INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

Department of Electronics & Electrical Engineering

EE657: Pattern Recognition and Machine Learning

Assignment No. 01

Name: Shubham

Roll Number: 150102079

Branch: Electronics and Communication Engineering

Title: Bayesian Decision Theory and Polynomial Regression

Date: 08-04-2018

Assignment - 1.P1

Objective: Learn the parameters μ5, μ6, Σ5, Σ6 and π by maximizing the likelihood for the given dataset,

$$\begin{aligned} \pi &= Pr(r = C_5) & Pr(x|C_5) &= \mathcal{N}(x|\mu_5,\ \textstyle\sum_5) \\ 1 - \pi &= Pr(r = C_6) & Pr(x|C_6) &= \mathcal{N}(x|\mu_6,\ \textstyle\sum_6) \end{aligned}$$

Use Bayes decision criterion to classify the test data. Estimate the misclassification rates of both classes and populate the 2x2 confusion matrix.

Dataset Description: *Optical Recognition of Handwritten Digits Dataset* from the *UCI repository*. The original dataset is transformed into data-points of dimension = 64.

Summary: Given training and testing sets consist of 777 and 333 data-points respectively. In this experiment, have evaluated class apriori probabilities, estimate of means and covariance matrices of Class 5 & 6 using below given formulas in textbook: (Here i = 5 & 6)

$$\begin{aligned} \hat{P}(C_i) &= \frac{\sum_t r_i^t}{N} \\ \boldsymbol{m}_i &= \frac{\sum_t r_i^t \boldsymbol{x}^t}{\sum_t r_i^t} \\ \mathbf{S}_i &= \frac{\sum_t r_i^t (\boldsymbol{x}^t - \boldsymbol{m}_i)(\boldsymbol{x}^t - \boldsymbol{m}_i)^T}{\sum_t r_i^t} \end{aligned}$$

After estimating these parameters, done classification based on Bayes decision criterion using the derived discriminant function in the textbook,

$$g_i(\boldsymbol{x}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i) + \log P(C_i)$$

for different variations of covariance matrices:

A. <u>For estimated covariance matrices Σ5 and Σ6</u>

The above given discriminant function $g_i(x)$ is used.

B. <u>For equal covariance matrices Σ5 = Σ6 = Σ</u>

Used Σ = (π5*Σ5)+(π6*Σ6)

Discriminant function gets reduced as follows:

$$g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^T \mathbf{S}^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) + \log \hat{P}(C_i)$$

C. <u>For equal covariance matrices Σ5 = Σ6 = Σ</u>

Σ = a*I (where a is an random integer between 1 and 10, I is Identity matrix)

The modified discriminant function $g_i(x)$ given above is used.

Observations:

```
D:\6TH SEMESTER\Course\EE 657 Pattern Recognition and Machine Learning\Assignmen
t-1-8th April\P1_data>python P1.py

Loading Dataset...
Number of training samples:  (777,)
Number of testing samples:  (333,)

Estimating Π5 and Π6...
Π5 =  0.5096525096525096
Π6 =  0.4903474903474903 7

Estimating Mean_5 and Mean_6...

Estimating Σ5 and Σ6...

Number of Class 5 datapoints in training set =  396
Number of Class 6 datapoints in training set =  381
Number of Class 5 datapoints in testing set =  155
Number of Class 6 datapoints in testing set =  178

Classification of test data using Bayes decision criterion...

A.
For estimated Σ5 and Σ6
Accuracy =  0.7717717717717718
Class 5 Accuracy =  0.6838709677419355
Class 6 Accuracy =  0.848314606741573

Confusion Matrix:
[['True Class 5', 'False Class 6'], ['False Class 5', 'True Class 6']]
[[106  49]
 [ 27 151]]

B.
For Σ = (Π5*Σ5)+(Π6*Σ6)
Accuracy =  0.8558558558558559
Class 1 Accuracy =  0.864516129032258
Class 2 Accuracy =  0.848314606741573

Confusion Matrix:
[['True Class 5', 'False Class 6'], ['False Class 5', 'True Class 6']]
[[134  21]
 [ 27 151]]

C.
For Σ = a*I (I = Identity Matrix)
Accuracy =  0.8138138138138138
Class 1 Accuracy =  0.8580645161290322
Class 2 Accuracy =  0.7752808988764045

Confusion Matrix:
[['True Class 5', 'False Class 6'], ['False Class 5', 'True Class 6']]
[[133  22]
 [ 40 138]]
```

Note: Computed estimates of means and covariance matrices of Class 5 & 6 can be checked for evaluation by un-commenting lines in the script P1.py under "Summarizing" section.

#print(M1,M2)

#print(C1,C2)

Conclusions:

1. Evaluated parameters are as follows:

$\pi = \pi 5 = 0.5096525096525096$ $\quad$ $1-\pi = \pi 6 = 0.49034749034749037$

2. $\mu 5$, $\mu 6$, $\Sigma 5$, $\Sigma 6$ have dimensions (64,1), (64,1), (64,64), (64,64) and hence can be checked using python script P1.py.

3. Testing data was classified using Bayes decision criterion with different variations of covariance matrices. Testing data was **best classified** for the case when equal covariance matrices($\Sigma = (\pi 5*\Sigma 5)+(\pi 6*\Sigma 6)$) was used for Class 5 and 6.

4. Reported accuracies and confusion matrices for case A, B and C are as follows:

A. <u>For estimated covariance matrices $\Sigma 5$ and $\Sigma 6$</u>

Test Accuracy: 0.7717717717717718
Class 1 Accuracy: 0.6838709677419355
Class 2 Accuracy: 0.848314606741573
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	106	49
Predicted Class 6	27	151

B. <u>For equal covariance matrices $\Sigma 5 = \Sigma 6 = \Sigma = (\pi 5*\Sigma 5)+(\pi 6*\Sigma 6)$</u>

Test Accuracy: 0.8558558558558559
Class 1 Accuracy: 0.864516129032258
Class 2 Accuracy: 0.848314606741573
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	134	21
Predicted Class 6	27	151

C. <u>For equal covariance matrices $\Sigma 5 = \Sigma 6 = \Sigma = a*I$</u>

Test Accuracy: 0.8138138138138138
Class 1 Accuracy: 0.8580645161290322
Class 2 Accuracy: 0.7752808988764045
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	133	22
Predicted Class 6	40	138

Assignment - 1.P2

Objective: Learn a binary classier for the given dataset taking class conditional densities as normal density. Estimate the misclassification rates of both classes, plot the discriminant function and iso-probability contours for different variations of covariance matrices.

Summary: Given training and testing sets consist of 310 and 90 data-points(dimension = 2) respectively. In this experiment, have evaluated class apriori probabilities, estimate of means and covariance matrices of Class 0 & 1 using same code as in P1.

After estimating these parameters, done classification based on Bayes decision criterion using the derived discriminant function in the textbook, for different variations of covariance matrices as mentioned in the objective.

$$\boxed{g_i(\boldsymbol{x}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i) + \log P(C_i)}$$

Firstly, classified the test dataset using estimated Σ0 and Σ1. Then tried various variations of covariance matrices as following cases:

(a) Equal diagonal Σ_s of equal variances along both dimensions, Σ0 = Σ1 = $\begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$

(b) Equal diagonal Σ_s with unequal variances along different dimensions, Σ0 = Σ1 = $\begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$

(c) Arbitrary Σ_s but shared by both classes, Σ0 = Σ1 = $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

(d) Different arbitrary Σ_s for the two classes.

In all the above cases, have plot discriminant function and iso-probability contours taking class conditional densities as normal density.

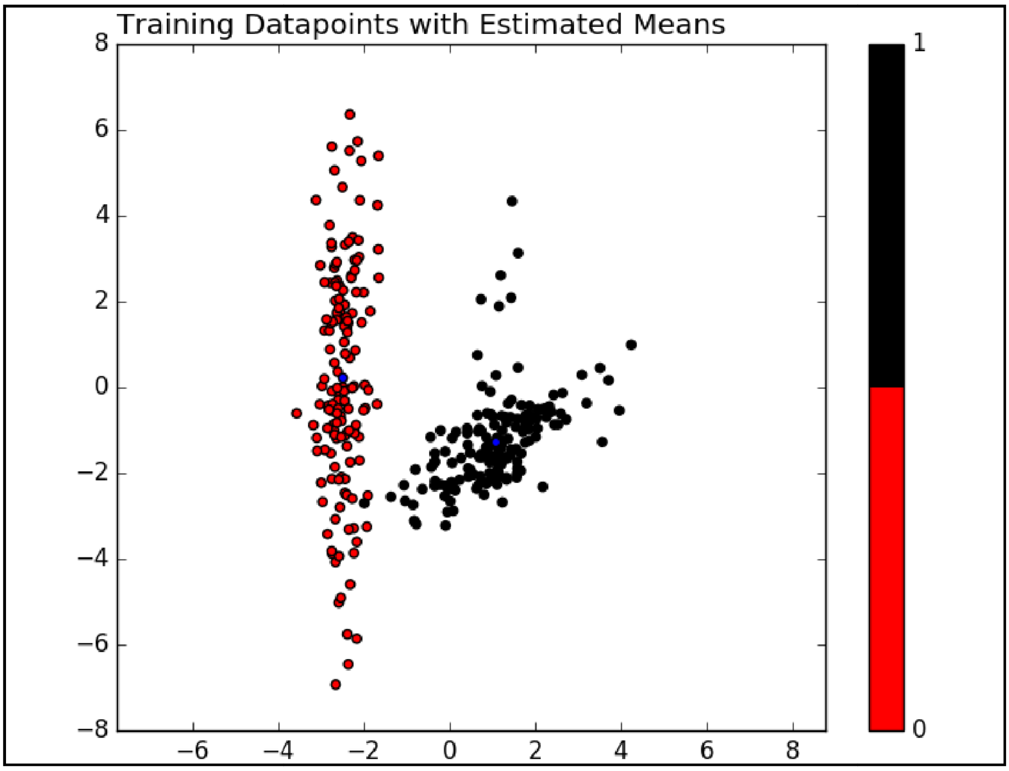
In cases of equal Σ_s, the discirminant function become as follows:

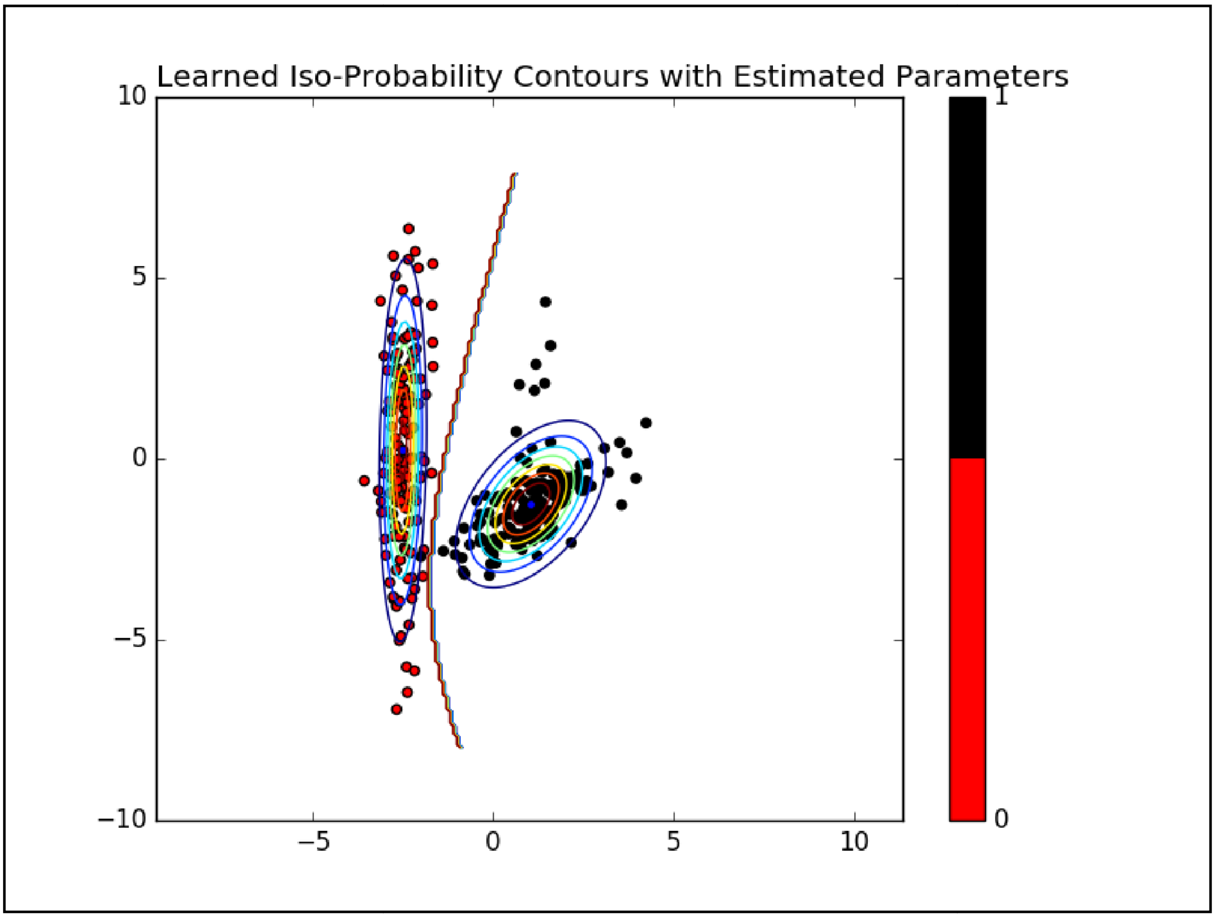
$$\boxed{g_i(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{m}_i)^T\mathbf{S}^{-1}(\boldsymbol{x} - \boldsymbol{m}_i) + \log \hat{P}(C_i)}$$

