Assignment I: Linear Regression

Sumedh Sopan Nagrale

Electrical Engineering

U00846175 , fall 2017

Wright State university

Contents

1 Regression:

Regression is a measure of the relation between the mean value of one variable (output) and corresponding values of the other variables(eg: time and Cost).
The choice of a good function depends on the type of application and dataset.

1.1 Linear Regression:

Modelling bivariate data as a function of a function and noise.
Bivariate data : (x_1, y_1) , (x_2, y_2),.....(x_n, y_n)
Model:

$$y_i = f(x_i) + E_i$$

where,
$f(x_i)$: Some Function
E_i : Random error

1.2 Hypothesis Function:

Hypothesis function gives a relation between the input and the output.
Example of hypothesis function:

1. lines : $y = ax + b + E$
2. polynomial : $y = ax^2 + bx + C + E$
3. others:
   (a) $y = \frac{a}{x} + b + E$
   (b) $y = asin(x) + b + E$

A multivariate hypothesis is given by:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n$$

1.2.1 Initial Values of θ

The ultimate goal is to find a optimal value of the $\theta's$. Selection of θ is random. However, If we have an idea of how the data looks like we can select θ values which will make our algorithm converge the optimal point faster.

1.3 Cost Function:

The cost function is defined by:

$$J(\theta_0, \theta_1) = \frac{1}{2M}\sum_{i=1}^{M}(h_\theta(x^i) - y^i)^2$$

The aim is to minimise the cost function $J(\theta)$[1], basically it is a summation of all the error our hypothesis has made. In other words we can say we are searching for values of the parameters or simply, θ which will fetch us a minimum value of Cost Function.

1.4 Mean Square Error:

Mean square error is the difference between hypothesis $h_\theta(x)$ and the output values(y). It shows how good our estimator is.

2 Optimization:

We want to minimise cost function over the training data samples. We want to experiment with different values of θ, So that the cost function keeps reducing so we end up in the minimum.

2.1 Gradient Descent:

Gradient Descent is used to find the minimum error by minimizing the cost function.

$$\theta_j = \theta_j - \alpha\frac{d}{d\theta_j}J(\theta)$$

where,
α = Learning Rate. [2] $\frac{d}{d\theta_j}J(\theta)$, on reaching the minima the slope becomes zero. This is a good place to stop.[3]

2.2 Batch Process:

If we are considering all the data points in the dataset to make a single update, Here, we are implementing Batch process.

[1] The value $\frac{1}{M}$ normalise the data.

[2] Selection of α depends on the dataset and the type of application.if the α is too small the convergence will be be slower. If the α value is too large, minima of the cost function could be skipped.

[3] Since real data is usually noisy, J will converge to a value. As in the current assignment it is converging in between [0.095,0.12].

3 Dataset:

3.1 Preprocessing:

3.1.1 Outlier:

An outlier is a point which greatly influences the regression hypothesis, i.e it changes the slopes vastly. In order to avoid this from happening, We can remove the outlier.

1. In Context of Flu Dataset:
   An outlier is present in the Respiratory Etiquette, The given data is within the range of 0 to 5, However, one of the datapoint has a value of 9. Which makes our hypothesis function change the values of estimated θ drastically. In order to avoid this from happening, We have to remove this datapoint.

3.1.2 NAN:

The presence of NAN in the dataset will create a problem of calculation of the gradient. Which will lead us to a result of NAN.
Two possible solutions are available, one is to remove the NAN other is to replace it with 0.

1. In Context of Flu Dataset:
   The NAN data, if used directly resulted in a output of NAN. which makes no sense. Since the gradient is a mathematical algorithm, presence of such data point lead to problem in calculations. I have removed it from the dataset. I didn't use 0 in place of NAN because that would be a modification of the dataset, which may lead us to a wrong analysis.

3.1.3 Normalisation/Feature Scaling:

Here, By Normalisation I mean Feature scaling. Feature scaling is used to bring the input variable or features of data in a Range which makes the comparison of the data in the dataset easier. The Feature scaling can be done in multiple way. One of the way is using the following formula:

$$X_{scaled} = a + (b - a)\frac{(X - X_{min})}{X_{max} - X_{min}}$$

.

1. In Context of Flu Dataset:
   I have converted the input variables within the range [0,1]. So as to increase the ease of comparison between the data point.

3.2 Dataset Bifurcation:

Dataset is bifurcated in order to compare the results with different ratio of testing data and test data.

3.2.1 Training Dataset:

Data used to learn the relationship between the independent and target variable.

3.2.2 Test Dataset:

Data which is used to estimate the accuracy.

1. In Context of Flu Dataset:

   Since we need to experiment with different ratios of training and test data.

   I have bifurcated Data into Training and Test data in the following ratio:

Dataset	Training	Test	Ratio
1	164	244	40:60
2	204	204	50:50
3	286	122	70:30
4	367	41	90:10

4 Linear Regression with Single Variable:

1. Hypothesis: The hypothesis function for the linear regression with a single variable is given

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

2. Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2M} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)^2$$
$$= \frac{1}{2M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 - y)^2$$

   where,

   x_1 : KnowlTrans

   y : Risk.

3. Gradient Descent: Gradient descent is an optimisation algorithm, which minimises the cost function stated above. i.e it finds the values of θ_0 and θ_1 such that the cost function is minimum.

$$\theta_0 = \theta_0 - \alpha \frac{d}{d\theta_0} J(\theta_0, \theta_1) = \theta_0 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 - y)$$

$$\theta_1 = \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_0, \theta_1) = \theta_1 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 - y) * x_1$$

4. Error Once, we find a minimum Cost function (J_{min}), the corresponding θ's are used to find the Mean square error in the test data. The error will show how good our model is.

   Mean square error(i)

$$MSE(i) = \frac{1}{M} (h_\theta(x^i) - y^i)^2$$

5 Quadratic Regression with One Variable

1. Hypothesis: The hypothesis function for the Quadratic Regression with One Variable is given by

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta x_1^2$$

2. Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2M} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)^2$$

$$= \frac{1}{2M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta x_1^2 - y)^2$$

where,

x_1 : KnowlTrans

y : Risk.

3. Gradient Descent: Gradient descent is an optimisation algorithm, which minimises the cost function stated above. i.e it finds the values of θ_0 and θ_1 such that the cost function is minimum.

$$\theta_0 = \theta_0 - \alpha \frac{d}{d\theta_0} J(\theta_0, \theta_1, \theta_2) = \theta_0 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 - y)$$

$$\theta_1 = \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_0, \theta_1, \theta_2) = \theta_1 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 - y) * x_1$$

$$\theta_2 = \theta_2 - \alpha \frac{d}{d\theta_2} J(\theta_0, \theta_1, \theta_2) = \theta_2 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 - y) * x_1^2$$

4. Error Once, we find a minimum Cost function (J_{min}), the corresponding θ's are used to find the Mean square error in the test data. The error will show how good our model is.
Mean square error(i)

$$MSE(i) = \frac{1}{M} (h_\theta(x^i) - y^i)^2$$

6 Linear Regression with Two Variables

1. Hypothesis: The hypothesis function for Linear Regression with Two Variables:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta x_2$$

2. Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2M} \sum_{i=1}^{M} (h_\theta(x^i) - y^i)^2$$

$$= \frac{1}{2M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta x_2 - y)^2$$

where,

x_1 : KnowlTrans

x_2 : RespEtiq

y : Risk.

