ENS409 Project 2 Report

Question 1

Find the best fit line in the least squares sense to the training data. Plot the data and the best fit line on the same plot. Compute the correlation coefficient r. How much of the original uncertainty is explained by the linear model?

Solution

Here, I have used the approach that was told in the lecture which is included in the 5-LS_Regression slide set. To define the line that fits to the dataset in the least squares sense, we need to determine the coefficients and to minimize the error we set the derivatives to zero. Then, by using the equations we came up with the coefficients we quantify how good is our fit to the original data by using r measure.

```
%%best fit line in the least square sense, correlation coefficient r
load('proj3data.mat');
Xtrain = x(1,1:450);
Ytrain = y(1,1:450);
coefficients(Xtrain, Ytrain);
```

Here, we load the data and selected first 75% as training data.

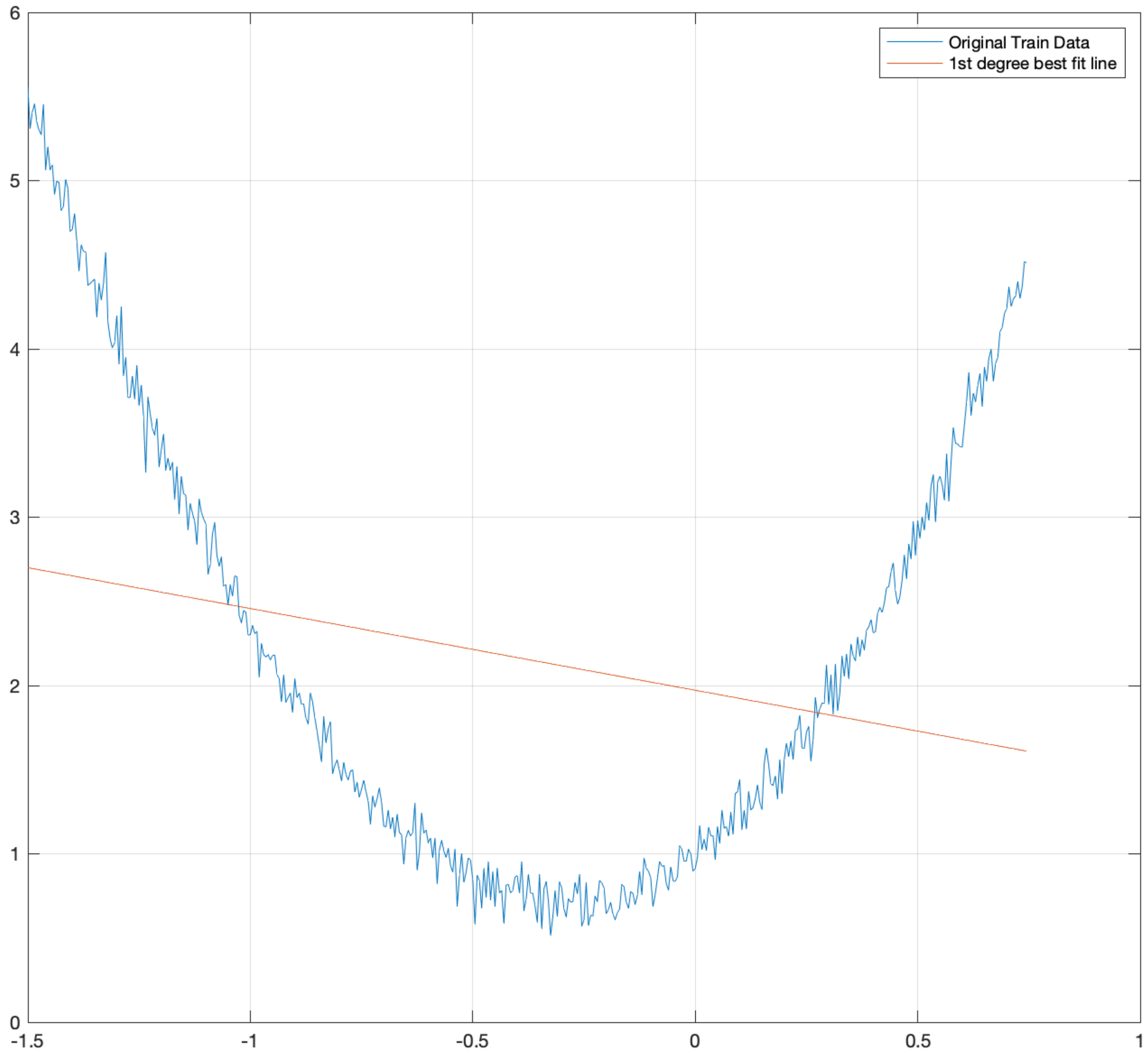
```
function coefficients(Xtrain, Ytrain)
    [n] = length(Xtrain);
    sum1 = sum(Xtrain);
    sum2 = sum(Ytrain);
    sum3 = sum(Xtrain.*Xtrain);
    sum4 = sum(Xtrain.*Ytrain);
    sum5 = sum(Ytrain.*Ytrain);

    a1 = (n*sum4 - (sum1*sum2))/(n*sum3 -(sum1*sum1));
    a0 = (sum2 - (a1*sum1))/n;
    lse = @(x)a1*x + a0;

    r = (n*sum4 - (sum1*sum2))/(sqrt(n*sum3 -
(sum1*sum1))*sqrt(n*sum5 -(sum2*sum2)));
    rsquare = r*r;
    % If the value of r=1, that corresponds to a perfect fit to the
data
    % if the value of r is close to unity,
    % good correlation of the fit and the original data.
    fprintf('The a1 coeffcient is %.6f', a1);
    fprintf('\n')
    fprintf('The a0 coeffcient is %.6f', a0);
```

```
    fprintf('\n')
    fprintf('The correlation coefficient r is %.6f', r);
    fprintf('\n')
    fprintf('The original uncertainty explained by the linear model
is %.6f', rsquare);
    fprintf('\n')
    plot(Xtrain, Ytrain, 'DisplayName','Original Train Data');
    hold on;
    plot(Xtrain, lse(Xtrain), 'DisplayName','1st degree best fit
line');
    grid on;
    legend;
    hold off;
end
```

After calculating the coefficients we also calculated the coefficient r

```
The a1 coeffcient is -0.485607
The a0 coeffcient is 1.971961
The correlation coefficient r is -0.239105
The original uncertainty explained by the linear model is 0.057171
```

If the value of r=1, that corresponds to a perfect fit to the data. However, since it is close to 0, this line does not represent the best linet hat can fit. As you can see, the original uncertainty explained by the linear model is also very low.



Question 2

Obtain the least squares polynomial y of degree $n = 2, 5, 10, 30$ which fits the training data. Then use MATLAB's polyfit function to find the regression polynomial y of the " same degree. Plot data, y and your approximation y on the same plot and look at the maximum of the absolute error and the relative error. Comment on the results.

Solution

```
%% Obtain the least squares polynomial
load('proj3data.mat');
Xtrain = x(1,1:450);
Ytrain = y(1,1:450);
leastsquarepolynomial(Xtrain, Ytrain);
```

Here, we load the data and selected first 75% as training data. Then, I will generate matrixes that are going to be similar to the following matrix.

Polynomial Regression

- By arranging the terms, we obtain three equations with three unknowns. These equations can be written in the matrix form as:

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 \\ \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i^3 & \sum_{i=1}^{n} x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i^2 y_i \end{bmatrix}$$

- By solving this matrix equation, we can obtain a_0, a_1, and a_2
- This technique can be extended for a polynomial of n^{th} order starting with the expression
  $y = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + e(x)$
  and following the same procedure described above.

