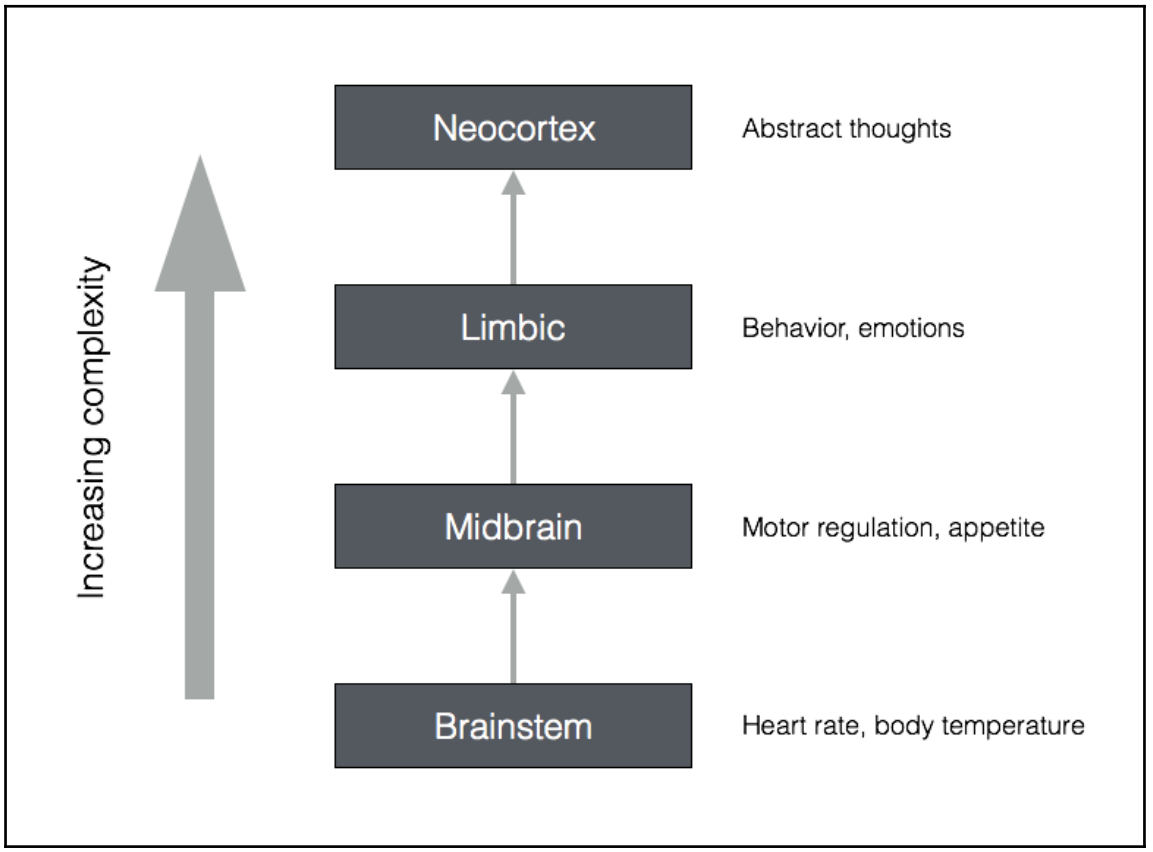
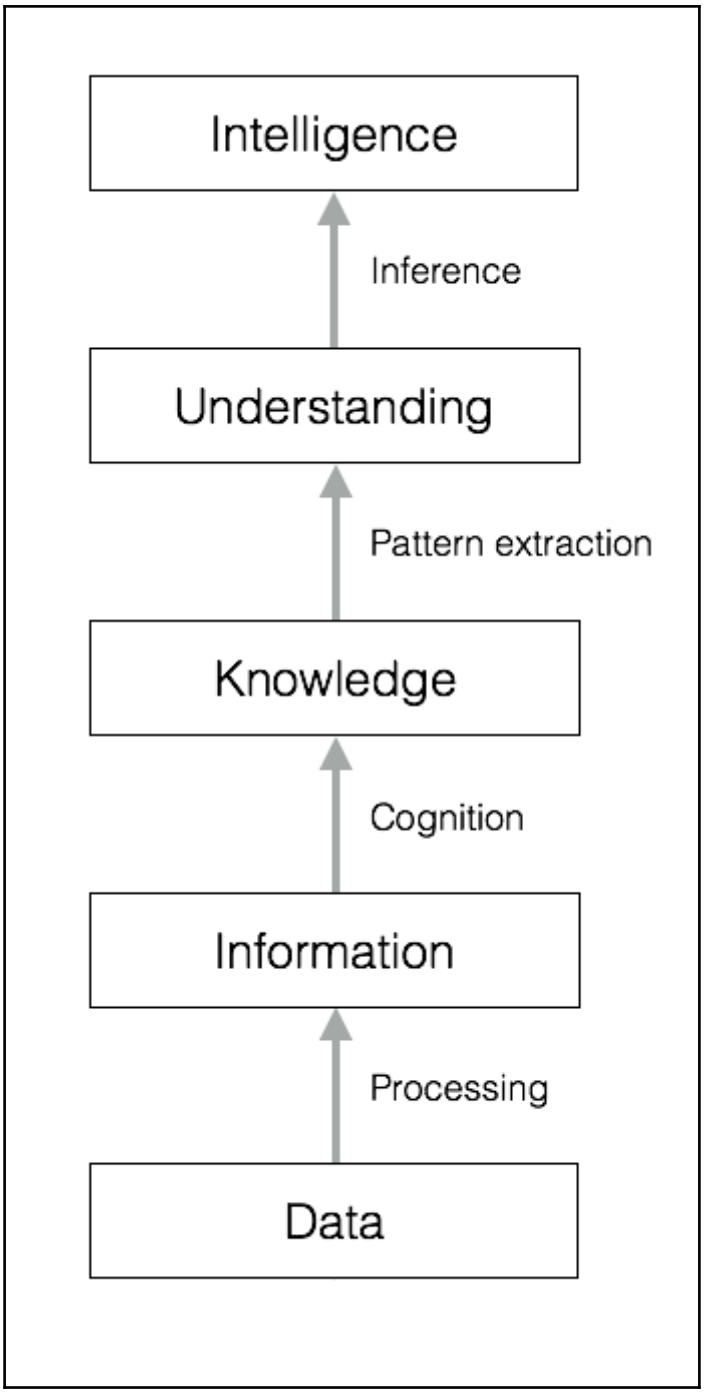
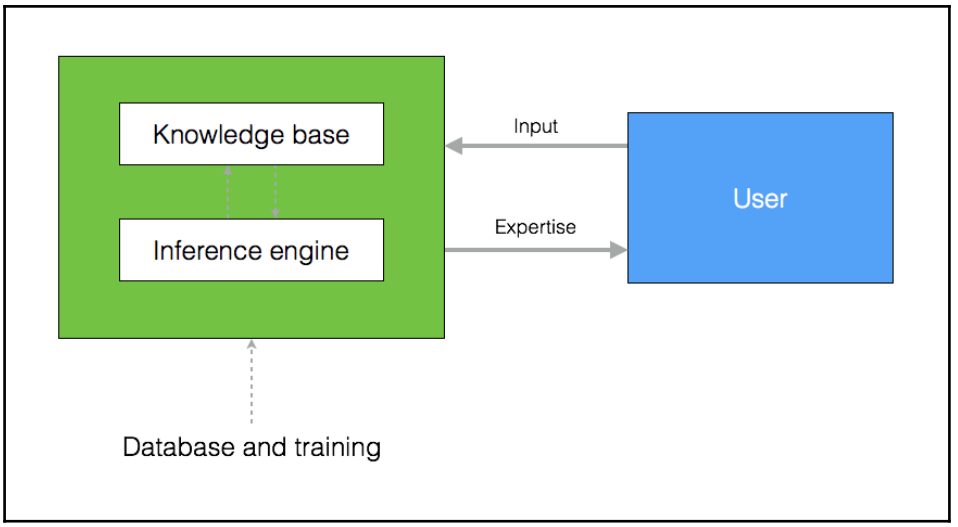
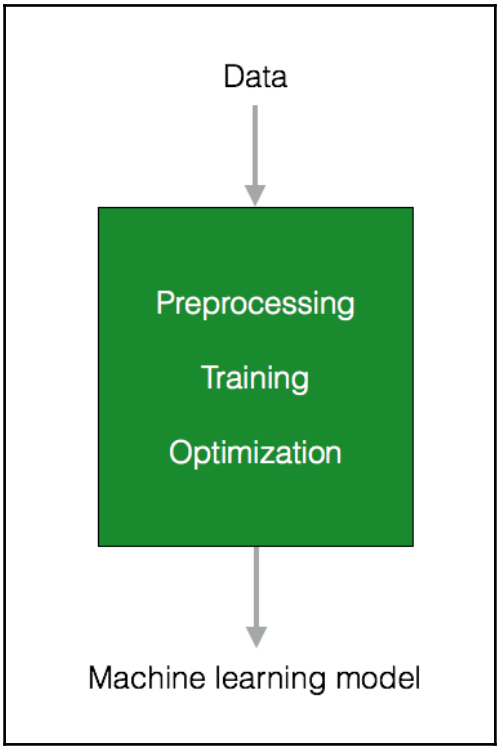
Graphics Bundle

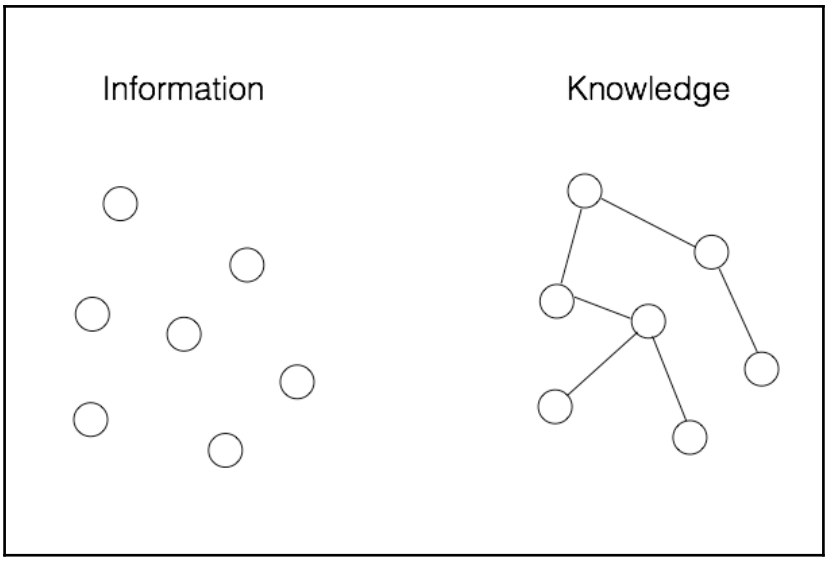
Chapter 1: Introduction to Artificial Intelligence

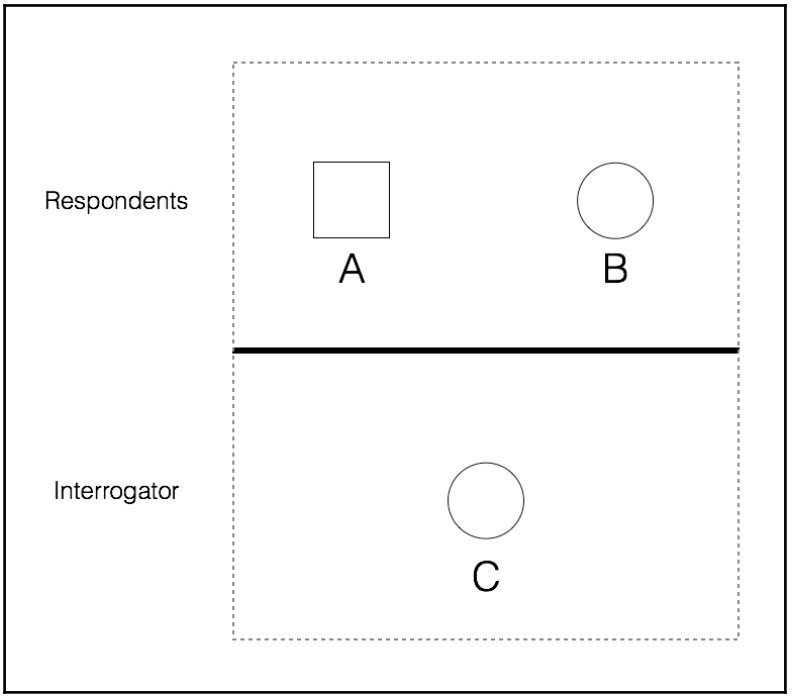


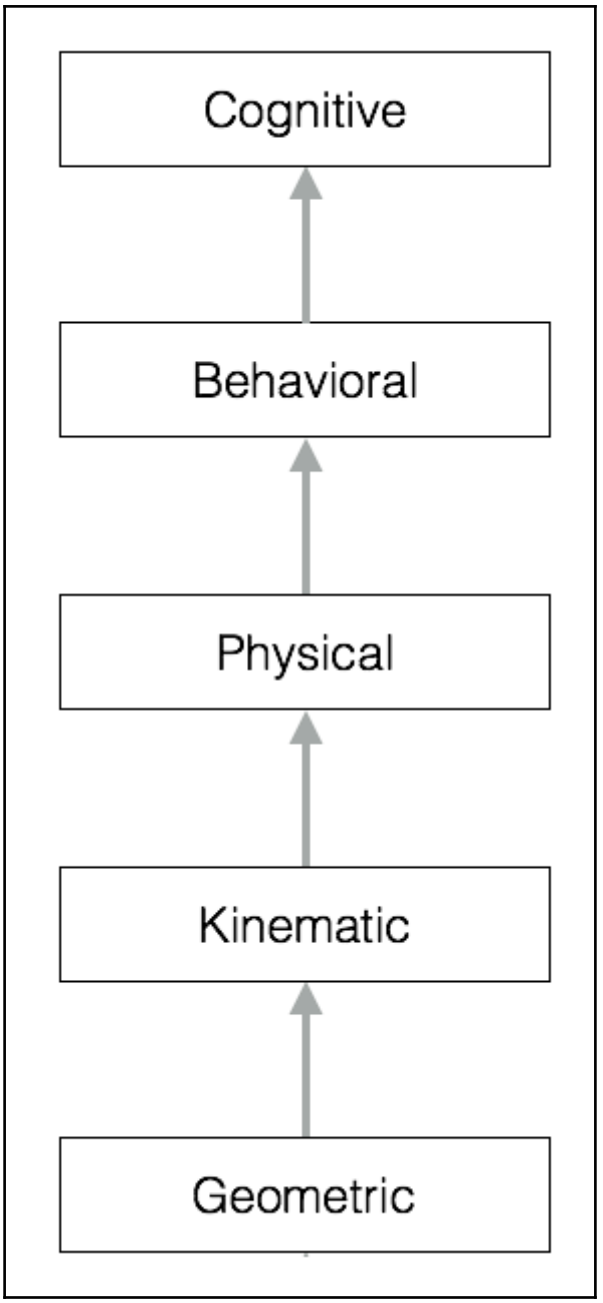

Intelligence
Inference
Understanding
Pattern extraction
Knowledge
Cognition
Information
Processing
Data

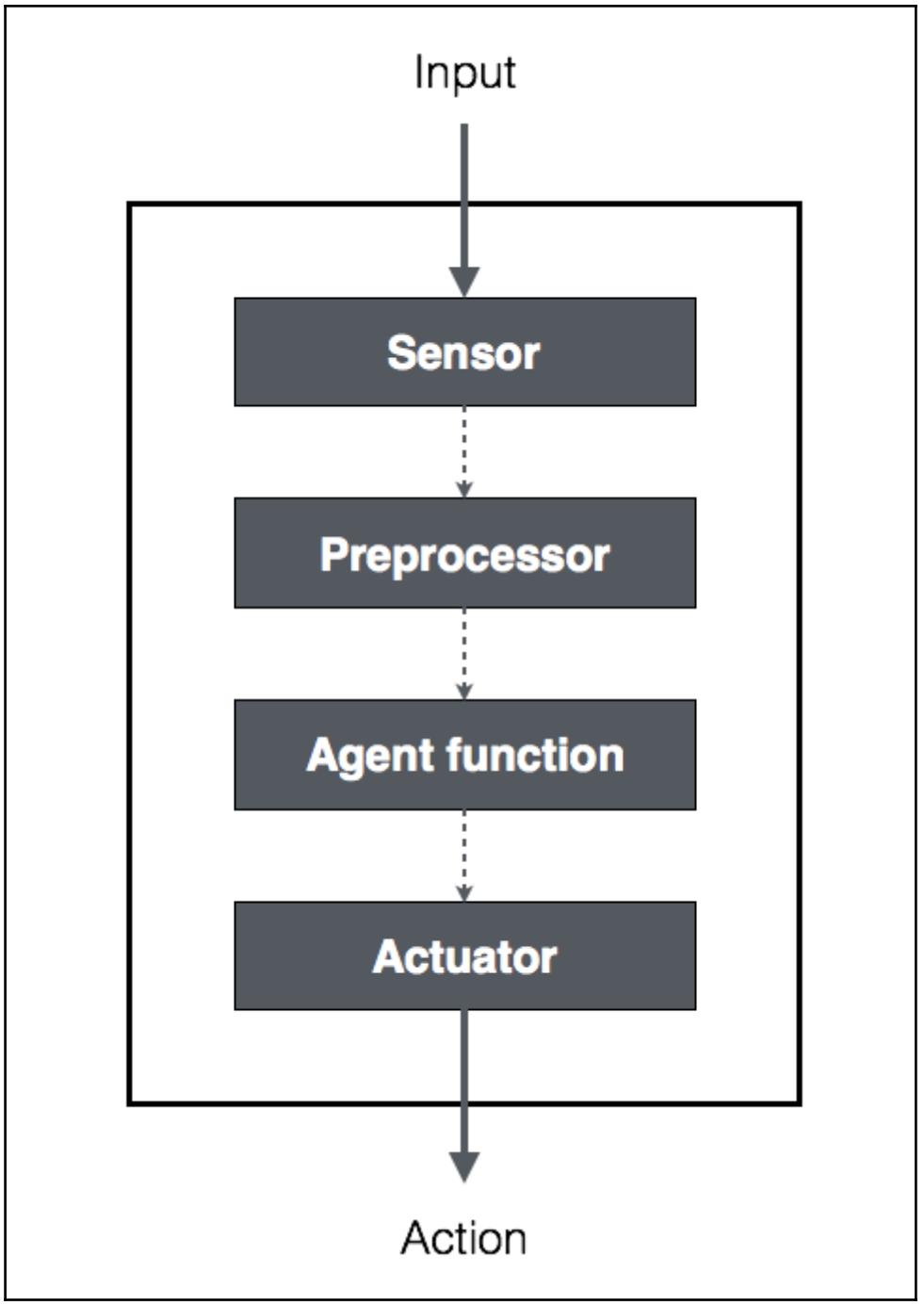

Knowledge base
Inference engine
Input
Expertise
User
Database and training

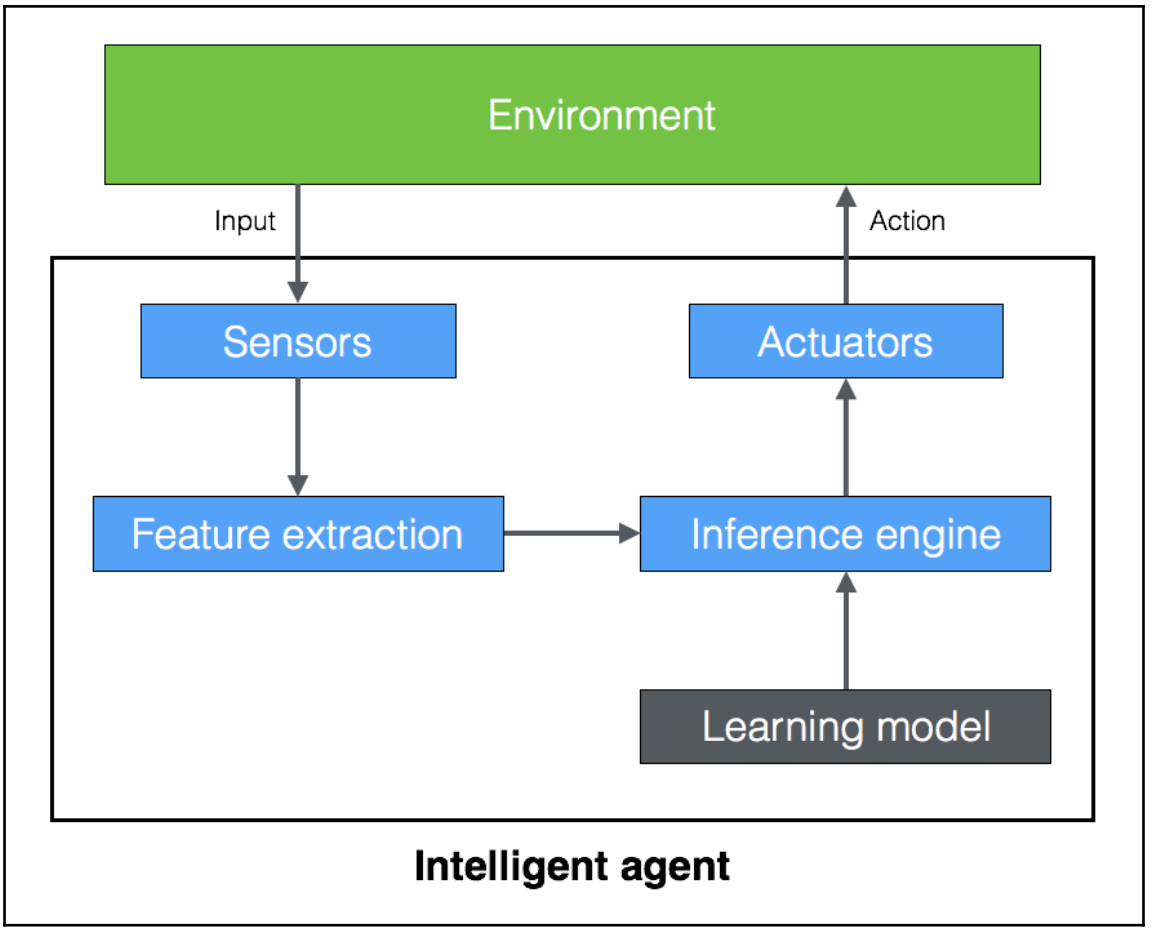

Data
Preprocessing
Training
Optimization
Machine learning model


Information
Knowledge


Respondents
A
B
Interrogator
C


Cognitive
Behavioral
Physical
Kinematic
Geometric


Input
Sensor
Preprocessor
Agent function
Actuator
Action




```
>>> print(house_prices.data)
[[  6.32000000e-03   1.80000000e+01   2.31000000e+00 ...,   1.53000000e+01
    3.96900000e+02   4.98000000e+00]
 [  2.73100000e-02   0.00000000e+00   7.07000000e+00 ...,   1.78000000e+01
    3.96900000e+02   9.14000000e+00]
 [  2.72900000e-02   0.00000000e+00   7.07000000e+00 ...,   1.78000000e+01
    3.92830000e+02   4.03000000e+00]
 ...,
 [  6.07600000e-02   0.00000000e+00   1.19300000e+01 ...,   2.10000000e+01
    3.96900000e+02   5.64000000e+00]
 [  1.09590000e-01   0.00000000e+00   1.19300000e+01 ...,   2.10000000e+01
    3.93450000e+02   6.48000000e+00]
 [  4.74100000e-02   0.00000000e+00   1.19300000e+01 ...,   2.10000000e+01
    3.96900000e+02   7.88000000e+00]]
```

```
>>> print(house_prices.target)
[ 24.   21.6  34.7  33.4  36.2  28.7  22.9  27.1  16.5  18.9  15.   18.9
  21.7  20.4  18.2  19.9  23.1  17.5  20.2  18.2  13.6  19.6  15.2  14.5
  15.6  13.9  16.6  14.8  18.4  21.   12.7  14.5  13.2  13.1  13.5  18.9
  20.   21.   24.7  30.8  34.9  26.6  25.3  24.7  21.2  19.3  20.   16.6
  14.4  19.4  19.7  20.5  25.   23.4  18.9  35.4  24.7  31.6  23.3  19.6
  18.7  16.   22.2  25.   33.   23.5  19.4  22.   17.4  20.9  24.2  21.7
  22.8  23.4  24.1  21.4  20.   20.8  21.2  20.3  28.   23.9  24.8  22.9
  23.9  26.6  22.5  22.2  23.6  28.7  22.6  22.   22.9  25.   20.6  28.4
  21.4  38.7  43.8  33.2  27.5  26.5  18.6  19.3  20.1  19.5  19.5  20.4
  19.8  19.4  21.7  22.8  18.8  18.7  18.5  18.3  21.2  19.2  20.4  19.3
  22.   20.3  20.5  17.3  18.8  21.4  15.7  16.2  18.   14.3  19.2  19.6
  23.   18.4  15.6  18.1  17.4  17.1  13.3  17.8  14.   14.4  13.4  15.6
  11.8  13.8  15.6  14.6  17.8  15.4  21.5  19.6  15.3  19.4  17.   15.6
  13.1  41.3  24.3  23.3  27.   50.   50.   50.   22.7  25.   50.   23.8
  23.8  22.3  17.4  19.1  23.1  23.6  22.6  29.4  23.2  24.6  29.9  37.2
  39.8  36.2  37.9  32.5  26.4  29.6  50.   32.   29.8  34.9  37.   30.5
  36.4  31.1  29.1  50.   33.3  30.3  34.6  34.9  32.9  24.1  42.3  48.5
  50.   22.6  24.4  22.5  24.4  20.   21.7  19.3  22.4  28.1  23.7  25.
  23.3  28.7  21.5  23.   26.7  21.7  27.5  30.1  44.8  50.   37.6  31.6
  46.7  31.5  24.3  31.7  41.7  48.3  29.   24.   25.1  31.5  23.7  23.3
```

```
>>> print(digits.images[4])
[[  0.   0.   0.   1.  11.   0.   0.   0.]
 [  0.   0.   0.   7.   8.   0.   0.   0.]
 [  0.   0.   1.  13.   6.   2.   2.   0.]
 [  0.   0.   7.  15.   0.   9.   8.   0.]
 [  0.   5.  16.  10.   0.  16.   6.   0.]
 [  0.   4.  15.  16.  13.  16.   1.   0.]
 [  0.   0.   0.   3.  15.  10.   0.   0.]
 [  0.   0.   0.   2.  16.   4.   0.   0.]]
```