3. Gradient Descent: Gradient descent is an optimisation algorithm, which minimises the cost function stated above. i.e it finds the values of θ_0 and θ_1 such that the cost function is minimum.

$$\theta_0 = \theta_0 - \alpha \frac{d}{d\theta_0} J(\theta_0, \theta_1, \theta_2) = \theta_0 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta_2 x_2 - y)$$

$$\theta_1 = \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_0, \theta_1, \theta_2) = \theta_1 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta_2 x_2 - y) * x_1$$

$$\theta_2 = \theta_2 - \alpha \frac{d}{d\theta_2} J(\theta_0, \theta_1, \theta_2) = \theta_2 - \alpha \frac{1}{M} \sum_{i=1}^{M} (\theta_0 + \theta_1 x_1 + \theta_2 x_2 - y) * x_2$$

4. Error Once, we find a minimum Cost function (J_{min}), the corresponding θ's are used to find the Mean square error in the test data. The error will show how good our model is.
   Mean square error(i)

$$MSE(i) = \frac{1}{M}(h_\theta(x^i) - y^i)^2$$

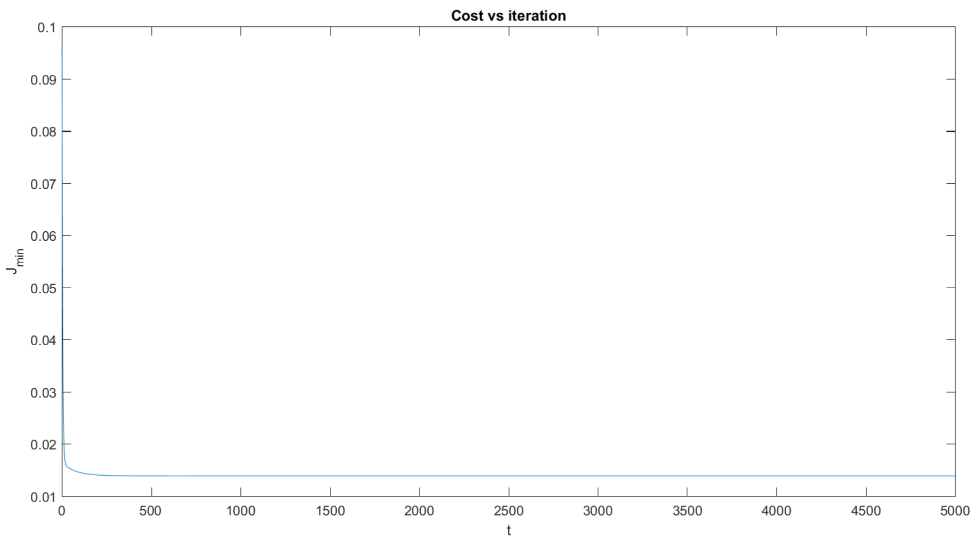
7 Answers to the Questions, plots, results and Observations:

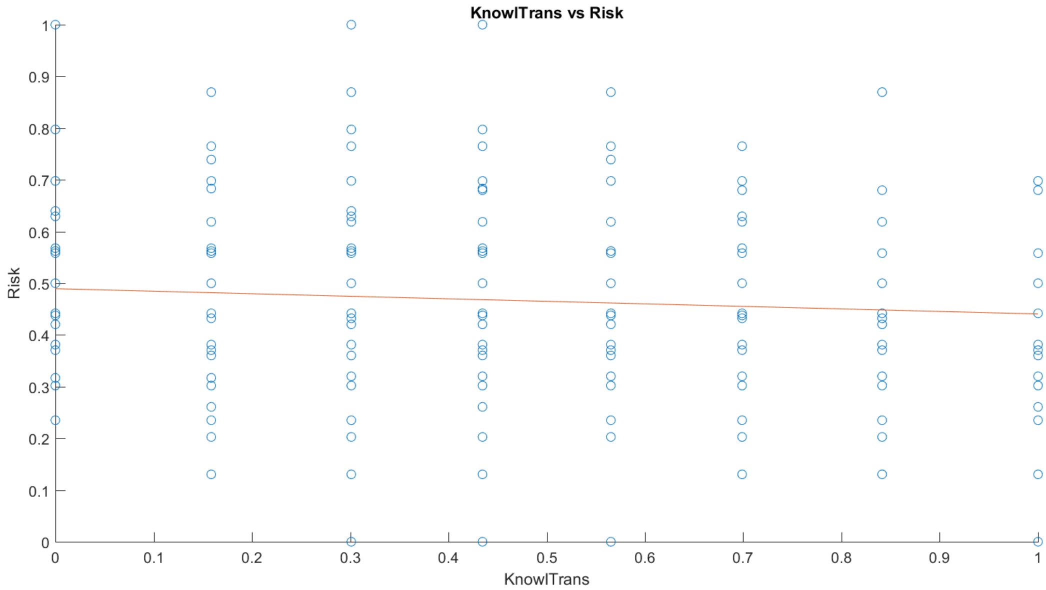
7.1 Linear Regression with One Variable:

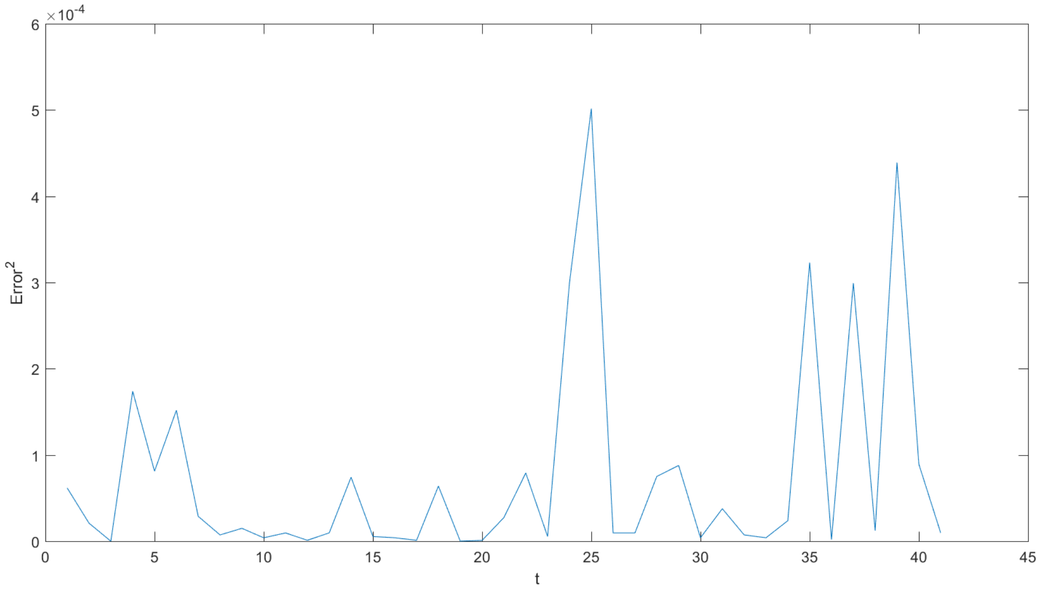
Q) Can you map the Knowledge of flu transmission (KnowlTrans) to the perceived risk (Risk) of contracting influenza? Note, here Risk is the target variable.