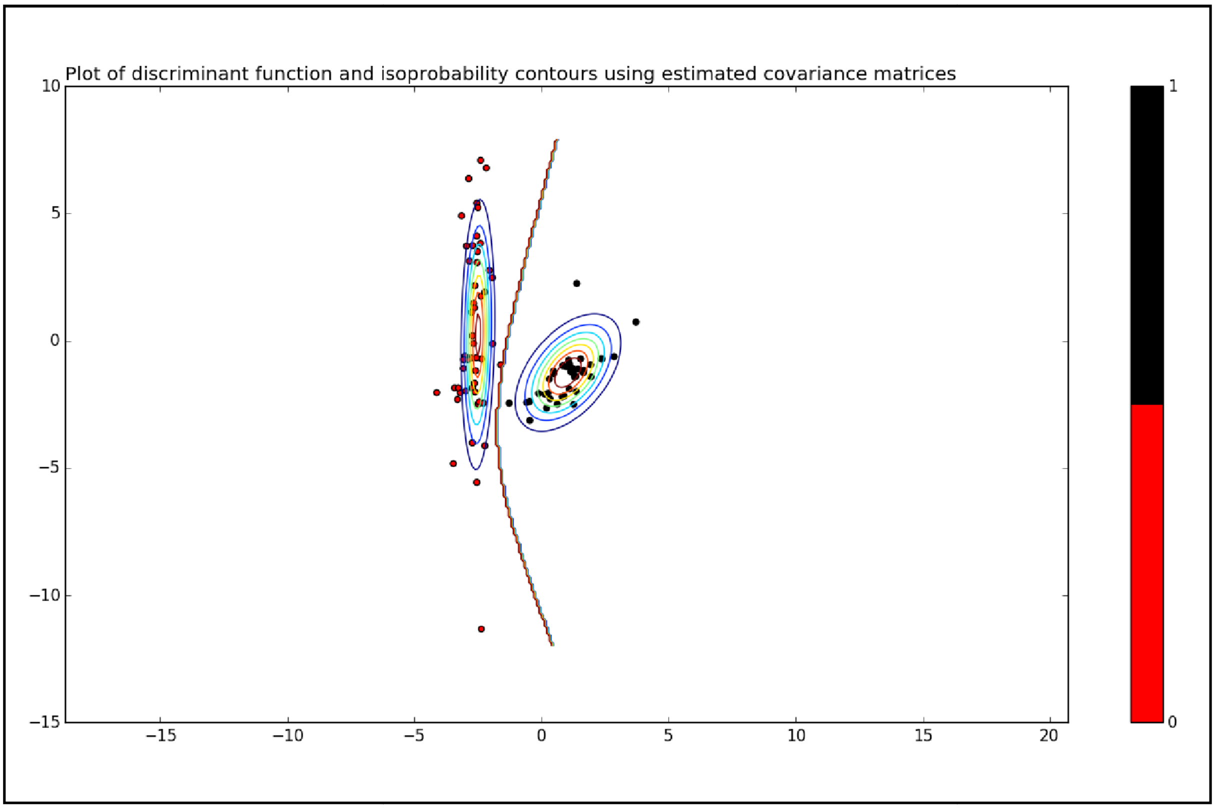
Observations:

Note: While running the script for evaluation, keep closing the generated plot(figure_1) for further execution of the script.

```
D:\6TH SEMESTER\Course\EE 657 Pattern Recognition and Machine Learning\Assignmen
t-1-8th April\P2_data>python P2.py

Loading Dataset...
Number of training samples:  (310,)
Number of testing samples:  (90,)

Estimating p0 and p1...
p0 = p = 0.4838709677419355
p1 = 1-p = 0.5161290322580645

Estimating Mean_0 and Mean_1...
Estimate of Mean_0 =
[-2.48646619  0.23701453]
Estimate of Mean_1 =
[ 1.056773  -1.2525509]

Estimating Σ0 and Σ1...
Estimate of Σ0 =
[[ 0.10758923  0.10686359]
 [ 0.10686359  7.04930364]]
Estimate of Σ1 =
[[ 1.03771527  0.57882397]
 [ 0.57882397  1.28070394]]

Training Accuracy:  0.9967741935483871
Class 0 Accuracy:  1.0
Class 1 Accuracy:  0.99375

Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[150   0]
 [  1 159]]

Classification using estimated Σ0 and Σ1...
Test Accuracy:  1.0
Class 0 Accuracy:  1.0
Class 1 Accuracy:  1.0

Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[50  0]
 [ 0 40]]
```







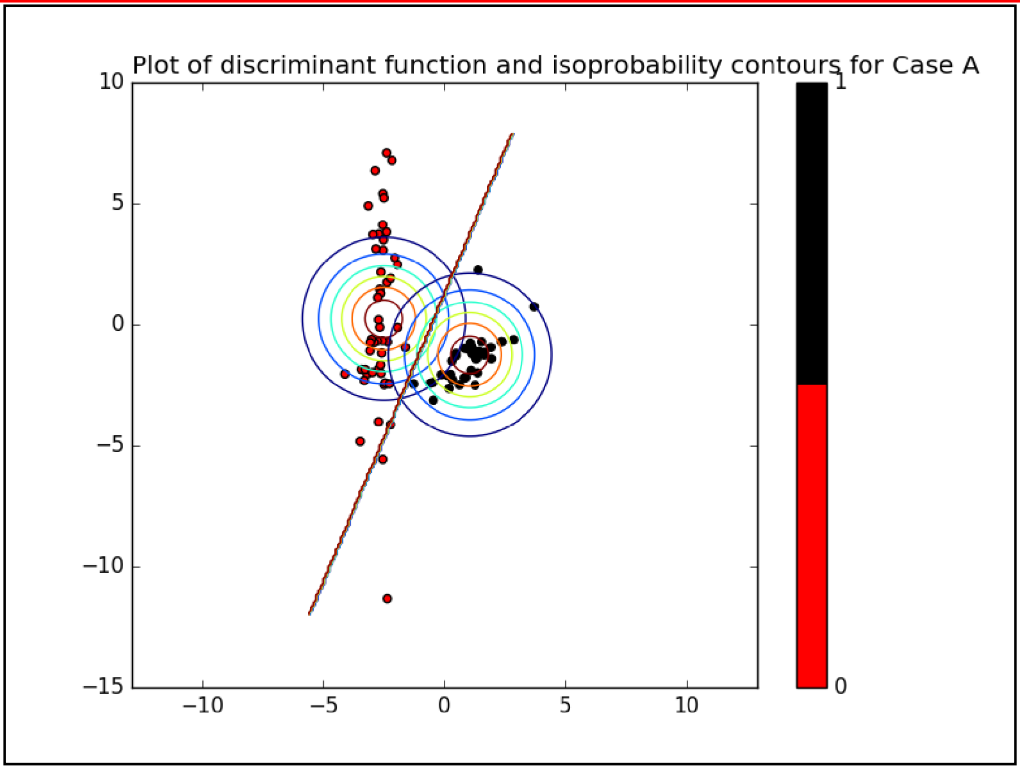
The above plot shows classification on test data using estimated Σ0 and Σ1

Different variations of covariance matrices as described in summary(3 observations each):

<u>Case A:</u>

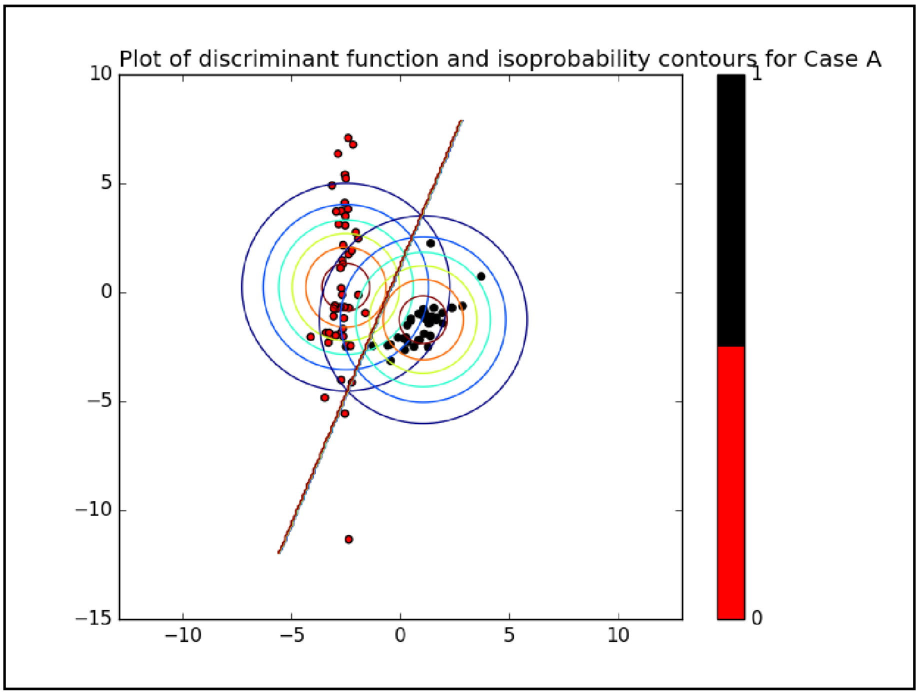
a = 3

```
A. Classification using Σ0 = Σ1 = Σ of the form
a 0
0 a
Σ =
[[3 0]
 [0 3]]

Test Accuracy:  0.9666666666666667
Class 0 Accuracy:  0.94
Class 1 Accuracy:  1.0

Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[47  3]
 [ 0 40]]
```