Chapter 2: Classification and Regression Using Supervised Learning

```
Label mapping:
black --> 0
green --> 1
red --> 2
white --> 3
yellow --> 4

Labels = ['green', 'red', 'black']
Encoded values = [1, 2, 0]

Encoded values = [3, 0, 4, 1]
Decoded labels = ['white', 'black', 'yellow', 'green']
```

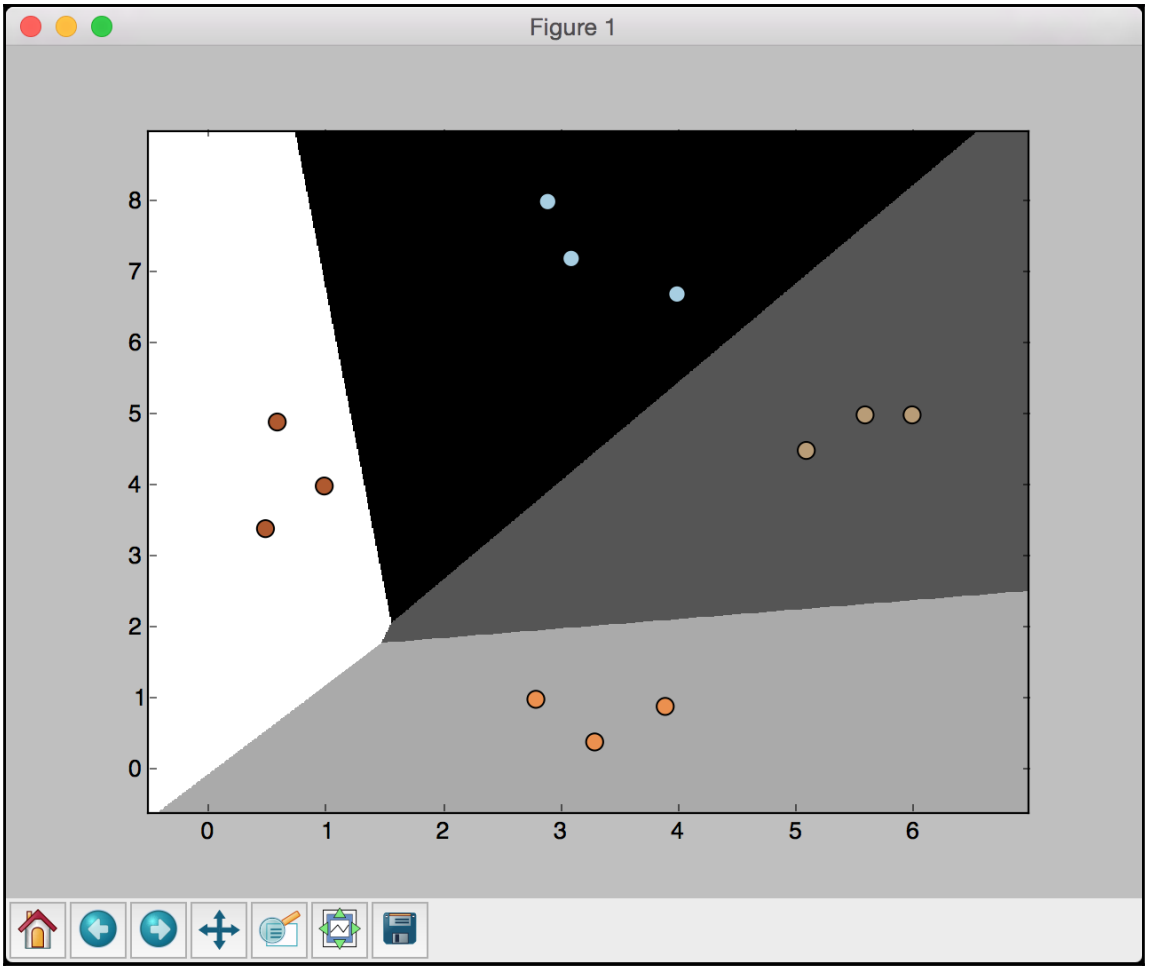


Figure 1
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6

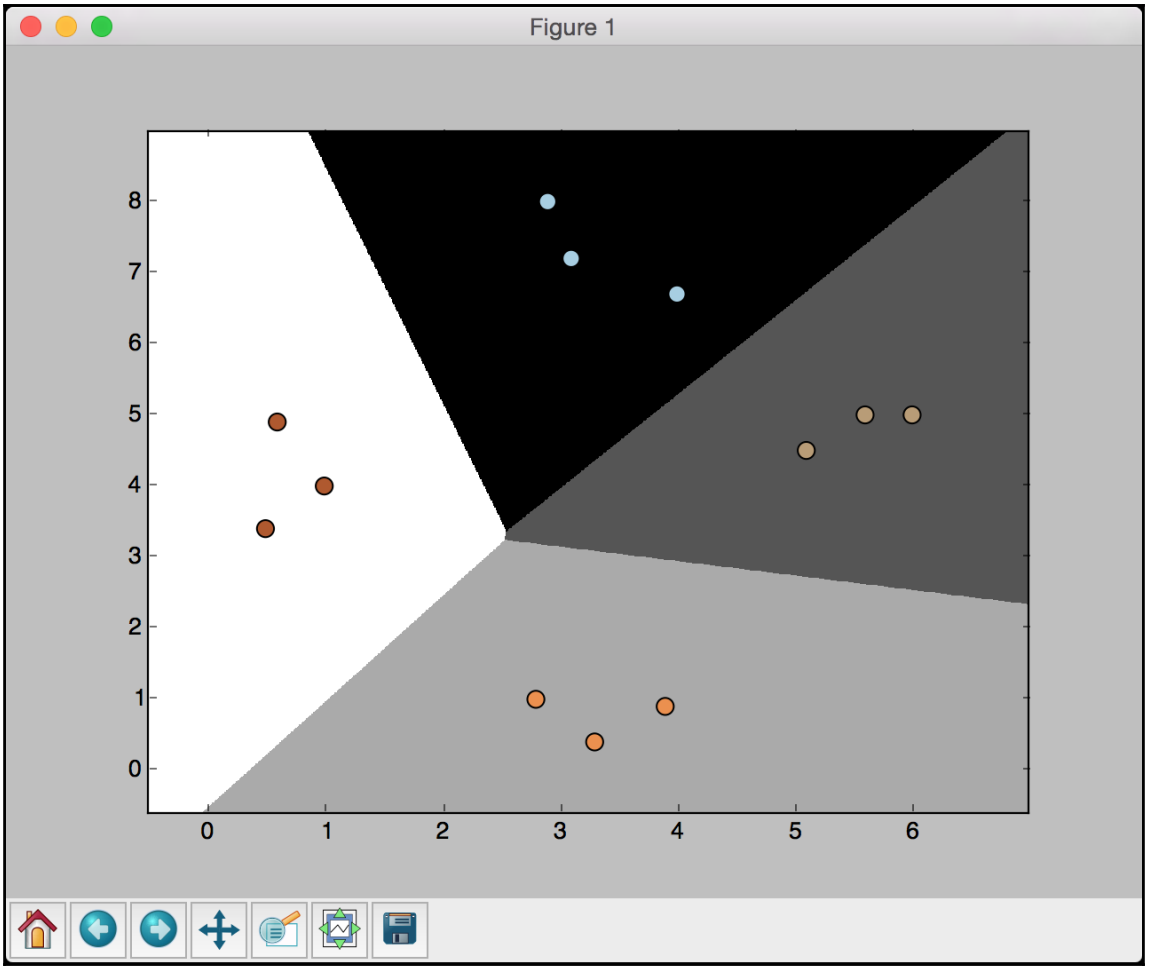


Figure 1
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6

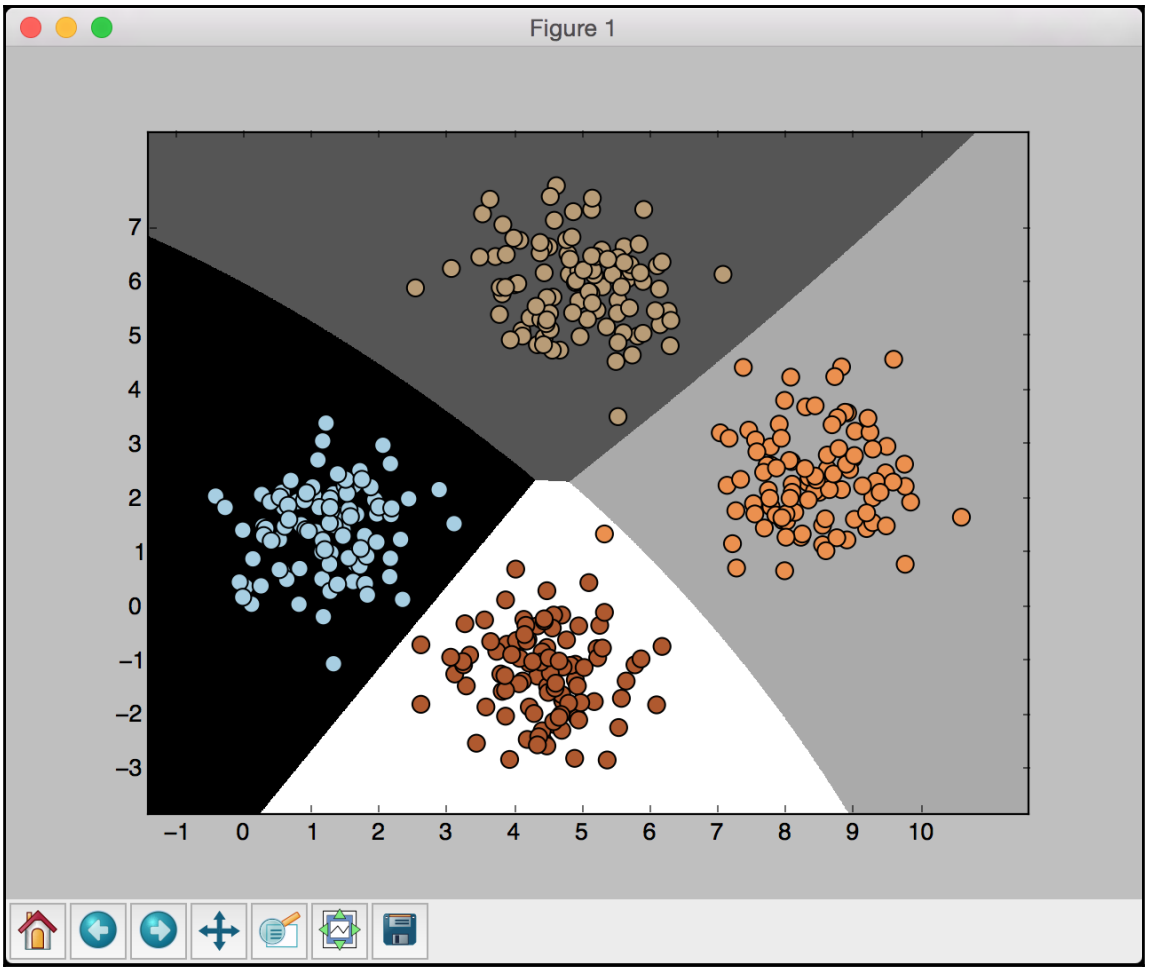


Figure 1
7
6
5
4
3
2
1
0
−1
−2
−3
−1
0
1
2
3
4
5
6
7
8
9
10

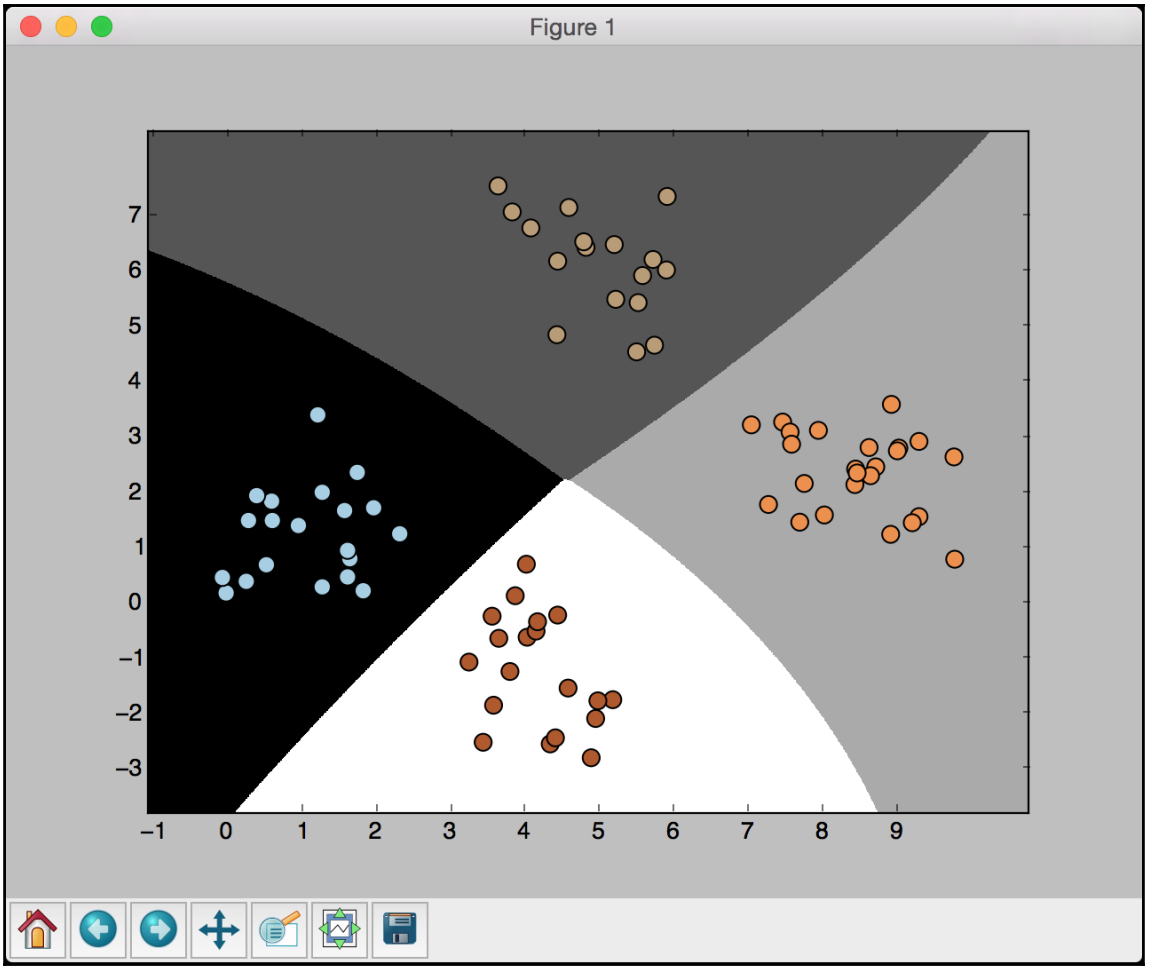


Figure 1
7
6
5
4
3
2
1
0
−1
−2
−3
−1
0
1
2
3
4
5
6
7
8
9

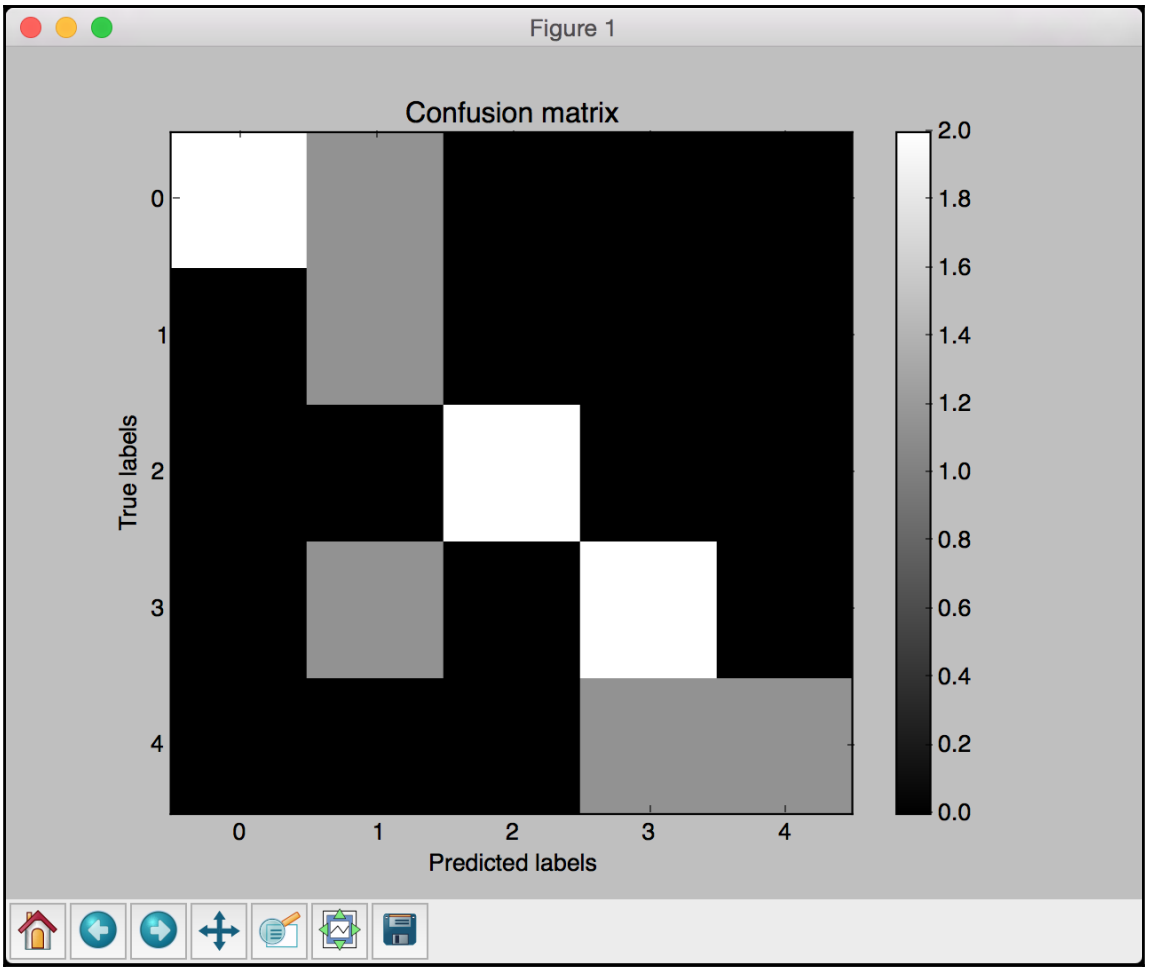
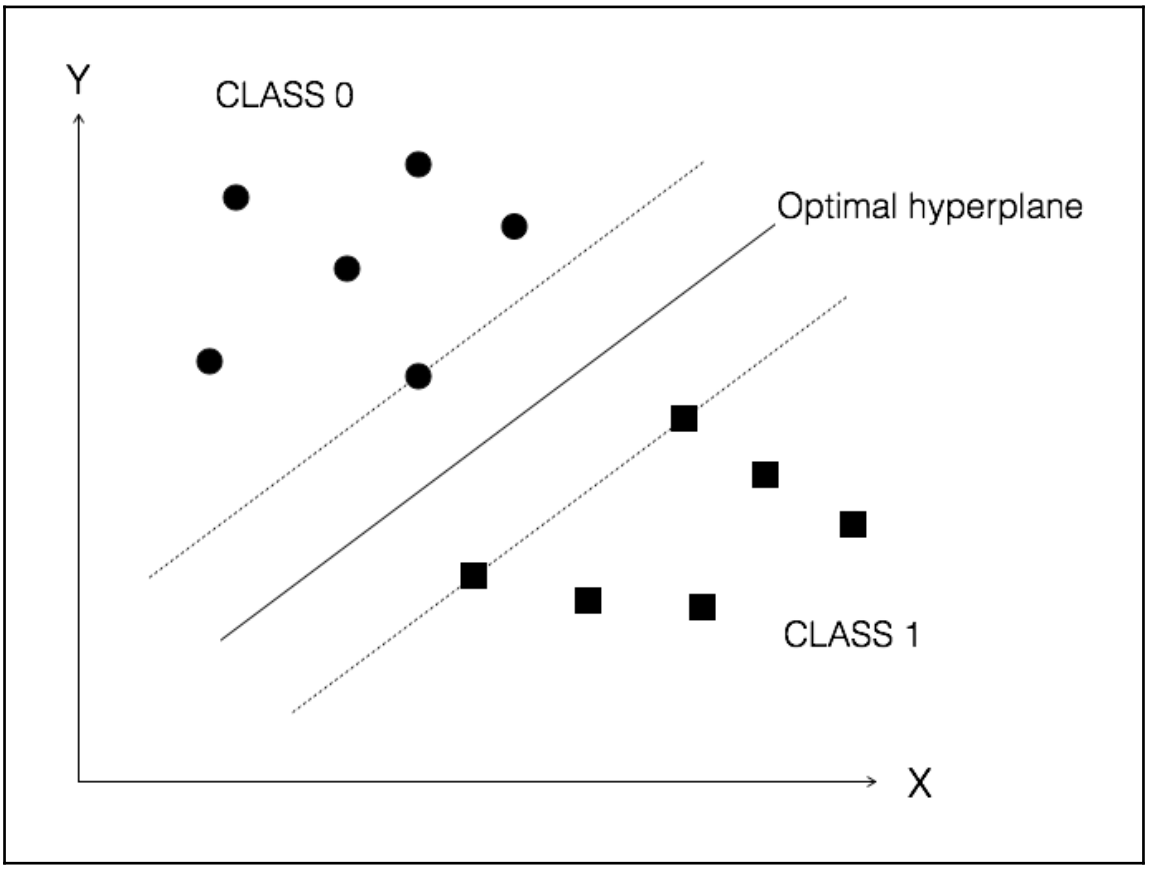


Figure 1
Confusion matrix
True labels
Predicted labels
0
1
2
3
4
0
1
2
3
4
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0

	precision	recall	f1-score	support
Class-0	1.00	0.67	0.80	3
Class-1	0.33	1.00	0.50	1
Class-2	1.00	1.00	1.00	2
Class-3	0.67	0.67	0.67	3
Class-4	1.00	0.50	0.67	2
avg / total	0.85	0.73	0.75	11