→ Yes, Knowledge of flu transmission can be mapped to the perceived risk of contracting influenza. where risk is the target variable. The plots and results are shown subsequent section

7.1.1 Plots:







7.1.2 Results:

Dataset	Training	Test	Ratio	$Train : J_{min}$	θ_0	θ_1	$J_{min}Test$	$Error^2$	Iteration
1	367	41	90:10	0.0139	0.4219	-0.0489	0.0031	0.0005012	4219

7.2 Observation:

1. The Normalised dataset leads us to a smaller value of Cost Function.

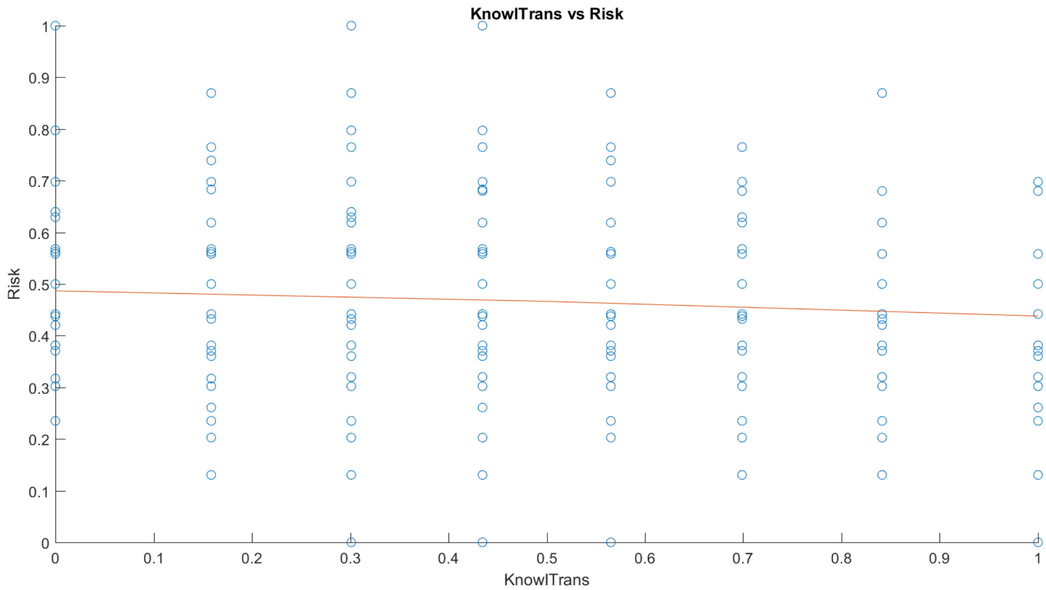
7.3 Quadratic Regression with One Variable:

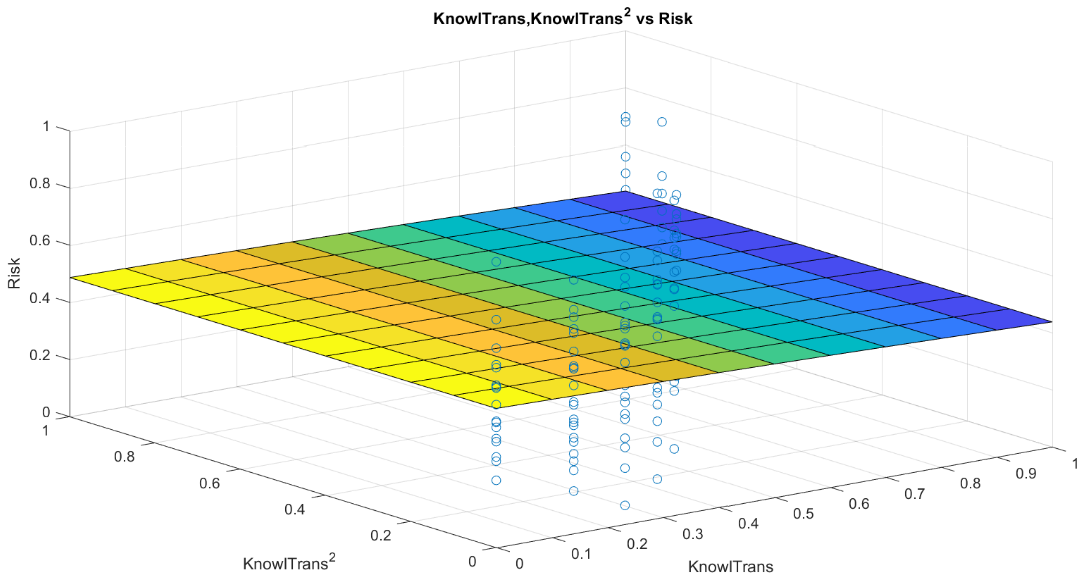
Q)Can you show the results of mapping the KnowlTrans to the Risk (y) using a quadratic expression? $\rightarrow$ Yes, Results of mapping the KnowlTrans to the Risk (y) using a quadratic expression can be shown. please look into the plot section

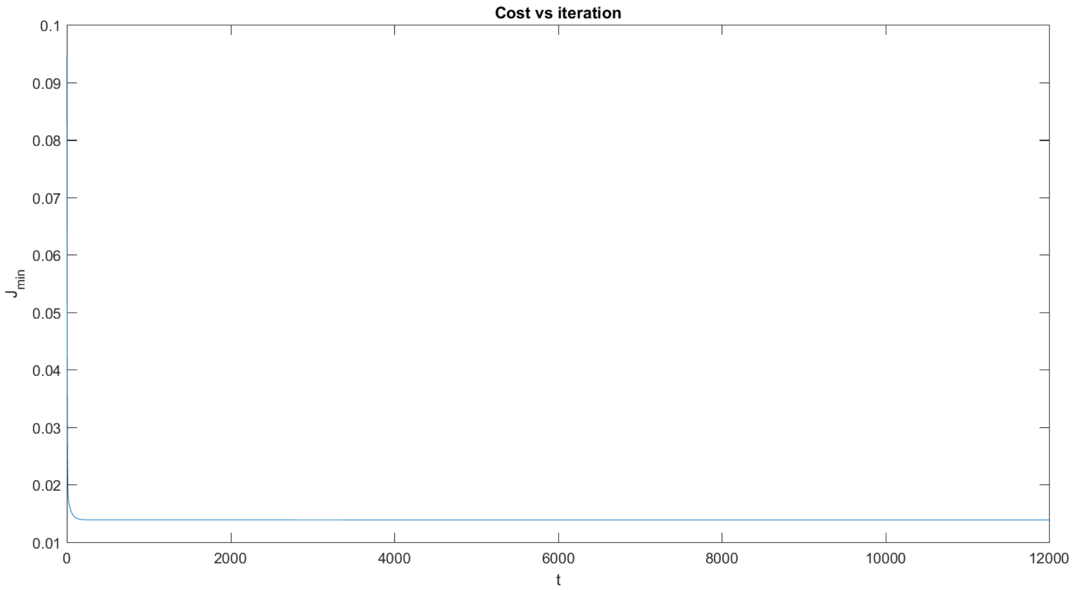
Q) Does the quadratic expression improve the performance of the algorithm from (1)? Please use graphs or other visualization techniques to compare the performance of 1 and 2.

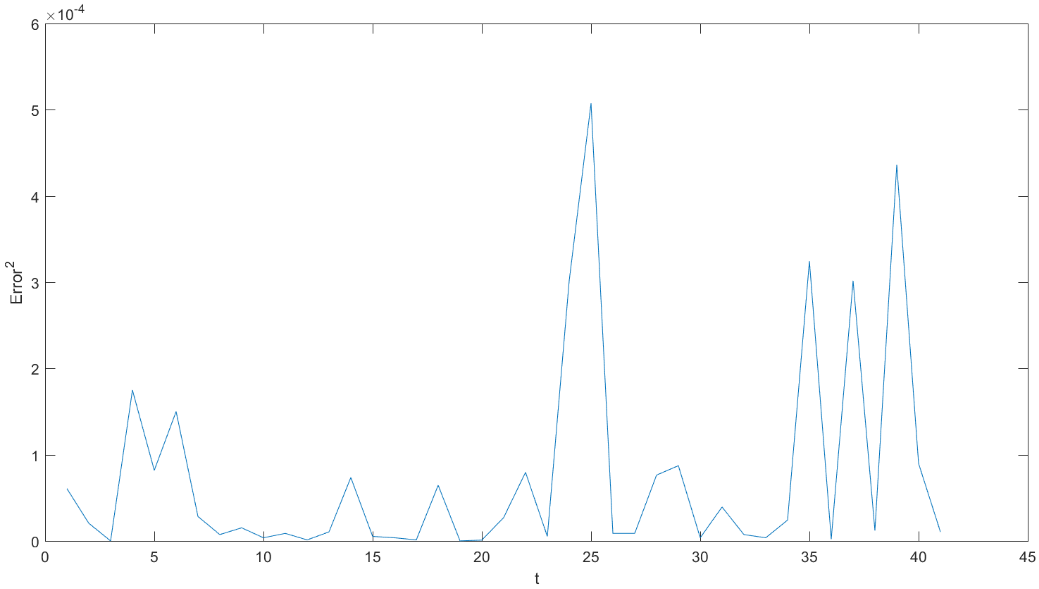
→ It has three parameters, θ_0, θ_1 and θ_2, that are to be estimated along with the Quadratic term which makes it slower for the Quadratic expression to converge. This can be verified from the table in the result section, where one can see the number of iteration is more quadratic expression than others to reach a same value.