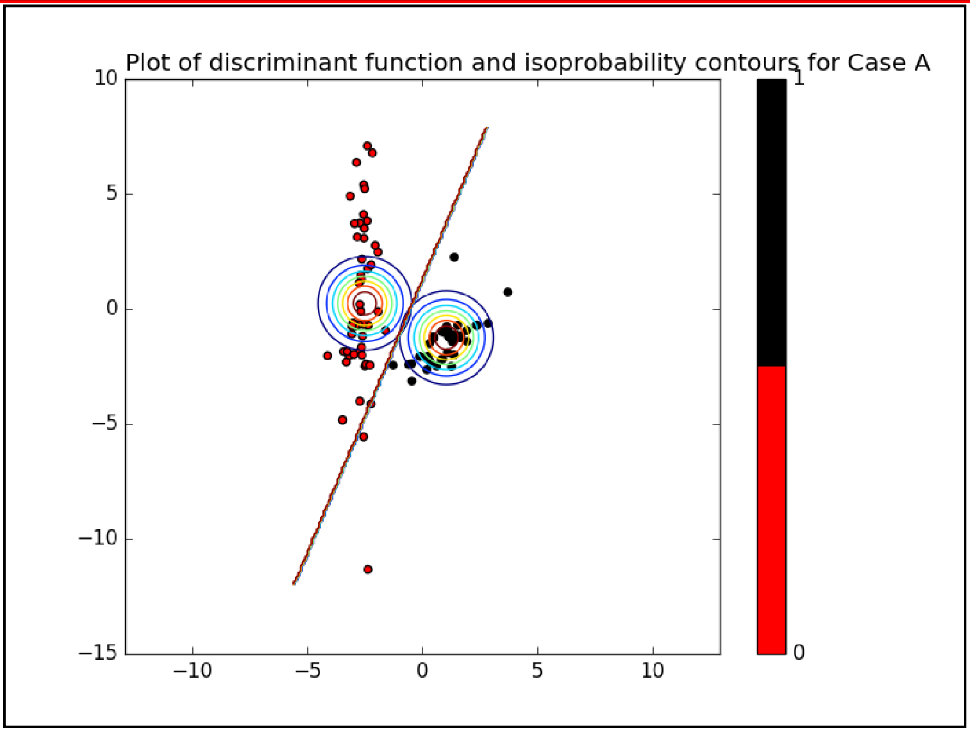
a = 6

```
A. Classification using Σ0 = Σ1 = Σ of the form
a 0
0 a
Σ =
[[6 0]
 [0 6]]

Test Accuracy:  0.9666666666666667
Class 0 Accuracy:  0.94
Class 1 Accuracy:  1.0

Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[47  3]
 [ 0 40]]
```



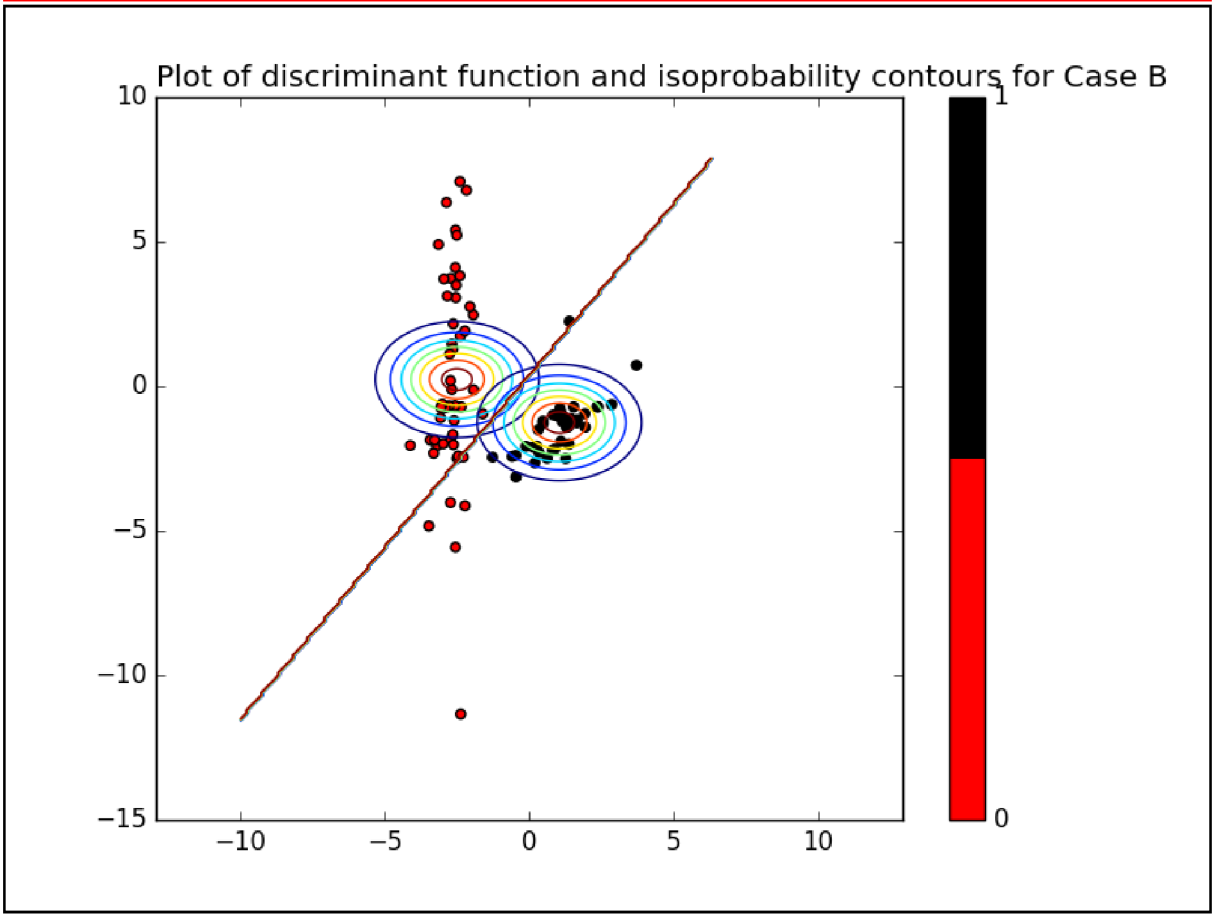
a = 1

```
A. Classification using Σ0 = Σ1 = Σ of the form
a 0
0 a
Σ =
[[1 0]
 [0 1]]

Test Accuracy:  0.9666666666666667
Class 0 Accuracy:  0.94
Class 1 Accuracy:  1.0

Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[47  3]
 [ 0 40]]
```

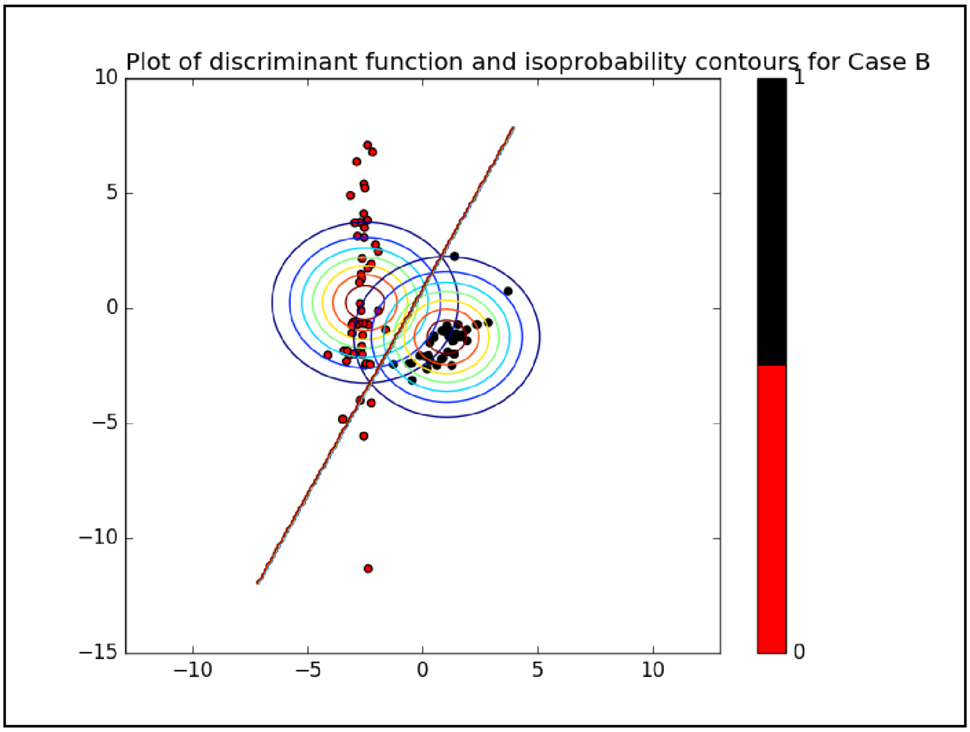


Case B:

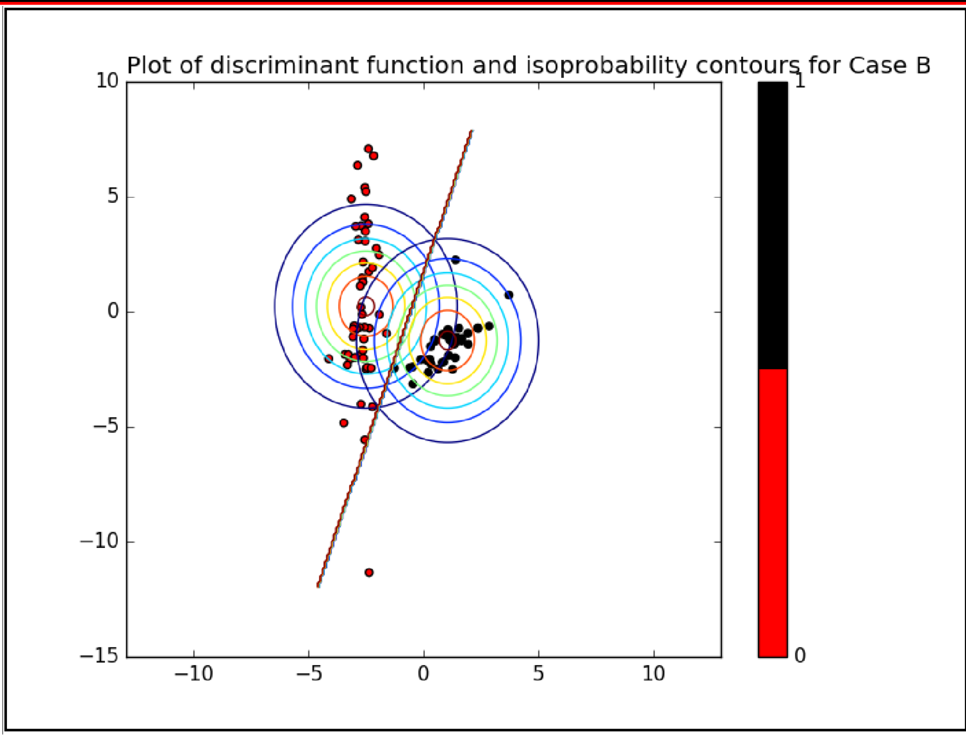
```
B. Classification using Σ0 = Σ1 = Σ of the form
a 0
0 b
Σ =
[[2 0]
 [0 1]]

Test Accuracy:  0.9222222222222223
Class 0 Accuracy:  0.88
Class 1 Accuracy:  0.975
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[44  6]
 [ 1 39]]
```




```
B. Classification using Σ0 = Σ1 = Σ of the form
a 0
0 b
Σ =
[[4 0]
 [0 3]]

Test Accuracy:  0.9555555555555556
Class 0 Accuracy:  0.92
Class 1 Accuracy:  1.0
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[46  4]
 [ 0 40]]
```