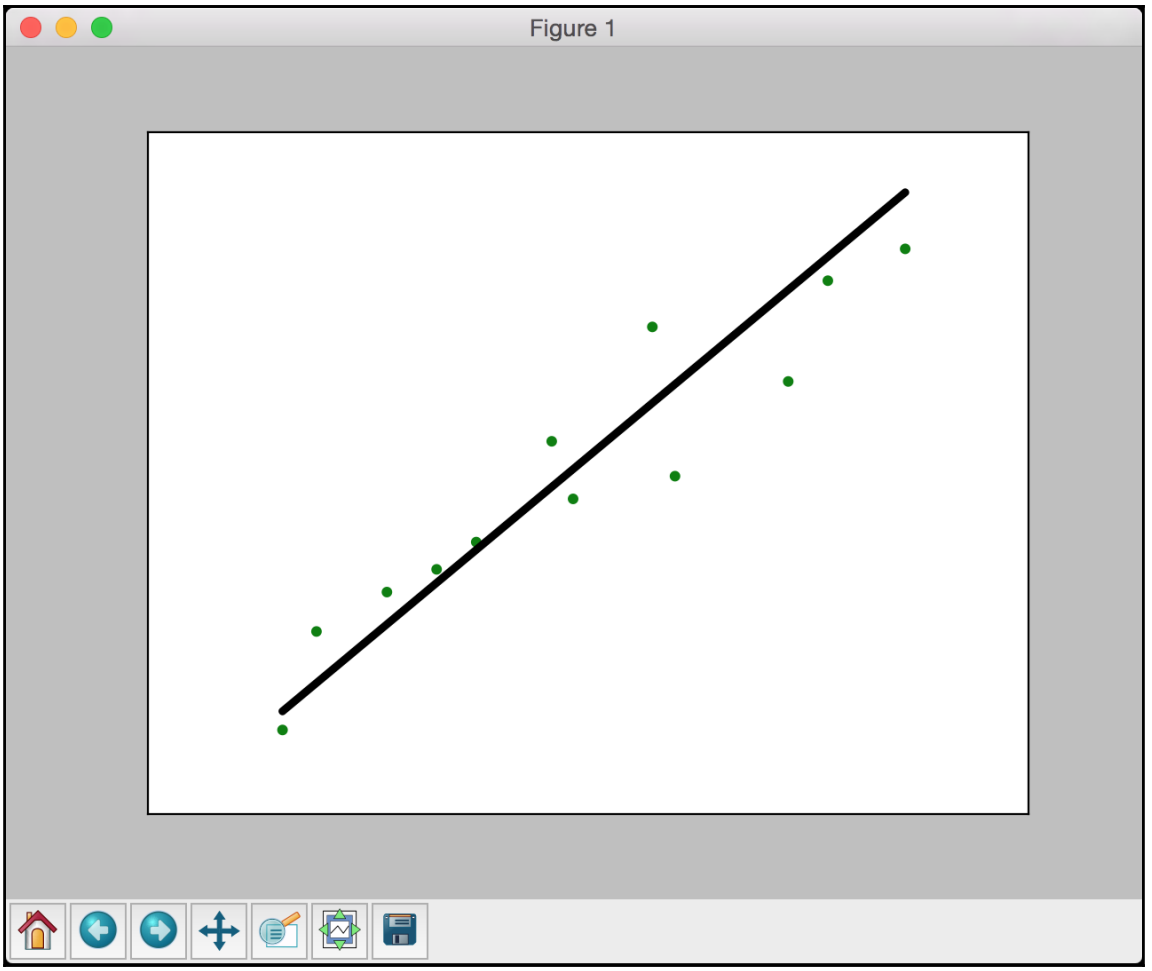
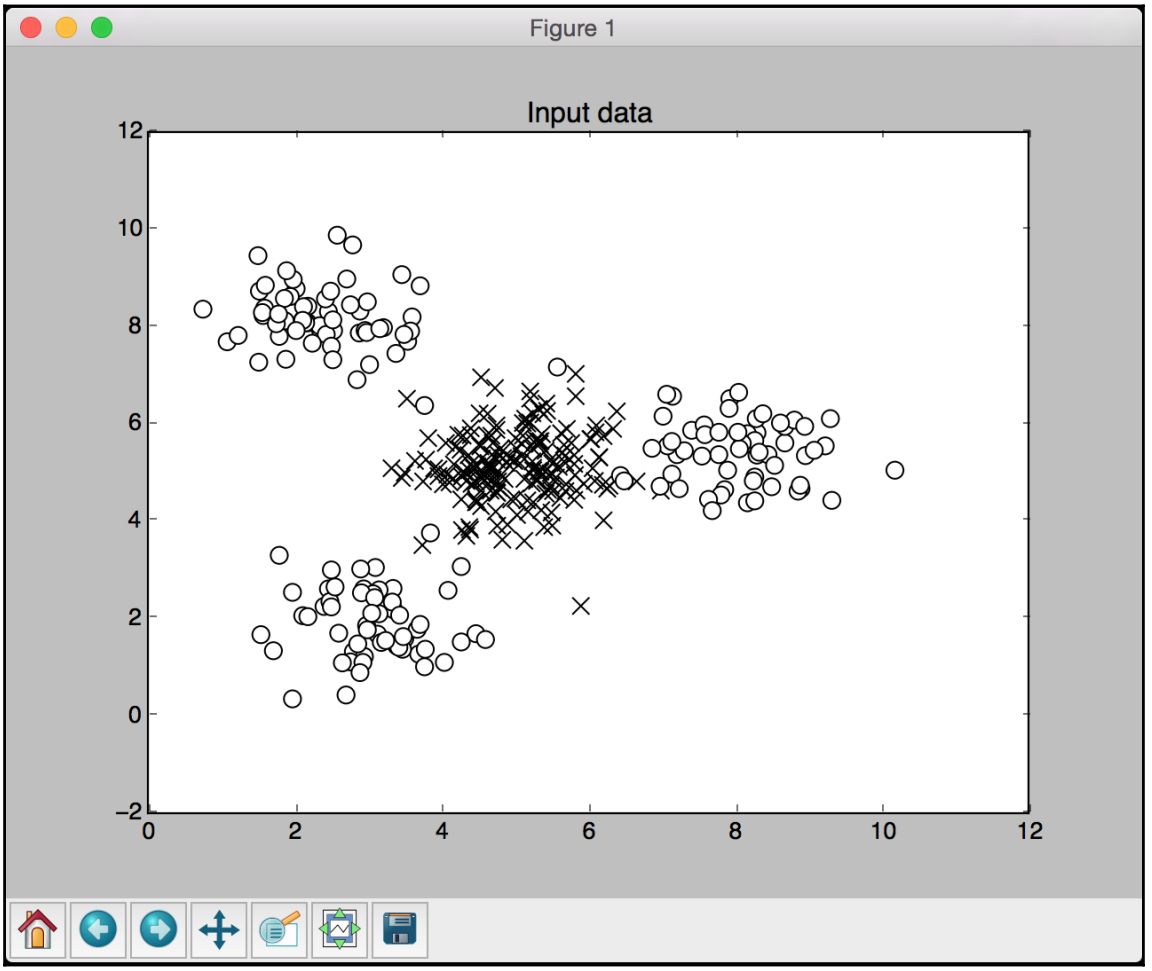


Figure 1

Chapter 3 : Predictive Analytics with Ensemble Learning



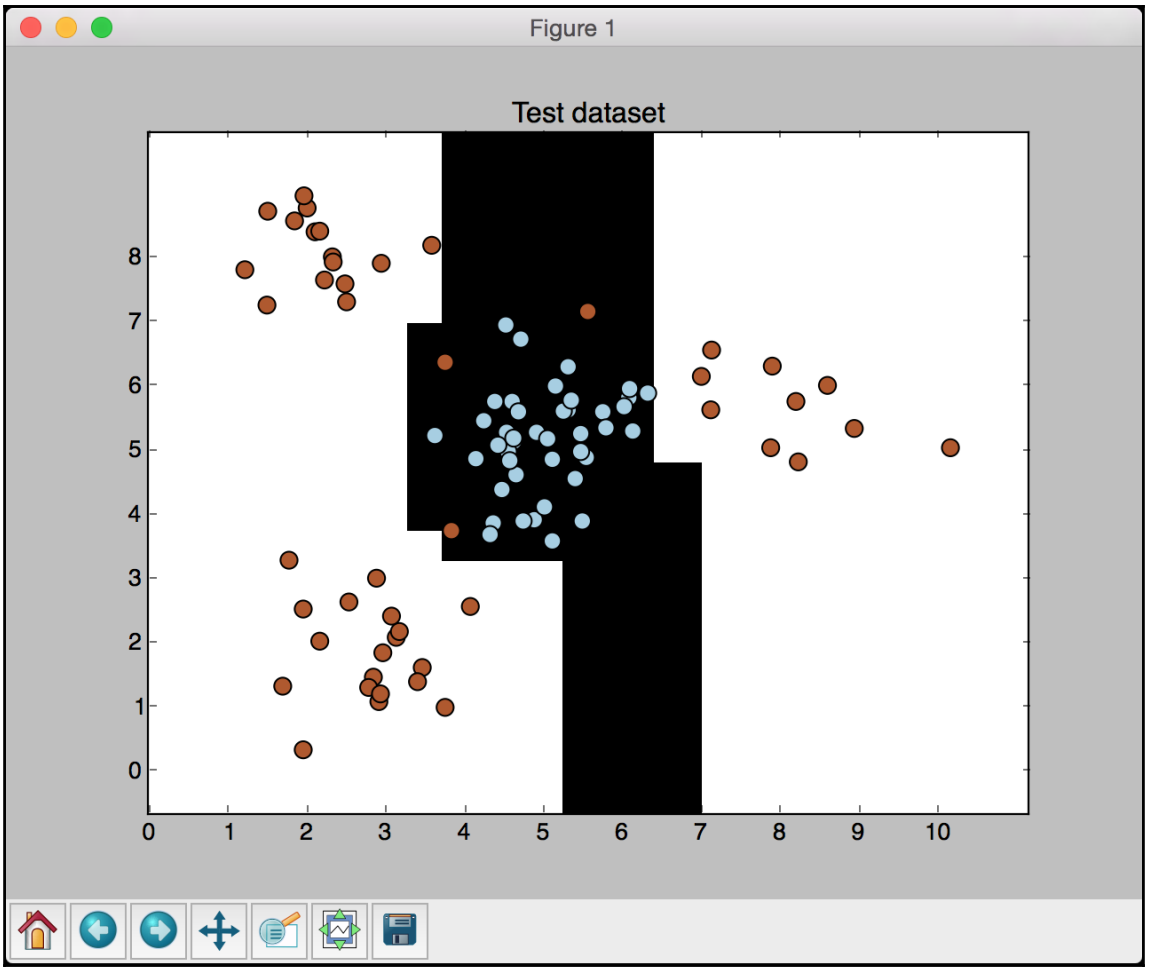


Figure 1
Test dataset
0
1
2
3
4
5
6
7
8
9
10

```
########################################

Classifier performance on training dataset

             precision    recall  f1-score   support

    Class-0       0.99      1.00      1.00       137
    Class-1       1.00      0.99      1.00       133

avg / total       1.00      1.00      1.00       270

########################################

########################################

Classifier performance on test dataset

             precision    recall  f1-score   support

    Class-0       0.93      1.00      0.97        43
    Class-1       1.00      0.94      0.97        47

avg / total       0.97      0.97      0.97        90

########################################
```

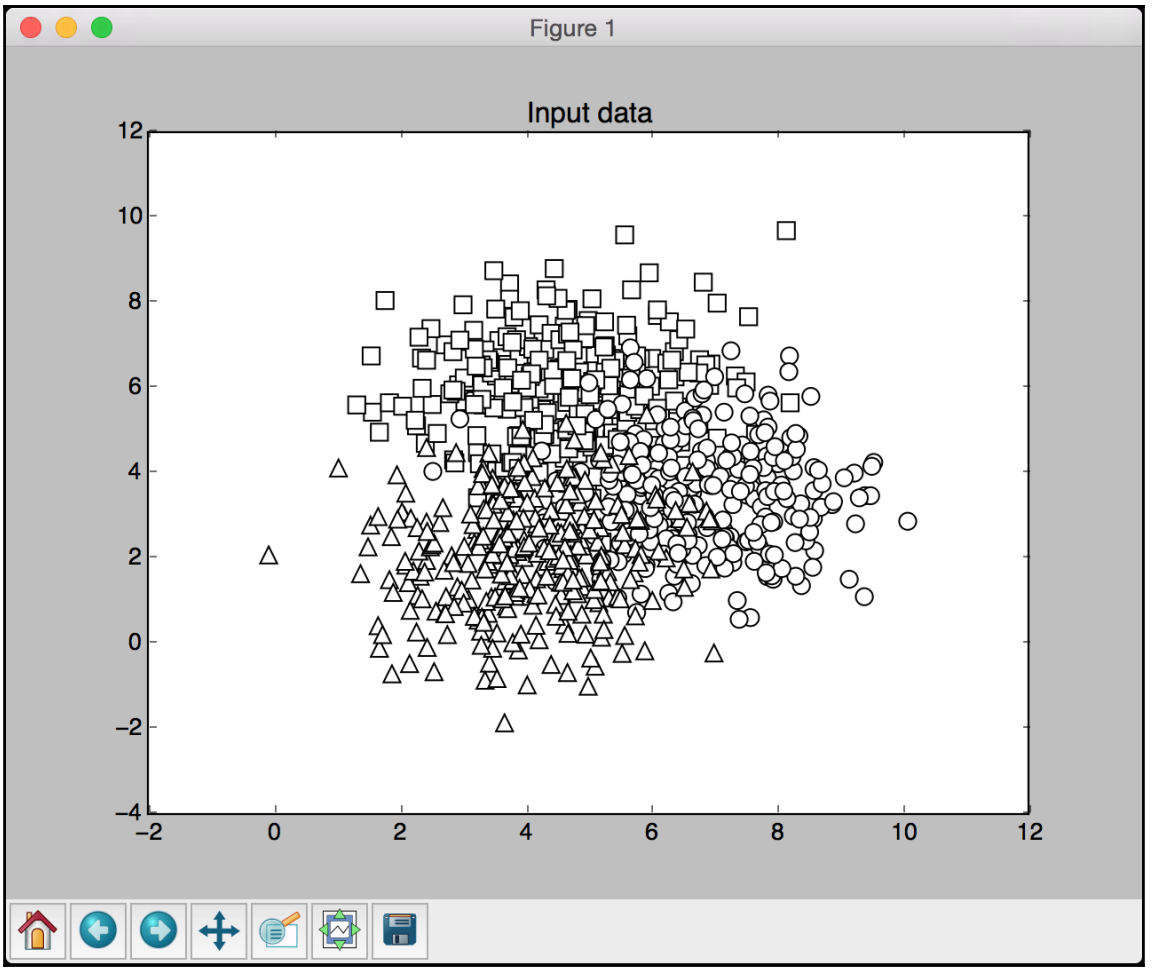


Figure 1
Input data
12
10
8
6
4
2
0
−2
−4
−2
0
2
4
6
8
10
12

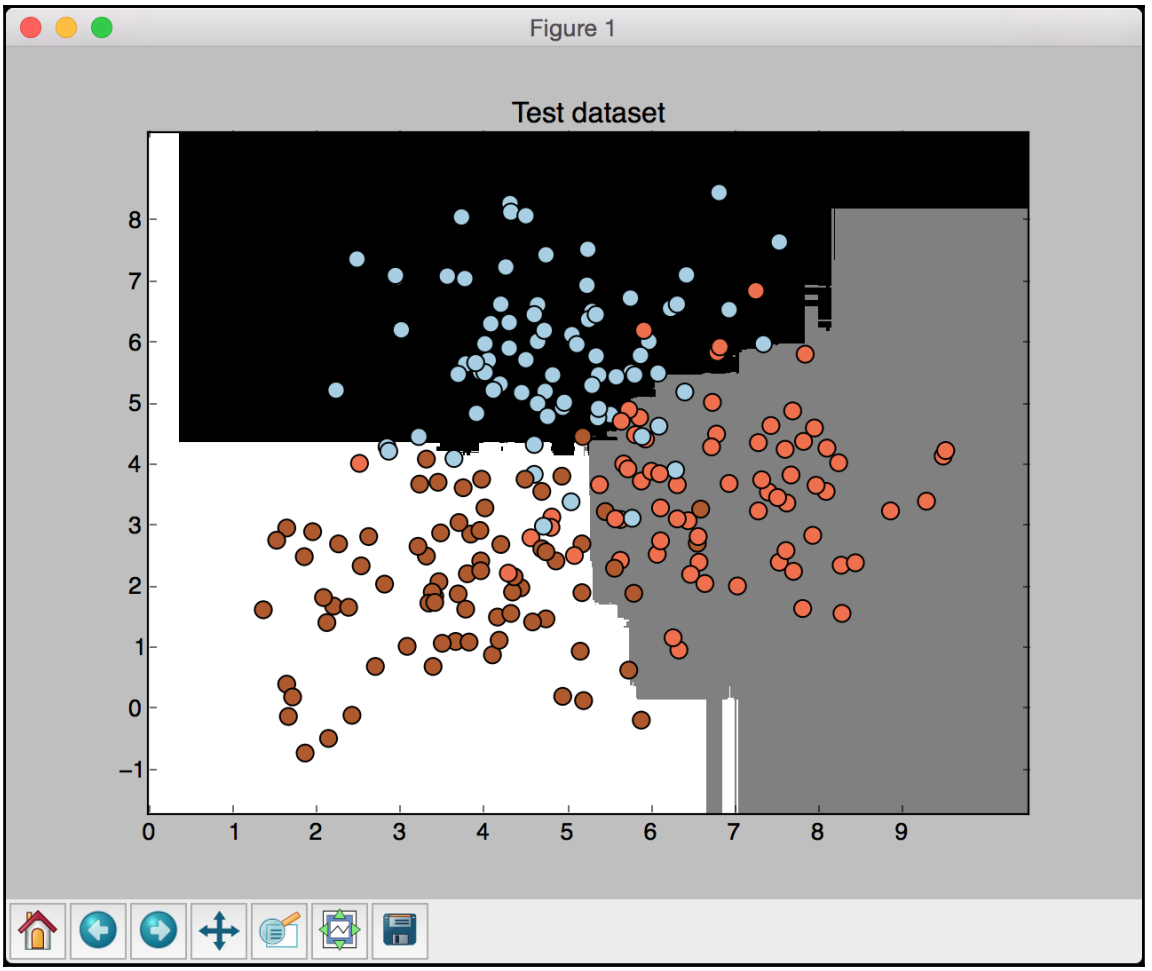


Figure 1
Test dataset
8
7
6
5
4
3
2
1
0
−1
0
1
2
3
4
5
6
7
8
9

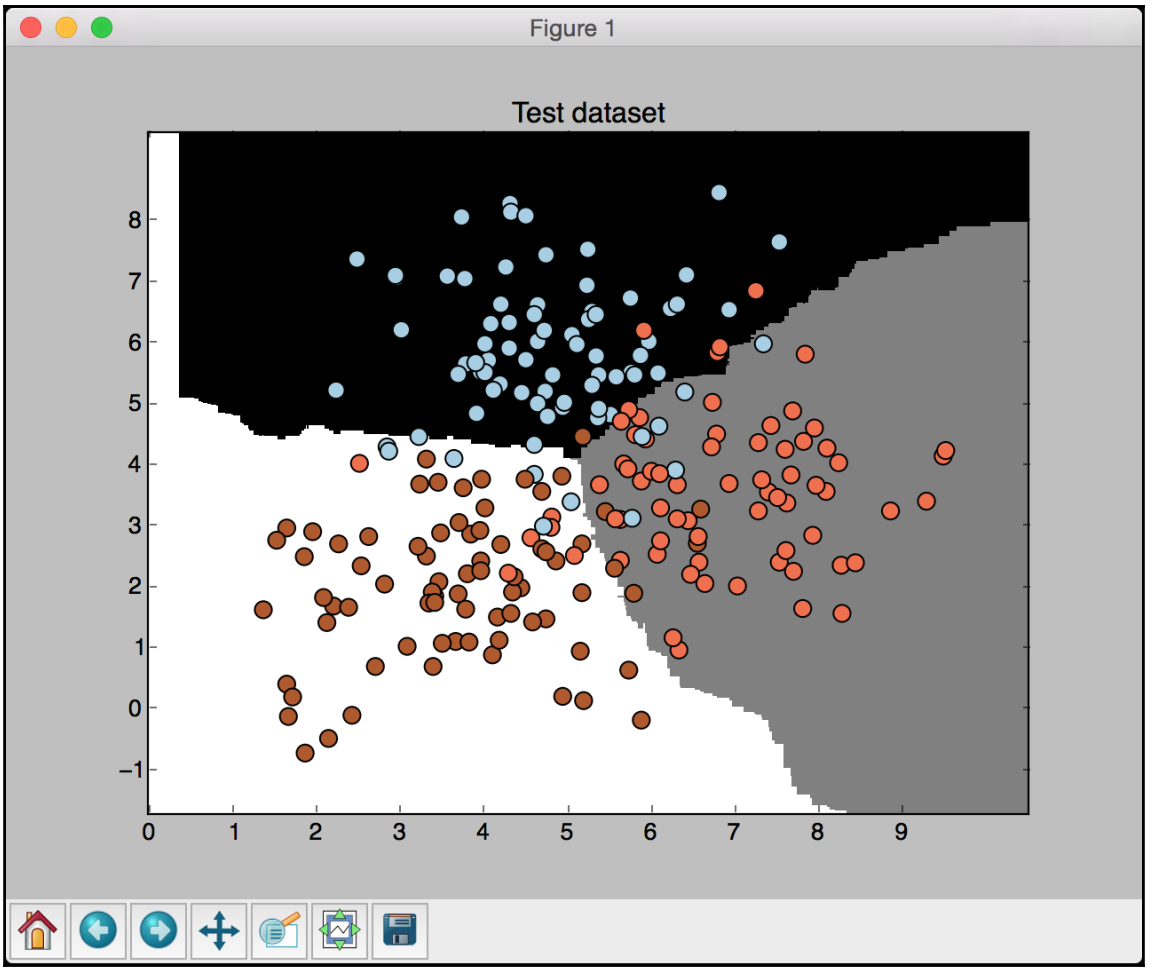


Figure 1
Test dataset
8
7
6
5
4
3
2
1
0
−1
0
1
2
3
4
5
6
7
8
9

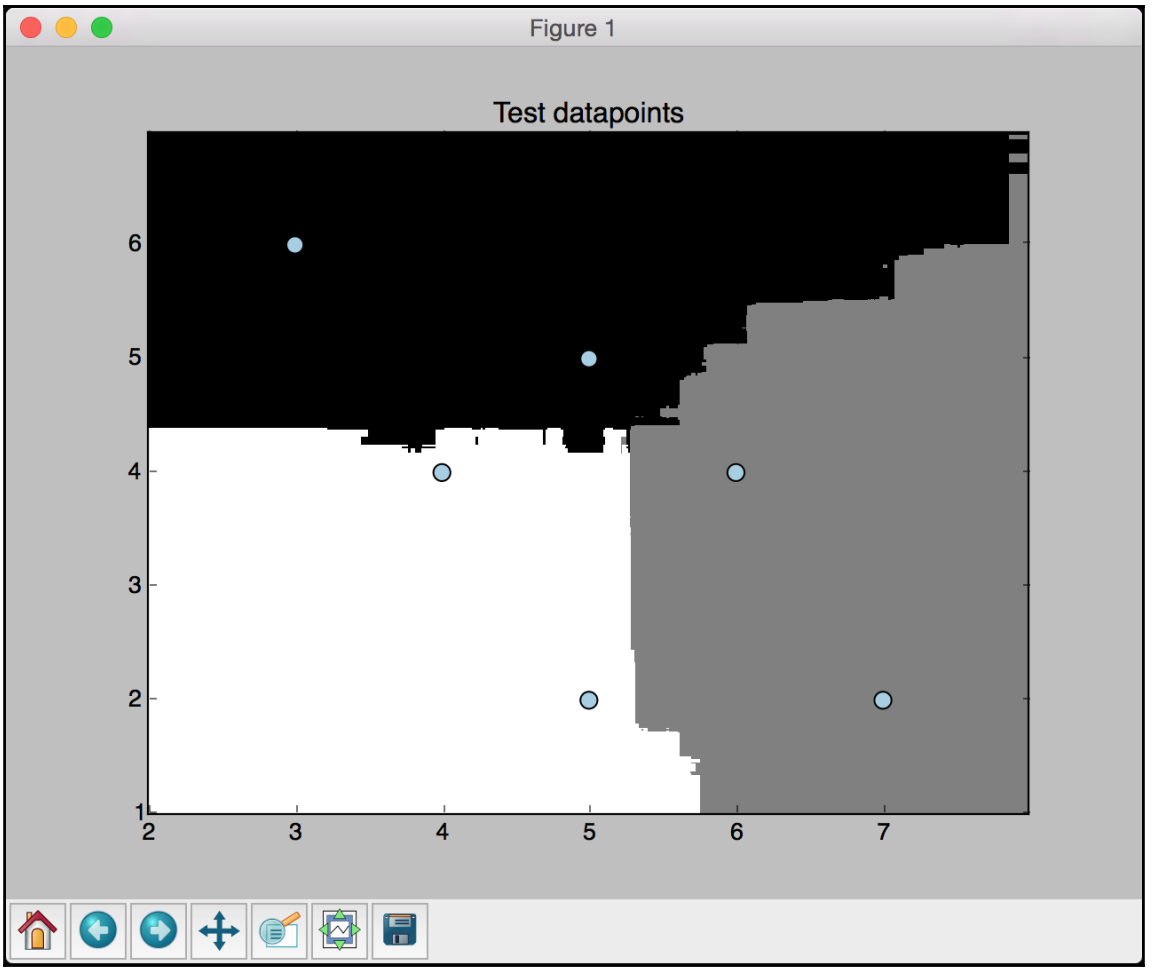


Figure 1
Test datapoints
6
5
4
3
2
1
2
3
4
5
6
7

```
Datapoint: [5 5]
Probabilities: [ 0.81427532  0.08639273  0.09933195]
Predicted class: Class-0

Datapoint: [3 6]
Probabilities: [ 0.93574458  0.02465345  0.03960197]
Predicted class: Class-0

Datapoint: [6 4]
Probabilities: [ 0.12232404  0.7451078   0.13256816]
Predicted class: Class-1

Datapoint: [7 2]
Probabilities: [ 0.05415465  0.70660226  0.23924309]
Predicted class: Class-1

Datapoint: [4 4]
Probabilities: [ 0.20594744  0.15523491  0.63881765]
Predicted class: Class-2

Datapoint: [5 2]
Probabilities: [ 0.05403583  0.0931115   0.85285267]
Predicted class: Class-2
```