7.3.1 Plots:









7.3.2 Result

Quadratic Regression with One Variable:

Dataset	Training	Test	Ratio	$Train : J_{min}$	θ_0	θ_1	θ_2	$J_{min}Test$	$Error^2$	Iteration
1	367	41	90:10	0.0139	0.4867	-0.0325	-0.0162	0.0031	0.000521	12000

7.4 Observation:

1. A polynomial term has a quadratic (squared) or above term turns a linear regression model into a curve. Since it is the term X that is being squared the model still qualifies as a linear function.

2. From the visual inspection, we can see the dataset is pretty scattered. As one would anticipate quadratic linear

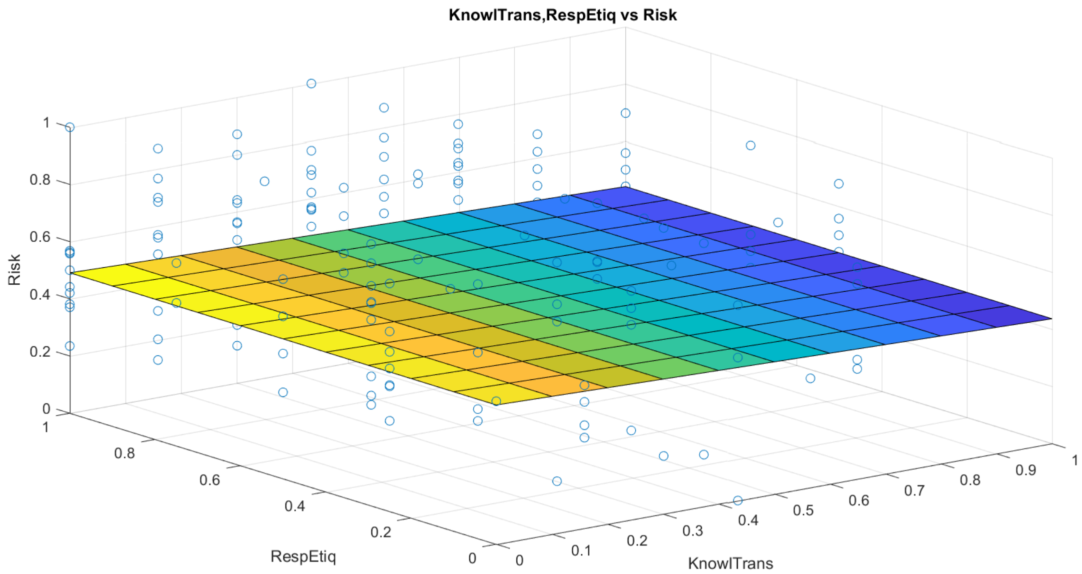
model fails to fit the dataset.

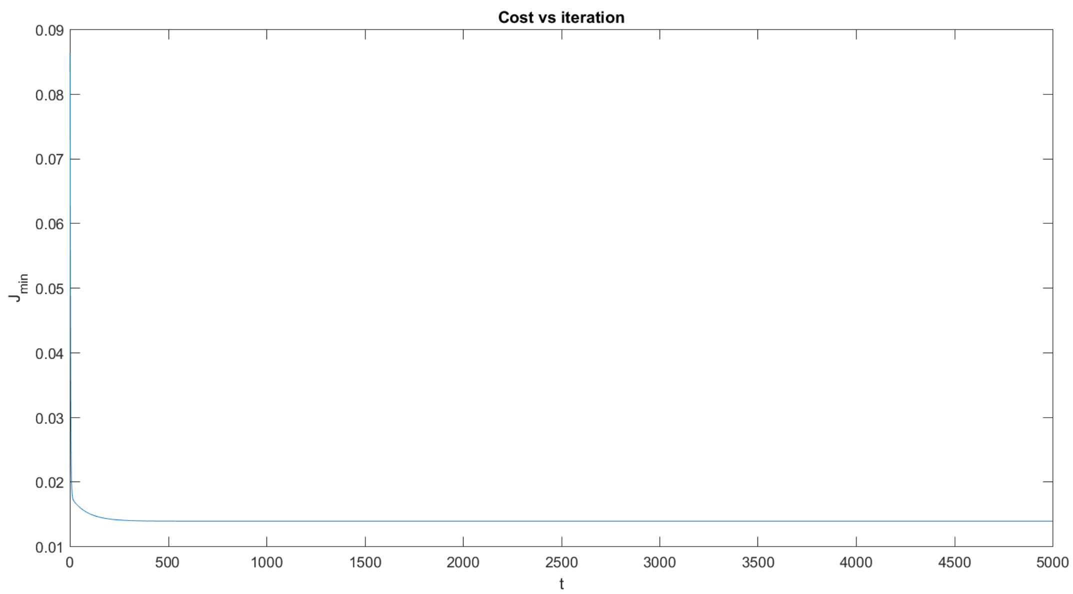
7.5 Linear Regression with Two Variables:

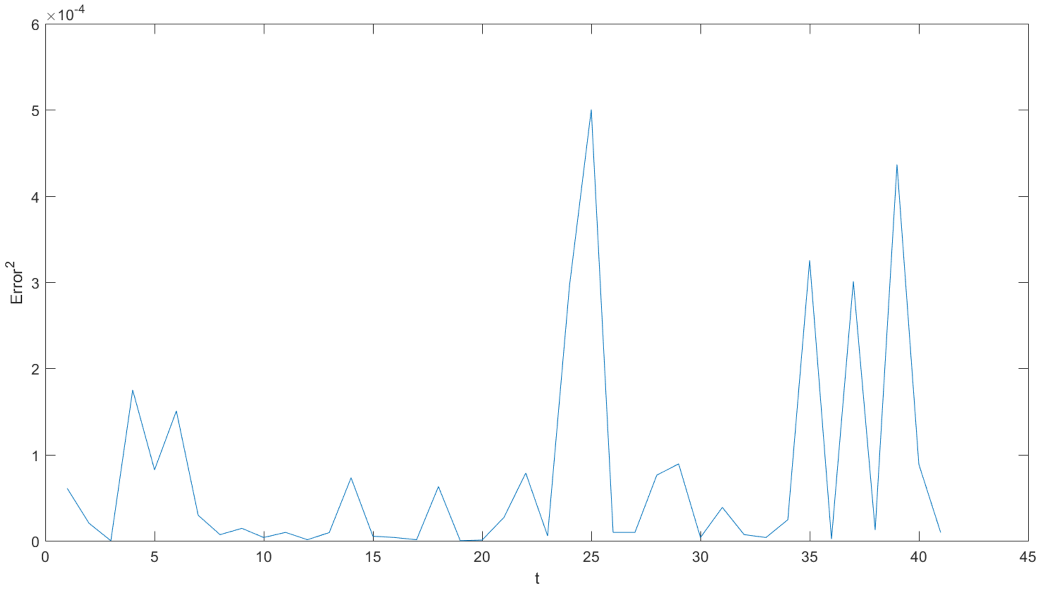
Q) Does adding respiratory etiquette (RespEtiq) to the model improve the performance of the previous models? Please use graphs or other visualization techniques to compare the performance of 1, 2 and 3.