I will use the technique that was explained here and generalize it to 2nd, 5th, 10th and 30th degree polynomials.

```
function leastsquarepolynomial(Xtrain, Ytrain)
    n = length(Xtrain);
    sum1 = sum(Xtrain);
    sum2 = sum(Ytrain);
    sum3 = sum(Xtrain.*Xtrain);
    sum4 = sum(Xtrain.^3);
    sum5 = sum(Xtrain.^4);
    sum6 = sum(Xtrain.*Ytrain);
    sum7 = sum(Xtrain.^2.*Ytrain);

    secondpolmat = [n sum1 sum3;
                    sum1 sum3 sum4;
                    sum3 sum4 sum5];
    result = [sum2; sum6; sum7];

    fifthpolmat = zeros(6,6);
    tenthpolmat = zeros(11,11);
    thirtiethpolmat = zeros(31,31);
```

```
result5 = zeros(6,1);
result10 = zeros(11,1);
result30 = zeros(31,1);
for a = 1:6
    if(a == 1)
        result5(a,1) = sum(Ytrain);
    else
        result5(a,1) = sum((Xtrain.^(a-1)).*Ytrain);
    end
end
for a = 1:11
    if(a == 1)
        result10(a,1) = sum(Ytrain);
    else
        result10(a,1) = sum((Xtrain.^(a-1)).*Ytrain);
    end
end
for a = 1:31
    if(a == 1)
        result30(a,1) = sum(Ytrain);
    else
        result30(a,1) = sum((Xtrain.^(a-1)).*Ytrain);
    end
end

for i=1:6
    for j=1:6
        if(i+j -2 == 0)
            fifthpolmat(i,j) = n;
        else
            fifthpolmat(i,j) = sum(Xtrain.^(i+j-2));
        end
    end
end
for i=1:11
    for j=1:11
        if(i+j -2 == 0)
            tenthpolmat(i,j) = n;
        else
            tenthpolmat(i,j) = sum(Xtrain.^(i+j-2));
        end
    end
end

for i=1:31
    for j=1:31
        if(i+j -2 == 0)
            thirtiethpolmat(i,j) = n;
        else
            thirtiethpolmat(i,j) = sum(Xtrain.^(i+j-2));
        end
    end
end

coefficients2 = secondpolmat\result;
lse = @(x)coefficients2(3)*(x.^2) + coefficients2(2)*x +
coefficients2(1);
```

```
    coefficients5 = fifthpolmat\result5;
    lse5 = @(x)coefficients5(6)*(x.^5) + coefficients5(5)*(x.^4) +
coefficients5(4)*(x.^3) + coefficients5(3)*(x.^2) +
coefficients5(2)*x + coefficients5(1);

    coefficients10 = tenthpolmat\result10;
    lse10 = @(x)coefficients10(11)*(x.^10)
+coefficients10(10)*(x.^9) +coefficients10(9)*(x.^8) ...
        +coefficients10(8)*(x.^7) +coefficients10(7)*(x.^6)
+coefficients10(6)*(x.^5) ...
        +coefficients10(5)*(x.^4) +coefficients10(4)*(x.^3)
+coefficients10(3)*(x.^2) ...
        + coefficients10(2)*x + coefficients10(1);

    coefficients30 = thirtiethpolmat\result30;
    lse30 = @(x)coefficients30(31)*(x.^30)+
coefficients30(30)*(x.^29) + coefficients30(29)*(x.^28) +
coefficients30(28)*(x.^27)...
        + coefficients30(27)*(x.^26) + coefficients30(26)*(x.^25) +
coefficients30(25)*(x.^24)...
        + coefficients30(24)*(x.^23) + coefficients30(23)*(x.^22) +
coefficients30(22)*(x.^21)...
        + coefficients30(21)*(x.^20) + coefficients30(20)*(x.^19) +
coefficients30(19)*(x.^18)...
        + coefficients30(18)*(x.^17) + coefficients30(17)*(x.^16) +
coefficients30(16)*(x.^15)...
        + coefficients30(15)*(x.^14) + coefficients30(14)*(x.^13)+
coefficients30(13)*(x.^12)...
        + coefficients30(12)*(x.^11) + coefficients30(11)*(x.^10) +
coefficients30(10)*(x.^9)...
        + coefficients30(9)*(x.^8) + coefficients30(8)*(x.^7) +
coefficients30(7)*(x.^6)...
        + coefficients30(6)*(x.^5) + coefficients30(5)*(x.^4) +
coefficients30(4)*(x.^3)...
        + coefficients30(3)*(x.^2) + coefficients30(2)*x +
coefficients30(1);

    p2 = polyfit(Xtrain, Ytrain, 2);
    p5 = polyfit(Xtrain, Ytrain, 5);
    p10 = polyfit(Xtrain, Ytrain, 10);
    p30 = polyfit(Xtrain, Ytrain, 30);
    f2 = polyval(p2,Xtrain);
    f5 = polyval(p5,Xtrain);
    f10 = polyval(p10,Xtrain);
    f30 = polyval(p30,Xtrain);

    abserror30 = abs(Ytrain - lse30(Xtrain));
    abserror10 = abs(Ytrain - lse10(Xtrain));
    abserror5 = abs(Ytrain - lse5(Xtrain));
    abserror2 = abs(Ytrain - lse(Xtrain));

    relative30 = abserror30./abs(Ytrain);
    relative10 = abserror10./abs(Ytrain);
    relative5 = abserror5./abs(Ytrain);
    relative2 = abserror2./abs(Ytrain);
```

```
fprintf('Second Degree Least Square Polynomial\n');
fprintf('The a2 coeffcient is %.6f', coefficients2(3));
fprintf('\n')
fprintf('The a1 coeffcient is %.6f', coefficients2(2));
fprintf('\n')
fprintf('The a0 coeffcient is %.6f', coefficients2(1));
fprintf('\n\n')

fprintf('Fifth Degree Least Square Polynomial\n');
fprintf('The a5 coeffcient is %.6f', coefficients5(6));
fprintf('\n')
fprintf('The a4 coeffcient is %.6f', coefficients5(5));
fprintf('\n')
fprintf('The a3 coeffcient is %.6f', coefficients5(4));
fprintf('\n')
fprintf('The a2 coeffcient is %.6f', coefficients5(3));
fprintf('\n')
fprintf('The a1 coeffcient is %.6f', coefficients5(2));
fprintf('\n')
fprintf('The a0 coeffcient is %.6f', coefficients5(1));
fprintf('\n\n')

fprintf('Tenth Degree Least Square Polynomial\n');
fprintf('The a10 coeffcient is %.6f', coefficients10(11));
fprintf('\n')
fprintf('The a9 coeffcient is %.6f', coefficients10(10));
fprintf('\n')
fprintf('The a8 coeffcient is %.6f', coefficients10(9));
fprintf('\n')
fprintf('The a7 coeffcient is %.6f', coefficients10(8));
fprintf('\n')
fprintf('The a6 coeffcient is %.6f', coefficients10(7));
fprintf('\n')
fprintf('The a5 coeffcient is %.6f', coefficients10(6));
fprintf('\n')
fprintf('The a4 coeffcient is %.6f', coefficients10(5));
fprintf('\n')
fprintf('The a3 coeffcient is %.6f', coefficients10(4));
fprintf('\n')
fprintf('The a2 coeffcient is %.6f', coefficients10(3));
fprintf('\n')
fprintf('The a1 coeffcient is %.6f', coefficients10(2));
fprintf('\n')
fprintf('The a0 coeffcient is %.6f', coefficients10(1));
fprintf('\n\n')

fprintf('Thirtieth Degree Least Square Polynomial\n');
fprintf('The a30 coeffcient is %.6f', coefficients30(31));
fprintf('\n')
fprintf('The a29 coeffcient is %.6f', coefficients30(30));
fprintf('\n')
fprintf('The a28 coeffcient is %.6f', coefficients30(29));
fprintf('\n')
fprintf('The a27 coeffcient is %.6f', coefficients30(28));
fprintf('\n')
fprintf('The a26 coeffcient is %.6f', coefficients30(27));
fprintf('\n')
```

```
    fprintf('The a25 coeffcient is %.6f', coefficients30(26));
    fprintf('\n')
    fprintf('The a24 coeffcient is %.6f', coefficients30(25));
    fprintf('\n')
    fprintf('The a23 coeffcient is %.6f', coefficients30(24));
    fprintf('\n')
    fprintf('The a22 coeffcient is %.6f', coefficients30(23));
    fprintf('\n')
    fprintf('The a21 coeffcient is %.6f', coefficients30(22));
    fprintf('\n')
    fprintf('The a20 coeffcient is %.6f', coefficients30(21));
    fprintf('\n')
    fprintf('The a19 coeffcient is %.6f', coefficients30(20));
    fprintf('\n')
    fprintf('The a18 coeffcient is %.6f', coefficients30(19));
    fprintf('\n')
    fprintf('The a17 coeffcient is %.6f', coefficients30(18));
    fprintf('\n')
    fprintf('The a16 coeffcient is %.6f', coefficients30(17));
    fprintf('\n')
    fprintf('The a15 coeffcient is %.6f', coefficients30(16));
    fprintf('\n')
    fprintf('The a14 coeffcient is %.6f', coefficients30(15));
    fprintf('\n')
    fprintf('The a13 coeffcient is %.6f', coefficients30(14));
    fprintf('\n')
    fprintf('The a12 coeffcient is %.6f', coefficients30(13));
    fprintf('\n')
    fprintf('The a11 coeffcient is %.6f', coefficients30(12));
    fprintf('\n')
    fprintf('The a10 coeffcient is %.6f', coefficients30(11));
    fprintf('\n')
    fprintf('The a9 coeffcient is %.6f', coefficients30(10));
    fprintf('\n')
    fprintf('The a8 coeffcient is %.6f', coefficients30(9));
    fprintf('\n')
    fprintf('The a7 coeffcient is %.6f', coefficients30(8));
    fprintf('\n')
    fprintf('The a6 coeffcient is %.6f', coefficients30(7));
    fprintf('\n')
    fprintf('The a5 coeffcient is %.6f', coefficients30(6));
    fprintf('\n')
    fprintf('The a4 coeffcient is %.6f', coefficients30(5));
    fprintf('\n')
    fprintf('The a3 coeffcient is %.6f', coefficients30(4));
    fprintf('\n')
    fprintf('The a2 coeffcient is %.6f', coefficients30(3));
    fprintf('\n')
    fprintf('The a1 coeffcient is %.6f', coefficients30(2));
    fprintf('\n')
    fprintf('The a0 coeffcient is %.6f', coefficients30(1));
    fprintf('\n\n')
    fprintf('The 2nd degree max absolute error is %.6f',
max(abserror2));
    fprintf('\n')
    fprintf('The 2nd degree max absolute relative error is %.6f',
max(relative2));
```