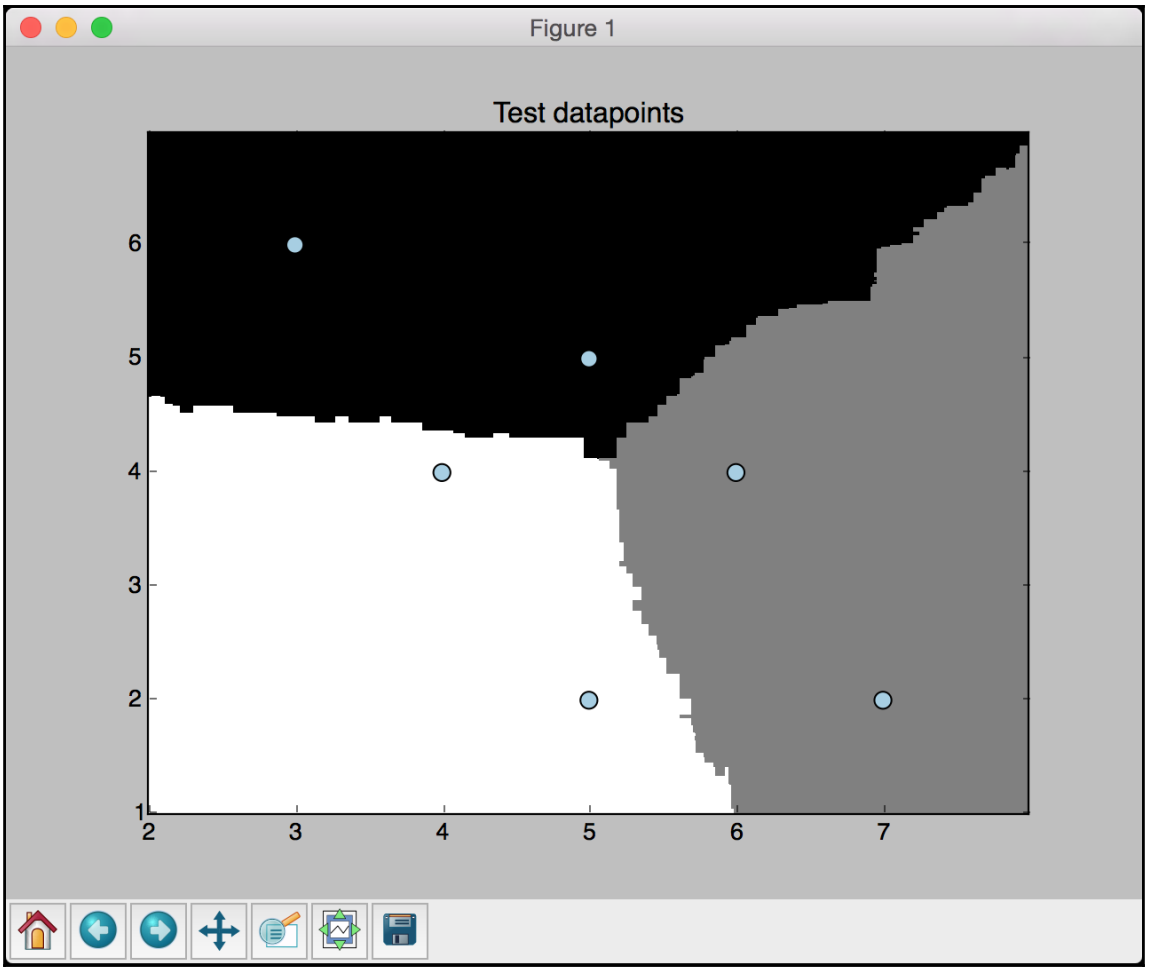
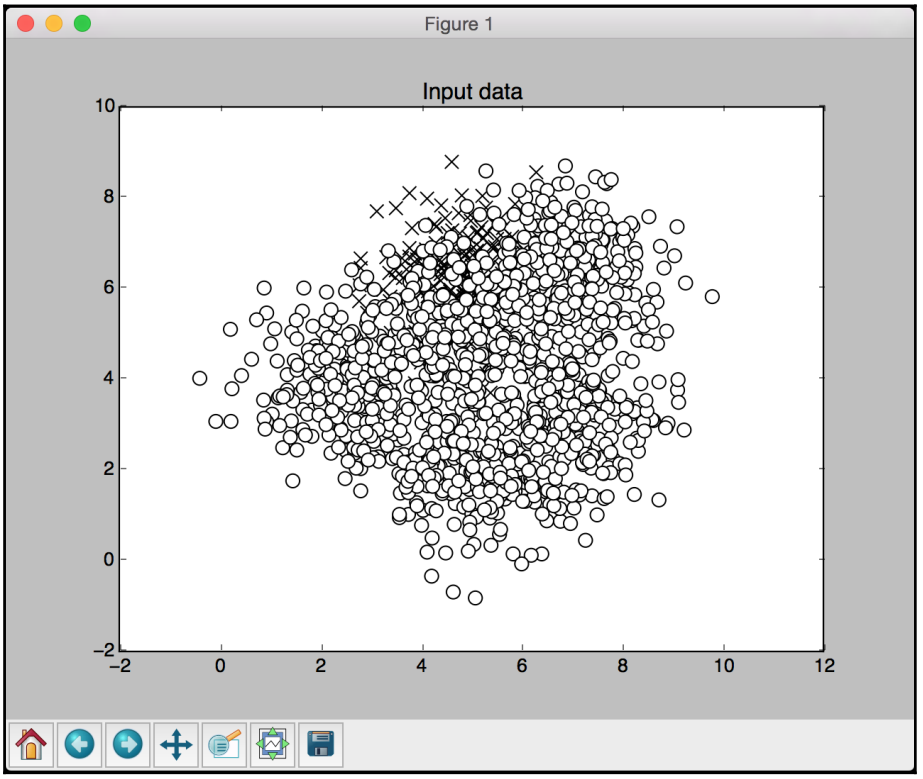
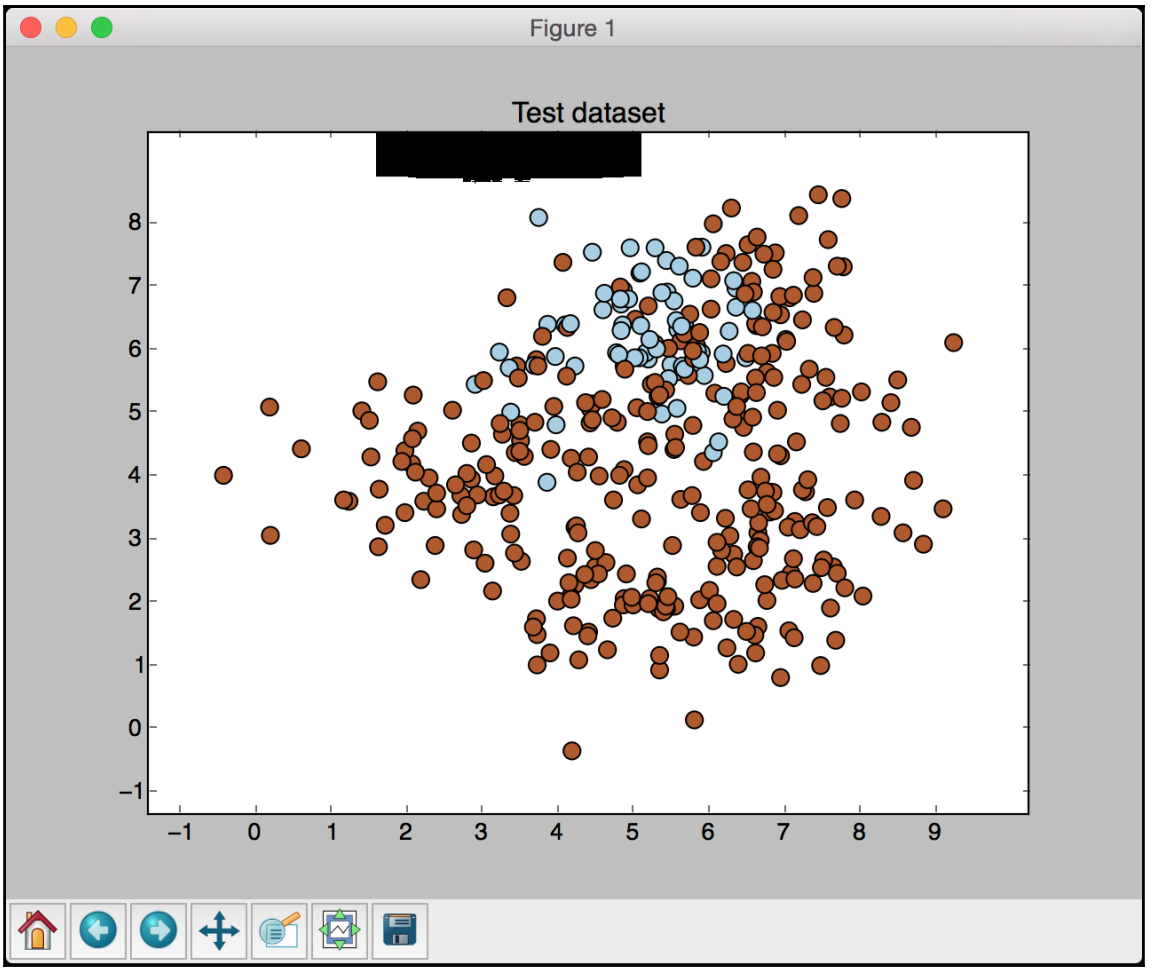


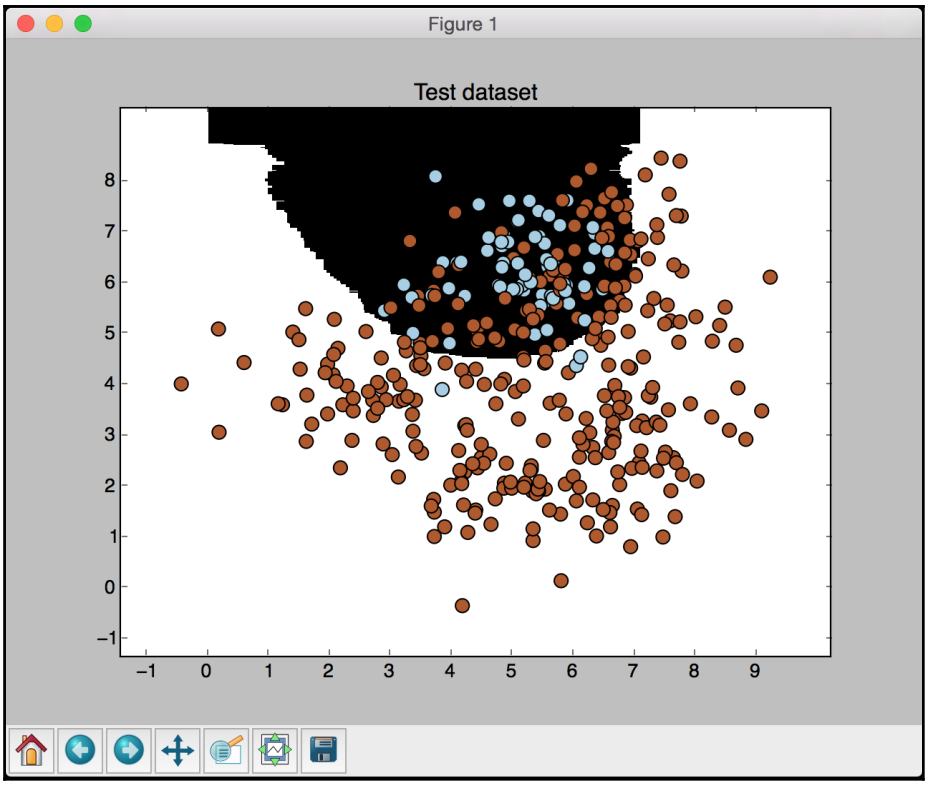
Figure 1
Test datapoints
6
5
4
3
2
1
2
3
4
5
6
7

```
Datapoint: [5 5]
Probabilities: [ 0.48904419  0.28020114  0.23075467]
Predicted class: Class-0

Datapoint: [3 6]
Probabilities: [ 0.66707383  0.12424406  0.20868211]
Predicted class: Class-0

Datapoint: [6 4]
Probabilities: [ 0.25788769  0.49535144  0.24676087]
Predicted class: Class-1

Datapoint: [7 2]
Probabilities: [ 0.10794013  0.6246677   0.26739217]
Predicted class: Class-1

Datapoint: [4 4]
Probabilities: [ 0.33383778  0.21495182  0.45121039]
Predicted class: Class-2

Datapoint: [5 2]
Probabilities: [ 0.18671115  0.28760896  0.52567989]
Predicted class: Class-2
```






```
########################################

Classifier performance on test dataset

             precision    recall  f1-score   support

    Class-0       0.00      0.00      0.00        69
    Class-1       0.82      1.00      0.90       306

avg / total       0.67      0.82      0.73       375

########################################
```




```
########################################

Classifier performance on test dataset

             precision    recall  f1-score   support

    Class-0       0.45      0.94      0.61        69
    Class-1       0.98      0.74      0.84       306

avg / total       0.88      0.78      0.80       375

########################################
```

```
##### Searching optimal parameters for precision_weighted

Grid scores for the parameter grid:
{'n_estimators': 100, 'max_depth': 2} --> 0.847
{'n_estimators': 100, 'max_depth': 4} --> 0.841
{'n_estimators': 100, 'max_depth': 7} --> 0.844
{'n_estimators': 100, 'max_depth': 12} --> 0.836
{'n_estimators': 100, 'max_depth': 16} --> 0.818
{'n_estimators': 25, 'max_depth': 4} --> 0.846
{'n_estimators': 50, 'max_depth': 4} --> 0.84
{'n_estimators': 100, 'max_depth': 4} --> 0.841
{'n_estimators': 250, 'max_depth': 4} --> 0.845

Best parameters: {'n_estimators': 100, 'max_depth': 2}

Performance report:

             precision    recall  f1-score   support

        0.0       0.94      0.81      0.87        79
        1.0       0.81      0.86      0.83        70
        2.0       0.83      0.91      0.87        76

avg / total       0.86      0.86      0.86       225
```

```
##### Searching optimal parameters for recall_weighted

Grid scores for the parameter grid:
{'n_estimators': 100, 'max_depth': 2} --> 0.84
{'n_estimators': 100, 'max_depth': 4} --> 0.837
{'n_estimators': 100, 'max_depth': 7} --> 0.841
{'n_estimators': 100, 'max_depth': 12} --> 0.834
{'n_estimators': 100, 'max_depth': 16} --> 0.816
{'n_estimators': 25, 'max_depth': 4} --> 0.843
{'n_estimators': 50, 'max_depth': 4} --> 0.836
{'n_estimators': 100, 'max_depth': 4} --> 0.837
{'n_estimators': 250, 'max_depth': 4} --> 0.841

Best parameters: {'n_estimators': 25, 'max_depth': 4}

Performance report:

             precision    recall  f1-score   support

        0.0       0.93      0.84      0.88        79
        1.0       0.85      0.86      0.85        70
        2.0       0.84      0.92      0.88        76

avg / total       0.87      0.87      0.87       225
```

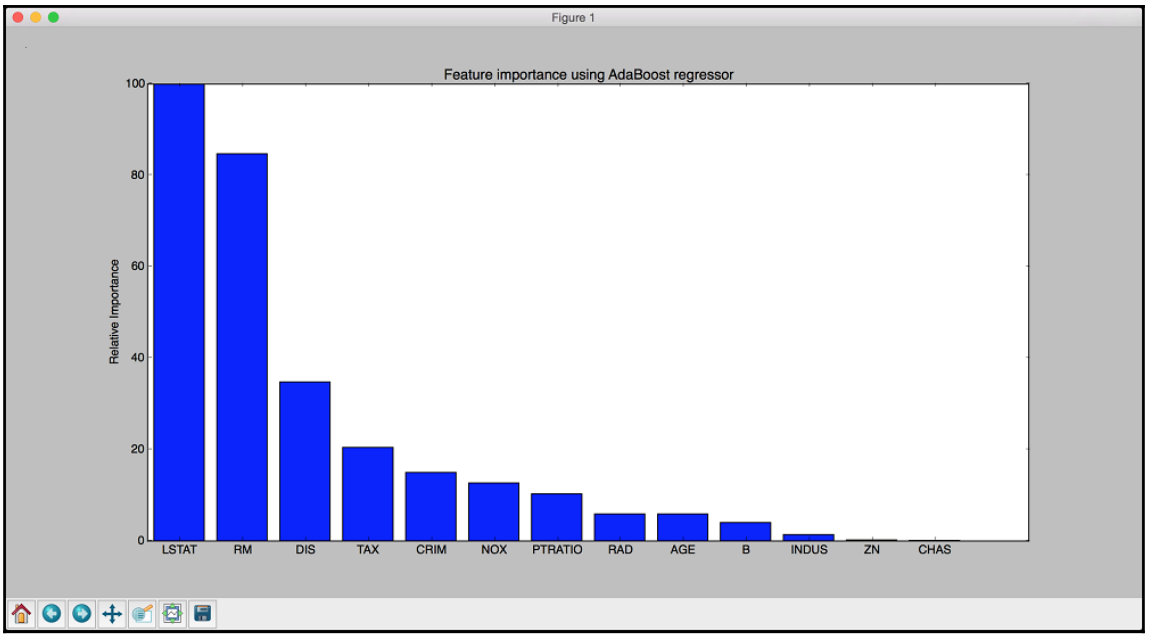
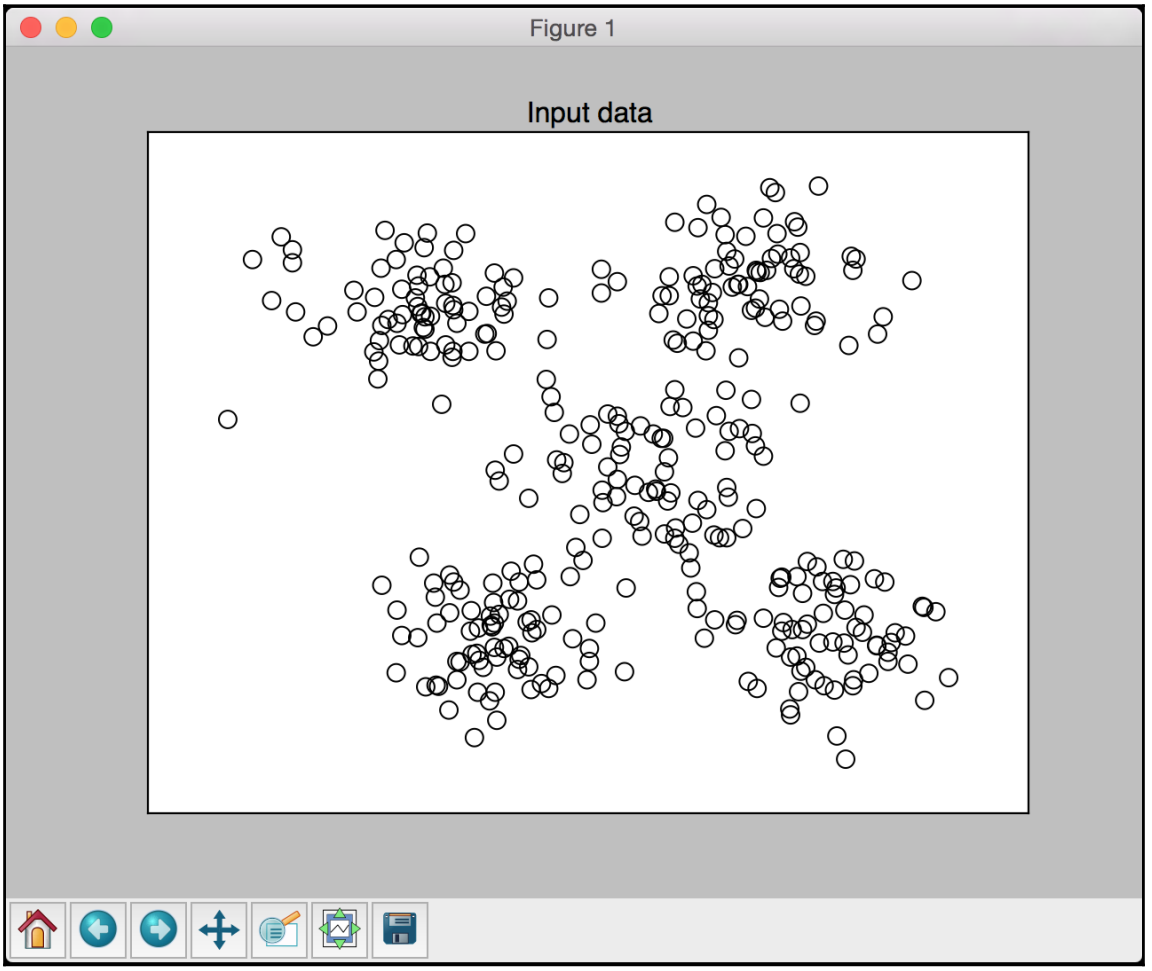


Figure 1
Feature importance using AdaBoost regressor
Relative Importance
100
80
60
40
20
0
LSTAT
RM
DIS
TAX
CRIM
NOX
PTRATIO
RAD
AGE
B
INDUS
ZN
CHAS