→ From the result section it can be seen that the number of iteration taken by the linear model with two parameter is less as compared to linear model with one parameter and Quadratic expression.Thus, by adding RespEtiq in the linear model performance is enhanced.

7.5.1 Plots:







7.5.2 Results

Linear Regression with Two Variables:

Dataset	Training	Test	Ratio	$Train : J_{min}$	θ_0	θ_1	θ_2	$J_{min}Test$	$Error^2$	Iteration
1	367	41	90:10	0.0139	0.4868	-0.050	0.0042	0.0031	0.000504	3794

7.6 Creating a training and test dataset:

a. Experiment using different ratios of training and test data.

Linear Regression with One Variable:

Dataset	Training	Test	Ratio	$TrainingData : J_{min}$	θ_0	θ_1	$J_{min}Test$	$Error^2$	iteration
1	164	244	40:60	0.0121	0.5174	-0.0549	0.0698	0.0021	3657
2	204	204	50:50	0.0124	0.5174	-0.0610	0.0320	0.0013	2974
3	286	122	70:30	0.0131	0.4997	-0.0582	0.0138	0.0010	3999
4	367	41	90:10	0.0140	0.4891	-0.0493	0.0031	0.0005	4219

Quadratic Regression with One Variable:

Dataset	Training	Test	Ratio	$Train : J_{min}$	θ_0	θ_1	θ_2	$J_{min}Test$	$Error^2$	Iteration
1	164	244	40:60	0.0134	0.5198	0.0303	-0.1163	0.0454	0.0016	12000
2	204	204	50:50	0.0124	0.5040	0.0030	-0.059	0.0316	0.0013	12000
3	286	122	70:30	0.130	0.4873	0.0174	-0.0762	0.0138	0.0009	12000
4	367	41	90:10	0.0139	0.4867	-0.0325	-0.0162	0.0031	0.0005	12000

Linear Regression with Two Variables:

Dataset	Training	Test	Ratio	$Train : J_{min}$	θ_0	θ_1	θ_2	$J_{min}Test$	$Error^2$	Iteration
1	164	244	40:60	0.0133	0.5632	-0.0767	-0.0379	0.0455	0.0017	3116
2	204	204	50:50	0.0124	0.5295	-0.0494	-0.0248	0.0317	0.0013	3695
3	286	122	70:30	0.0130	0.4863	-0.0649	0.0217	0.0137	0.0009	3114
4	367	41	90:10	0.0139	0.4868	-0.05	0.0042	0.0031	0.0005	3794

Above, gives a tabular comparison of different ratio of dataset.

7.7 Observation:

1. Cost function for the test dataset decreases with increase in the training dataset.
2. Mean square error decreases as training is increased.
3. Cost Function generated over the iteration for training dataset is somewhat same for all the model for a given ratio of dataset. Only the number of iteration differs.

8 Goodness of fit:

8.1 R2

R-square is defined as

$$R - squared = 1 - MSE/sum((y - mean(y))^2)$$

Dataset	Training	Test	Ratio	R square	Goodness of fit
1	367	41	90:10	-0.032	Line fits worse than horizontal line
2	367	41	90:10	-0.0375	Quadratic fits worse than horizontal line
2	367	41	90:10	-0.0315	Linear model with two variable fits worse than horizontal line

Since R square is negative, They are not a good fit for the given data. However, Comparatively linear model can be choosen over quadratic model.

b. What was the best proportion from the set of experiments you conducted?

The best proportion from the set of experiment is the 90:10 ratio of Training to test data. Which is obvious, since larger the training data more correct the model gets by updating the values of parameters thus reducing the error caused in the output From the experiment, we also know:

1. The Cost function for the test data decreases as the proportion of training data increases.

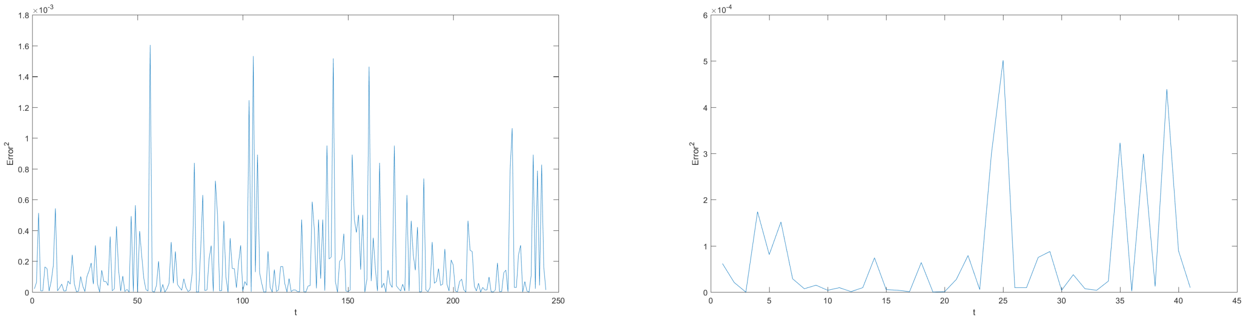
2. The mean Square error(MSE) is also decreasing.

c. How do you evaluate the performance for each of the three sections above?

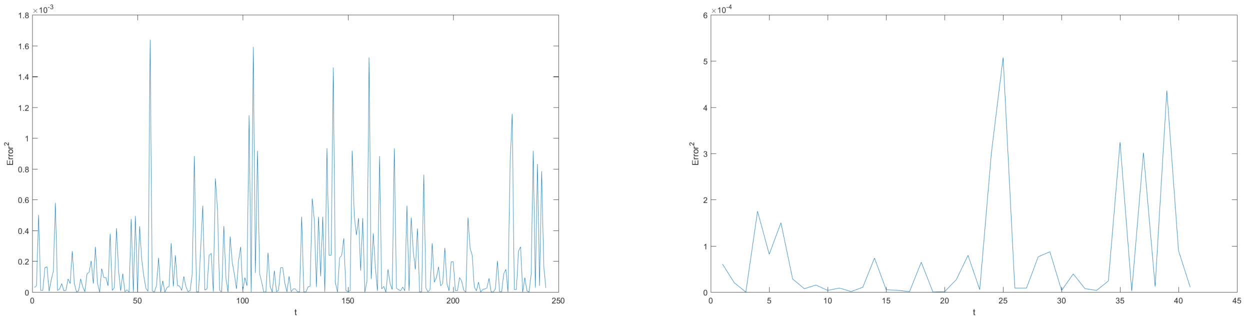
1. Linear Regression with single variable: Linear regression with single variable has better performance than Quadratic Regression in terms of number of iterations.

2. Quadratic Regression with single variable:The plots of θ shows that it takes longer to converge to a value. As compared to either of the linear regression, Quadratic regression is a poorer fit to the dataset in terms of number of iterations.