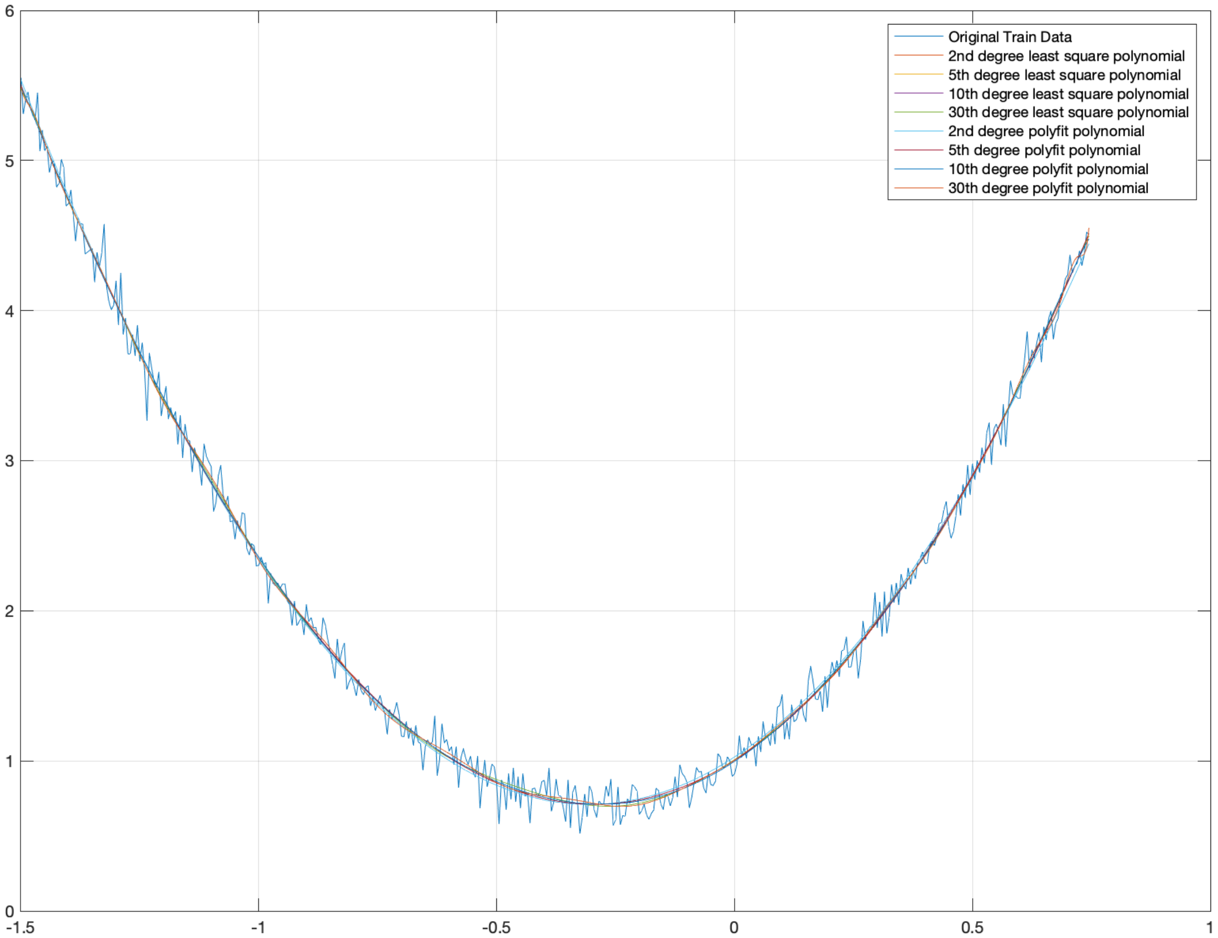
```
    fprintf('\n')
    fprintf('The 5th degree max absolute error is %.6f',
max(abserror5));
    fprintf('\n')
    fprintf('The 5th degree max absolute relative error is %.6f',
max(relative5));
    fprintf('\n')
    fprintf('The 10th degree max absolute error is %.6f',
max(abserror10));
    fprintf('\n')
    fprintf('The 10th degree max absolute relative error is %.6f',
max(relative10));
    fprintf('\n')
    fprintf('The 30th degree max absolute error is %.6f',
max(abserror30));
    fprintf('\n')
    fprintf('The 30th degree max absolute relative error is %.6f',
max(relative30));
    fprintf('\n')
    plot(Xtrain, Ytrain, 'DisplayName','Original Train Data');
    hold on;
%     plot(Xtrain, lse(Xtrain), 'DisplayName','2nd degree least
square polynomial');
%     grid on;
    plot(Xtrain, lse5(Xtrain), 'DisplayName','5th degree least
square polynomial');
    grid on;
%     plot(Xtrain, lse10(Xtrain), 'DisplayName','10th degree least
square polynomial');
%     grid on;
%     plot(Xtrain, lse30(Xtrain), 'DisplayName','30th degree least
square polynomial');
%     grid on;
%     plot(Xtrain, f2, 'DisplayName','2nd degree polyfit
polynomial');
%     grid on;
    plot(Xtrain, f5, 'DisplayName','5th degree polyfit polynomial');
    grid on;
%     plot(Xtrain, f10, 'DisplayName','10th degree polyfit
polynomial');
%     grid on;
%     plot(Xtrain, f30, 'DisplayName','30th degree polyfit
polynomial');
%     grid on;
    legend;
    hold off;
end
```

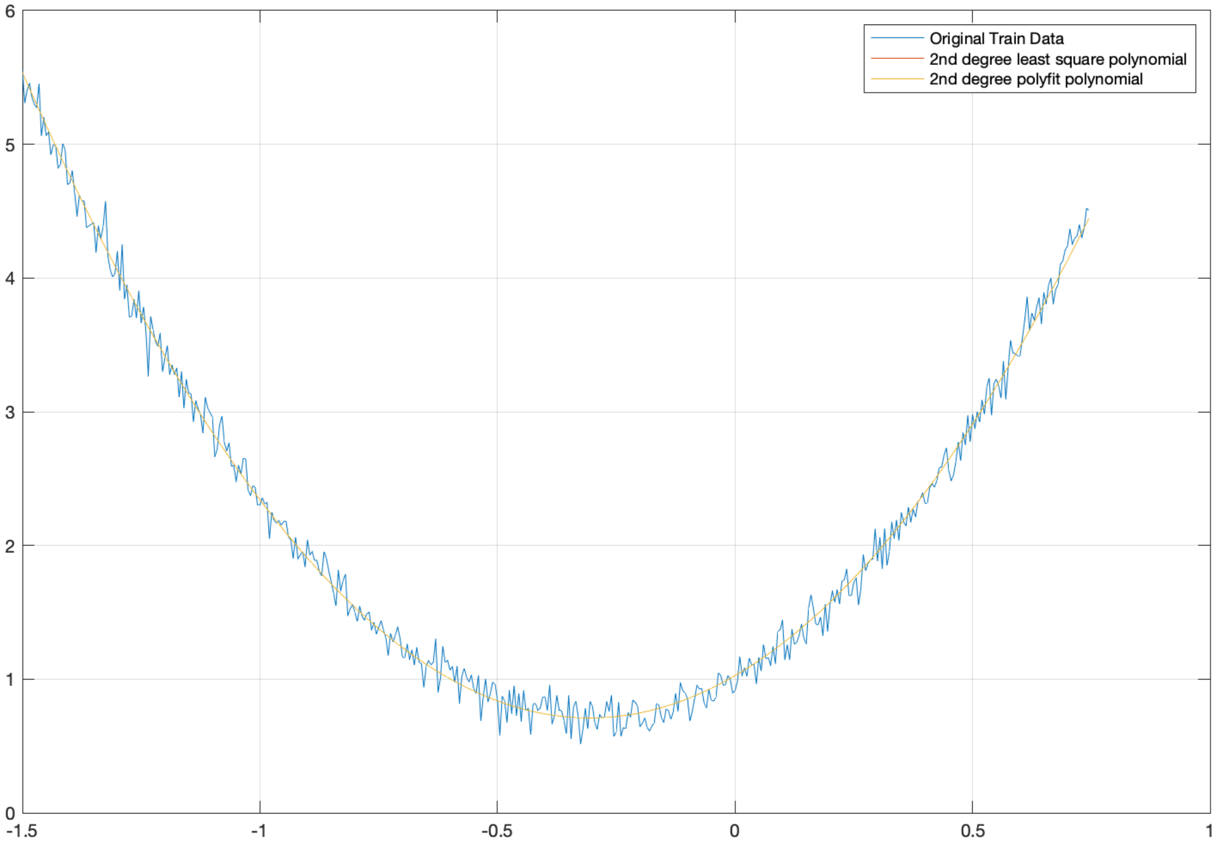
In the function I generated matrixes so that I can find the corresponding coefficients. Then, I subtracted the approximation with true value so that I can receive the absolute and relative errors. Then, I printed out the max error of each one for each polynomial. I also plotted the general version with polyfit polynomials and each function separately.

