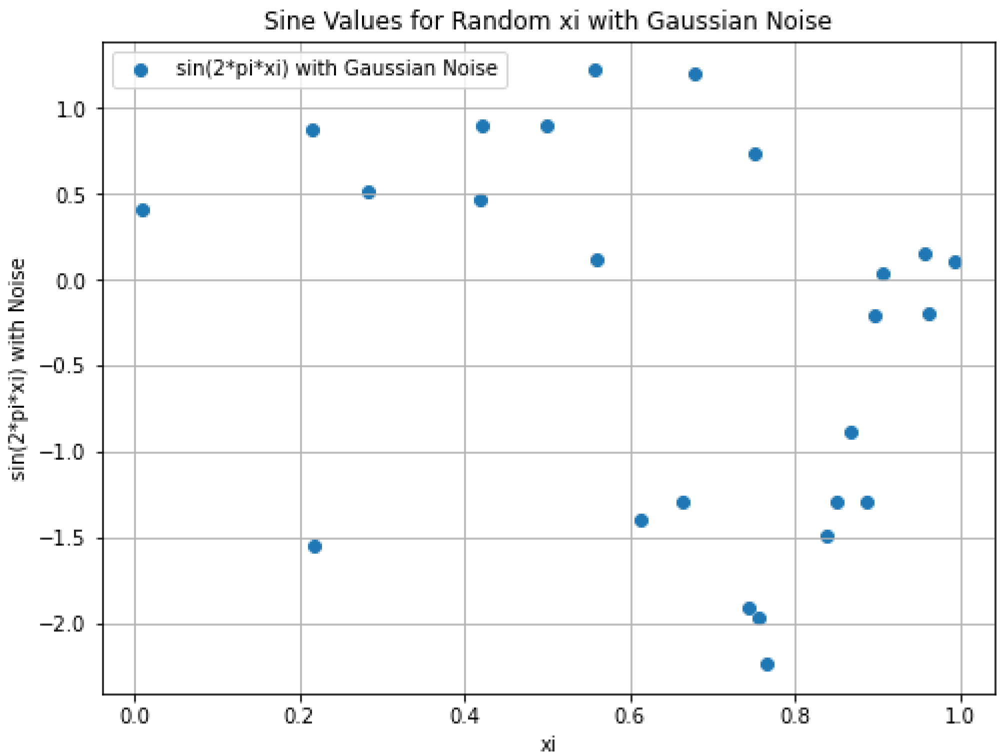
Predictive Distribution for Random Weight Vectors
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*x)
sin(2*pi*xi) with Noise
xi

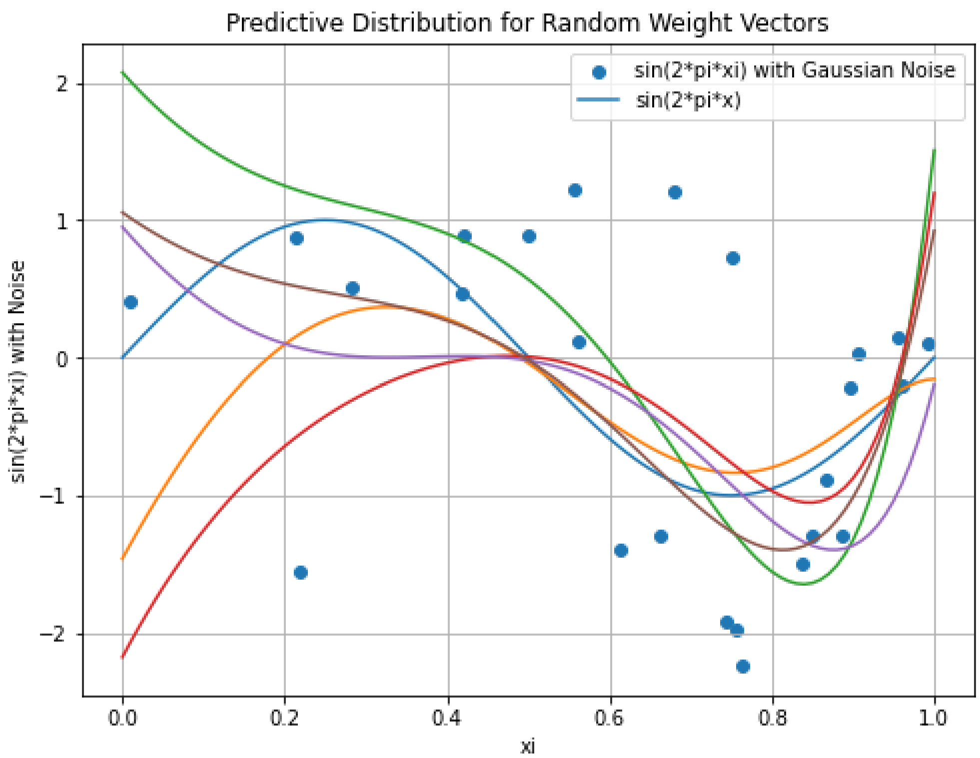
For N = 15:


Sine Values for Random xi with Gaussian Noise
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*xi) with Noise
xi


Predictive Distribution for Random Weight Vectors
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*x)
sin(2*pi*xi) with Noise
xi

For N = 25:


Sine Values for Random xi with Gaussian Noise
sin(2*pi*xi) with Gaussian Noise
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
sin(2*pi*xi) with Noise
0.0
0.2
0.4
0.6
0.8
1.0
xi


Predictive Distribution for Random Weight Vectors
sin(2*pi*xi) with Gaussian Noise
sin(2*pi*x)
2
1
0
−1
−2
sin(2*pi*xi) with Noise
0.0
0.2
0.4
0.6
0.8
1.0
xi