Chapter 4: Detecting Patterns with Unsupervised Learning



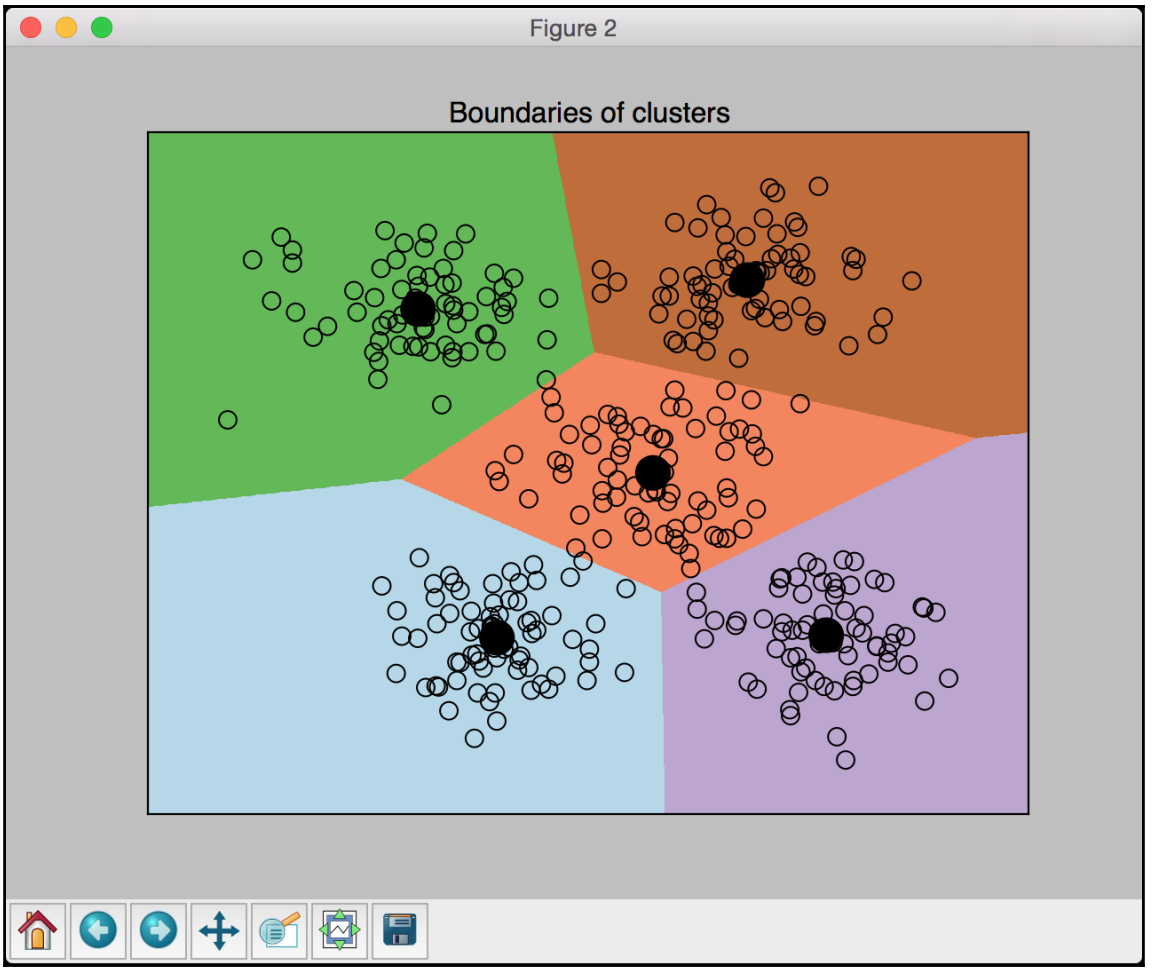
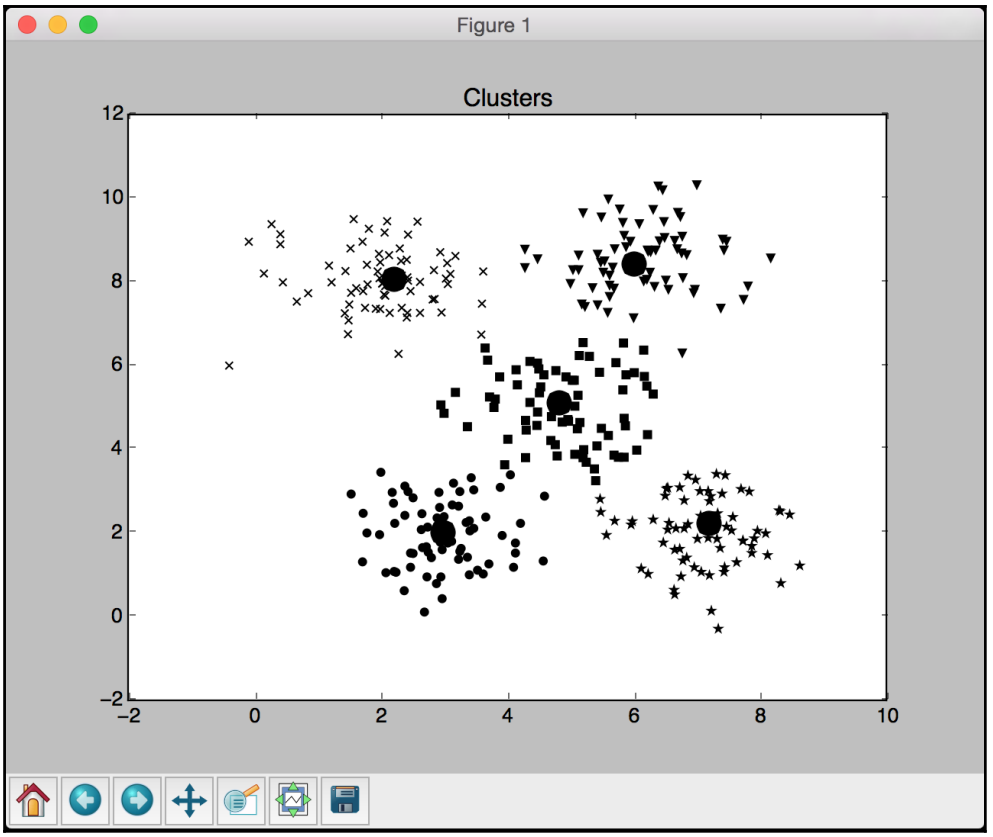


Figure 2
Boundaries of clusters




```
Centers of clusters:
 [[ 2.95568966  1.95775862]
 [ 7.17563636  2.18145455]
 [ 2.17603774  8.03283019]
 [ 5.97960784  8.39078431]
 [ 4.81044444  5.07111111]]

Number of clusters in input data = 5
```

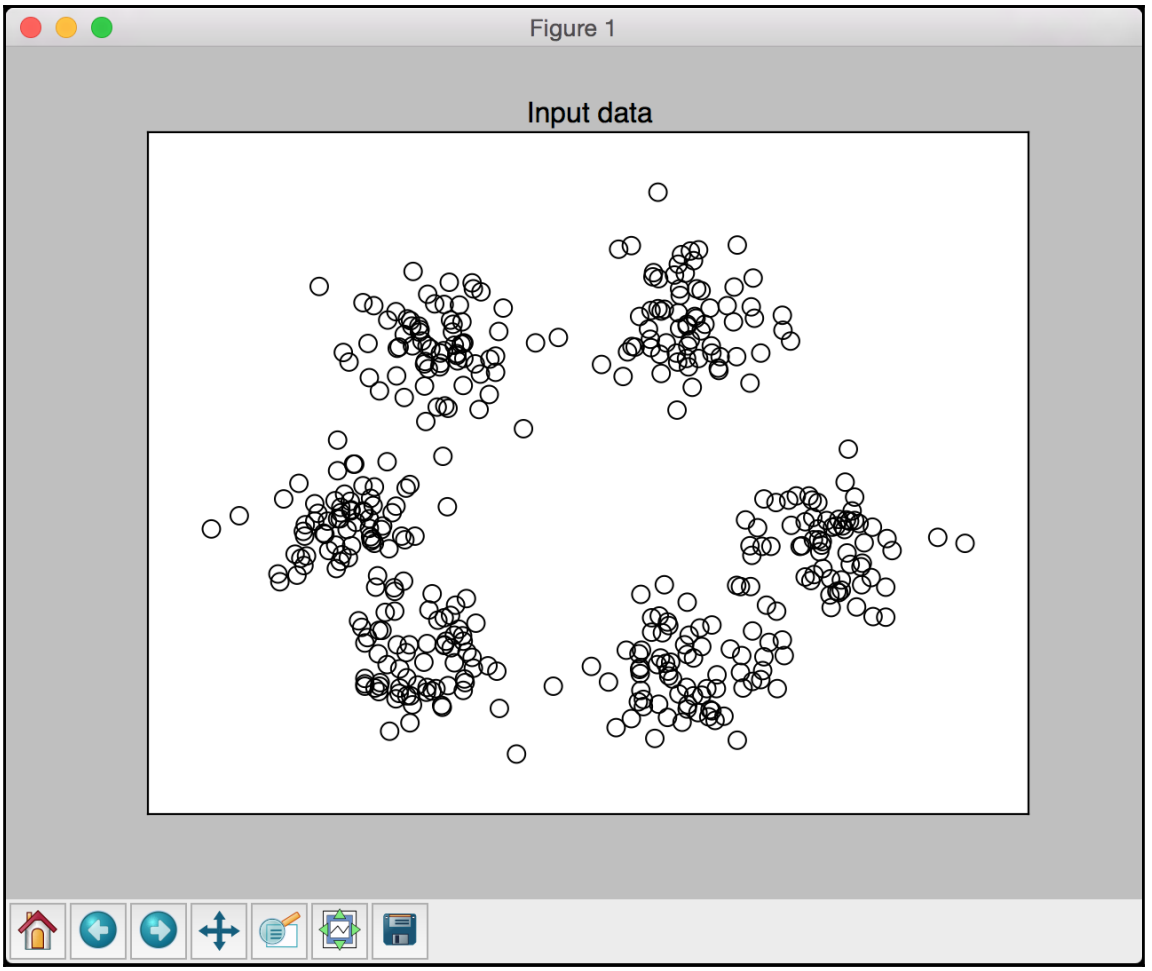


Figure 1
Input data

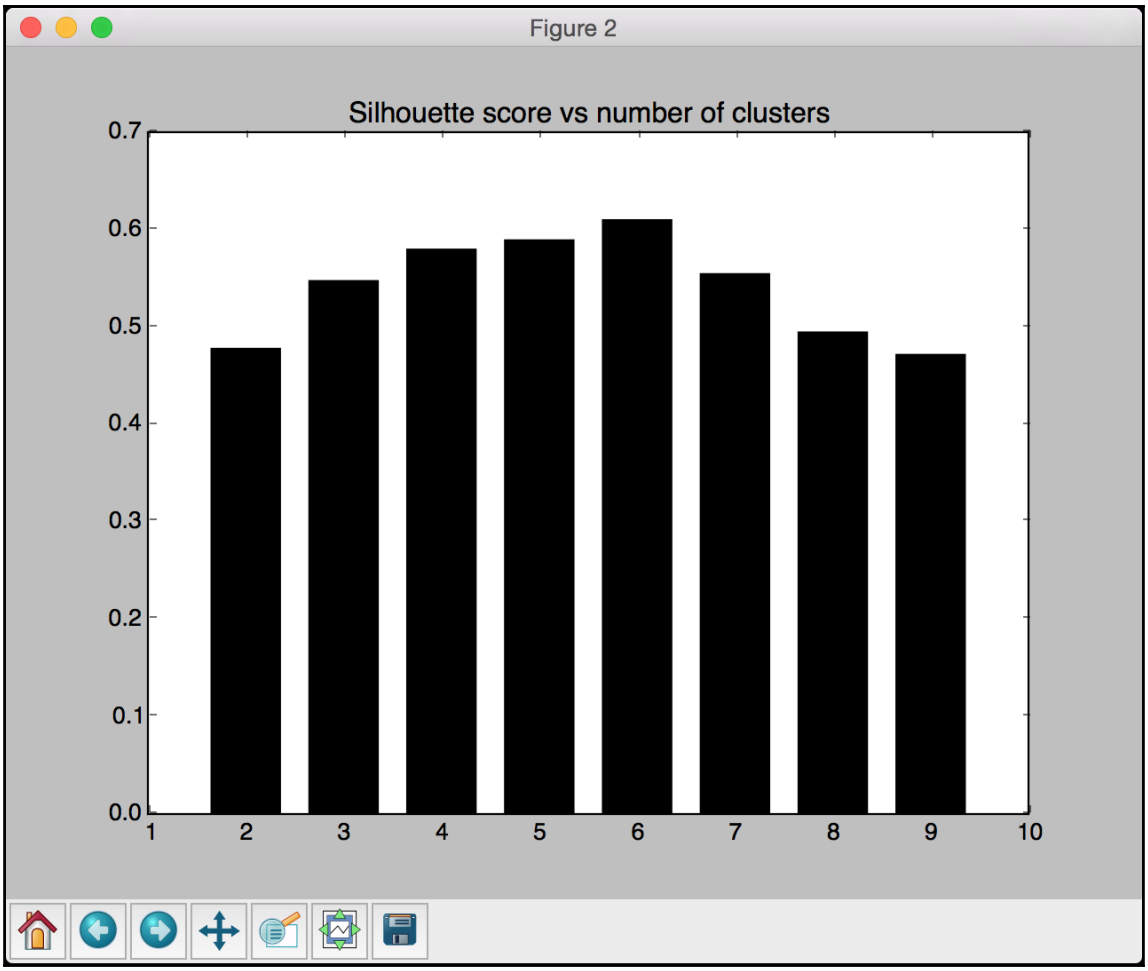
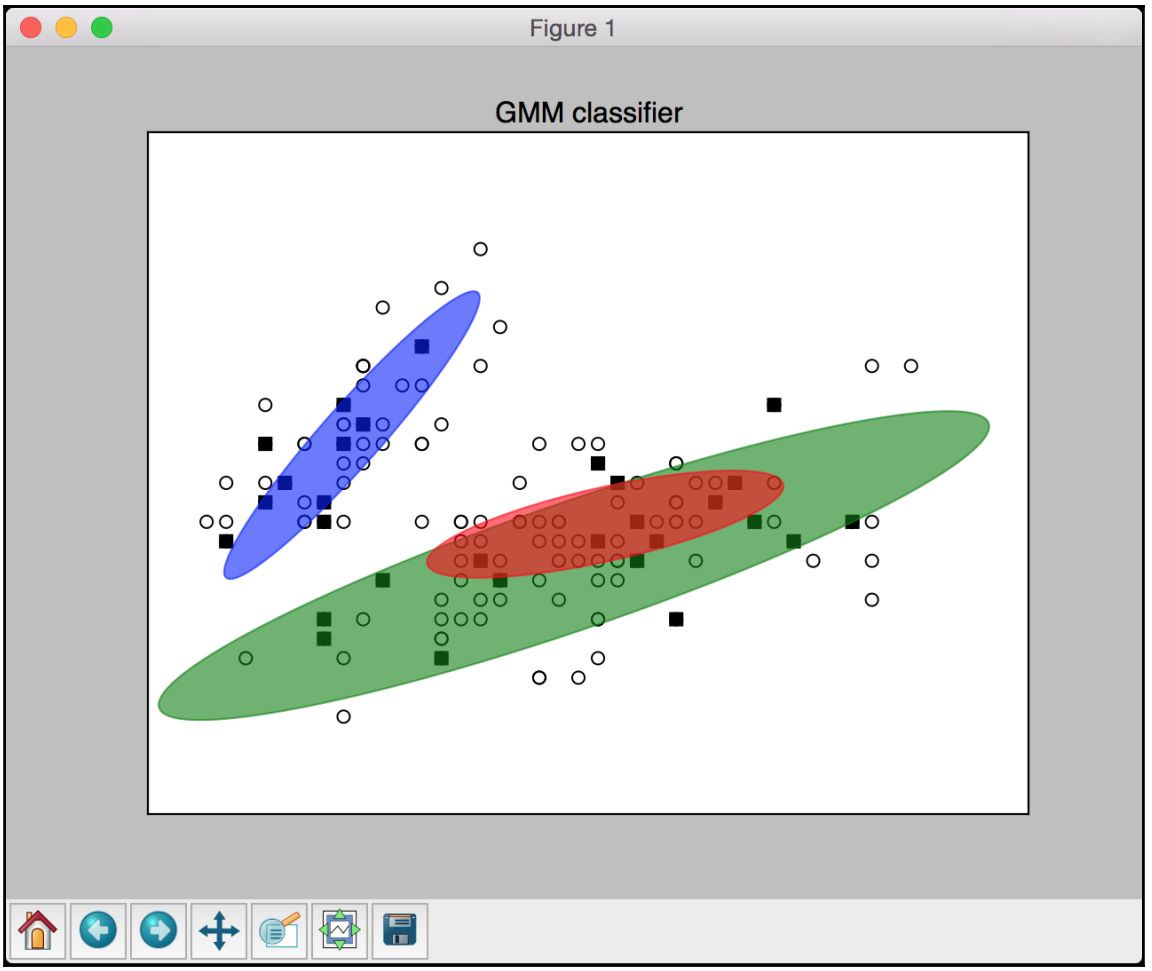


Figure 2
Silhouette score vs number of clusters
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
6
7
8
9
10

```
Number of clusters = 2
Silhouette score = 0.477626248705

Number of clusters = 3
Silhouette score = 0.547174241173

Number of clusters = 4
Silhouette score = 0.579480188969

Number of clusters = 5
Silhouette score = 0.589003263565

Number of clusters = 6
Silhouette score = 0.609690411895

Number of clusters = 7
Silhouette score = 0.554310234032

Number of clusters = 8
Silhouette score = 0.494433661954

Number of clusters = 9
Silhouette score = 0.471414689437

Optimal number of clusters = 6
```




```
Clustering of stocks based on difference in opening and closing quotes:

Cluster 1 ==> Kraft Foods
Cluster 2 ==> CVS, Walgreen
Cluster 3 ==> Amazon, Yahoo
Cluster 4 ==> Cablevision
Cluster 5 ==> Pfizer, Sanofi-Aventis, GlaxoSmithKline, Novartis
Cluster 6 ==> HP, General Electrics, 3M, Microsoft, Cisco, IBM, Texas instruments, Dell
Cluster 7 ==> Coca Cola, Kimberly-Clark, Pepsi, Procter Gamble, Kellogg, Colgate-Palmolive
Cluster 8 ==> Comcast, Wells Fargo, Xerox, Home Depot, Wal-Mart, Marriott, Navistar, DuPont de Nemours, A
merican express, Ryder, JPMorgan Chase, AIG, Time Warner, Bank of America, Goldman Sachs
Cluster 9 ==> Canon, Unilever, Mitsubishi, Apple, Mc Donalds, Boeing, Toyota, Caterpillar, Ford, Honda, S
AP, Sony
Cluster 10 ==> Valero Energy, Exxon, ConocoPhillips, Chevron, Total
Cluster 11 ==> Raytheon, General Dynamics, Lookheed Martin, Northrop Grumman
```

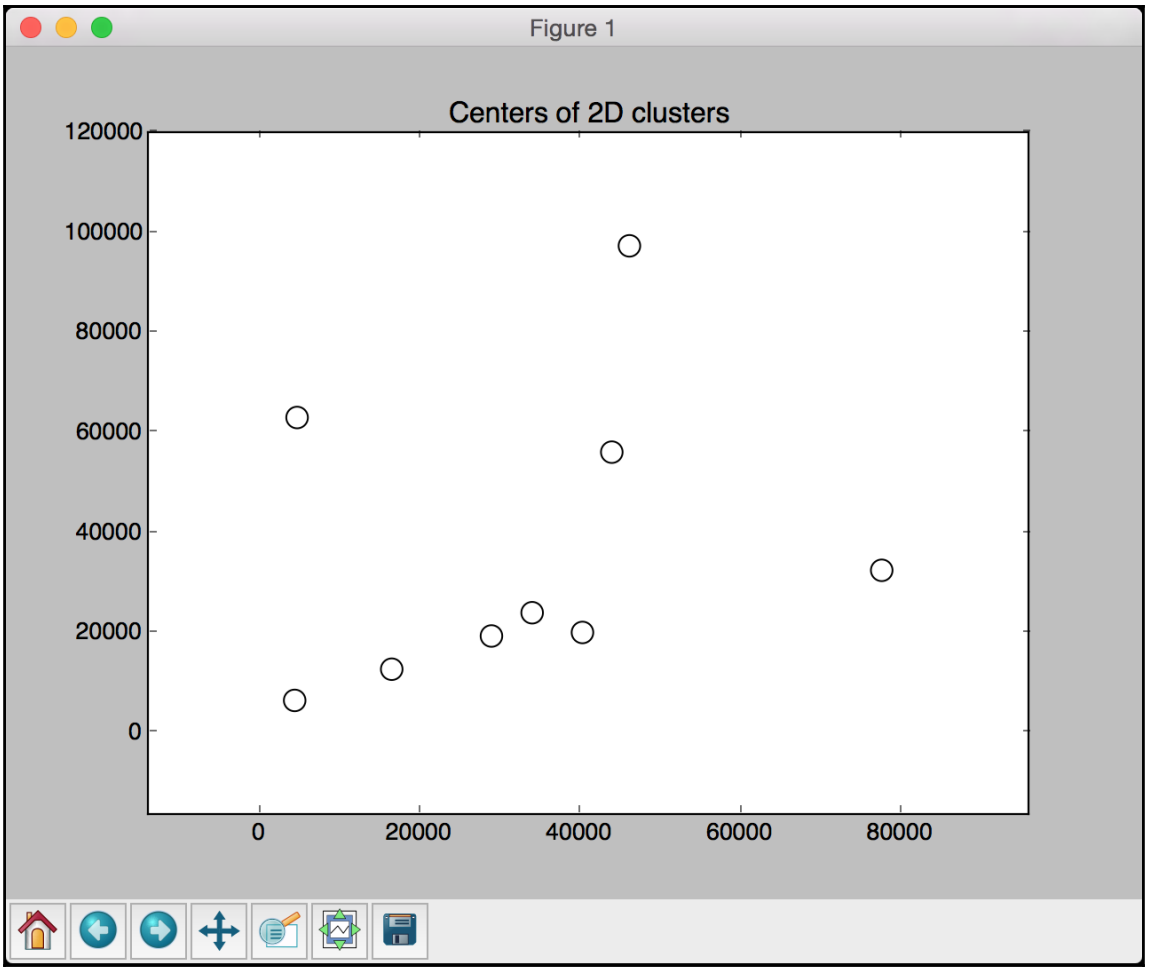


Figure 1
Centers of 2D clusters
120000
100000
80000
60000
40000
20000
0
0
20000
40000
60000
80000

```
Number of clusters in input data = 9

Centers of clusters:
Tsh     Tan     Hal     Tur     Tub     Swe
9823    4637    6539    2607    2228    1239
38589   44199   56158   5030    24674   4125
7852    4939    63081   134     40066   1332
35314   16745   12775   66900   1298    5613
22617   77873   32543   1005    21035   837
104972  29186   19415   16016   5060    9372
38741   40539   20120   35059   255     50710
28333   34263   24065   5575    4229    18076
14987   46397   97393   1127    37315   3235
```

Chapter 5: Building Recommender Systems

```
Predicted output:
 [1 2 2 0 2 2 0 2 0 1 2 0 2 1 0 0 2 2 2 1 0 2 0 1 2 1 1 1 0 0 1 2 1 0 0 0 2
 1 1 0 2 0 0 0 1 2 0 2 1 0 1 0 0 0 2 1 1 1 1 1 0 1 2 2 2 0 2 0 2 2 0 1 2 0
 2 0 2 0 1 0 2 2 1 1 1 2 0 0 0 0 2 2 0 2 1 1 2 0 1 1 2 1 1 0 1 0 2 2 2 0 0
 1 2 1 1 0 2 0 0 0 0 0 2 2 1 1 1 2 0 2 2 1 0 2 0 0 0 1 1 2 2 2 2 2 2 1 1 0
 2 0]

Score: 0.893333333333

Indices of selected features: 13, 15, 18, 19, 21, 23, 24
```