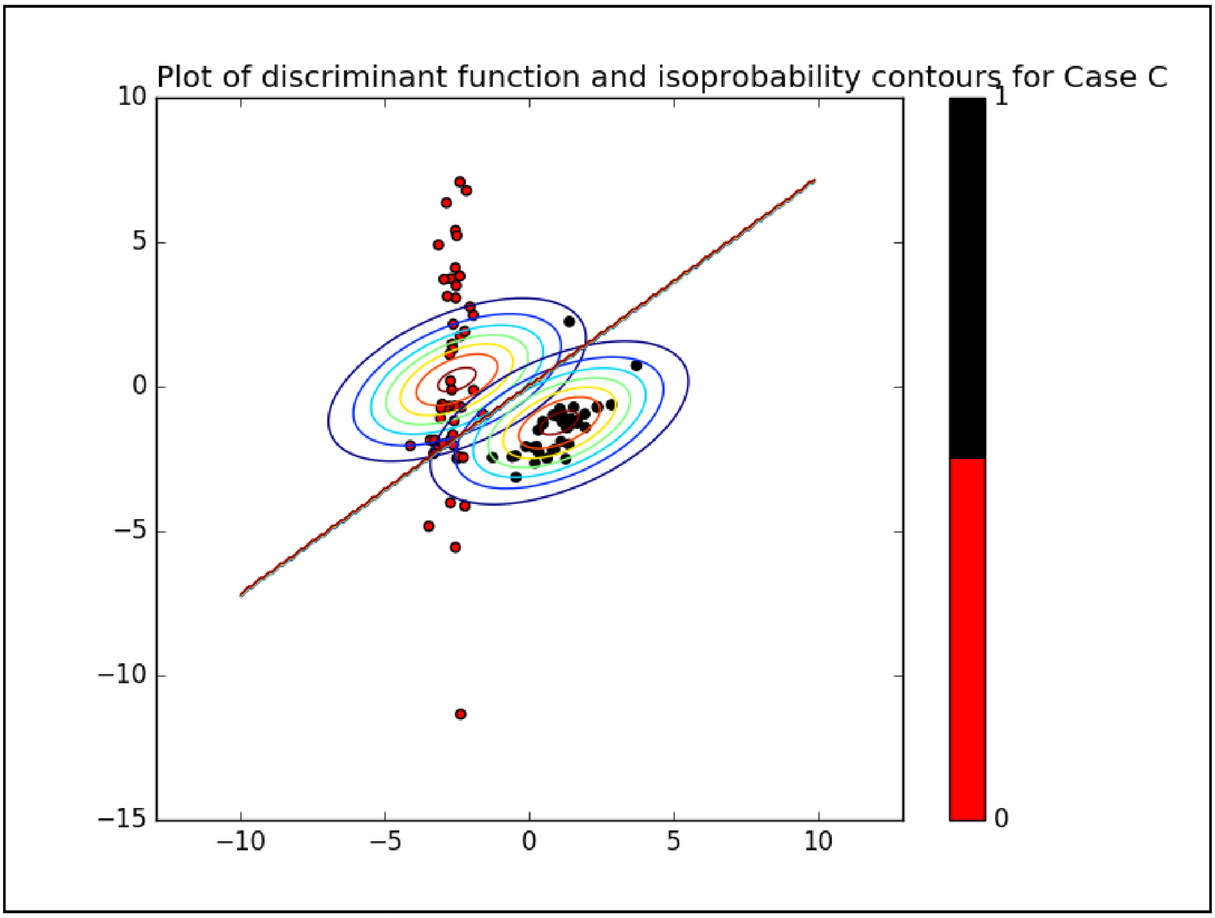



```
B. Classification using Σ0 = Σ1 = Σ of the form
a 0
0 b
Σ =
[[4 0]
 [0 5]]

Test Accuracy:  0.9888888888888889
Class 0 Accuracy:  0.98
Class 1 Accuracy:  1.0
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[49  1]
 [ 0 40]]
```

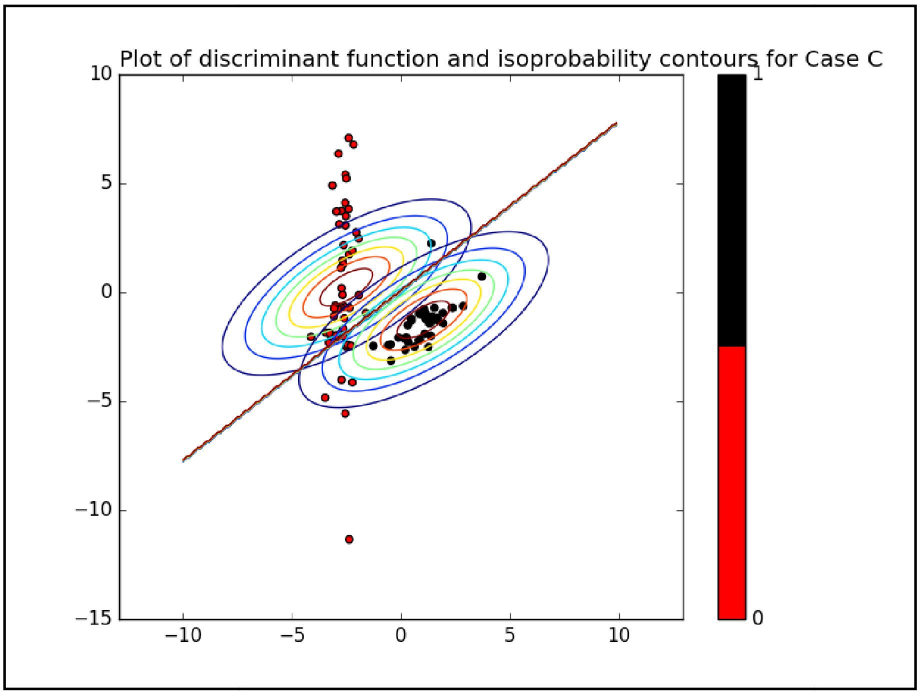


Case C:

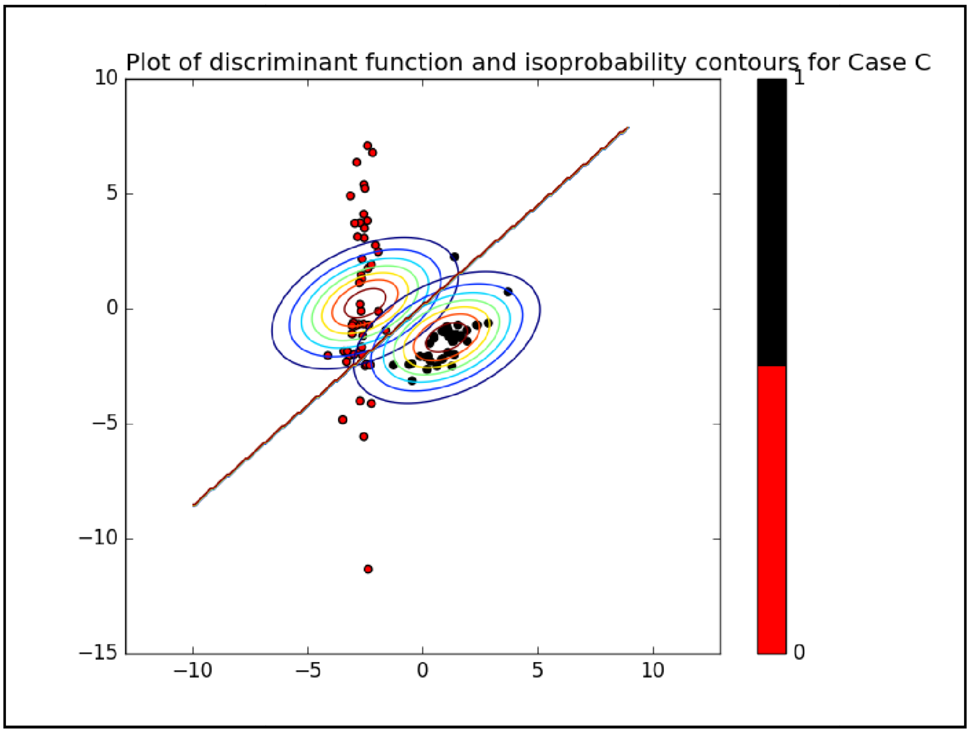
```
C. Classification using Σ0 = Σ1 = Σ of the form
a b
c d
Σ =
[[ 5.          1.61276161]
 [ 1.61276161  2.        ]]
Test Accuracy:  0.888888888888888
Class 0 Accuracy:  0.82
Class 1 Accuracy:  0.975
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[41  9]
 [ 1 39]]
```




```
C. Classification using Σ0 = Σ1 = Σ of the form
a b
c d
Σ =
[[ 8.          3.84666089]
 [ 3.84666089  4.        ]]
Test Accuracy:  0.888888888888888
Class 0 Accuracy:  0.82
Class 1 Accuracy:  0.975
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[41  9]
 [ 1 39]]
```

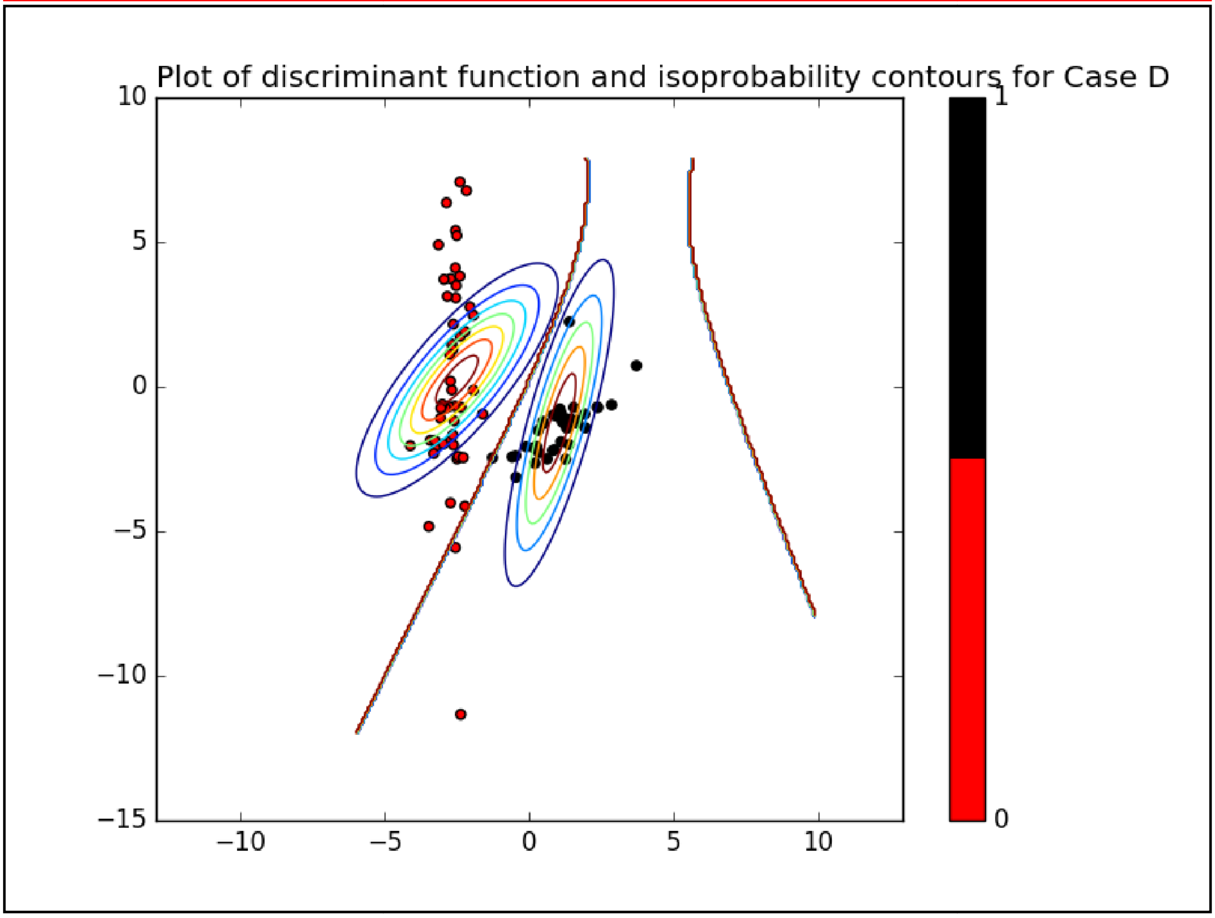



```
C. Classification using Σ0 = Σ1 = Σ of the form
a b
c d
Σ =
[[ 4.          1.18793939]
 [ 1.18793939  2.        ]]
Test Accuracy:  0.9
Class 0 Accuracy:  0.84
Class 1 Accuracy:  0.975
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[42  8]
 [ 1 39]]
```