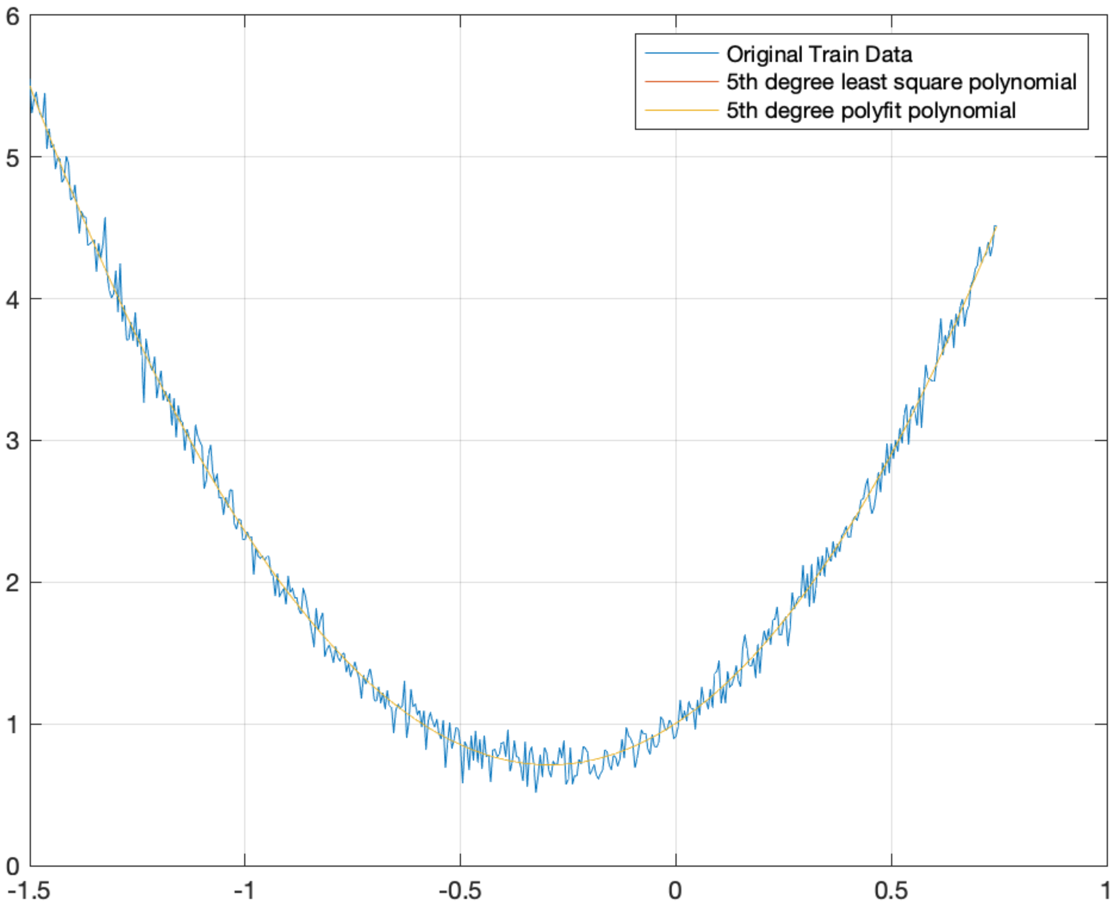
```
Second Degree Least Square Polynomial
The a2 coeffcient is 3.384460
The a1 coeffcient is 2.069660
The a0 coeffcient is 1.026455

Fifth Degree Least Square Polynomial
The a5 coeffcient is -0.007952
The a4 coeffcient is 0.035831
The a3 coeffcient is 0.170830
The a2 coeffcient is 3.487860
The a1 coeffcient is 2.008638
The a0 coeffcient is 1.004065

Tenth Degree Least Square Polynomial
The a10 coeffcient is -1.439715
The a9 coeffcient is -6.168453
The a8 coeffcient is -7.342729
The a7 coeffcient is 2.211211
The a6 coeffcient is 8.739649
The a5 coeffcient is 2.169396
The a4 coeffcient is -2.983950
The a3 coeffcient is -0.855657
The a2 coeffcient is 3.838799
The a1 coeffcient is 2.095053
The a0 coeffcient is 0.996610
```

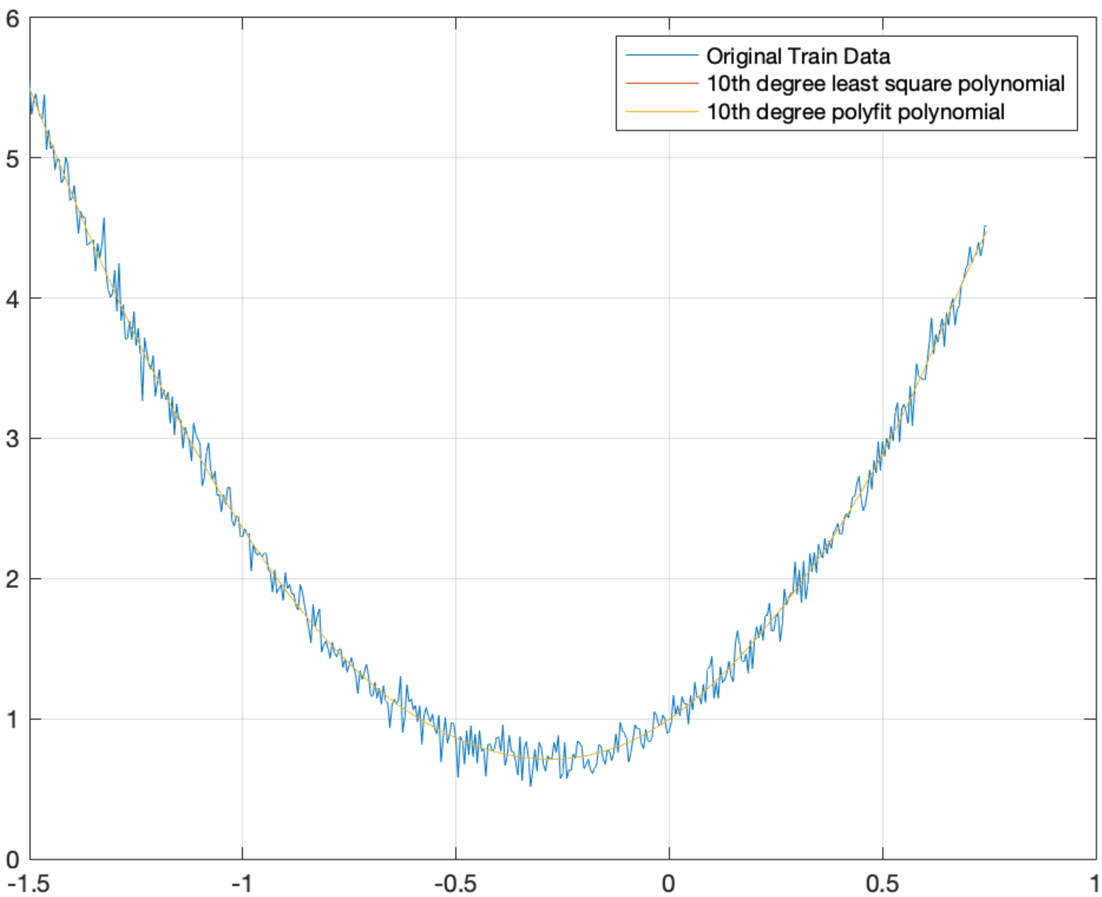
```
Thirtieth Degree Least Square Polynomial
The a30 coeffcient is 0.404228
The a29 coeffcient is 1.366222
The a28 coeffcient is 0.920820
The a27 coeffcient is -0.550190
The a26 coeffcient is -1.874082
The a25 coeffcient is -2.475354
The a24 coeffcient is 9.530547
The a23 coeffcient is 12.940417
The a22 coeffcient is -6.969332
The a21 coeffcient is -6.124877
The a20 coeffcient is -37.457732
The a19 coeffcient is -34.860530
The a18 coeffcient is 40.789692
The a17 coeffcient is -113.125289
The a16 coeffcient is -195.758612
The a15 coeffcient is -219.466450
The a14 coeffcient is -552.513099
The a13 coeffcient is 320.292212
The a12 coeffcient is 1600.108020
The a11 coeffcient is 201.642041
The a10 coeffcient is -1441.393269
The a9 coeffcient is -393.101725
The a8 coeffcient is 631.015161
The a7 coeffcient is 179.920534
The a6 coeffcient is -145.633100
The a5 coeffcient is -30.246643
The a4 coeffcient is 16.886433
The a3 coeffcient is 1.151396
The a2 coeffcient is 2.738052
The a1 coeffcient is 2.083830
The a0 coeffcient is 1.007841

The 2nd degree max absolute error is 0.364582
The 2nd degree max absolute relative error is 0.431232
The 5th degree max absolute error is 0.363292
The 5th degree max absolute relative error is 0.456748
The 10th degree max absolute error is 0.365874
The 10th degree max absolute relative error is 0.473936
The 30th degree max absolute error is 0.347372
The 30th degree max absolute relative error is 0.496635
```

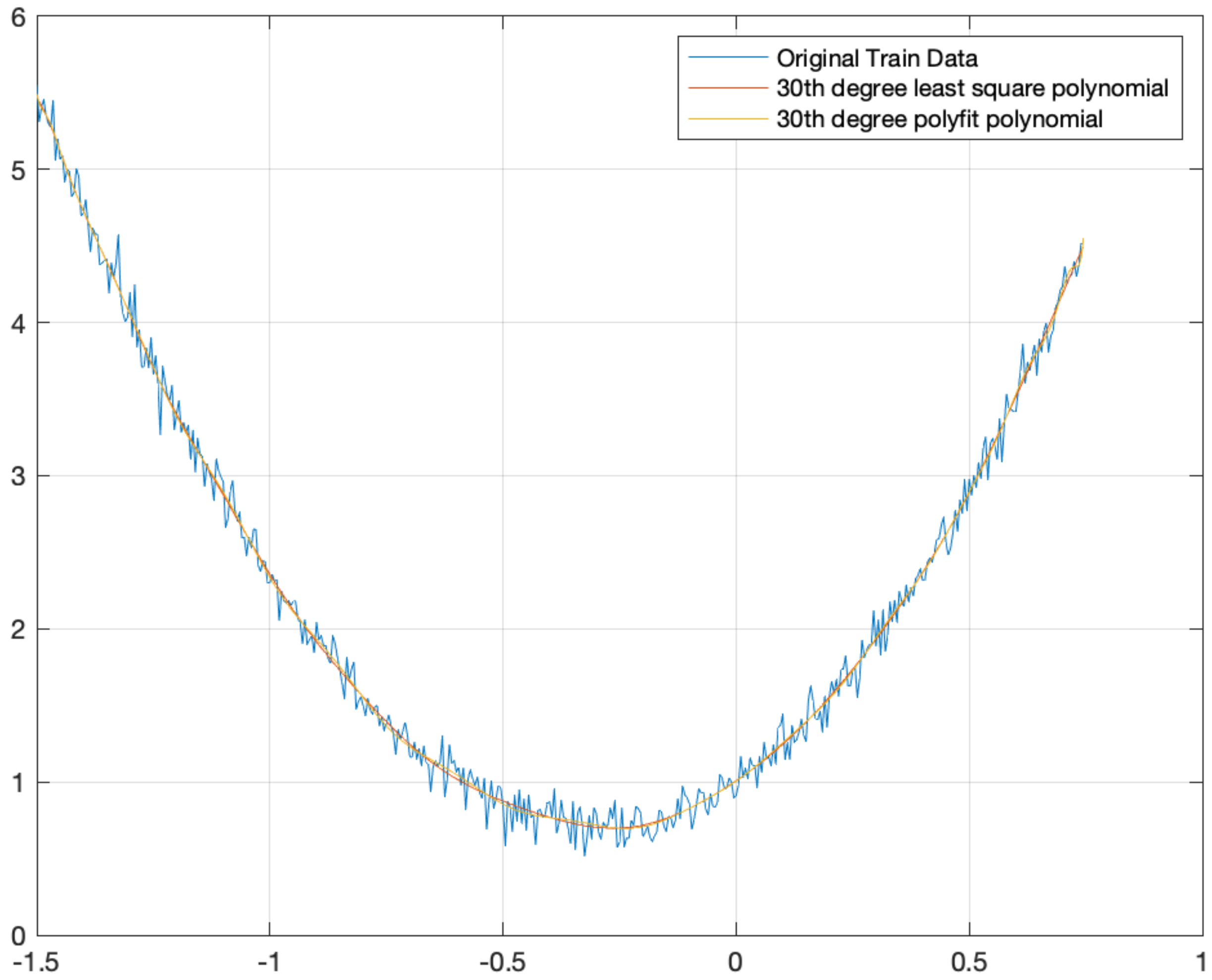
Now, let’s take a look at the figures.