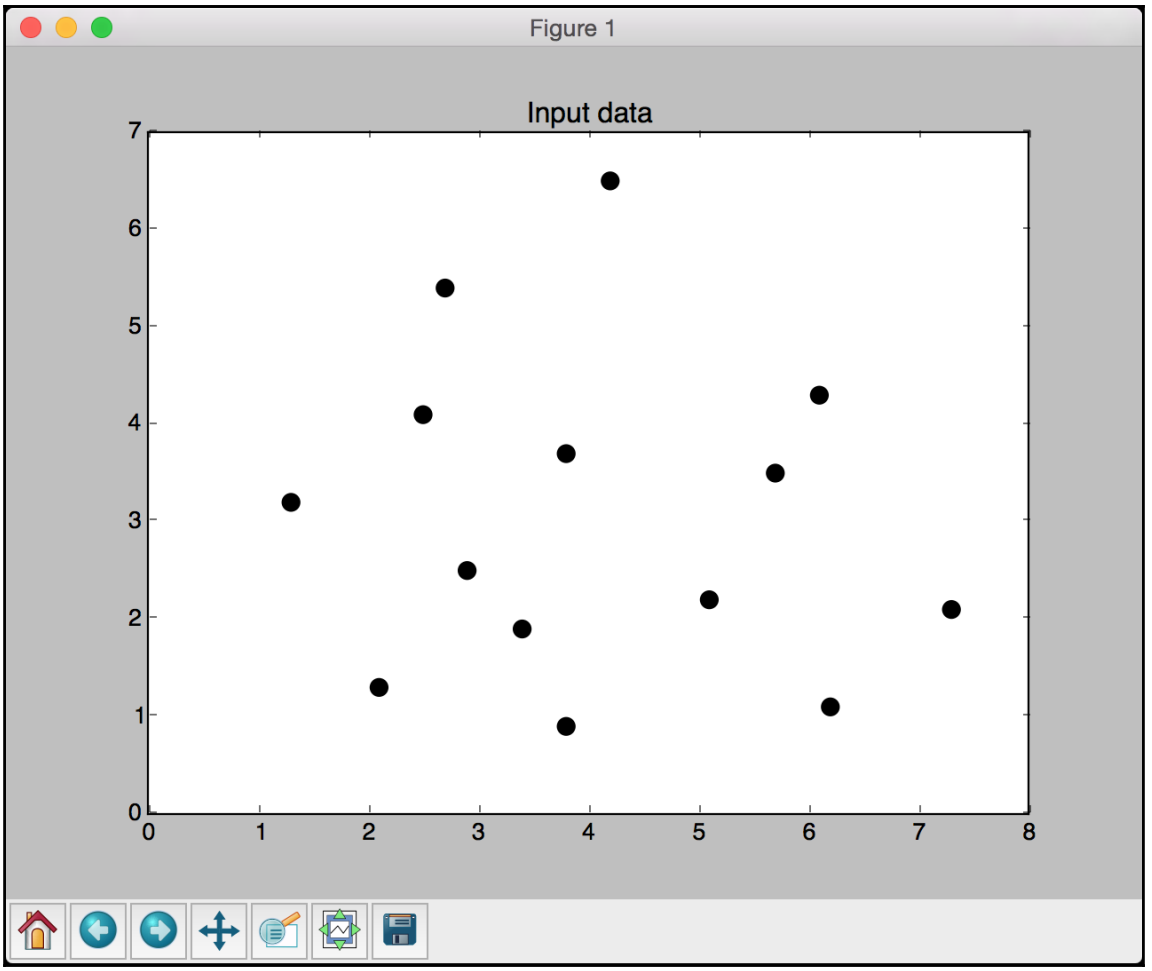
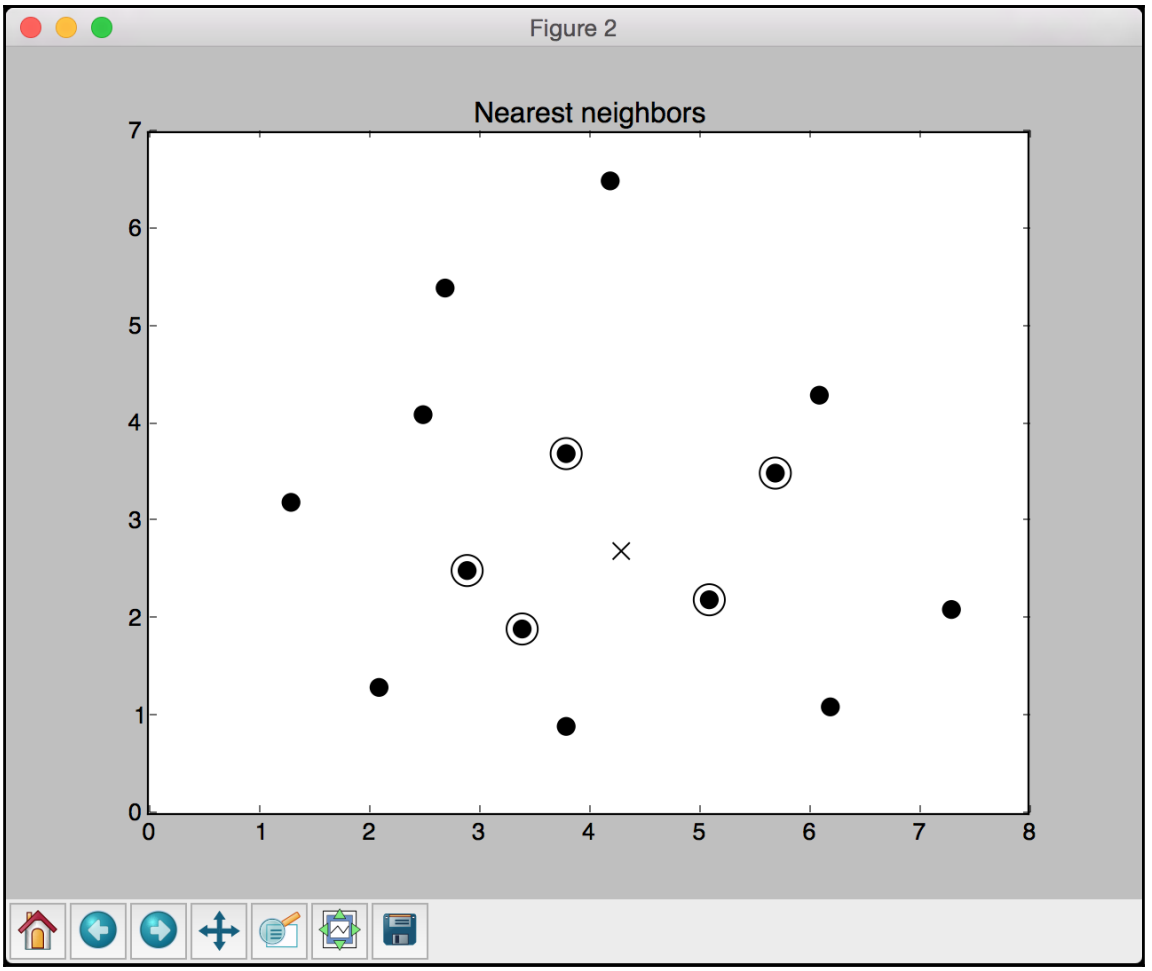


Figure 1
Input data
0
1
2
3
4
5
6
7
8




```
K Nearest Neighbors:
1 ==> [ 5.1  2.2]
2 ==> [ 3.8  3.7]
3 ==> [ 3.4  1.9]
4 ==> [ 2.9  2.5]
5 ==> [ 5.7  3.5]
```

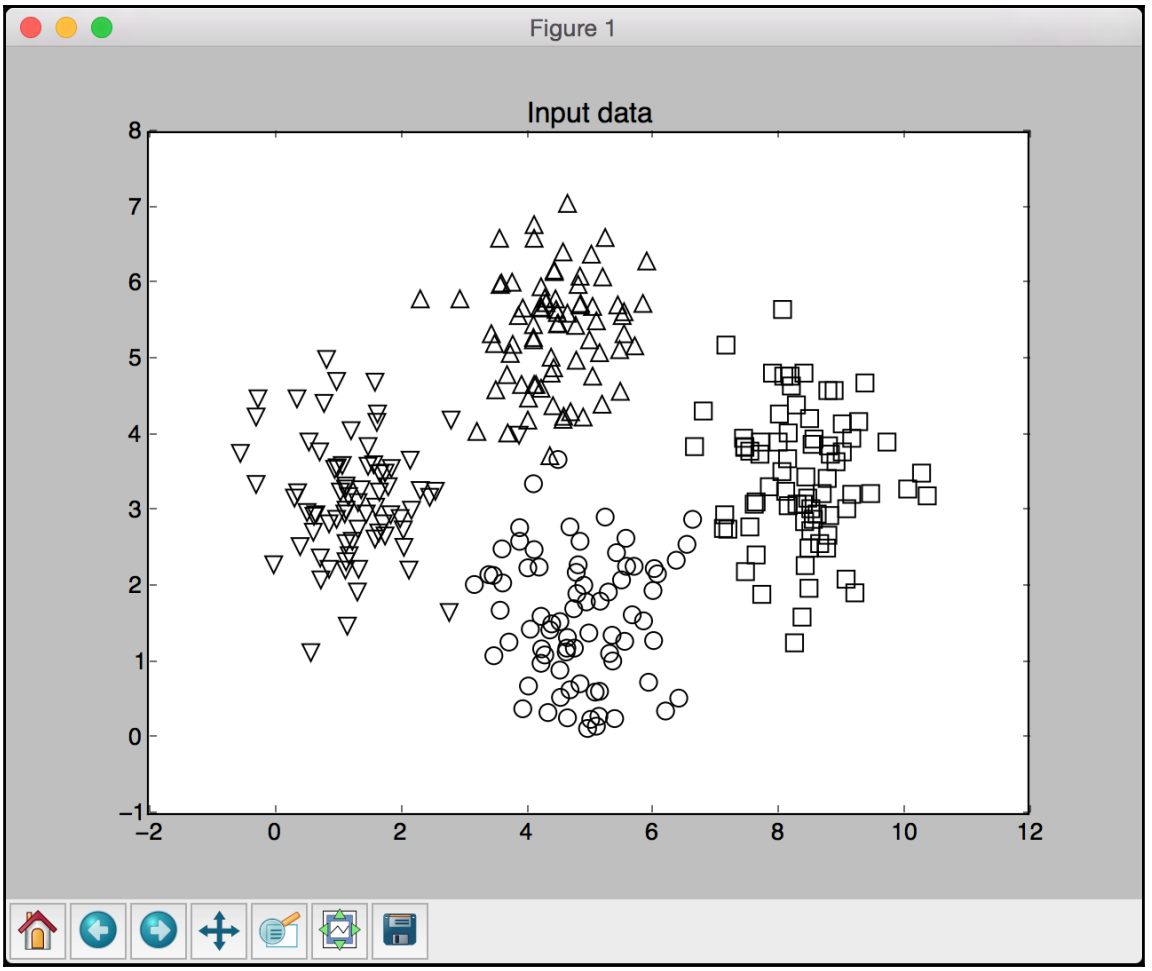


Figure 1
Input data
8
7
6
5
4
3
2
1
0
−1
−2
0
2
4
6
8
10
12

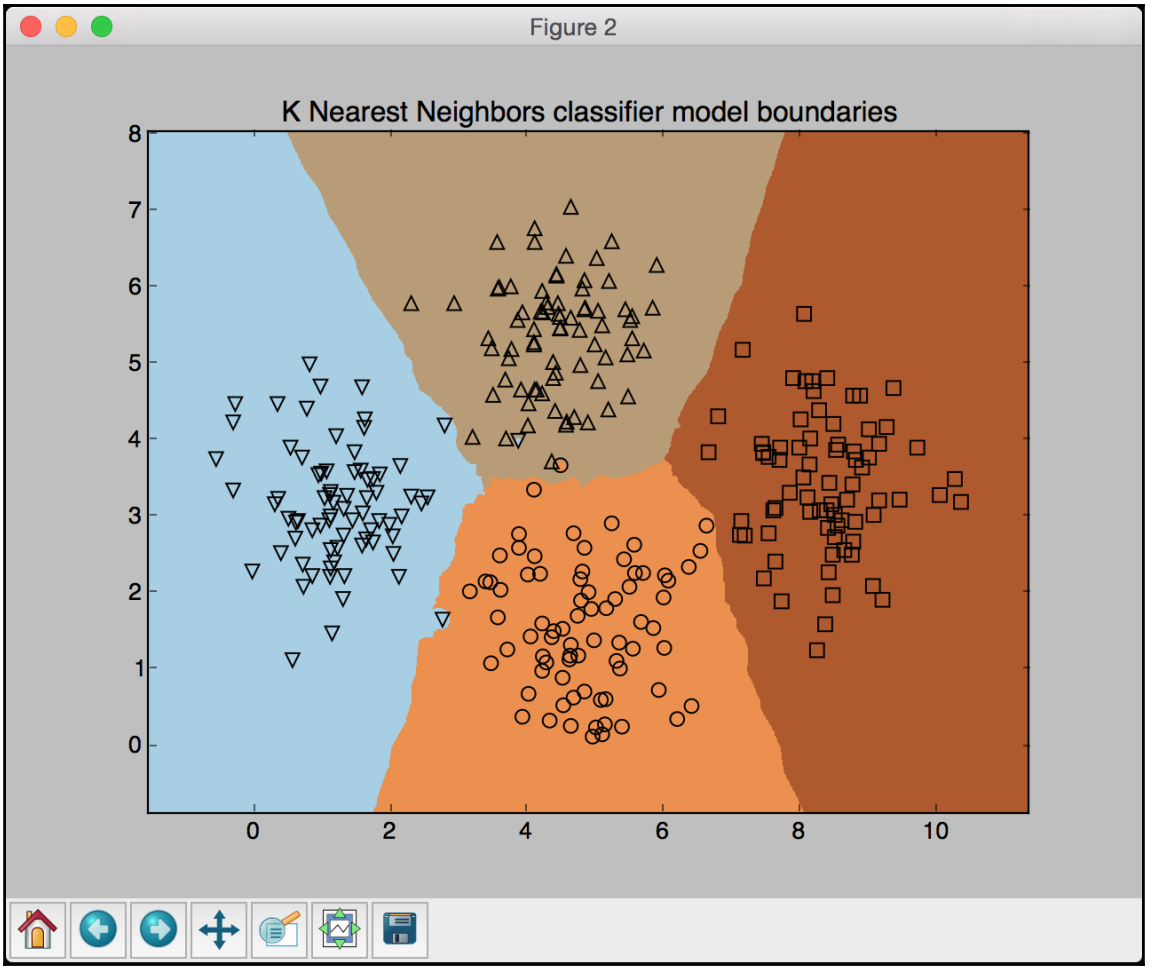


Figure 2
K Nearest Neighbors classifier model boundaries
8
7
6
5
4
3
2
1
0
0
2
4
6
8
10

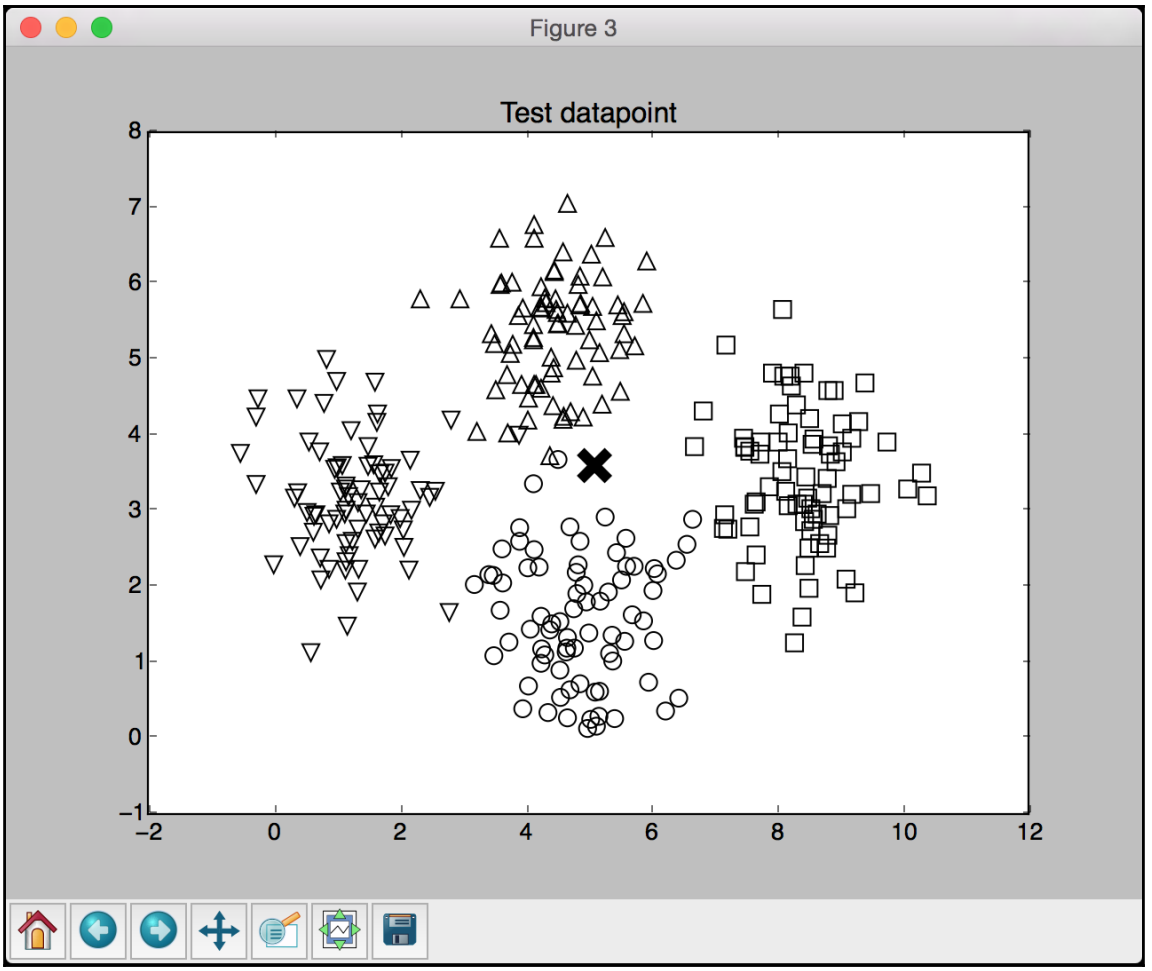
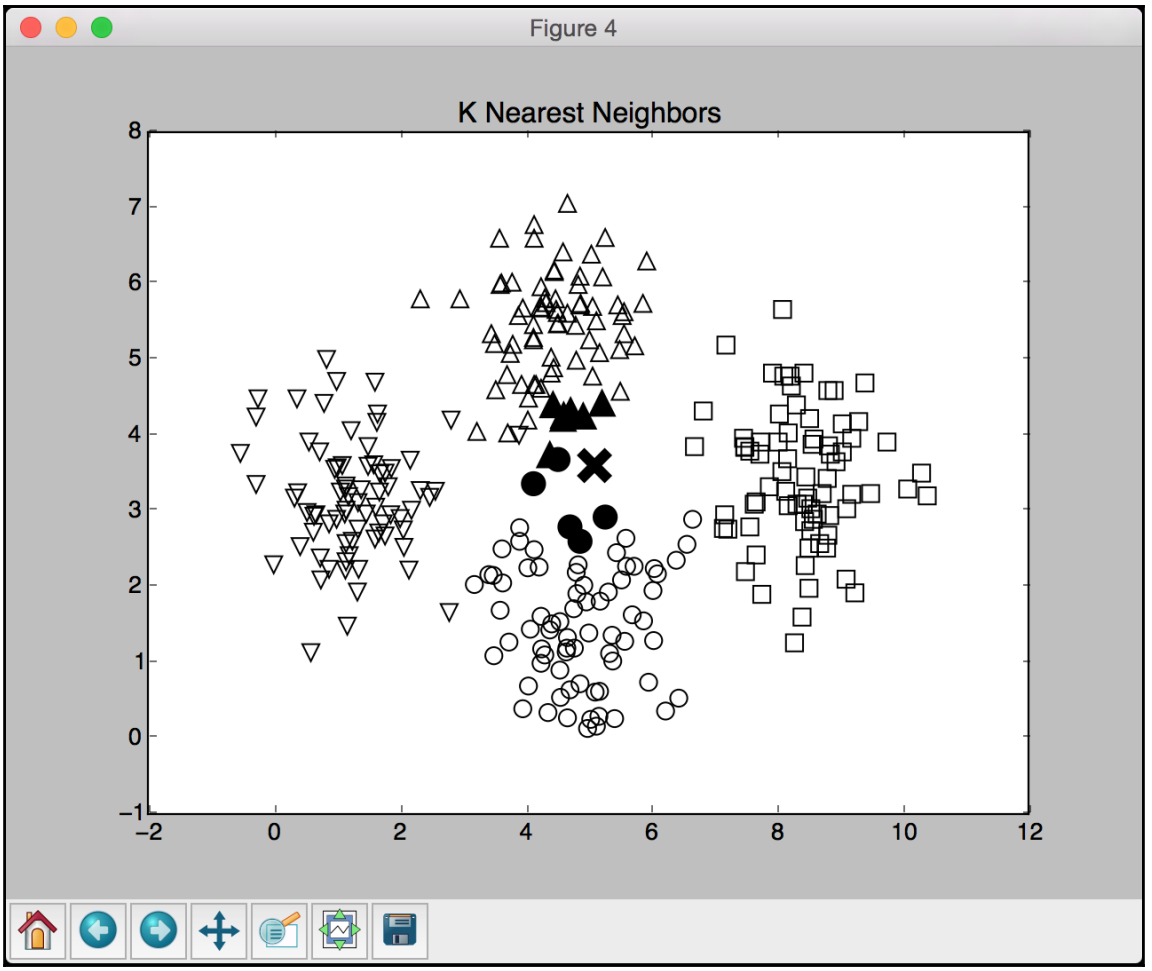


Figure 3
Test datapoint
8
7
6
5
4
3
2
1
0
−1
−2
0
2
4
6
8
10
12



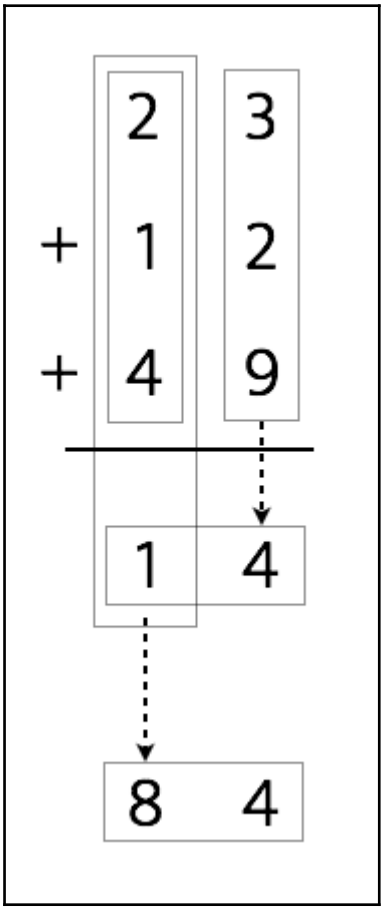

```
Users similar to Bill Duffy:

User                        Similarity score
-----------------------------------------
David Smith                  0.99
Samuel Miller                0.88
Adam Cohen                   0.86
```

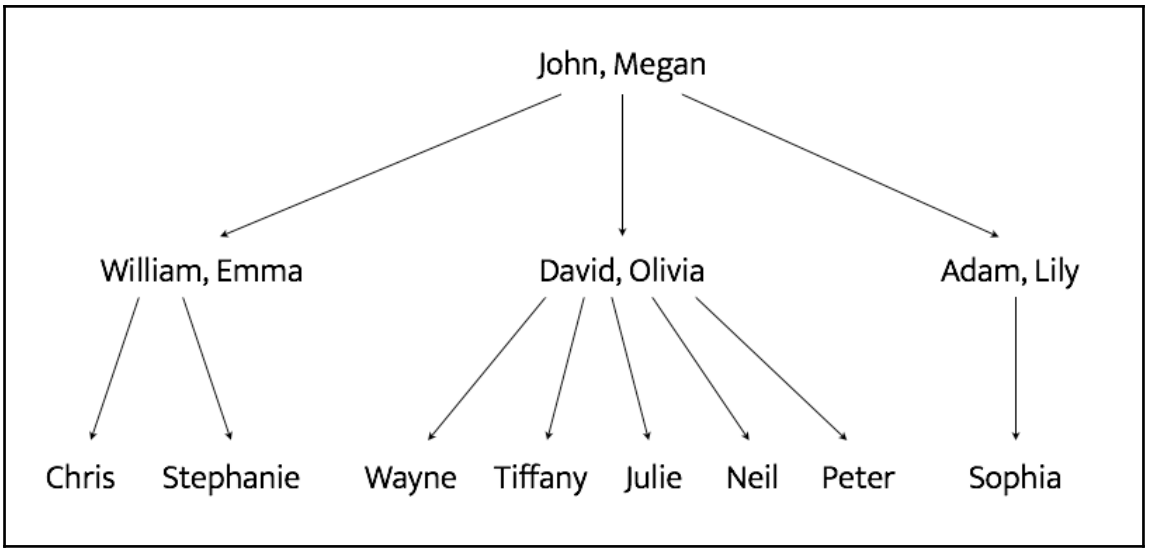
```
Users similar to Clarissa Jackson:

User                     Similarity score
-----------------------------------------
Chris Duncan              1.0
Bill Duffy                0.83
Samuel Miller             0.73
```

```
Movie recommendations for Chris Duncan:
1. Vertigo
2. Goodfellas
3. Scarface
4. Roman Holiday
```