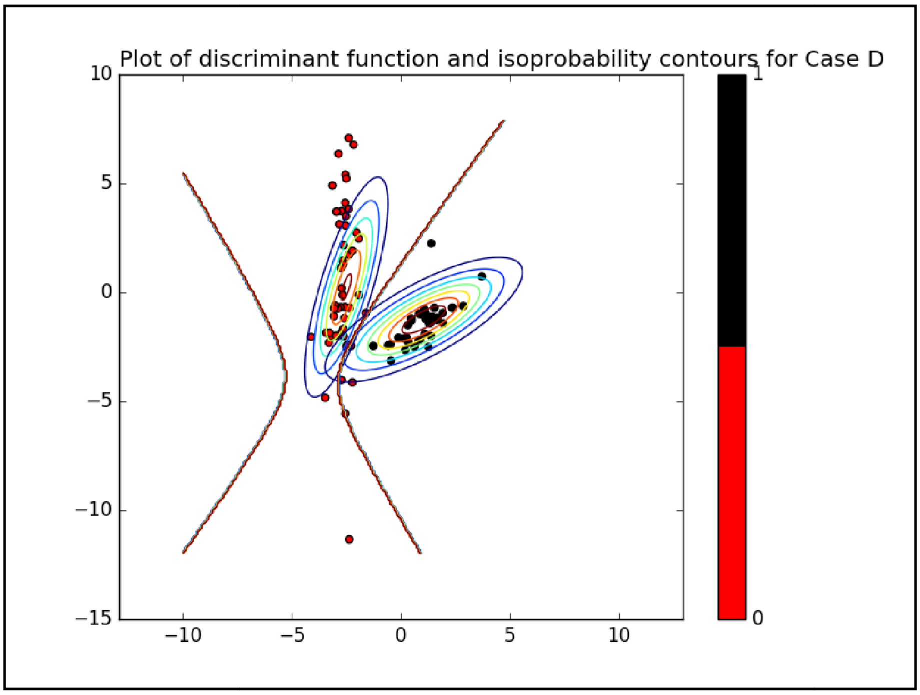


Case D:

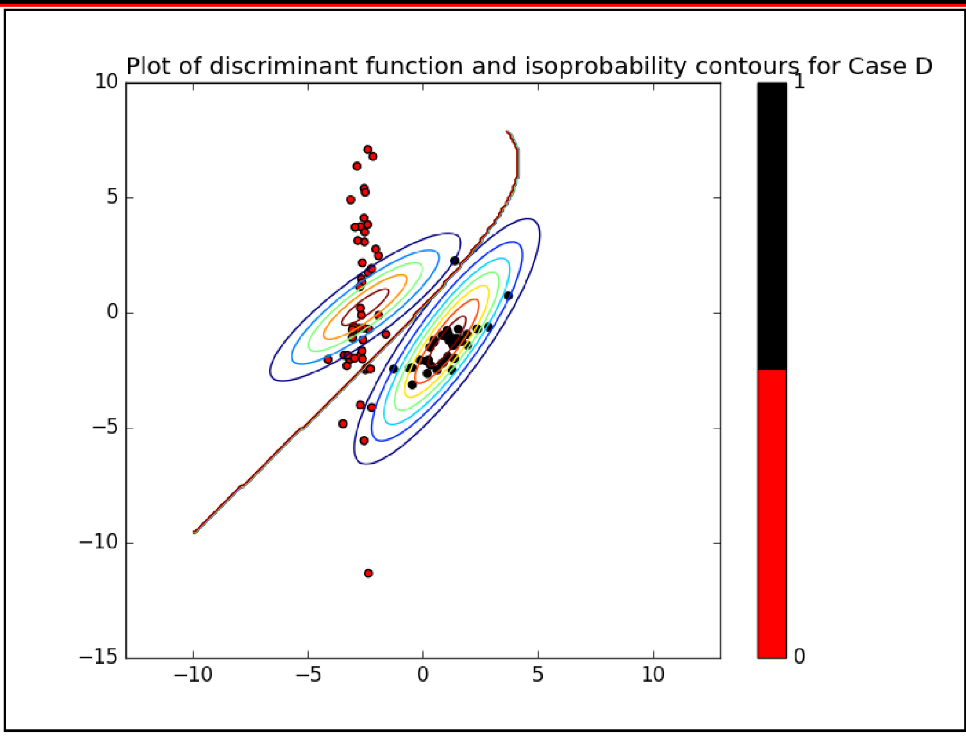
```
D. Classification using different Σ0 and Σ1 of the form
a b
c d
Σ1 =
[[ 3.          2.77128129]
 [ 2.77128129  4.        ]]
Σ2 =
[[ 1.   2.4]
 [ 2.4  9. ]]
Test Accuracy:  0.9666666666666667
Class 0 Accuracy:  0.96
Class 1 Accuracy:  0.975
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[48  2]
 [ 1 39]]
```




```
D. Classification using different Σ0 and Σ1 of the form
a b
c d
Σ1 =
[[ 1.         2.010771]
 [ 2.010771  7.       ]]
Σ2 =
[[ 5.          2.40333102]
 [ 2.40333102  2.        ]]
Test Accuracy:  0.9555555555555556
Class 0 Accuracy:  0.92
Class 1 Accuracy:  1.0
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[46  4]
 [ 0 40]]
```




```
D. Classification using different Σ0 and Σ1 of the form
a b
c d
Σ1 =
[[ 5.          3.36949551]
 [ 3.36949551  3.        ]]
Σ2 =
[[ 4.          4.60360728]
 [ 4.60360728  7.        ]]
Test Accuracy:  0.9
Class 0 Accuracy:  0.84
Class 1 Accuracy:  0.975
Confusion Matrix:

[['True Class 0', 'False Class 1'], ['False Class 0', 'True Class 1']]
[[42  8]
 [ 1 39]]
```



Conclusions:

1. Evaluated parameters are as follows:

p0 =0.4838709677419355 p1 = 0.5161290322580645

2. $\mu 0 = [-2.48646619 \;\; 0.23701453]^T$
   $\mu 1 = [\; 1.056773 \;\; -1.2525509]^T$
   $\Sigma 0 = [[0.10758923 \; 0.10686359]$
   $[0.10686359 \; 7.04930364]]$
   $\Sigma 1 = [[1.03771527 \; 0.57882397]$
   $[0.57882397 \; 1.28070394]]$

3. Testing data was classified using Bayes decision criterion with different variations of covariance matrices. Testing data was **best classified** for the case when estimates of covariance matrices Σ0 and Σ1 were used.

4. Training samples are well-fitted by the estimated gaussian.
Training Accuracy = 0.9967741935483871
Class 1 Accuracy: 1.0
Class 2 Accuracy: 0.99375

5. Reported accuracies and confusion matrices for all cases are as follows:

For estimated covariance matrices Σ0 and Σ1

Test Accuracy: 1.0
Class 1 Accuracy: 1.0
Class 2 Accuracy: 1.0
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	50	0
Predicted Class 6	0	40

A. Classification using $\Sigma 0 = \Sigma 1 = \Sigma = \begin{bmatrix} a & 0 \\ 0 & a \end{bmatrix}$ (Best case from 3 observations)

Test Accuracy: 0.967
Class 1 Accuracy: 0.94
Class 2 Accuracy: 1.0
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	47	3
Predicted Class 6	0	40

B. Classification using $\Sigma 0 = \Sigma 1 = \Sigma = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}$ (Best case from 3 observations)

Test Accuracy: 0.989
Class 1 Accuracy: 0.98
Class 2 Accuracy: 1.0
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	49	1
Predicted Class 6	0	40

C. Classification using $\Sigma 0 = \Sigma 1 = \Sigma = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ (Best case from 3 observations)

Test Accuracy: 0.9
Class 1 Accuracy: 0.84
Class 2 Accuracy: 0.975
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	42	8
Predicted Class 6	1	39

D. Classification using different Σ0 and Σ1 (Best case from 3 observations)

Test Accuracy: 0.967
Class 1 Accuracy: 0.96
Class 2 Accuracy: 0.975
Confusion Matrix:

	Actual Class 5	Actual Class 6
Predicted Class 5	48	2
Predicted Class 6	1	39

6. Learnt the effect of variations of covariance matrices on classification score. Covariance matrices controls the orientation of the Gaussian and hence can have effect on the accuracy score. The two classes 0 and 1 have well separated datapoints with few outliers.

7. The deterministic function has shape of a parabola for the cases where $\Sigma 0 \neq \Sigma 1$, and special cases where $\Sigma 0 = \Sigma 1$,it is a straight line.

8. Iso probability curves have shape of ellipse and in special cases where all elements of covariance matrices are zero, except the diagonal elements which are equal, they are circles.

Assignment - 1.P3

Objective: We wish to understand the association between an employee's age and education, as well as the calendar year, on his wage. Perform polynomial regression on age vs wage, year vs wage, plot education vs wage. Provide description of your observations on the variation of wage as a function of each these attributes. Can we get an accurate prediction of a particular man's wage from one of these 3 attributes alone?

Dataset Description: Wage dataset contains the income survey information for a group of males from Atlantic region of the United States.

Summary: For attributes education, age and year, found the best polynomial order(k) which regresses the data well given an error metric. In this experiment, used root mean squared error as loss function.

Used root mean squared error in two ways:

a. MSE between each wage[i] and attribute[i], where attributes are education, age and year.

b. MSE between each wage[i] and average(wage[attribute[j]]) where j contains domain of attribute, where attributes are education, age and year.

Using the best polynomial order, plotted the estimate of deterministic function f(attribute[i]) i.e. (w.T)*x.

Observations:

Note: Average of wage[attribute[j]] is shown as '*'. While running the script, keep closing the open figure, for further execution of the code