Original Train Data
5th degree least square polynomial
5th degree polyfit polynomial
6
5
4
3
2
1
0
-1.5
-1
-0.5
0
0.5
1


Original Train Data
10th degree least square polynomial
10th degree polyfit polynomial
6
5
4
3
2
1
0
-1.5
-1
-0.5
0
0.5
1



When we check the results absolute and relative errors are very small. We need to test them on the test set, so that we can decide which polynomials overfitted the training data and will have a poor performance on the test data.

Question 3

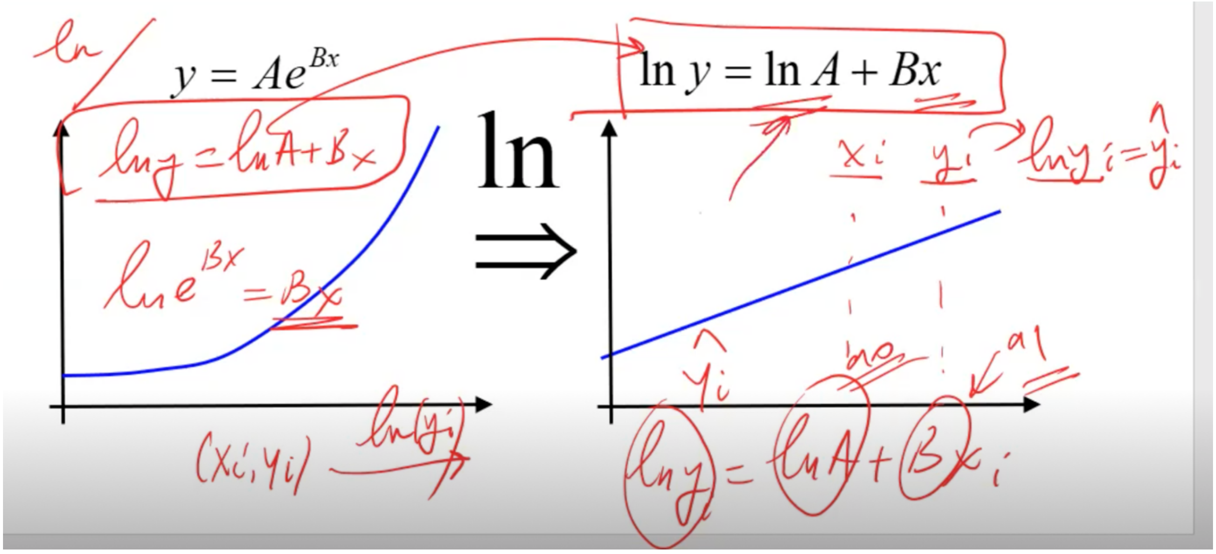
Obtain the least squares exponential fit y_e using the training data, where an exponential polynomial has the form $y = ae^{bx}$. Plot the data and the approximation y_e on the same plot. Compute the error of your approximation as in part 2. Comment on the results.

Solution

We can linearize this by taking the natural logarithm of both sides.

After this transformation, we can fit the linear regression model to this linearized model

So, I used the following approach which was told by Mustafa Hocam.




```
%% Obtain the least squares exponential fit
load('proj3data.mat');
Xtrain = x(1,1:450);
Ytrain = y(1,1:450);
leastsquareexponential(Xtrain, Ytrain);
```

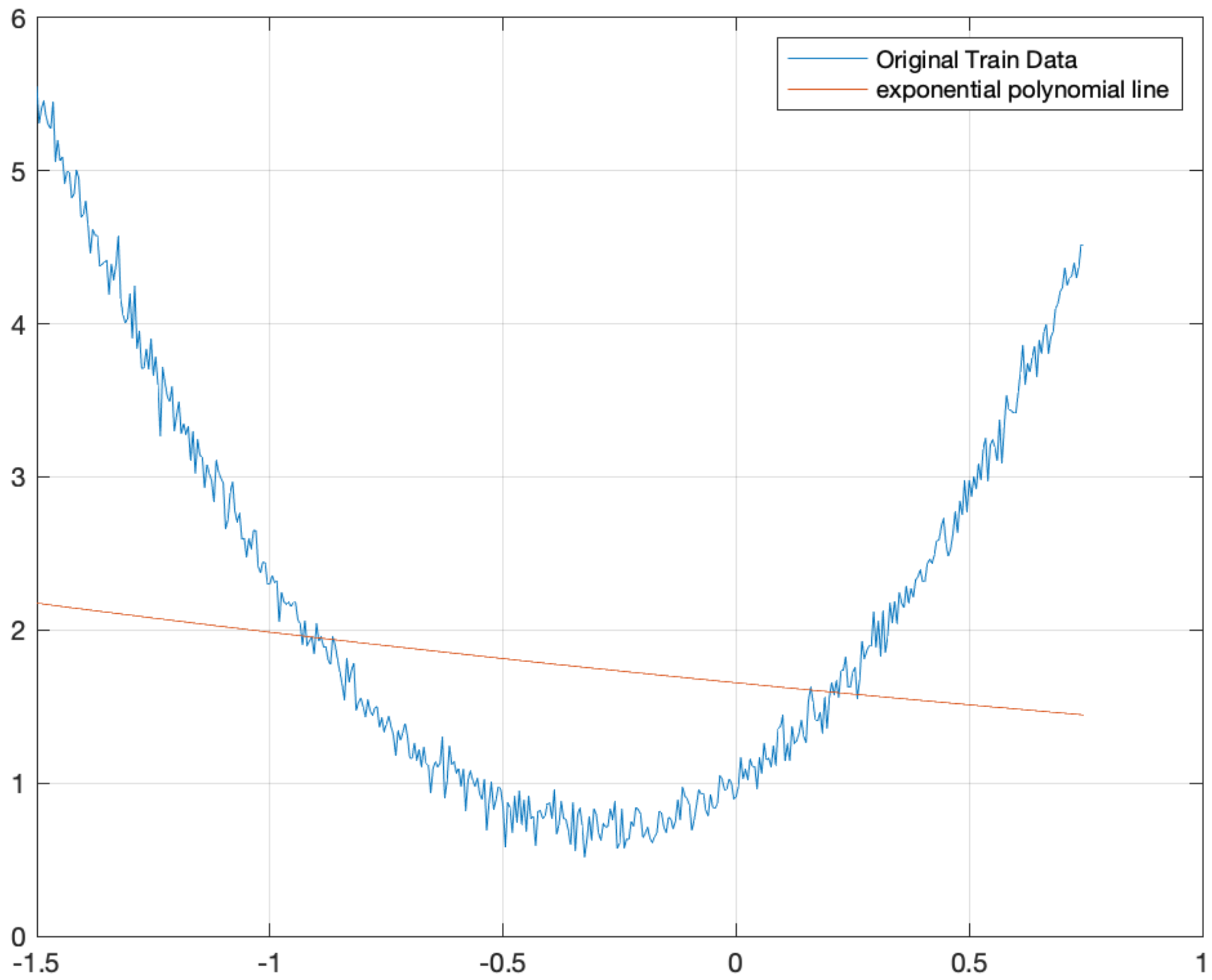
Here, we load the data and selected first 75% as training data. Then, I will find the coefficients by using the above formula.

```
function leastsquareexponential(Xtrain, Ytrain)
   n = length(Xtrain);
   logy = log(Ytrain);

   sum1 = sum(Xtrain);
   sum2 = sum(logy);
   sum3 = sum(Xtrain.*Xtrain);
   sum4 = sum(Xtrain.*logy);

   a1 = (n*sum4 - (sum1*sum2))/(n*sum3 -(sum1*sum1));
   a0 = (sum2 - (a1*sum1))/n;

   A = exp(a0);
   B = a1;
   lse = @(x)A*exp(B*x);
   abserror = abs(Ytrain - lse(Xtrain));
   relative = abserror./abs(Ytrain);
   fprintf('The A coeffcient is %.6f', A);
   fprintf('\n')
   fprintf('The B coeffcient is %.6f', B);
   fprintf('\n\n')
   fprintf('The max absolute error is %.6f', max(abserror));
   fprintf('\n')
   fprintf('The max absolute relative error is %.6f',
max(relative));
```

```
    plot(Xtrain, Ytrain, 'DisplayName','Original Train Data');
    hold on;
    plot(Xtrain, lse(Xtrain), 'DisplayName','exponential polynomial
line');
    grid on;
    legend;
    hold off;

end
```

Here, I followed the formulas given in the above screenshot where I linearized the function then use the approach that I have used in the first question. After that, I converted the results to find A and B.

```
The A coeffcient is 1.655401
The B coeffcient is -0.182035

The max absolute error is 3.377489
The max absolute relative error is 2.410702
```



It seems like the exponential line does not fit to the training data. Thus, we kind of have an underfitting problem.