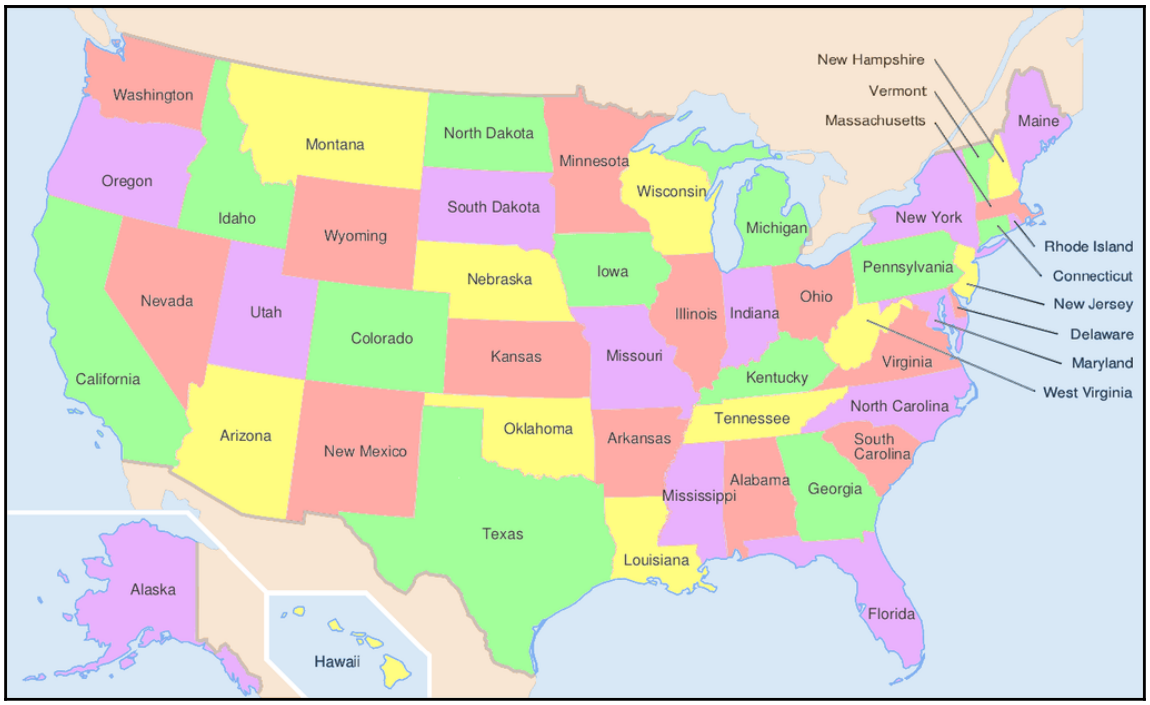
```
Movie recommendations for Julie Hammel:
1. The Apartment
2. Vertigo
3. Raging Bull
```

Chapter 6 : Logic Programming




John, Megan
William, Emma
David, Olivia
Adam, Lily
Chris
Stephanie
Wayne
Tiffany
Julie
Neil
Peter
Sophia

```
List of Megan's grandchildren:
Chris
Sophia
Peter
Stephanie
Julie
Tiffany
Neil
Wayne

List of David's siblings:
William
Adam

List of Tiffany's uncles:
William
Adam

List of all spouses:
Husband: Adam <==> Wife: Lily
Husband: David <==> Wife: Olivia
Husband: John <==> Wife: Megan
Husband: William <==> Wife: Emma
```


New Hampshire
Vermont
Massachusetts
Maine
Washington
Montana
North Dakota
Minnesota
Oregon
Wisconsin
South Dakota
Idaho
Michigan
New York
Wyoming
Rhode Island
Pennsylvania
Nebraska
Iowa
Connecticut
Nevada
Utah
Illinois
Indiana
Ohio
New Jersey
Colorado
Delaware
Kansas
Missouri
Maryland
Virginia
California
Kentucky
West Virginia
North Carolina
Arizona
Oklahoma
Tennessee
New Mexico
Arkansas
South Carolina
Alabama
Georgia
Mississippi
Texas
Louisiana
Alaska
Florida
Hawaii

```
Is Nevada adjacent to Louisiana?:
No

List of states adjacent to Oregon:
Washington
California
Nevada
Idaho

List of coastal states adjacent to Mississippi:
Alabama
Louisiana

List of 7 states that border a coastal state:
Georgia
Pennsylvania
Massachusetts
Wisconsin
Maine
Oregon
Ohio

List of states that are adjacent to Arkansas and Kentucky:
Missouri
Tennessee
```

Name	Pet	Car color	Country
Steve	dog	blue	France
Jack	cat	green	Canada
Matthew	rabbit	yellow	USA
Alfred	parrot	black	Australia

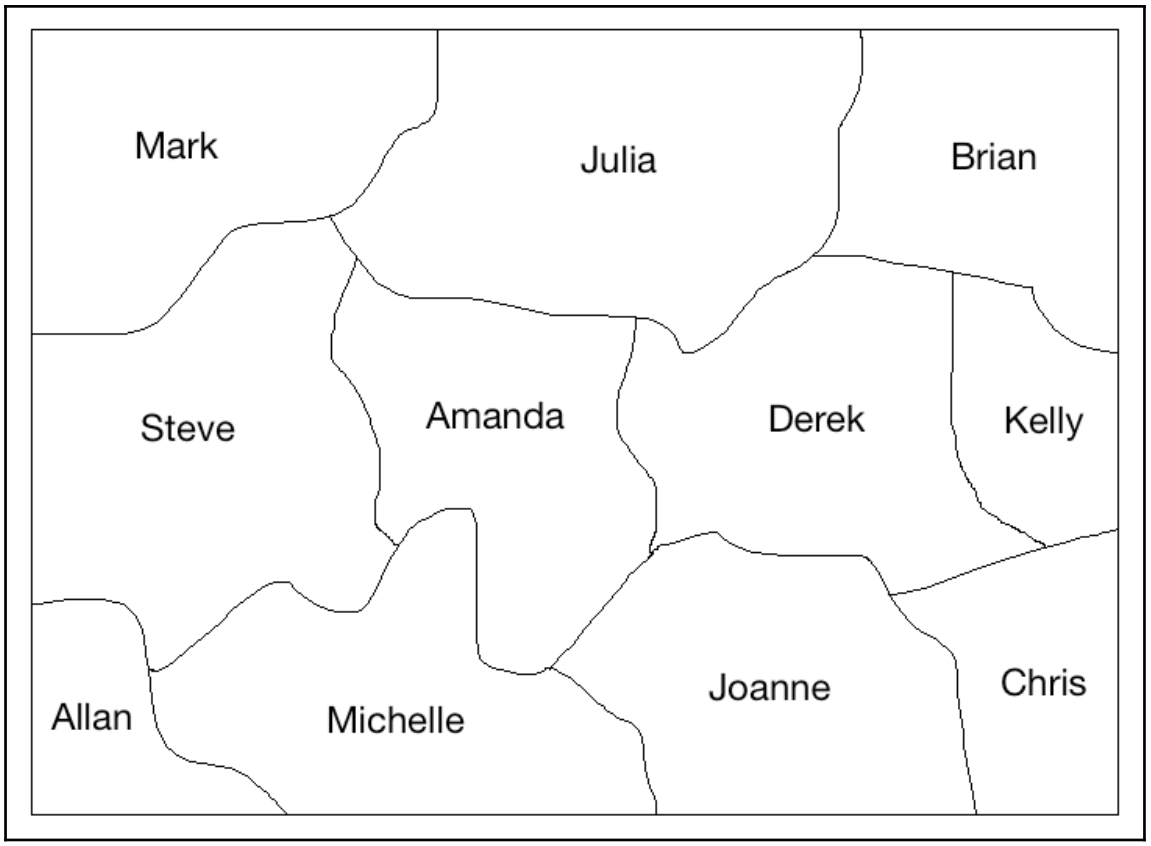
```
Matthew is the owner of the rabbit

Here are all the details:

Name            Pet             Color           Country
=====================================================

Steve           dog             blue            France

Jack            cat             ~_9             Canada

Matthew         rabbit          ~_11            USA

Alfred          ~_13            black           Australia
```

Chapter 7: Heuristic Search Techniques

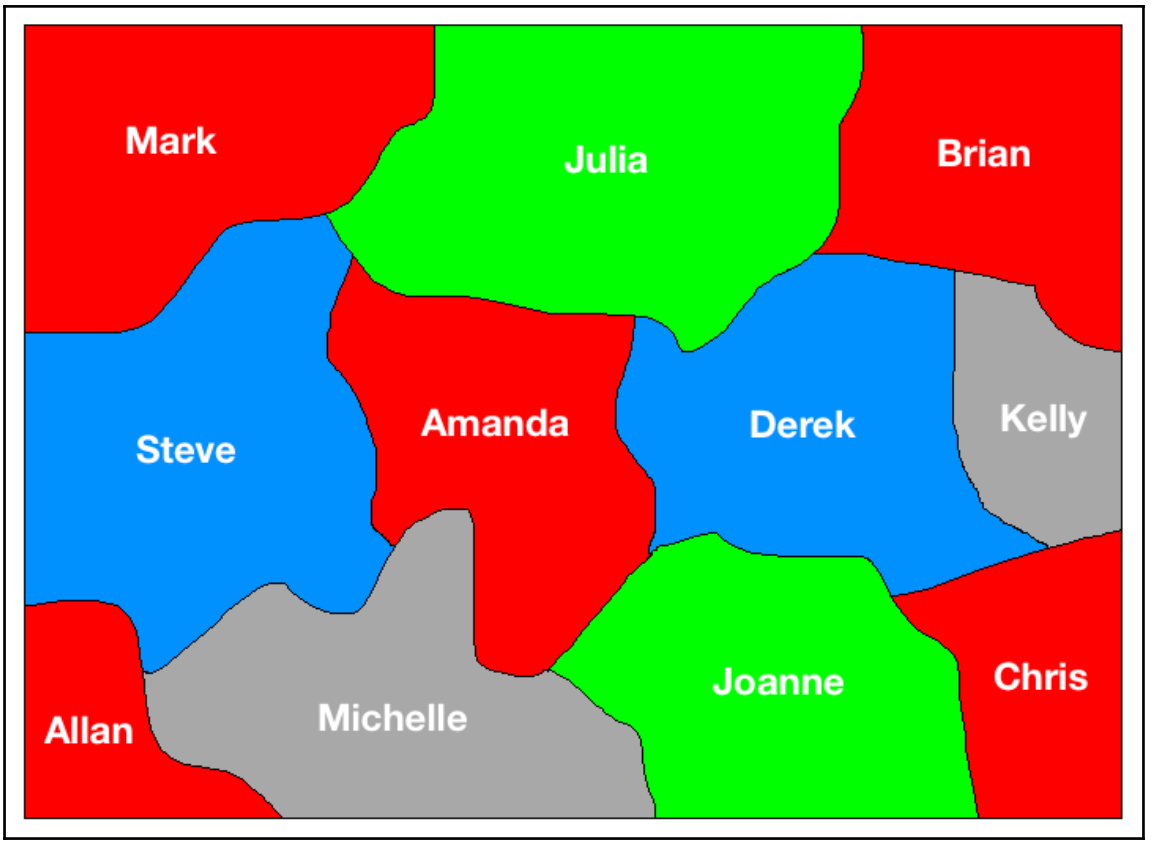
```
Path to the solution:
(None, '')
('A', 'A')
('r', 'Ar')
('t', 'Art')
('i', 'Arti')
('f', 'Artif')
('i', 'Artifi')
('c', 'Artific')
('i', 'Artifici')
('a', 'Artificia')
('l', 'Artificial')
(' ', 'Artificial ')
('I', 'Artificial I')
('n', 'Artificial In')
('t', 'Artificial Int')
('e', 'Artificial Inte')
('l', 'Artificial Intel')
('l', 'Artificial Intell')
('i', 'Artificial Intelli')
('g', 'Artificial Intellig')
('e', 'Artificial Intellige')
('n', 'Artificial Intelligen')
('c', 'Artificial Intelligenc')
('e', 'Artificial Intelligence')
```

```
Path to the solution:
(None, 'Artificial Inte')
('l', 'Artificial Intel')
('l', 'Artificial Intell')
('i', 'Artificial Intelli')
('g', 'Artificial Intellig')
('e', 'Artificial Intellige')
('n', 'Artificial Intelligen')
('c', 'Artificial Intelligenc')
('e', 'Artificial Intelligence')
(' ', 'Artificial Intelligence ')
('w', 'Artificial Intelligence w')
('i', 'Artificial Intelligence wi')
('t', 'Artificial Intelligence wit')
('h', 'Artificial Intelligence with')
(' ', 'Artificial Intelligence with ')
('P', 'Artificial Intelligence with P')
('y', 'Artificial Intelligence with Py')
('t', 'Artificial Intelligence with Pyt')
('h', 'Artificial Intelligence with Pyth')
('o', 'Artificial Intelligence with Pytho')
('n', 'Artificial Intelligence with Python')
```

```
Solutions:

Normal: {'Patricia': 2, 'John': 1, 'Anna': 3, 'Tom': 4}

Most constrained variable: {'Patricia': 2, 'John': 3, 'Anna': 1, 'Tom': 2}

Highest degree variable: {'Patricia': 2, 'John': 1, 'Anna': 3, 'Tom': 4}

Least constraining value: {'Patricia': 2, 'John': 1, 'Anna': 3, 'Tom': 4}

Most constrained variable and least constraining value: {'Patricia': 2, 'John': 3, 'Anna': 1, 'Tom': 2}

Highest degree and least constraining value: {'Patricia': 2, 'John': 1, 'Anna': 3, 'Tom': 4}

Minimum conflicts: {'Patricia': 4, 'John': 1, 'Anna': 3, 'Tom': 4}
```

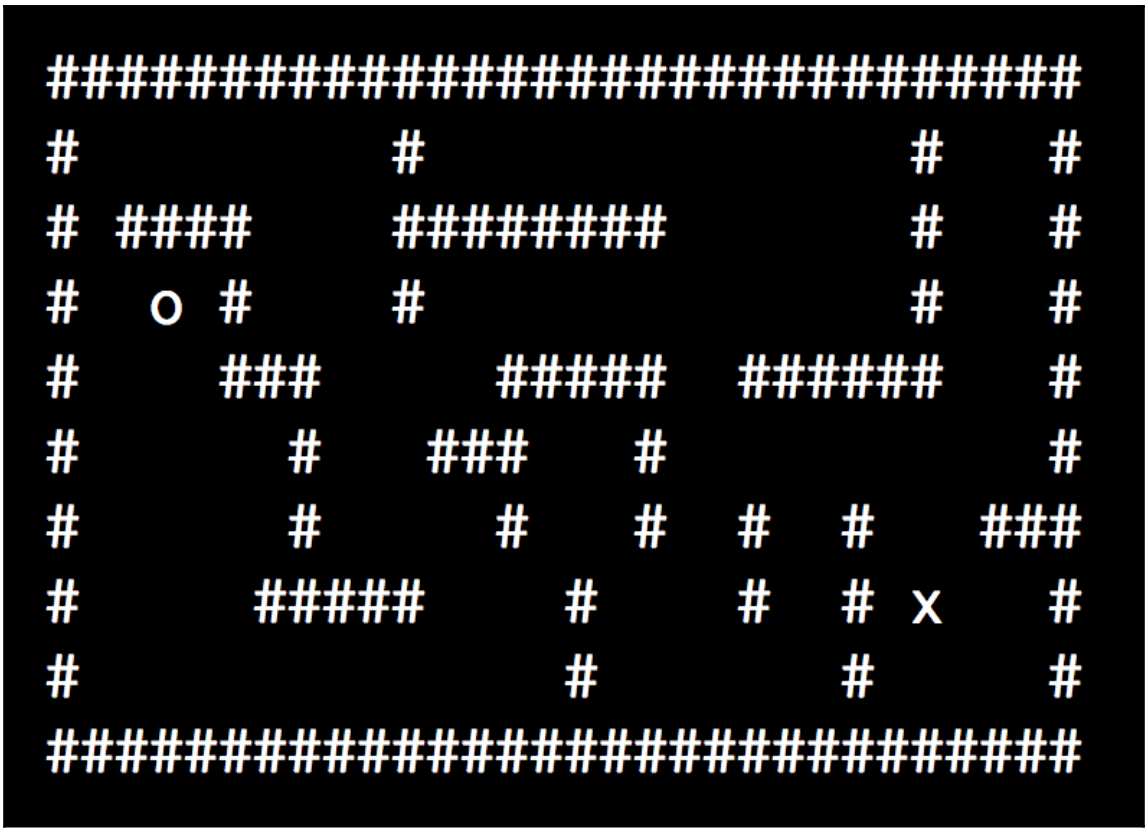

Mark
Julia
Brian
Steve
Amanda
Derek
Kelly
Allan
Michelle
Joanne
Chris

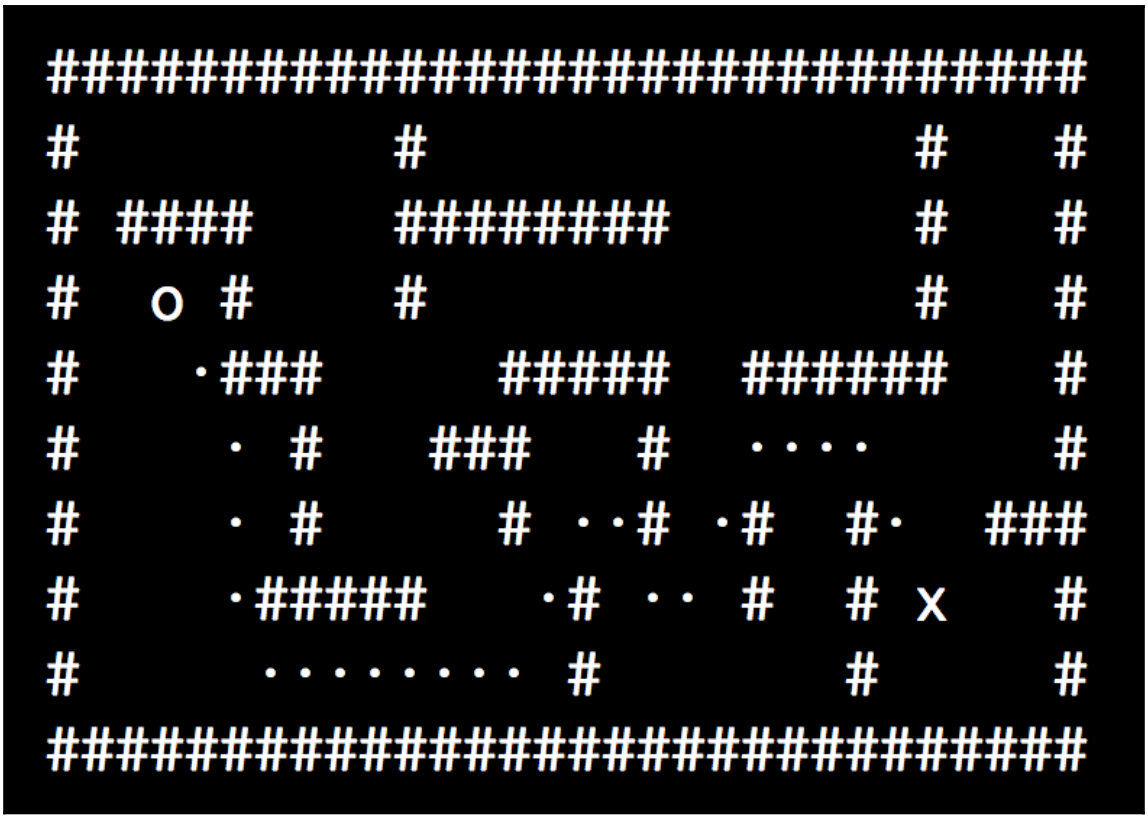
```
Color mapping:

Derek ==> blue
Michelle ==> gray
Allan ==> red
Steve ==> blue
Julia ==> green
Amanda ==> red
Joanne ==> green
Mark ==> red
Kelly ==> gray
Brian ==> red
```


Mark
Julia
Brian
Steve
Amanda
Derek
Kelly
Allan
Michelle
Joanne
Chris

```
Initial configuration
1-e-2
6-3-4
7-5-8

After moving 2 into the empty space
1-2-e
6-3-4
7-5-8

After moving 4 into the empty space
1-2-4
6-3-e
7-5-8

After moving 3 into the empty space
1-2-4
6-e-3
7-5-8

After moving 6 into the empty space
1-2-4
e-6-3
7-5-8
```

```
After moving 2 into the empty space
e-2-3
1-4-6
7-5-8

After moving 1 into the empty space
1-2-3
e-4-6
7-5-8

After moving 4 into the empty space
1-2-3
4-e-6
7-5-8

After moving 5 into the empty space
1-2-3
4-5-6
7-e-8

After moving 8 into the empty space. Goal achieved!
1-2-3
4-5-6
7-8-e
```


o
x


o
x

```
Starting the evolution process

Evaluated 500 individuals

===== Generation 0
Evaluated 297 individuals
Min = 58.0 , Max = 75.0
Average = 70.43 , Standard deviation = 2.91

===== Generation 1
Evaluated 303 individuals
Min = 63.0 , Max = 75.0
Average = 72.44 , Standard deviation = 2.16

===== Generation 2
Evaluated 310 individuals
Min = 65.0 , Max = 75.0
Average = 73.31 , Standard deviation = 1.6

===== Generation 3
Evaluated 273 individuals
Min = 67.0 , Max = 75.0
Average = 73.76 , Standard deviation = 1.41
```

```
===== Generation 57
Evaluated 306 individuals
Min = 68.0 , Max = 75.0
Average = 74.02 , Standard deviation = 1.27

===== Generation 58
Evaluated 276 individuals
Min = 69.0 , Max = 75.0
Average = 74.15 , Standard deviation = 1.18

===== Generation 59
Evaluated 288 individuals
Min = 69.0 , Max = 75.0
Average = 74.12 , Standard deviation = 1.24

==== End of evolution

Best individual:
 [1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0
, 1, 0, 0, 0, 1]

Number of ones: 45
```

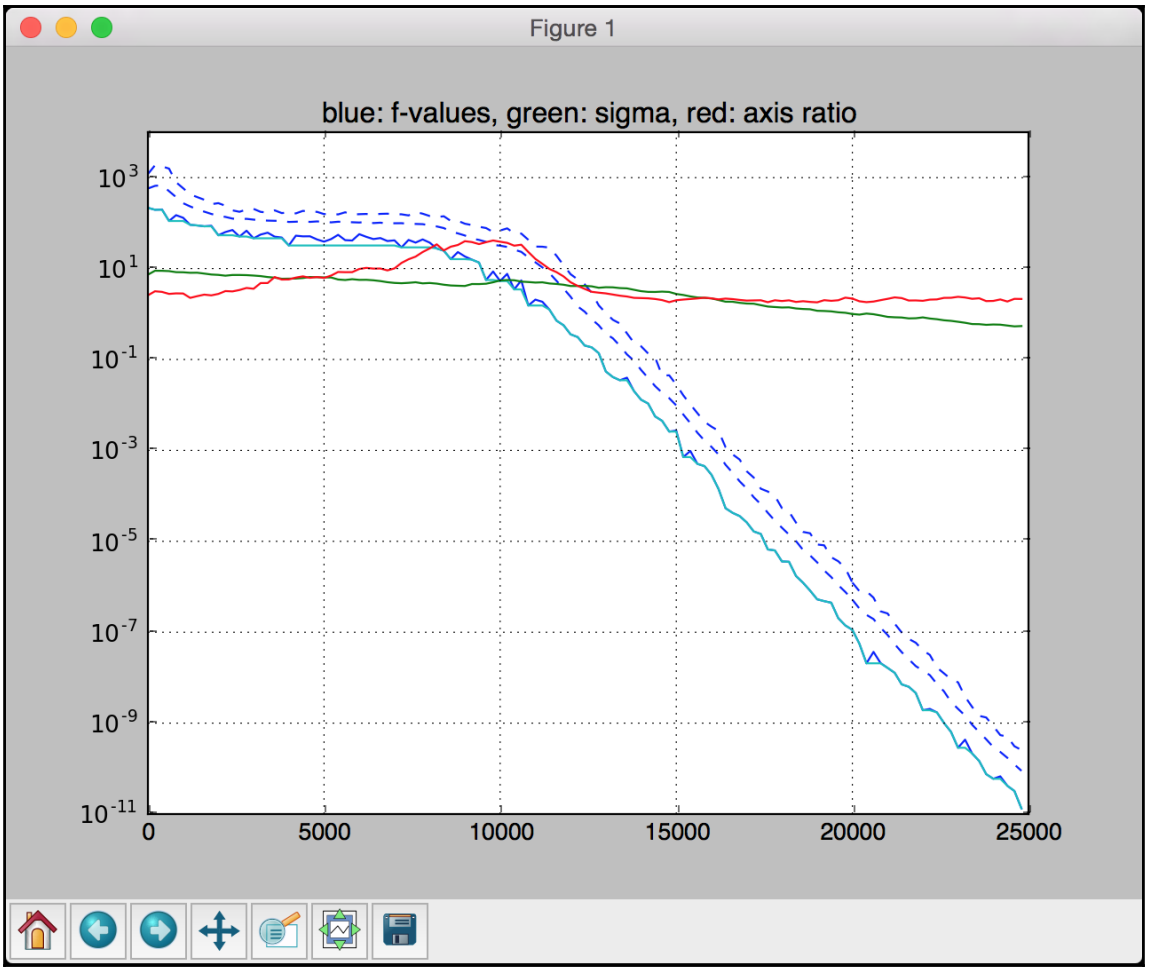


Figure 1
blue: f-values, green: sigma, red: axis ratio
10^3
10^1
10^-1
10^-3
10^-5
10^-7
10^-9
10^-11
0
5000
10000
15000
20000
25000