<u>Contents:</u>

1.a. Education vs Wage

1.b. Education vs Wage

2.a. Age vs Wage

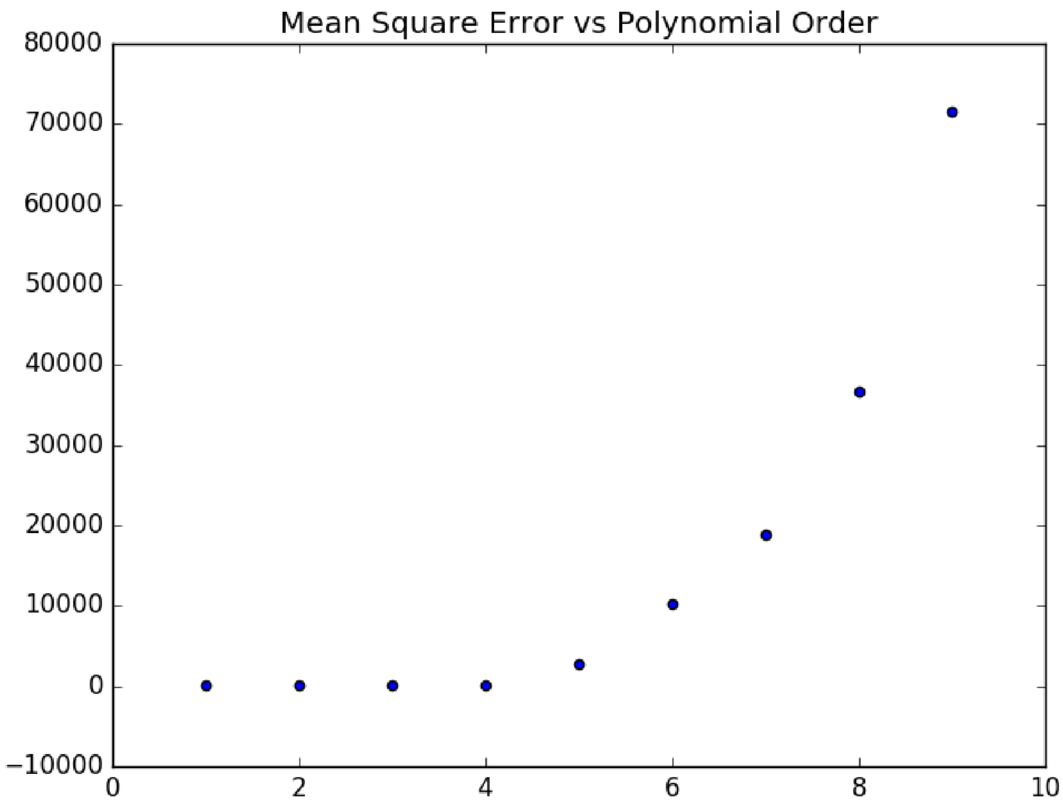
2.b. Age vs Wage

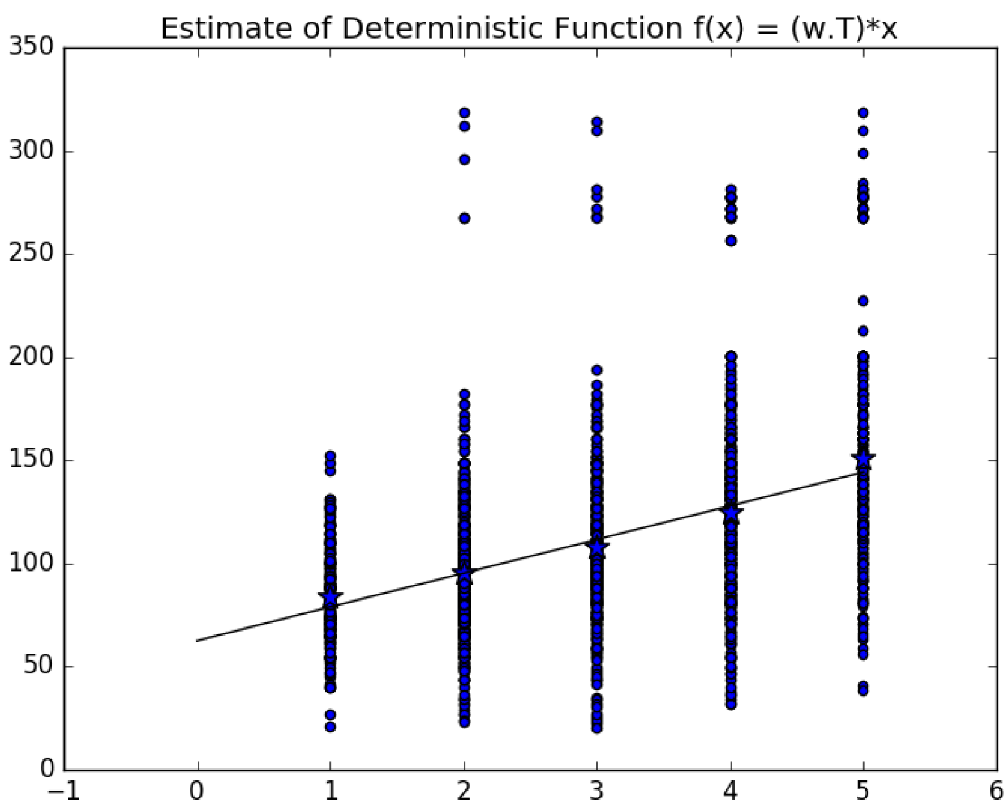
3.a. Year vs Wage

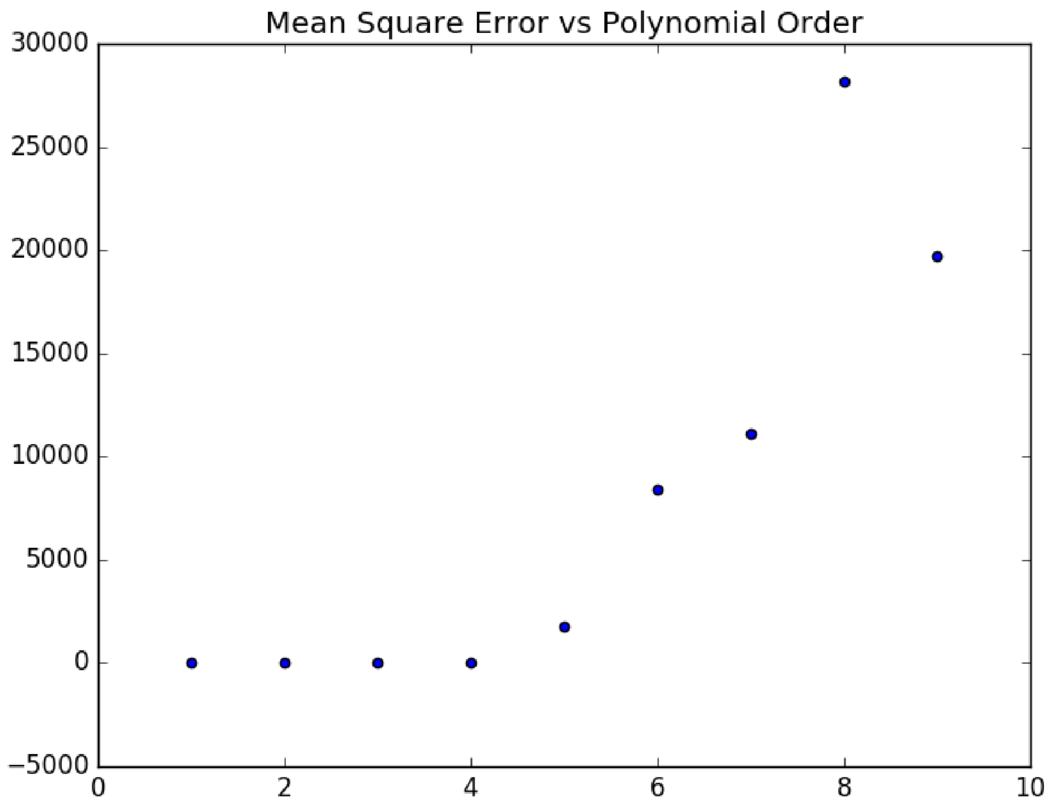
3.b. Year vs Wage

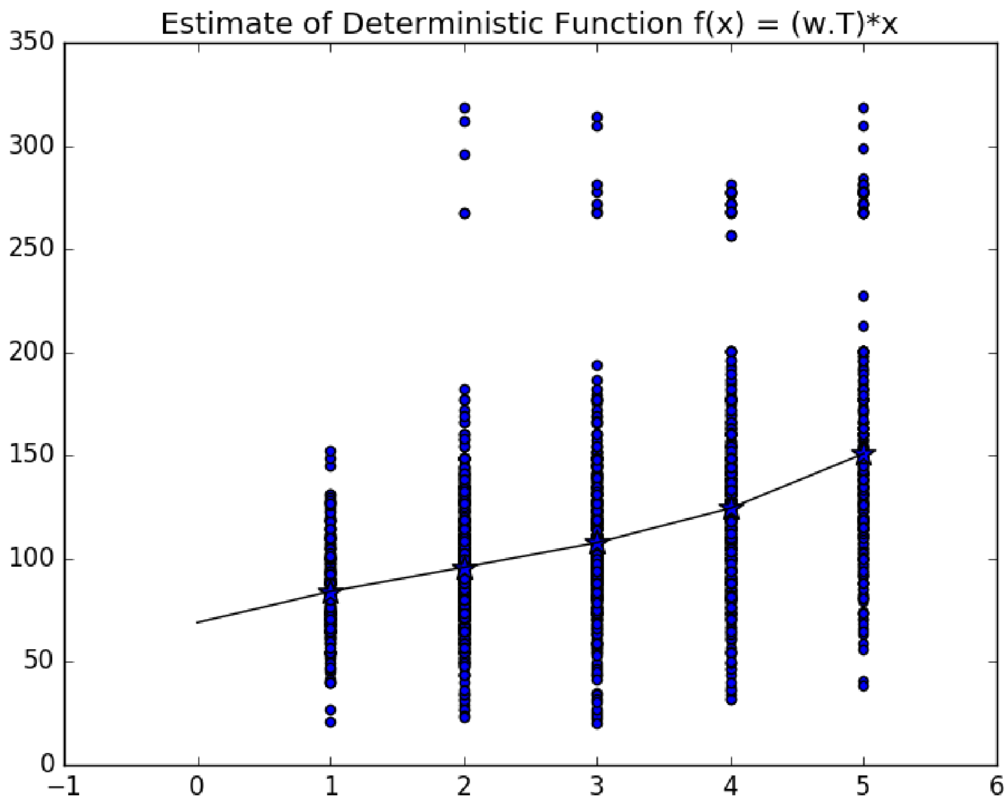
1.a. and 1.b.

```
D:\6TH SEMESTER\Course\EE 657 Pattern Recognition and Machine Learning\Assignmen
t-1-8th April\Problem_3>python P3.py

Loading Dataset...

Education vs Wage...
Predicting best polynomial order(k)...
Using MSE between each wage[i] and estimate of f(education[i]) as metric
Mean Square Error vs Polynomial Order
MSE =  80.3179522287  for polynomial order =  1
MSE =  83.4118973415  for polynomial order =  2
MSE =  81.8077313319  for polynomial order =  3
MSE =  81.6622876618  for polynomial order =  4
MSE =  2745.47848079  for polynomial order =  5
MSE =  10222.140233  for polynomial order =  6
MSE =  18817.8430149  for polynomial order =  7
MSE =  36695.6369106  for polynomial order =  8
MSE =  71478.7552223  for polynomial order =  9
Best polynomial order is  1  with MSE equal to  80.3179522287
Learned w is:
[[ 62.54440413]
 [ 16.33196149]]

Education vs Wage...
Predicting best polynomial order(k)...
Using MSE between each wage[i] and average(estimate of f(wage[education[j]]) as
metric
Mean Square Error vs Polynomial Order
MSE =  9.9459902817  for polynomial order =  1
MSE =  3.37406291115  for polynomial order =  2
MSE =  0.0875281330565  for polynomial order =  3
MSE =  2.71099771607e-09  for polynomial order =  4
MSE =  1747.01233487  for polynomial order =  5
MSE =  8421.9820059  for polynomial order =  6
MSE =  11136.8375047  for polynomial order =  7
MSE =  28162.7083952  for polynomial order =  8
MSE =  19746.8345974  for polynomial order =  9
Best polynomial order is  4  with MSE equal to  2.71099771607e-09
Learned w is:
[[  6.90224599e+01]
 [  1.78379257e+01]
 [ -3.22398882e+00]
 [  4.38411791e-01]
 [  2.96062951e-02]]
```


Mean Square Error vs Polynomial Order
80000
70000
60000
50000
40000
30000
20000
10000
0
−10000
0
2
4
6
8
10

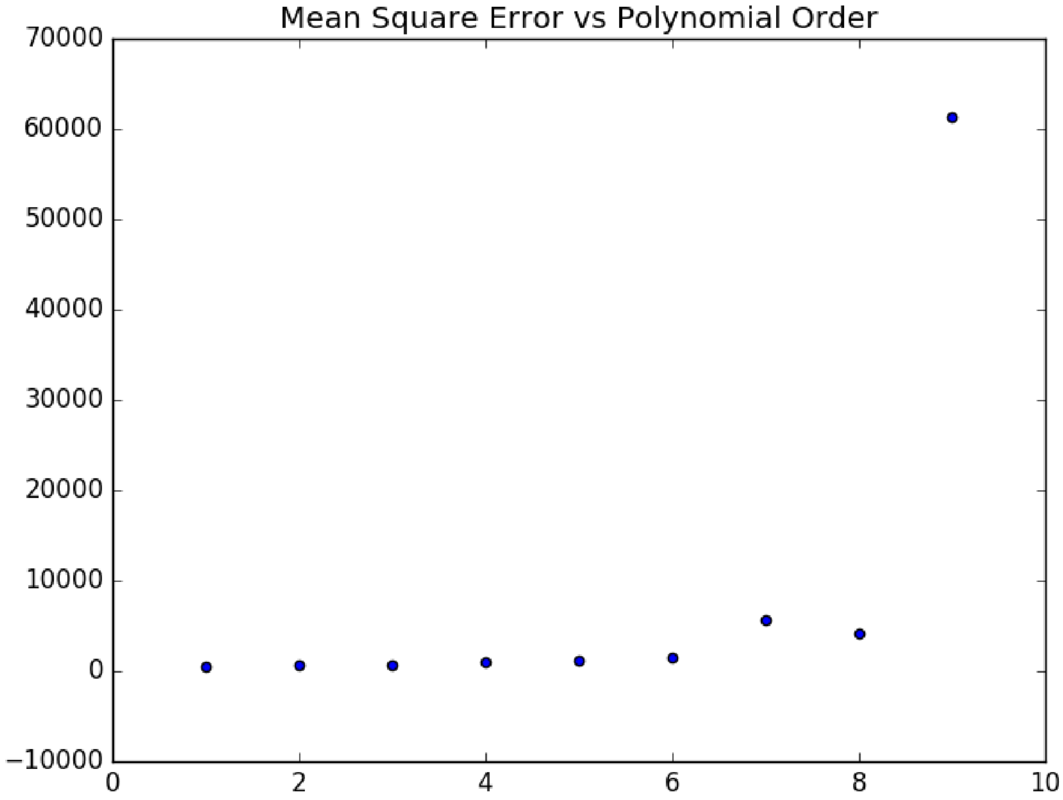

Estimate of Deterministic Function f(x) = (w.T)*x
350
300
250
200
150
100
50
0
−1
0
1
2
3
4
5
6