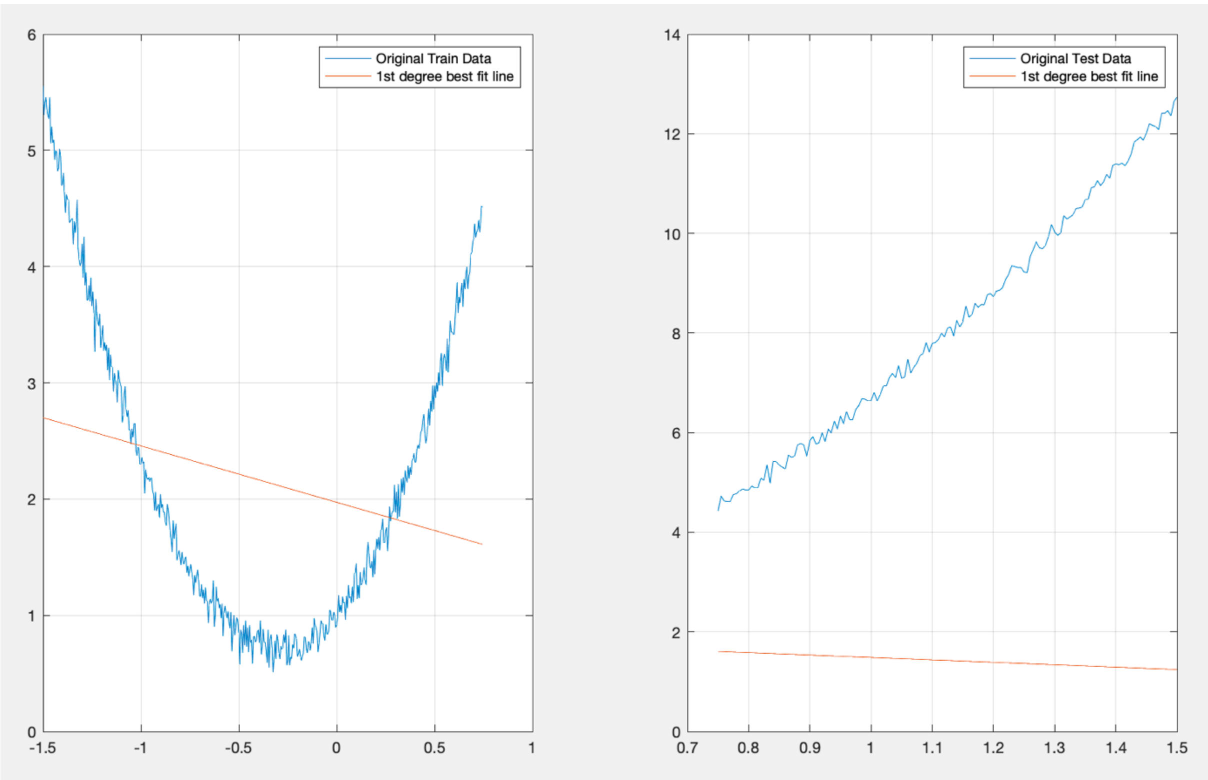
Question 4

Evaluate your models (part 1., 2., and 3.) on the test data and compute the errors in the models' predictions. Plot the errors on the same plot. Which model you would choose and why?

Solution

Part 1

```
The a1 coeffcient is -0.485607
The a0 coeffcient is 1.971961
The correlation coefficient r is -0.239105
The original uncertainty explained by the linear model is 0.057171

Test Data
The max absolute error is 11.492170
The max absolute relative error is 0.902357
```