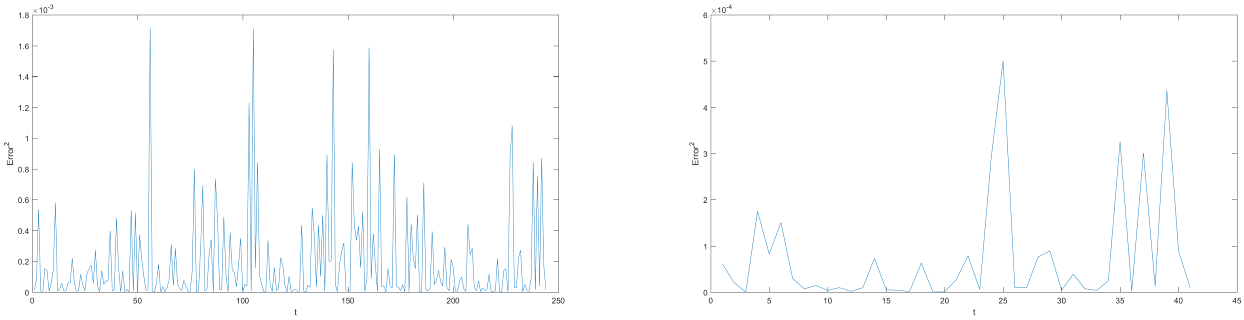
3. Linear Regression with two variable takes least number of iterations to reach to the min J value as compared to other models hence, For the given dataset its a better choice.



(a) Linear regression with single variable: Mean Square error for ratio of 40:60 and 90:10(Less error)



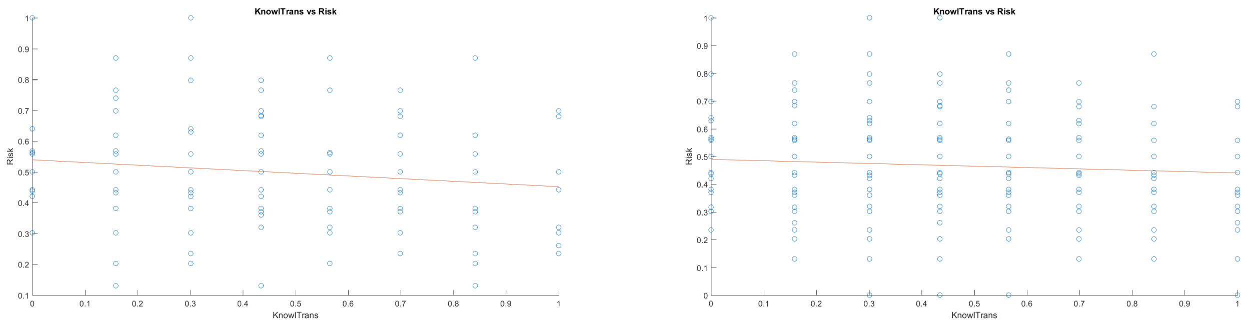
(b) Quadratic regression with single variable: Mean Square error for ratio of 40:60 and 90:10(Less error)



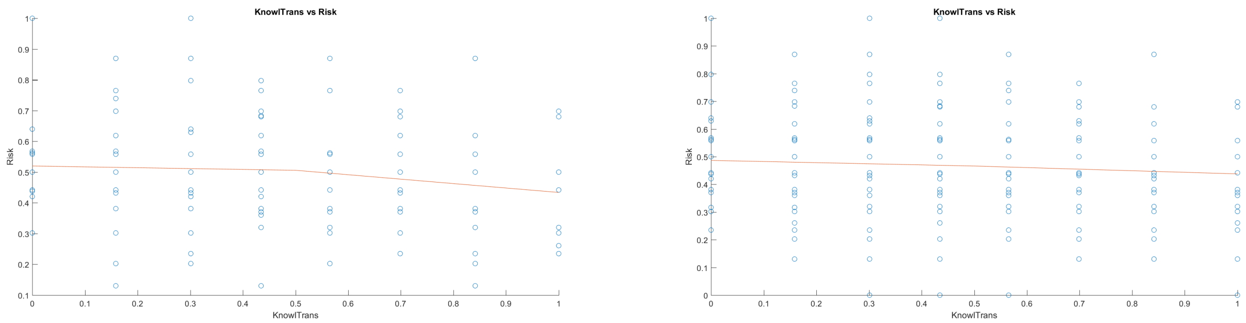
(c) Linear Regression with two variable Mean Square error for ratio of 40:60 and 90:10(Less error)

Performance Observation: Mean square Error see figures

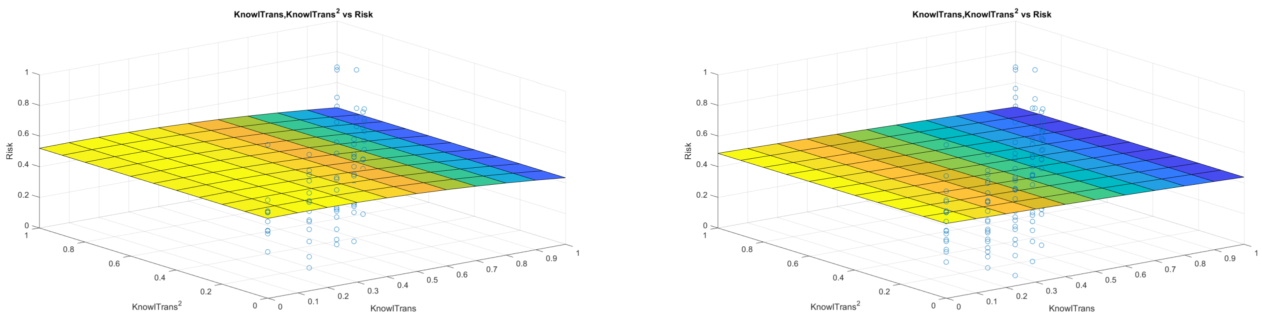
Performance Observation: Curve fitting: see figures



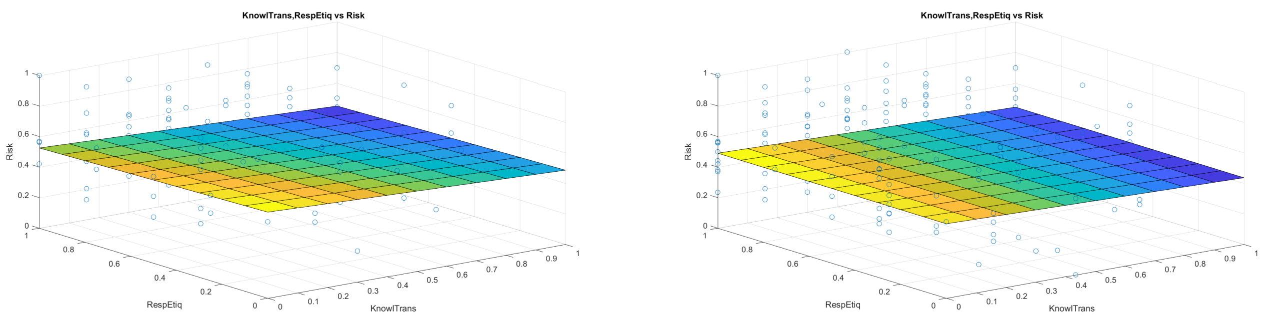
(a) Linear regression with single variable: Line fitting on training data of 40:60 and 90:10 ratio. As seen above the 90:10 is a better hypothesis



(b) Quadratic regression with single variable: parabolic Curve fitting on training data of 40:60 and 90:10 ratio. As seen above the 90:10 is a better hypothesis



(c) Quadratic regression with single variable:Parabolic Curve fitting on training data of 40:60 and 90:10 ratio. As seen above the 90:10 is a better hypothesis



(d) Linear regression with two variable: Plane fitting on training data of 40:60 and 90:10 ratio. As seen above the 90:10 is a better hypothesis