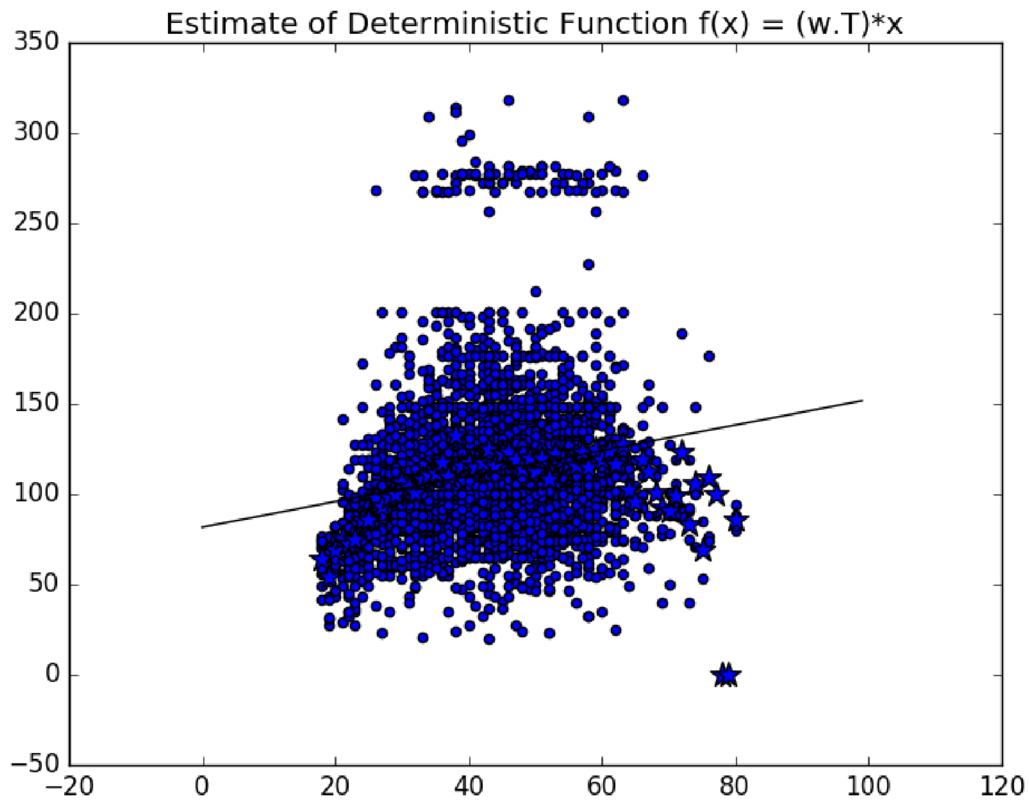

Mean Square Error vs Polynomial Order
30000
25000
20000
15000
10000
5000
0
−5000
0
2
4
6
8
10

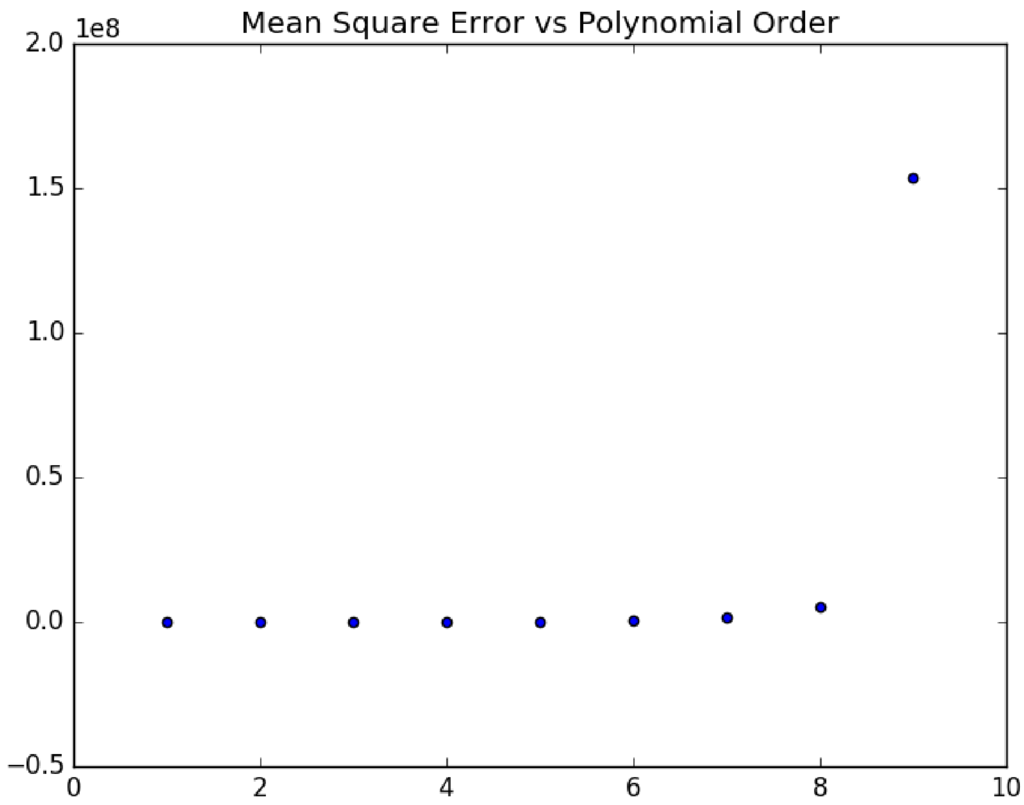

Estimate of Deterministic Function f(x) = (w.T)*x
350
300
250
200
150
100
50
0
−1
0
1
2
3
4
5
6

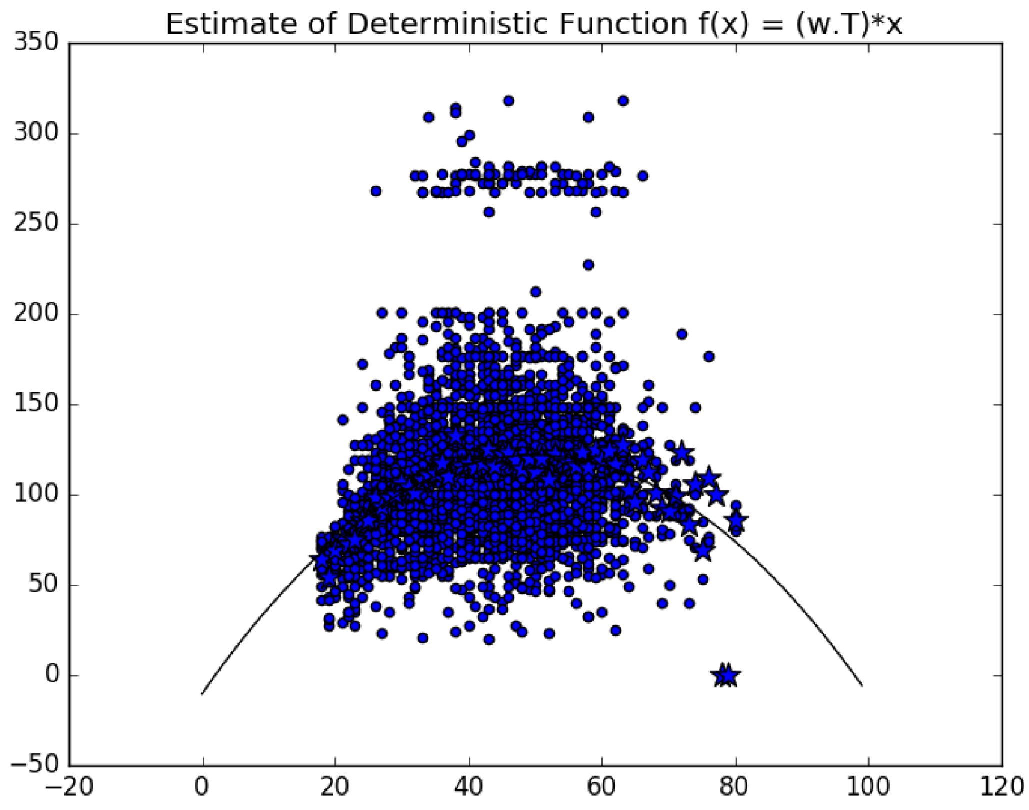
2.a. and 2.b.

```
Age vs Wage...
Predicting best polynomial order(k)...
Using MSE between each wage[i] and estimate of f(education[i]) as metric
Mean Square Error vs Polynomial Order
MSE =  543.653144355  for polynomial order =  1
MSE =  731.792078435  for polynomial order =  2
MSE =  738.887703023  for polynomial order =  3
MSE =  1041.75905395  for polynomial order =  4
MSE =  1222.03042894  for polynomial order =  5
MSE =  1439.36142732  for polynomial order =  6
MSE =  5658.28665891  for polynomial order =  7
MSE =  4127.21717136  for polynomial order =  8
MSE =  61342.1599467  for polynomial order =  9
Best polynomial order is  1  with MSE equal to  543.653144355
Learned w is:
[[ 81.70473544]
 [  0.70727593]]

Age vs Wage...
Predicting best polynomial order(k)...
Using MSE between each wage[i] and average(estimate of f(wage[education[j]]) as
metric
Mean Square Error vs Polynomial Order
MSE =  252.756990149  for polynomial order =  1
MSE =  131.09971152  for polynomial order =  2
MSE =  156.567450965  for polynomial order =  3
MSE =  139.211469977  for polynomial order =  4
P3.py:366: RuntimeWarning: overflow encountered in long_scalars
  X[j] = i**(j)
MSE =  10462.3067073  for polynomial order =  5
MSE =  274714.230557  for polynomial order =  6
MSE =  1726167.09789  for polynomial order =  7
MSE =  5106493.55022  for polynomial order =  8
MSE =  153775316.513  for polynomial order =  9
Best polynomial order is  2  with MSE equal to  131.09971152
Learned w is:
[[-10.42522426]
 [  5.29403003]
 [ -0.05300507]]
```


Mean Square Error vs Polynomial Order
70000
60000
50000
40000
30000
20000
10000
0
−10000
0
2
4
6
8
10

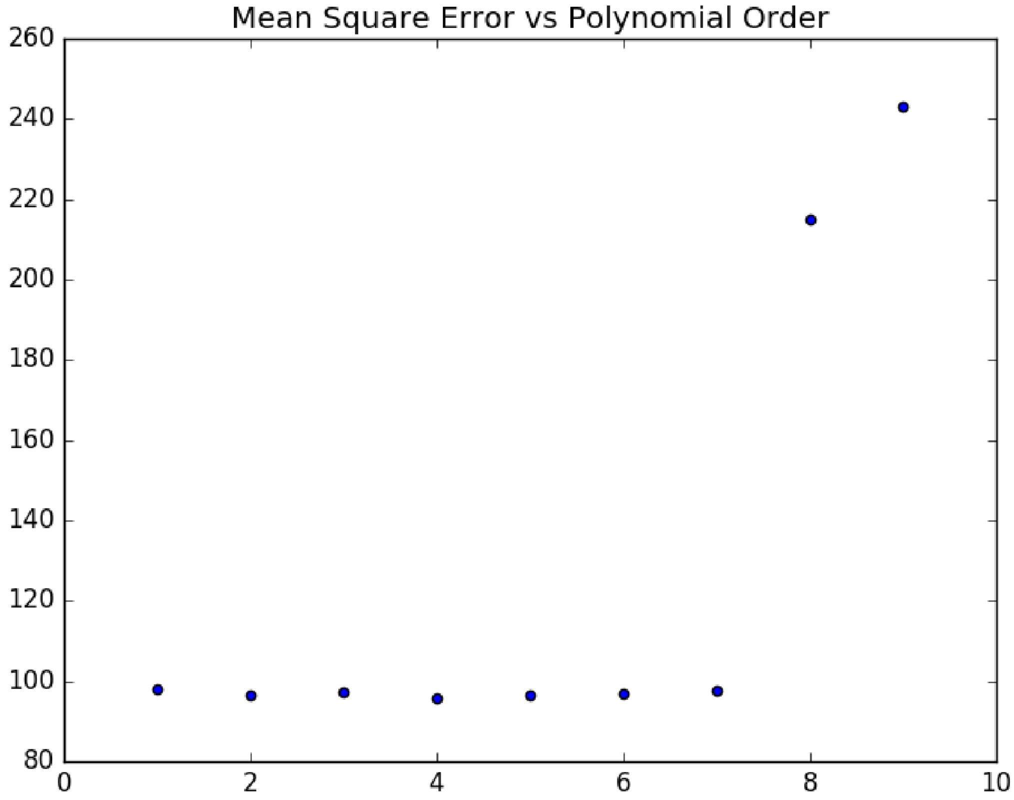

Estimate of Deterministic Function f(x) = (w.T)*x
350
300
250
200
150
100
50
0
−50
−20
0
20
40
60
80
100
120