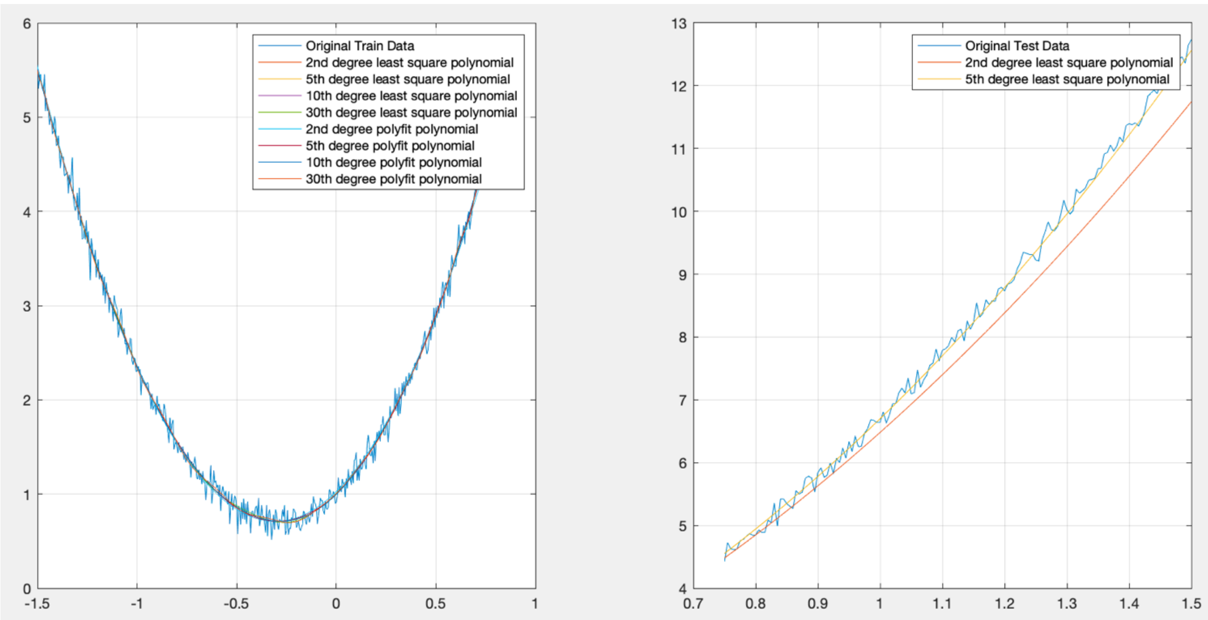


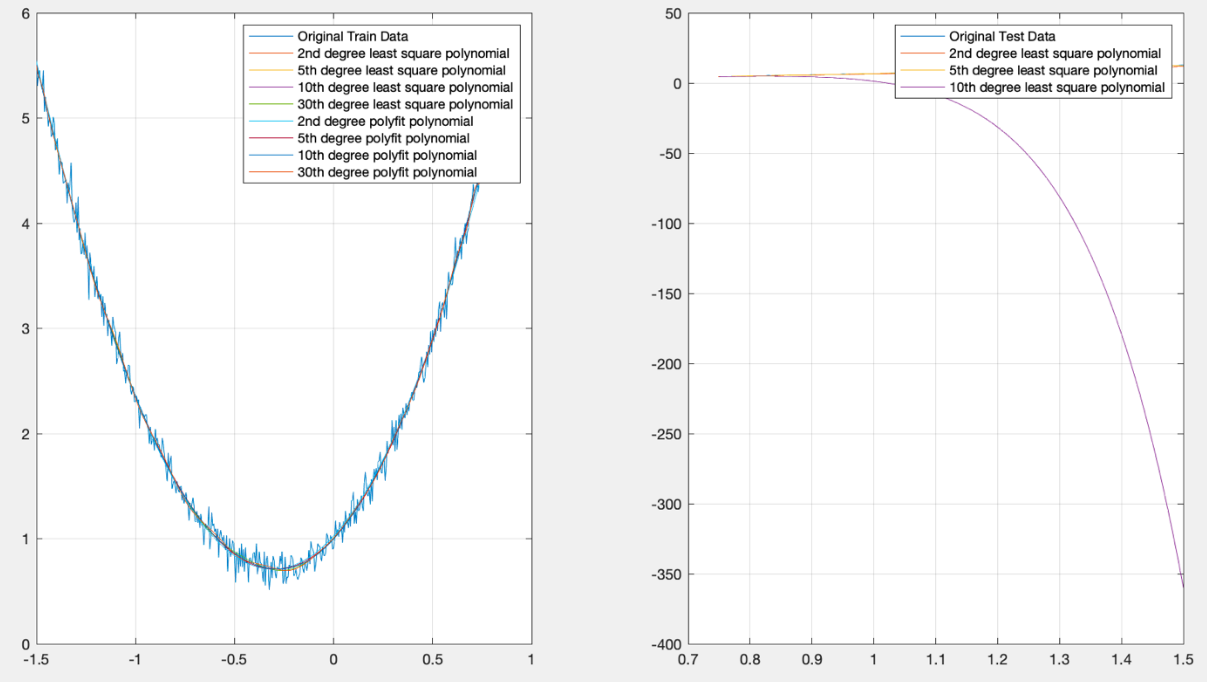
Part 2

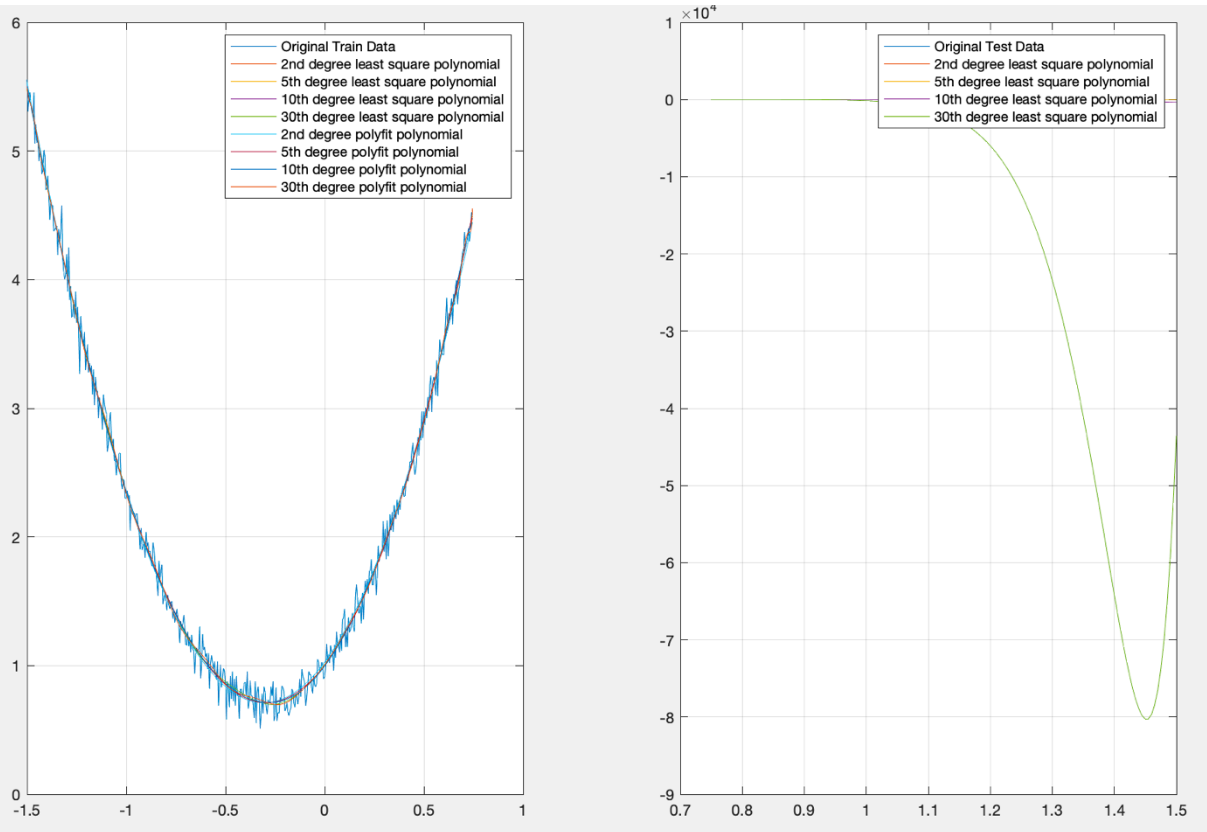
```
Second Degree Least Square Polynomial
The a2 coeffcient is 3.384460
The a1 coeffcient is 2.069660
The a0 coeffcient is 1.026455

Fifth Degree Least Square Polynomial
The a5 coeffcient is -0.007952
The a4 coeffcient is 0.035831
The a3 coeffcient is 0.170830
The a2 coeffcient is 3.487860
The a1 coeffcient is 2.008638
The a0 coeffcient is 1.004065

Tenth Degree Least Square Polynomial
The a10 coeffcient is -1.439715
The a9 coeffcient is -6.168453
The a8 coeffcient is -7.342729
The a7 coeffcient is 2.211211
The a6 coeffcient is 8.739649
The a5 coeffcient is 2.169396
The a4 coeffcient is -2.983950
The a3 coeffcient is -0.855657
The a2 coeffcient is 3.838799
The a1 coeffcient is 2.095053
The a0 coeffcient is 0.996610
```

```
Thirtieth Degree Least Square Polynomial
The a30 coeffcient is 0.404228
The a29 coeffcient is 1.366222
The a28 coeffcient is 0.920820
The a27 coeffcient is -0.550190
The a26 coeffcient is -1.874082
The a25 coeffcient is -2.475354
The a24 coeffcient is 9.530547
The a23 coeffcient is 12.940417
The a22 coeffcient is -6.969332
The a21 coeffcient is -6.124877
The a20 coeffcient is -37.457732
The a19 coeffcient is -34.860530
The a18 coeffcient is 40.789692
The a17 coeffcient is -113.125289
The a16 coeffcient is -195.758612
The a15 coeffcient is -219.466450
The a14 coeffcient is -552.513099
The a13 coeffcient is 320.292212
The a12 coeffcient is 1600.108020
The a11 coeffcient is 201.642041
The a10 coeffcient is -1441.393269
The a9 coeffcient is -393.101725
The a8 coeffcient is 631.015161
The a7 coeffcient is 179.920534
The a6 coeffcient is -145.633100
The a5 coeffcient is -30.246643
The a4 coeffcient is 16.886433
The a3 coeffcient is 1.151396
The a2 coeffcient is 2.738052
The a1 coeffcient is 2.083830
The a0 coeffcient is 1.007841
```

```
The 2nd degree max absolute error is 0.364582
The 2nd degree max absolute relative error is 0.431232
The 5th degree max absolute error is 0.363292
The 5th degree max absolute relative error is 0.456748
The 10th degree max absolute error is 0.365874
The 10th degree max absolute relative error is 0.473936
The 30th degree max absolute error is 0.347372
The 30th degree max absolute relative error is 0.496635

Test Data
The 2nd degree max absolute error is 1.001484
The 2nd degree max absolute relative error is 0.082060
The 5th degree max absolute error is 0.281954
The 5th degree max absolute relative error is 0.047746
The 10th degree max absolute error is 372.492219
The 10th degree max absolute relative error is 29.247832
The 30th degree max absolute error is 80281.857899
The 30th degree max absolute relative error is 6726.844606
```

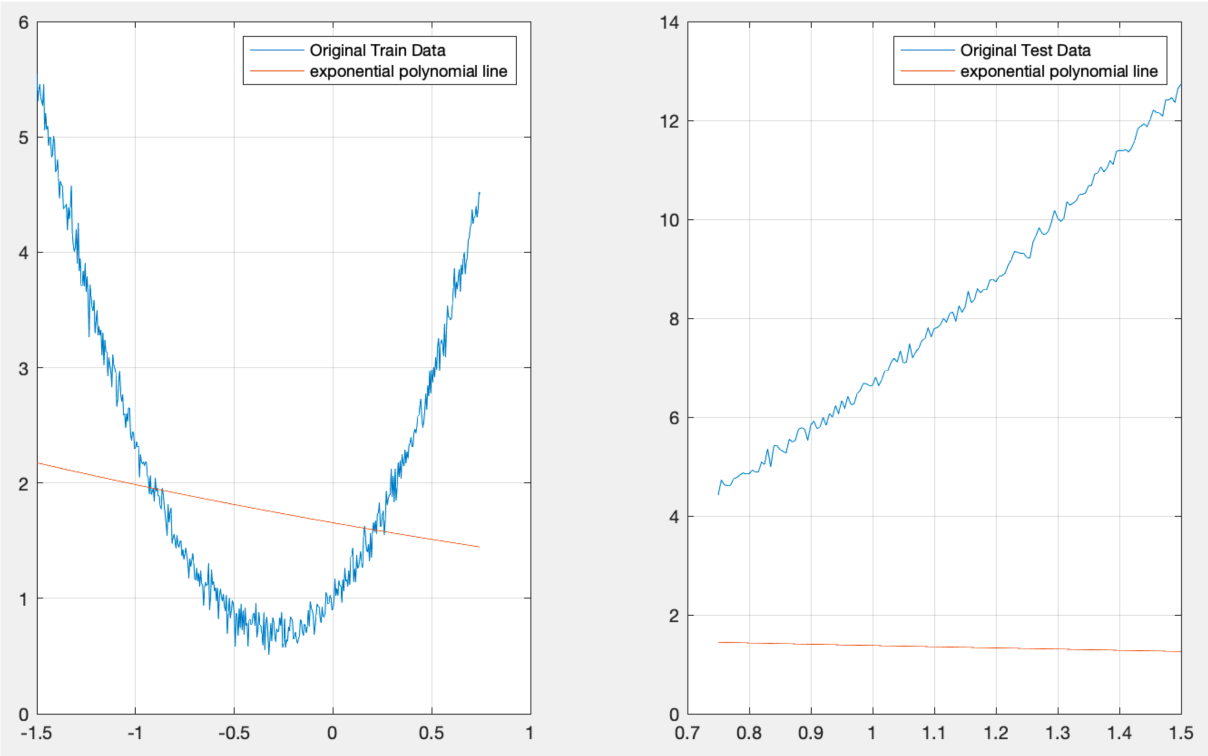



Original Train Data
2nd degree least square polynomial
5th degree least square polynomial
10th degree least square polynomial
30th degree least square polynomial
2nd degree polyfit polynomial
5th degree polyfit polynomial
10th degree polyfit polynomial
30th degree polyfit polynomial
Original Test Data
2nd degree least square polynomial
5th degree least square polynomial
10th degree least square polynomial