9 Matlab Code:

Listing 1: Main

```
function [] = Main()

for h = 3:1:3
hypothesis = h;
for dt = 4:1:4
    %30 Training Data%
    if(dt == 1)
    prefix = '40';
    SeperationRatio = 164;
    end
    %50 Training Data%
    if(dt == 2)
        prefix = '50';
    SeperationRatio = 204;
    end
    %70 Training Data%
    if(dt == 3)
        prefix = '70';
    SeperationRatio = 286;
    end
    if(dt == 4)
        prefix = 'Result';
     %90 Training Data%
    SeperationRatio = 367;
    end
%Initialise
[Theta,LearningRate,Epsilon,Iteration]=Initialise(hypothesis);
%FetchDataSet
fluDataSet = DatasetInitialise('fludata');
%Create DataSet for Training and Test
[TrainingData,TestData] = DataSetBifurcate(fluDataSet,SeperationRatio);
for i = 1:1:Iteration
        [row,column] = size(TrainingData);
        N = row;
        [dj1,dj2,dj3] = PartialDerivative(Theta,TrainingData,hypothesis);
        Theta(1) = Theta(1) - LearningRate*(1/N)*sum(dj1);
```

```
        Theta(2) = Theta(2) - LearningRate*(1/N)*sum(dj2);
        Theta(3) = Theta(3) - LearningRate*(1/N)*sum(dj3);
        theta1(i) = Theta(1);
        theta2(i) = Theta(2);
        theta3(i) = Theta(3);
        Cost(i) = (1/(2*N))* sum(MeanSquareError(Theta,hypothesis,TrainingData));
end
%plots
PlotFunction(theta1,theta2,theta3,Cost,hypothesis,prefix);
ind =find(Cost == min(Cost(:)))
t1 = theta1(ind(1));
t2 = theta2(ind(1));
t3 = theta3(ind(1));
PlotTrain(t1,t2,t3,TrainingData,hypothesis,prefix);
%Testing
T = [t1;t2;t3];
if(hypothesis == 1)

    filenameprefix = '/Plots1/hypothesis1';
    filenameprefix = strcat(filenameprefix,prefix);
end
if(hypothesis == 2)
    filenameprefix = '/Plots2/hypothesis2';
    filenameprefix = strcat(filenameprefix,prefix);
end
if(hypothesis == 3)
    filenameprefix = '/Plots3/hypothesis3';
    filenameprefix = strcat(filenameprefix,prefix);
end
[row,col] = size(TestData);
MSE = (1/N)*(MeanSquareError(T,hypothesis,TestData));
r2 = (R2(T,hypothesis,TestData));

f3 = figure
plot(MSE);
xlabel('t');
ylabel('Error^2')
f3.PaperUnits = 'inches';
f3.PaperPosition = [0 0 12 6];
```

```
f3name = strcat(filenameprefix,'error.png');
saveas(f3,[pwd f3name]);

CostTest = sum(MSE);
CostTrain(dt) = Cost(ind(1));
thetaTrain1(dt) = t1;
thetaTrain2(dt) = t2;
thetaTrain3(dt) = t3;
TestDataCost(dt) =CostTest;
ErrorTestdata(dt) = max(MSE);
ind1(dt) = ind(1);

display(CostTrain)
display(ind1)
display(thetaTrain1)
display(thetaTrain2)
display(thetaTrain3)

display(TestDataCost)
display(ErrorTestdata)
close all


end

end
```

Listing 2: DatasetInitialise

```
function [FluDataset] = DatasetInitialise( AlgorithmTestingOn )

if(AlgorithmTestingOn == 'fludata')
%This File fetches dataset from the xlsx file
%And converts it into a table.
SheetNumber = 1;
[CH,CV,ActualData] = xlsread('fluML.xlsx',SheetNumber);
rowHeadings = {'Student','Vaccin','HndWshQual','HndWshFreq','SocialDist','
    NotTchFace','RespEtiq','PrsnlDist','HandSanit','Risk','Complic','Barriers','
    Inefficacy','KnowlTrans','KnowlMgmt','Sick','Flu','Gender'};
Grad = cell2struct(ActualData, rowHeadings, 2);
```

```
Grad(1) = [];

FluDataset = struct2table(Grad);
FluDataset(any(isnan(FluDataset.Risk),2) == 1,:) = [];
FluDataset(any(isnan(FluDataset.KnowlTrans),2) == 1,:) = [];
FluDataset(any(isnan(FluDataset.RespEtiq),2) == 1,:) = [];
index0 = (find(FluDataset.RespEtiq== max(FluDataset.RespEtiq(:))));
FluDataset(index0,:) = [];
% index1 = find(FluDataset.Risk== max(FluDataset.Risk(:)))
% FluDataset(index1,:) = [];
clearvars -except FluDataset;
%Scaling::
% % FluDataset.Risk  = (((3*(FluDataset.Risk - min(FluDataset.Risk))/(max(
    FluDataset.Risk)-min(FluDataset.Risk)))- 1.5);
% FluDataset.KnowlTrans = (((2.9*(FluDataset.KnowlTrans - min(FluDataset.
    KnowlTrans))/(max(FluDataset.KnowlTrans)-min(FluDataset.KnowlTrans)))- 1.45)
    ;
% FluDataset.RespEtiq  = (((2.9*(FluDataset.RespEtiq - min(FluDataset.RespEtiq))
    /(max(FluDataset.RespEtiq)-min(FluDataset.RespEtiq)))- 1.45);
FluDataset.Risk  = ((((FluDataset.Risk - min(FluDataset.Risk))/(max(FluDataset.
    Risk)-min(FluDataset.Risk)))));
FluDataset.KnowlTrans = ((((FluDataset.KnowlTrans - min(FluDataset.KnowlTrans))/(
    max(FluDataset.KnowlTrans)-min(FluDataset.KnowlTrans)))));
FluDataset.RespEtiq  = ((((FluDataset.RespEtiq - min(FluDataset.RespEtiq))/(max(
    FluDataset.RespEtiq)-min(FluDataset.RespEtiq)))));
else

end
end
```

Listing 3: DataSetBifurcate

```
function [DataBifuractedTrain,DataBifuractedTest] = DataSetBifurcate(dataset,
    Number)
TraingfluData = dataset(1:(Number),:);
TestfluData = dataset(Number+1:end,:);
DataBifuractedTrain = TraingfluData;
DataBifuractedTest = TestfluData;
end
```

Listing 4: MeanSquareError

```
function [Cost] = MeanSquareError(Theta,hypothesis,TrainingData)
[row,column] = size(TrainingData);
for i = 1:1:row
    x = TrainingData.KnowlTrans(i);
    y = TrainingData.Risk(i);
    z = TrainingData.RespEtiq(i);
if(hypothesis == 1)
    Cost(i) = (Theta(1) + Theta(2)*x - y)^2;
end
if(hypothesis == 2)
    Cost(i) = (Theta(1) + Theta(2)*x + Theta(3)*x^2 - y)^2;
end
if(hypothesis == 3)
    Cost(i) = (Theta(1) + Theta(2)*x + Theta(3)*z - y)^2;
end
end
end
```

Listing 5: Initialise

```
function [Theta,alpha,epsilon,Iteration] = Initialise(hypothesis)
Theta = zeros(3,1);
epsilon = 0.00001;
if(hypothesis == 1)
%LearningRate: alpha
alpha = 0.1;
%Iteration = 8000;
Iteration = 1000;
end
if(hypothesis == 2)
alpha = 0.1;
%Iteration = 1000;
Iteration = 3700;
end
if(hypothesis == 3)
alpha = 0.1;
Iteration = 5000;
%Iteration = 1000;
end
```

```
end
```