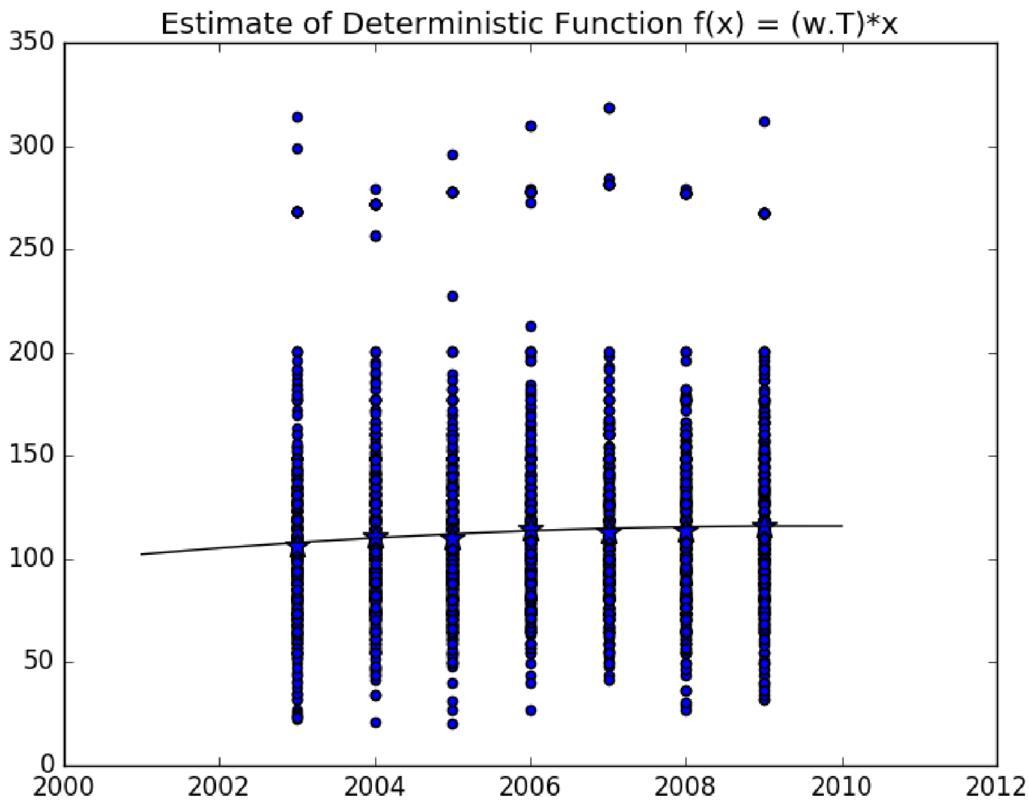

Mean Square Error vs Polynomial Order
1e8
2.0
1.5
1.0
0.5
0.0
−0.5
0
2
4
6
8
10

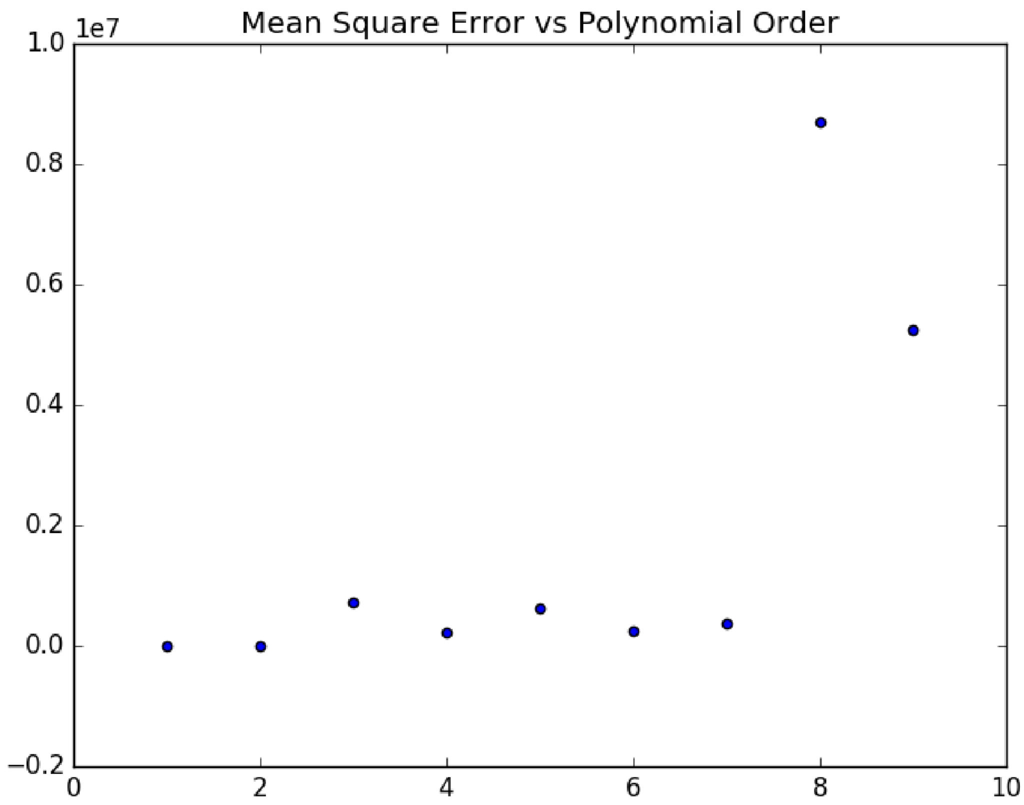

Estimate of Deterministic Function f(x) = (w.T)*x
350
300
250
200
150
100
50
0
−50
−20
0
20
40
60
80
100
120

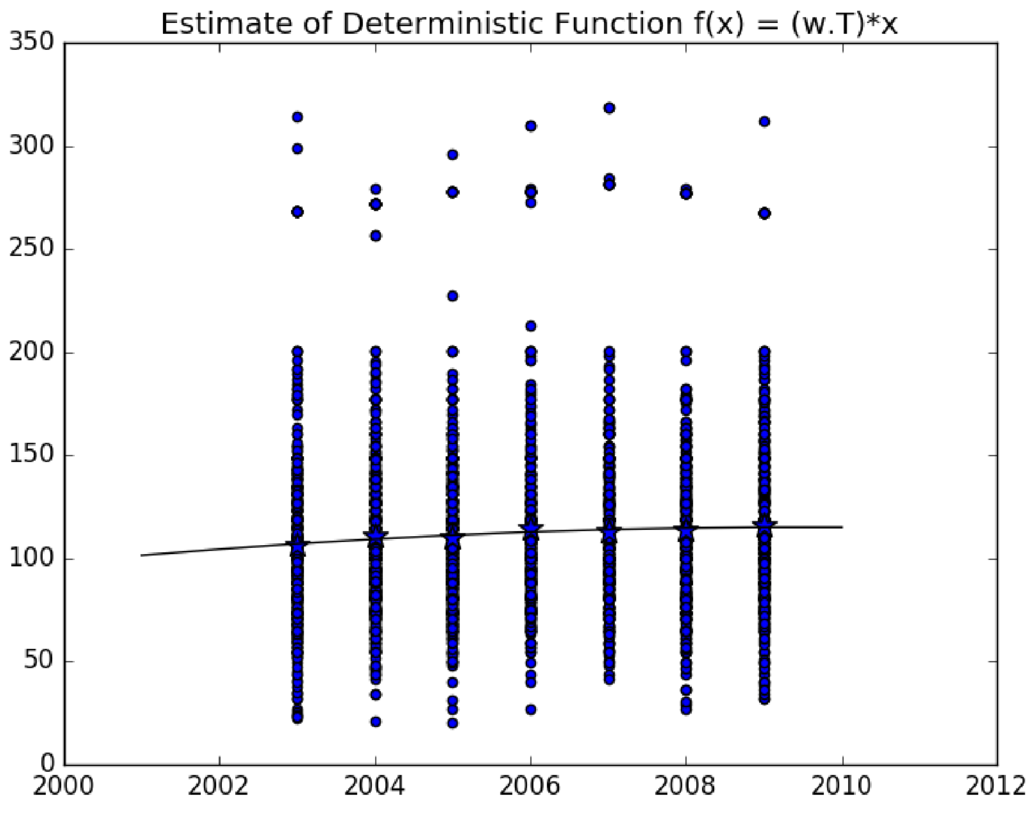
3.a. and 3.b.

```
Year vs Wage...
Predicting best polynomial order(k)...
Using MSE between each wage[i] and estimate of f(education[i]) as metric
Mean Square Error vs Polynomial Order
MSE =  98.0086130278  for polynomial order =  1
MSE =  96.4182752859  for polynomial order =  2
MSE =  97.0669682585  for polynomial order =  3
MSE =  95.9278387103  for polynomial order =  4
MSE =  96.5924900049  for polynomial order =  5
MSE =  96.760323504  for polynomial order =  6
MSE =  97.4509369164  for polynomial order =  7
MSE =  215.165942129  for polynomial order =  8
MSE =  243.138827543  for polynomial order =  9
Best polynomial order is  4  with MSE equal to  95.9278387103
Learned w is:
[[ -2.44902739e+05]
 [  9.04928914e+01]
 [  3.84640336e-02]
 [  1.52693870e-05]
 [ -1.32497119e-08]]

Year vs Wage...
Predicting best polynomial order(k)...
Using MSE between each wage[i] and average(estimate of f(wage[education[j]]) as
metric
Mean Square Error vs Polynomial Order
MSE =  3.66964230125  for polynomial order =  1
MSE =  3.28647290347  for polynomial order =  2
P3.py:566: RuntimeWarning: overflow encountered in long_scalars
  X[j] = i**(j)
MSE =  735204.707339  for polynomial order =  3
MSE =  220650.793464  for polynomial order =  4
MSE =  617020.824282  for polynomial order =  5
MSE =  256032.604561  for polynomial order =  6
MSE =  366510.358121  for polynomial order =  7
MSE =  8693304.27747  for polynomial order =  8
MSE =  5253789.214  for polynomial order =  9
Best polynomial order is  2  with MSE equal to  3.28647290347
Learned w is:
[[ -7.62431310e+05]
 [  7.58937713e+02]
 [ -1.88836529e-01]]
```


Mean Square Error vs Polynomial Order
260
240
220
200
180
160
140
120
100
80
0
2
4
6
8
10


Estimate of Deterministic Function f(x) = (w.T)*x
350
300
250
200
150
100
50
0
2000
2002
2004
2006
2008
2010
2012


Mean Square Error vs Polynomial Order
1e7
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
2
4
6
8
10


Estimate of Deterministic Function f(x) = (w.T)*x
350
300
250
200
150
100
50
0
2000
2002
2004
2006
2008
2010
2012

Conclusions:

1. For education vs wage, best polynomial order is 4 with MSE appx. equal to 0.

2. For age vs wage, best polynomial order is 2 with MSE appx. equal to 131.099.

3. For year vs wage, best polynomial order is 2 with MSE appx. equal to 3.286.

4. Can we get an accurate prediction of a particular man's wage from one of

these 3 attributes alone?

Yes, we can get an accurate prediction of a particular man's wage from one of these 3 attributes alone i.e. **education** as MSE of education vs wage is least as compared to other attributes vs wage.

5. It can be clearly seen from the regressed education vs wage plot, that as education level increases from <HS Grad to Advanced Degree, wage of the person also increases.

6. It can be clearly seen from the regressed age vs wage plot, that as age of the person increases, wage of the person initially increases then decreases after a peak age around 60 as man's work capacity decreses.

7. It can be clearly seen from the regressed year vs wage plot, that as year increases from 2003 through 2009, wage of the person increases.