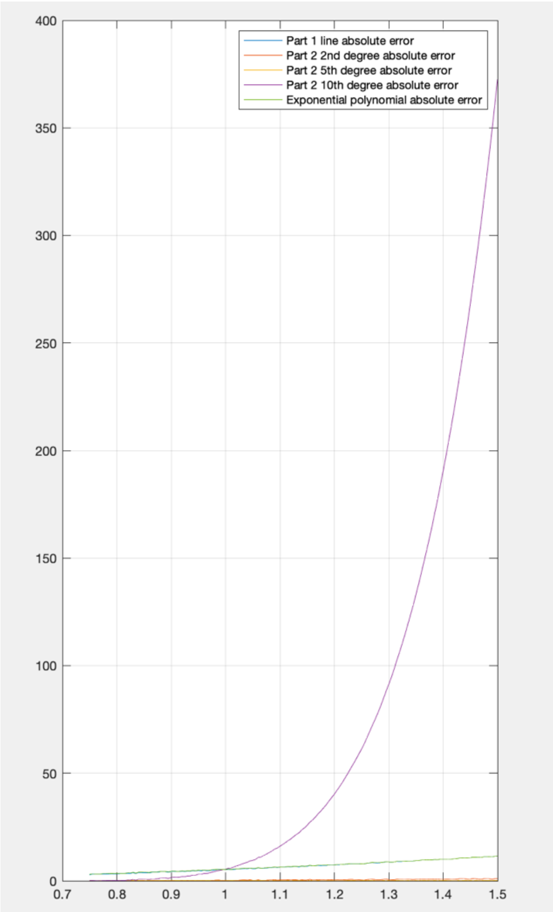

Original Train Data
2nd degree least square polynomial
5th degree least square polynomial
10th degree least square polynomial
30th degree least square polynomial
2nd degree polyfit polynomial
5th degree polyfit polynomial
10th degree polyfit polynomial
30th degree polyfit polynomial
Original Test Data
2nd degree least square polynomial
5th degree least square polynomial
10th degree least square polynomial
30th degree least square polynomial

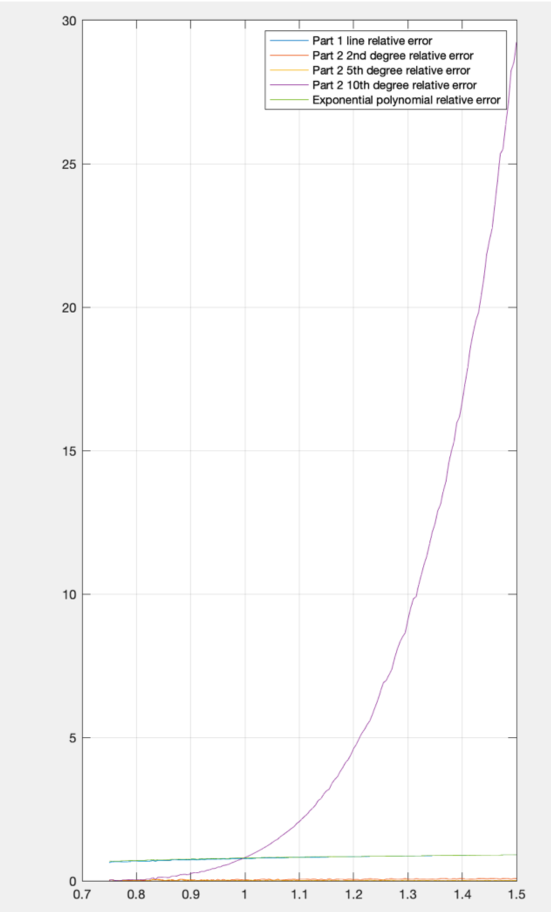
Part 3

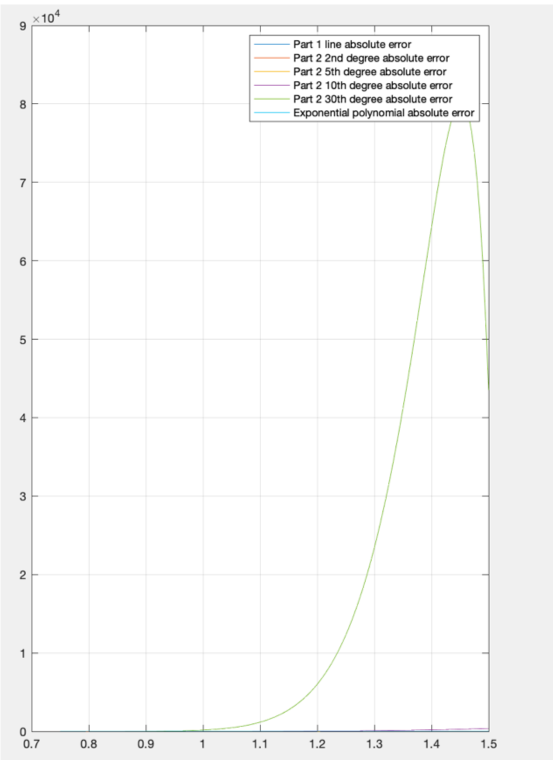
```
The A coeffcient is 1.655401
The B coeffcient is -0.182035

The max absolute error is 3.377489
The max absolute relative error is 2.410702

Test Data
The max absolute error is 11.475872
The max absolute relative error is 0.901078
```

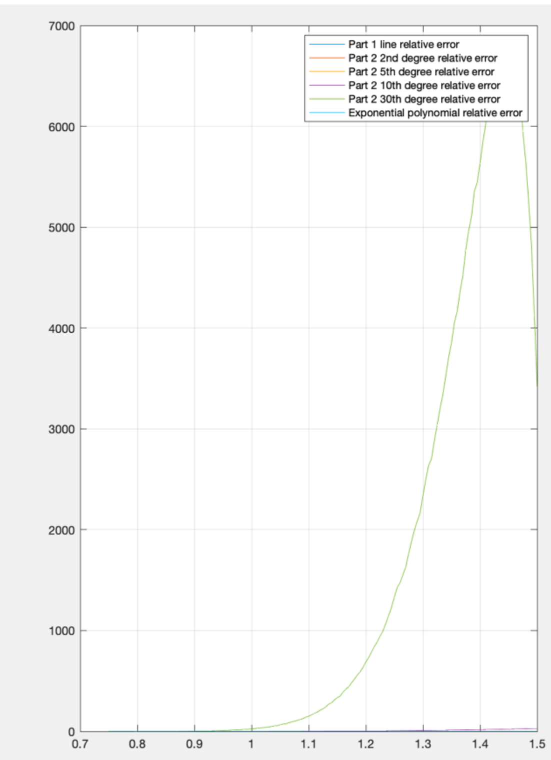


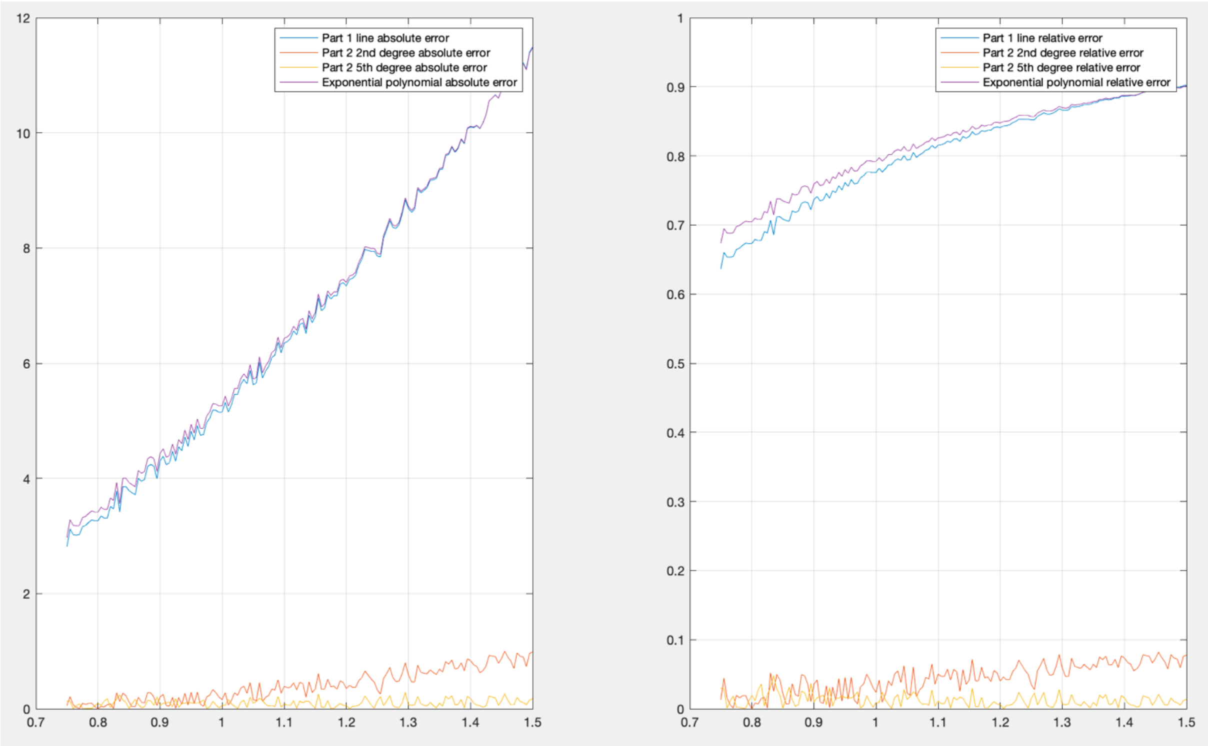
Printing the errors











When I check the plots and the errors on the test set. I would choose 2nd degree polynomial or 5th degree polynomial. As you can see 10th degree polynomial and 30th degree polynomial were overfitting the train data. Thus, their performance on the test data is very poor. Furthermore, I think exponential one and the linear regression the training dataset had a poor performance when we compare them with others. However, their performance on the test set is better than 10th degree polynomial and 30th degree polynomial.