Listing 6: PartialDerivative

```
function [dj1,dj2,dj3] = PartialDerivative(Theta,TrainingData,hypothesis)
[row,column] = size(TrainingData);
if(hypothesis == 1)
for i = 1:1:row
        xi = TrainingData.KnowlTrans(i);
        yi = TrainingData.Risk(i);
        dj1(i) = (Theta(1) + Theta(2)*xi - yi);
        dj2(i) = (Theta(1) + Theta(2)*xi  - yi)* xi;
        dj3(i) =0;
end
end
if(hypothesis == 2)
    for i = 1:1:row
        xi = TrainingData.KnowlTrans(i);
        yi = TrainingData.Risk(i);
        dj1(i) = (Theta(1) + Theta(2)*xi + Theta(3)*xi^2- yi);
        dj2(i) = (Theta(1) + Theta(2)*xi + Theta(3)*xi^2- yi)* xi;
        dj3(i) = (Theta(1) + Theta(2)*xi + Theta(3)*xi^2- yi)* xi^2;

    end
end

if(hypothesis == 3)
    for i = 1:1:row
        xi = TrainingData.KnowlTrans(i);
        yi = TrainingData.Risk(i);
        zi = TrainingData.RespEtiq(i);
        dj1(i) = (Theta(1) + Theta(2)*xi + Theta(3)*zi - yi);
        dj2(i) = (Theta(1) + Theta(2)*xi  + Theta(3)*zi - yi)* xi;
        dj3(i) = (Theta(1) + Theta(2)*xi  + Theta(3)*zi - yi)* zi;
    end
end


end
```

Listing 7: PlotFunction

```
function [] = PlotFunction(Theta1,Theta2,Theta3,Cost,hypothesis,prefix)
if(hypothesis == 1)
    filenameprefix = '/Plots1/hypothesis1';
    filenameprefix = strcat(filenameprefix,prefix);
end
if(hypothesis == 2)
    filenameprefix = '/Plots2/hypothesis2';
    filenameprefix = strcat(filenameprefix,prefix);
end
if(hypothesis == 3)
    filenameprefix = '/Plots3/hypothesis3';
    filenameprefix = strcat(filenameprefix,prefix);
end

f1 = figure;
plot(Theta1)
title({'\theta_0 vs iteration'});
xlabel('t');
ylabel('\theta_0')
f1.PaperUnits = 'inches';
f1.PaperPosition = [0 0 12 6];
%f1name = strcat('theta1.png');
f1name = strcat(filenameprefix,'theta1.png');
saveas(f1,[pwd f1name]);

f2 = figure;
plot(Theta2)
title({'\theta_1 vs iteration'});
xlabel('t');
ylabel('\theta_1')
f2.PaperUnits = 'inches';
f2.PaperPosition = [0 0 12 6];
f1name = strcat(filenameprefix,'theta2.png');
saveas(f2,[pwd f1name]);

f3 = figure;
plot(Theta3)
title({'\theta_2 vs iteration'});
```

```
xlabel('t');
ylabel('\theta_2')
f3.PaperUnits = 'inches';
f3.PaperPosition = [0 0 12 6];
f1name = strcat(filenameprefix,'theta3.png');
saveas(f3,[pwd f1name]);

f4 = figure;
plot(Cost)
title({'Cost vs iteration'});
xlabel('t');
ylabel('J_{min}')
f4.PaperUnits = 'inches';
f4.PaperPosition = [0 0 12 6];
f1name = strcat(filenameprefix,'Cost.png');
saveas(f4,[pwd f1name]);

end
```

Listing 8: PlotTrain

```
function [] = PlotTrain( Theta1,Theta2,Theta3,TrainingData,hypothesis,prefix)
if(hypothesis == 1)
    filenameprefix = '/Plots1/hypothesis1';
    filenameprefix = strcat(filenameprefix,prefix);
    f1 = figure
    scatter(TrainingData.KnowlTrans,TrainingData.Risk);
    hold on;
    [row,column] = size(TrainingData);
    x = 0:0.5:1;
    y = 0:0.5:1;
    plot(x,Theta1 + Theta2*x)
    title({'KnowlTrans vs Risk'});
    xlabel('KnowlTrans');
    ylabel('Risk')
    f1.PaperUnits = 'inches';
    f1.PaperPosition = [0 0 12 6];
    f1name = strcat(filenameprefix,'fitting.png');
    saveas(f1,[pwd f1name]);

```

```
%     figure
%     [X,Y] = meshgrid(-1.45:0.1:1.45,-1.45:0.1:1.45);
%     Z = Theta1 + Theta2*X - Y;
%     surf(X,Y,Z)
end

if(hypothesis == 2)
    filenameprefix = '/Plots2/hypothesis2';
    filenameprefix = strcat(filenameprefix,prefix);
    f1 = figure;
    scatter(TrainingData.KnowlTrans,TrainingData.Risk);
    hold on;
    [row,column] = size(TrainingData);
    x  = 0:0.5:1;
    y = 0:0.5:1;
    plot(x,Theta1 + Theta2*x + Theta3*x.^2)
    title({'KnowlTrans vs Risk'});
    xlabel('KnowlTrans');
    ylabel('Risk')
    f1.PaperUnits = 'inches';
    f1.PaperPosition = [0 0 12 6];
    f1name = strcat('fitting.png');
    f1name = strcat(filenameprefix,'fitting2D.png');
    saveas(f1,[pwd f1name]);

    f3 = figure;
    scatter3(TrainingData.KnowlTrans,(TrainingData.KnowlTrans).^2,TrainingData.Risk);
    hold on;
    [X,Y] = meshgrid(0:0.1:1,0:0.1:1);
    Z = Theta1 + Theta2*X + Theta3*X.^2;
    surf(X,Y,Z)
    title({' KnowlTrans,KnowlTrans^2 vs Risk'});
    xlabel('KnowlTrans');
    ylabel('KnowlTrans^2')
    zlabel('Risk')
    f3.PaperUnits = 'inches';
    f3.PaperPosition = [0 0 12 6];
    f1name = strcat(filenameprefix,'fitting3D.png');
```

```
    saveas(f3,[pwd f1name]);
end

if(hypothesis == 3)
    filenameprefix = '/Plots3/hypothesis3';
    filenameprefix = strcat(filenameprefix,prefix);
    [row,column] = size(TrainingData);
    [X,Y] = meshgrid(0:0.1:1,0:0.1:1);
    f3 = figure;
    scatter3(TrainingData.KnowlTrans,TrainingData.RespEtiq,TrainingData.Risk);
    hold on;
    Z = Theta3*Y + Theta1 + Theta2*X;
    %Z = (-Theta1-Theta2*X+Y)/Theta3;
    surf(X,Y,Z)
    title({' KnowlTrans,RespEtiq vs Risk'});
    xlabel('KnowlTrans');
    ylabel('RespEtiq')
    zlabel('Risk')
    f3.PaperUnits = 'inches';
    f3.PaperPosition = [0 0 12 6];
    f1name = strcat(filenameprefix,'fitting.png');
    saveas(f3,[pwd f1name]);
end

end
```

Listing 9: R2

```
function [ r2 ] = R2(Theta,hypothesis,TestData )
[row,column] = size(TestData);
for i = 1:1:row
    x = TestData.KnowlTrans(i);
    y = TestData.Risk(i);
    z = TestData.RespEtiq(i);
if(hypothesis == 1)
    r(i) = (Theta(1) + Theta(2)*x - y)^2;
end
if(hypothesis == 2)
    r(i) = (Theta(1) + Theta(2)*x + Theta(3)*x^2 - y)^2;
end
```

```
if(hypothesis == 3)
    r(i) = (Theta(1) + Theta(2)*x + Theta(3)*z - y)^2;
end
end
a = sum(r);
b= sum((-TestData.Risk + mean(TestData.Risk)).^2);
r2 = 1- (a/b)
end
```