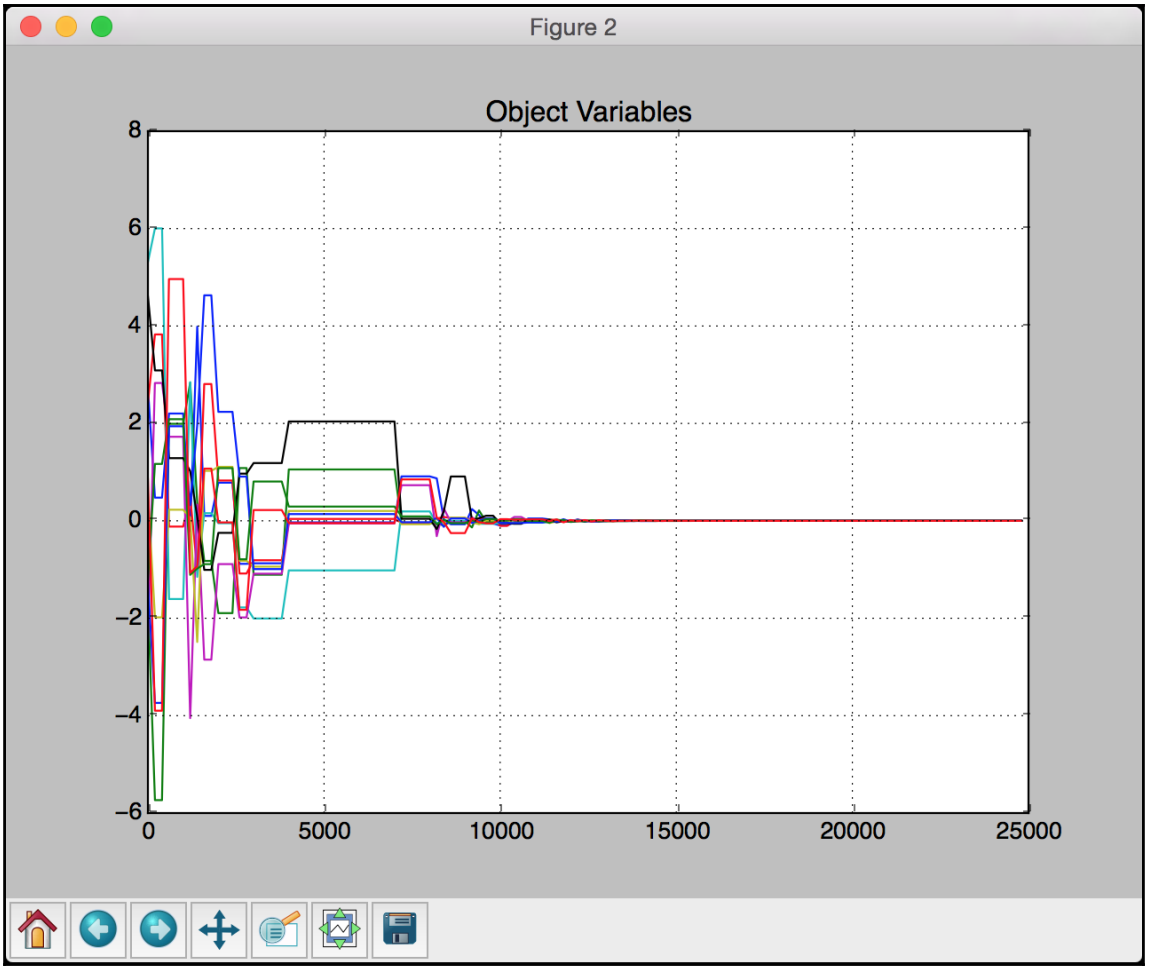


Figure 2
Object Variables
8
6
4
2
0
−2
−4
−6
0
5000
10000
15000
20000
25000

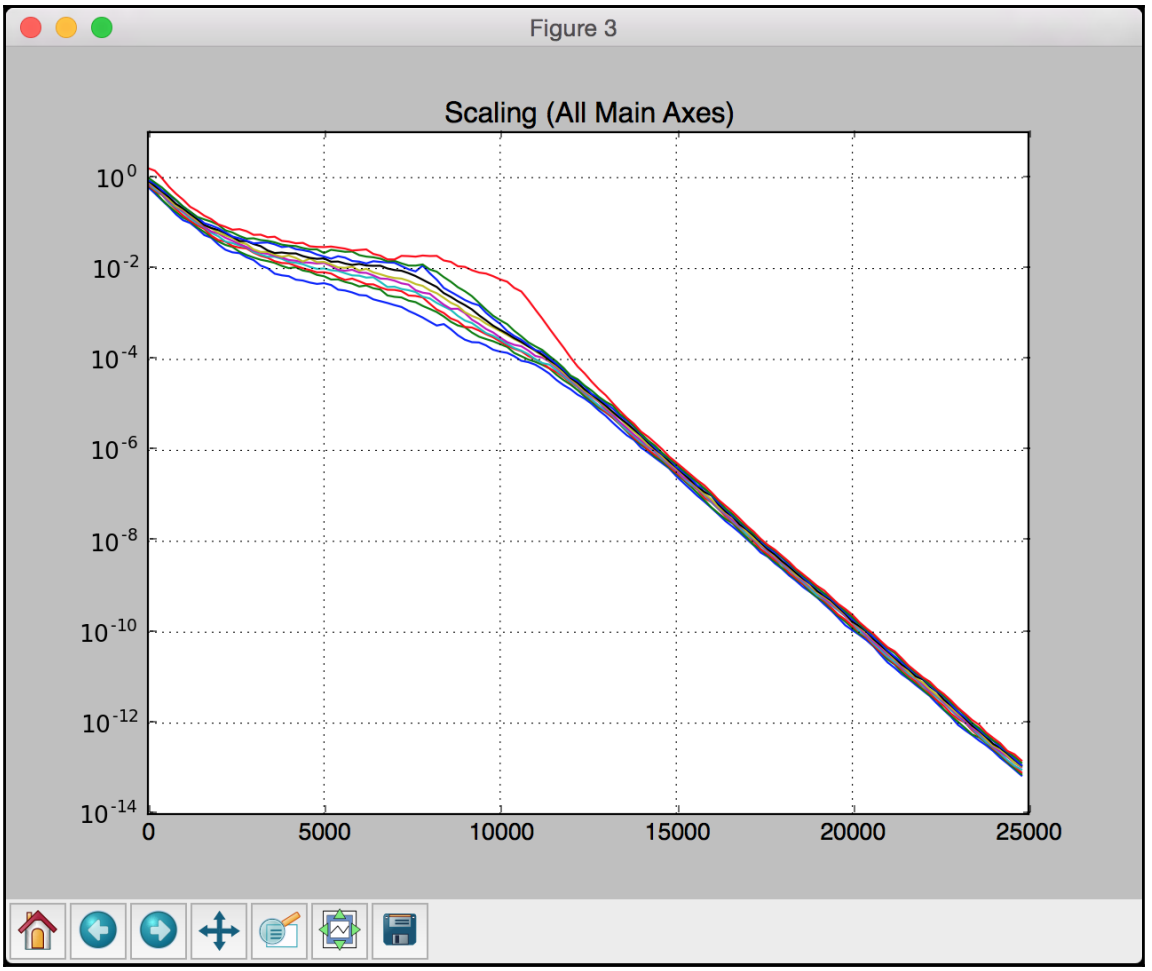


Figure 3
Scaling (All Main Axes)
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
0
5000
10000
15000
20000
25000

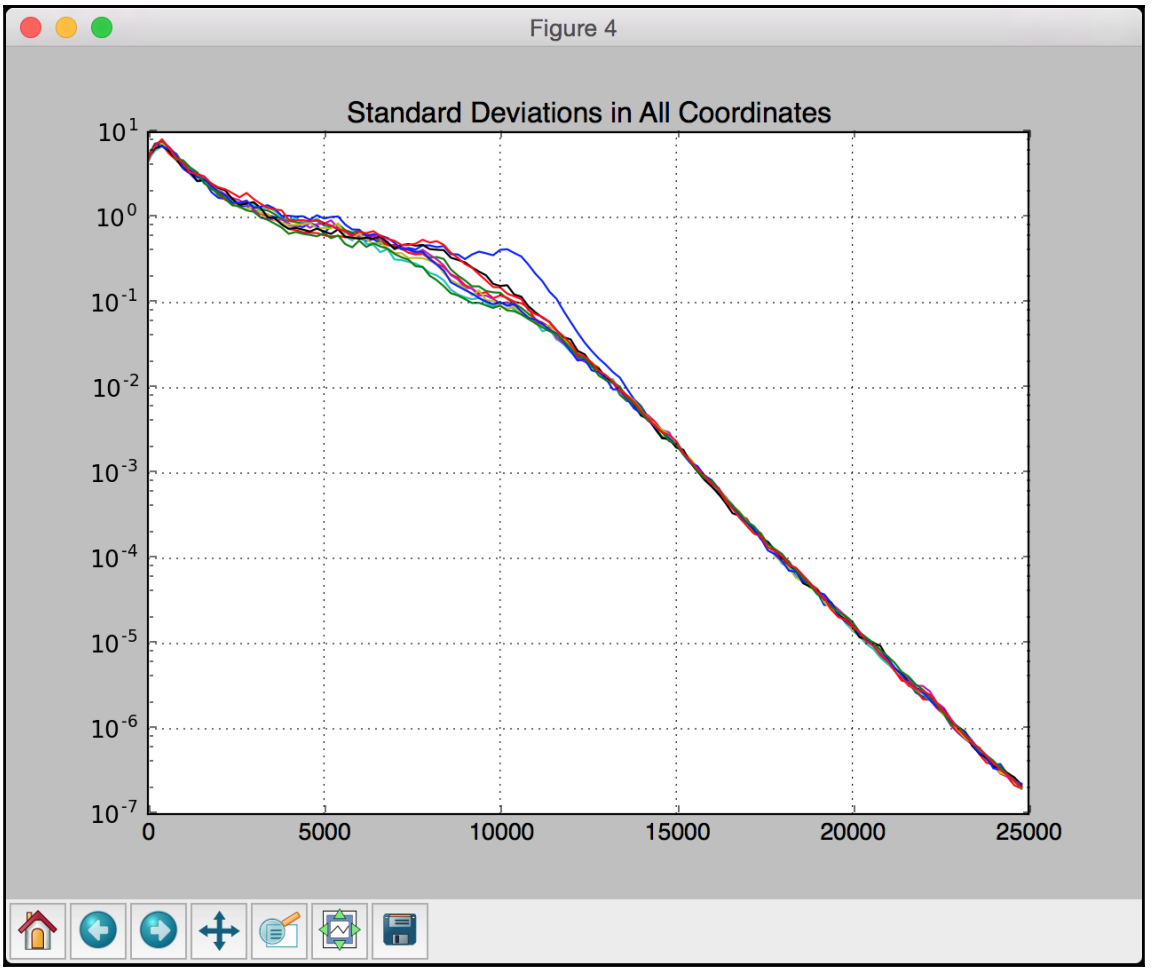


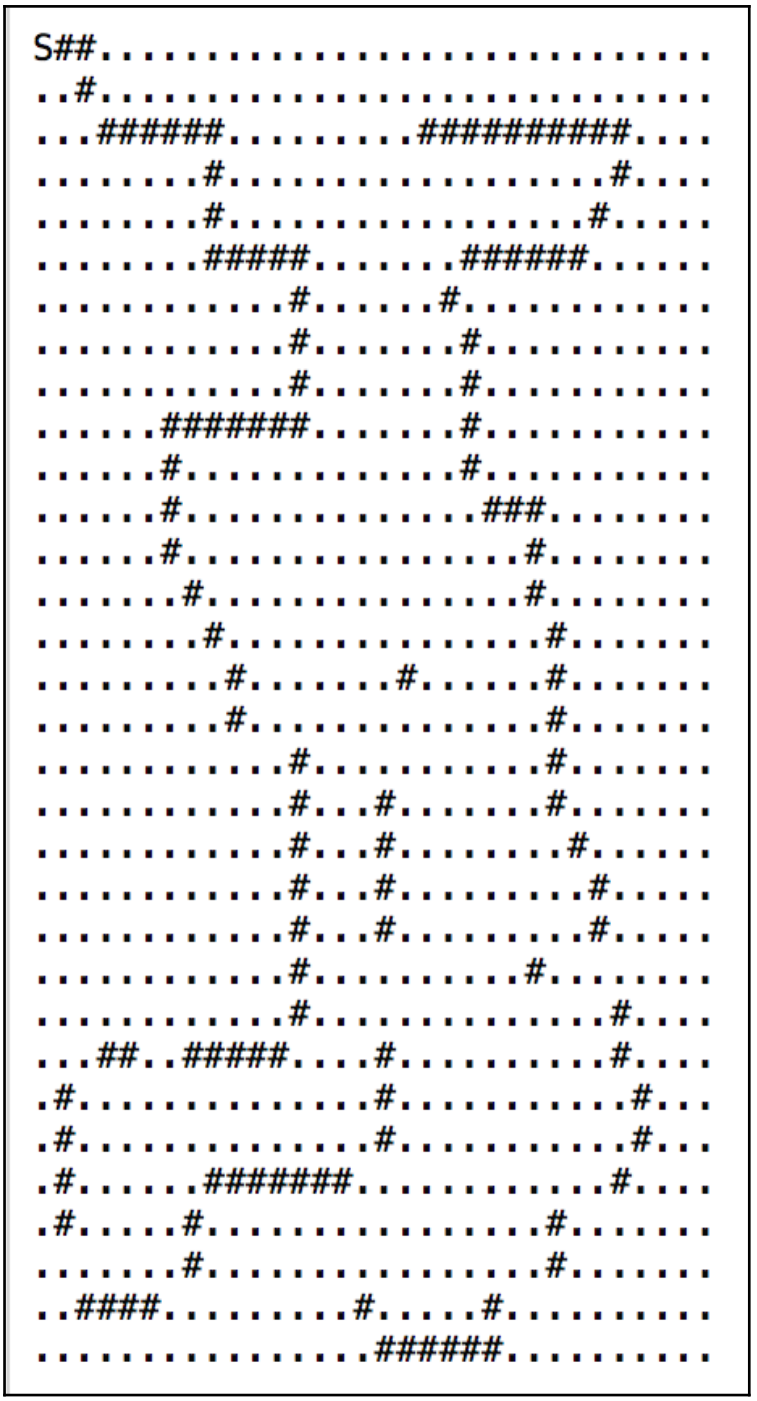
Figure 4
Standard Deviations in All Coordinates
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
0
5000
10000
15000
20000
25000

gen	evals	std	min	avg	max
0	200	188.36	217.082	576.281	1199.71
1	200	250.543	196.583	659.389	1869.02
2	200	273.081	199.455	683.641	1770.65
3	200	215.326	111.298	503.933	1579.3
4	200	133.046	149.47	373.124	790.899
5	200	75.4405	131.117	274.092	585.433
6	200	61.2622	91.7121	232.624	426.666
7	200	49.8303	88.8185	201.117	373.543
8	200	39.9533	85.0531	178.645	326.209
9	200	31.3781	87.4824	159.211	261.132
10	200	31.3488	54.0743	144.561	274.877
11	200	30.8796	63.6032	136.791	240.739
12	200	24.1975	70.4913	125.691	190.684
13	200	21.2274	50.6409	122.293	177.483
14	200	25.4931	67.9873	124.132	199.296
15	200	26.9804	46.3411	119.295	205.331
16	200	24.8993	56.0033	115.614	176.702
17	200	21.9789	61.4999	113.417	170.156
18	200	21.2823	50.2455	112.419	190.677
19	200	22.5016	48.153	111.543	166.2
20	200	21.1602	32.1864	106.044	171.899
21	200	23.3864	52.8601	107.301	163.617
22	200	23.1008	51.1226	109.628	185.777
23	200	22.0836	51.3058	106.402	179.673

100	200	2.38865e-07	1.12678e-07	5.18814e-07	1.23527e-06
101	200	1.49444e-07	5.56979e-08	3.3199e-07	7.98774e-07
102	200	1.11635e-07	2.07109e-08	2.41361e-07	7.96738e-07
103	200	9.50257e-08	3.69117e-08	1.94641e-07	5.75896e-07
104	200	5.63849e-08	2.09827e-08	1.26148e-07	2.887e-07
105	200	4.42488e-08	1.64212e-08	8.6972e-08	2.58639e-07
106	200	2.34933e-08	1.28302e-08	5.47789e-08	1.54658e-07
107	200	1.74434e-08	7.13185e-09	3.64705e-08	9.88235e-08
108	200	1.17157e-08	6.32208e-09	2.54673e-08	7.13075e-08
109	200	8.73027e-09	4.60369e-09	1.79681e-08	5.88066e-08
110	200	6.39874e-09	1.92573e-09	1.43229e-08	4.00087e-08
111	200	5.31196e-09	2.05551e-09	1.13736e-08	3.16793e-08
112	200	3.15607e-09	1.72427e-09	7.28548e-09	1.67727e-08
113	200	2.3789e-09	1.01164e-09	5.01177e-09	1.24541e-08
114	200	1.38424e-09	6.43112e-10	2.94696e-09	9.25819e-09
115	200	1.04172e-09	2.87571e-10	2.06068e-09	7.90436e-09
116	200	6.08685e-10	4.32905e-10	1.4704e-09	3.80221e-09
117	200	4.51515e-10	2.1538e-10	9.23627e-10	2.2759e-09
118	200	2.77204e-10	1.46869e-10	6.3507e-10	1.44637e-09
119	200	2.06475e-10	7.54881e-11	4.41427e-10	1.33167e-09
120	200	1.3138e-10	5.97282e-11	2.98116e-10	8.60453e-10
121	200	9.52385e-11	6.753e-11	2.32358e-10	5.45441e-10
122	200	7.55001e-11	4.1851e-11	1.72688e-10	5.05054e-10
123	200	5.52125e-11	3.2216e-11	1.23505e-10	3.10081e-10
124	200	4.38068e-11	1.32871e-11	8.94929e-11	2.57202e-10

		fitness				size			
gen	nevals	avg	max	min	std	avg	max	min	std
0	450	18.6918	47.1923	7.39087	6.27543	3.73556	7	2	1.62449
1	251	15.4572	41.3823	4.46965	4.54993	3.80222	12	1	1.81316
2	236	13.2545	37.7223	4.46965	4.06145	3.96889	12	1	1.98861
3	251	12.2299	60.828	4.46965	4.70055	4.19556	12	1	1.9971
4	235	11.001	47.1923	4.46965	4.48841	4.84222	13	1	2.17245
5	229	9.44483	31.478	4.46965	3.8796	5.56	19	1	2.43168
6	225	8.35975	22.0546	3.02133	3.40547	6.38889	15	1	2.40875
7	237	7.99309	31.1356	1.81133	4.08463	7.14667	16	1	2.57782
8	224	7.42611	359.418	1.17558	17.0167	8.33333	19	1	3.11127
9	237	5.70308	24.1921	1.17558	3.71991	9.64444	23	1	3.31365
10	254	5.27991	30.4315	1.13301	4.13556	10.5089	25	1	3.51898

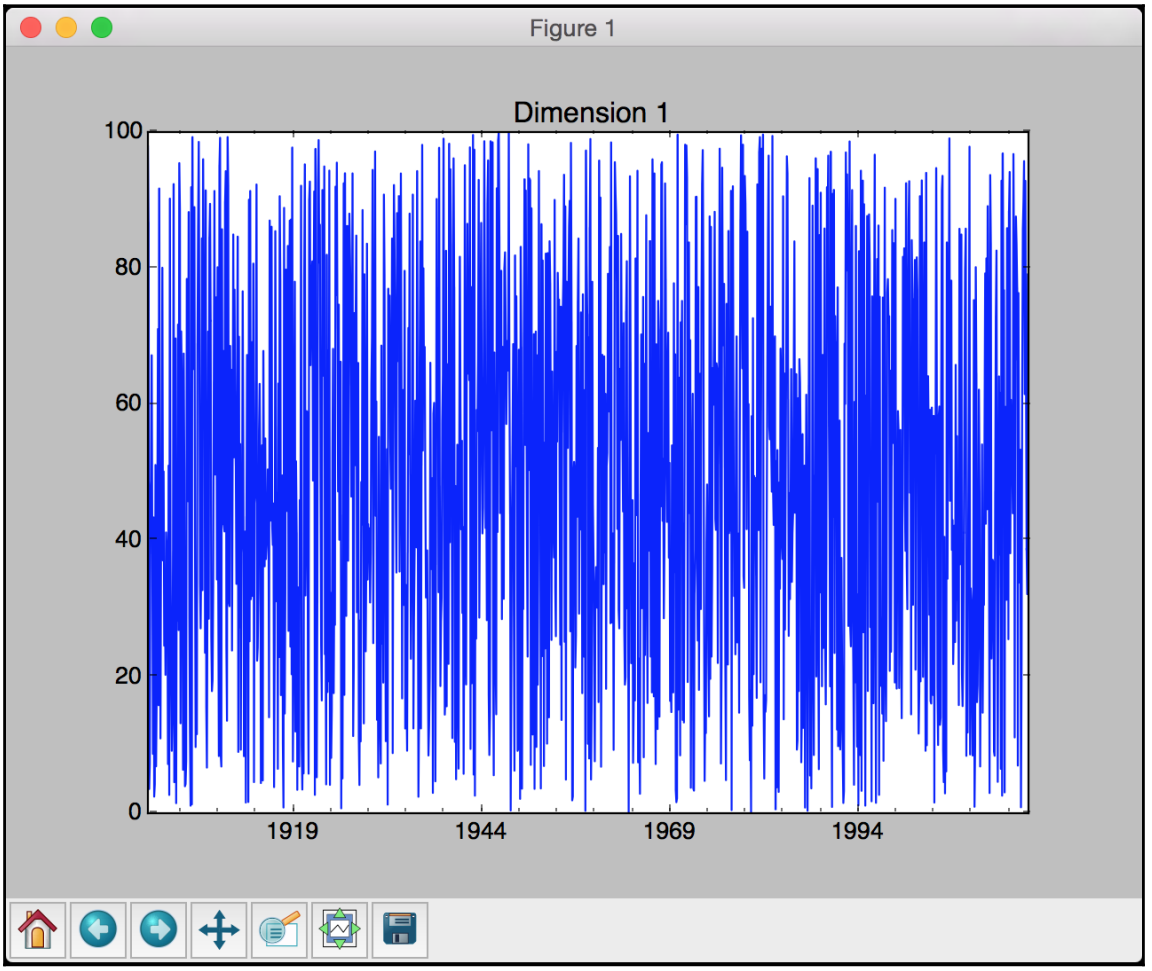
36	209	1.10464	22.0546	0.0474957	2.71898	26.4867	46	1	5.23289
37	258	1.61958	86.0936	0.0382386	6.1839	27.2111	45	3	4.75557
38	257	2.03651	70.4768	0.0342642	5.15243	26.5311	49	1	6.22327
39	235	1.95531	185.328	0.0472693	9.32516	26.9711	48	1	6.00345
40	234	1.51403	28.5529	0.0472693	3.24513	26.6867	52	1	5.39811
41	230	1.4753	70.4768	0.0472693	5.4607	27.1	46	3	4.7433
42	233	12.3648	4880.09	0.0396503	229.754	26.88	53	1	5.18192
43	251	1.807	86.0936	0.0396503	5.85281	26.4889	50	1	5.43741
44	236	9.30096	3481.25	0.0277886	163.888	26.9622	55	1	6.27169
45	231	1.73196	86.7372	0.0342642	6.8119	27.4711	51	2	5.27807
46	227	1.86086	185.328	0.0342642	10.1143	28.0644	56	1	6.10812
47	216	12.5214	4923.66	0.0342642	231.837	29.1022	54	1	6.45898
48	232	14.3469	5830.89	0.0322462	274.536	29.8244	58	3	6.24093
49	242	2.56984	272.833	0.0322462	18.2752	29.9267	51	1	6.31446
50	227	2.80136	356.613	0.0322462	21.0416	29.7978	56	4	6.50275
51	243	1.75099	86.0936	0.0322462	5.70833	29.8089	56	1	6.62379
52	253	10.9184	3435.84	0.0227048	163.602	29.9911	55	1	6.66833
53	243	1.80265	48.0418	0.0227048	4.73856	29.88	55	1	7.33084
54	234	1.74487	86.0936	0.0227048	6.0249	30.6067	55	1	6.85782
55	220	1.58888	31.094	0.0132398	3.82809	30.5644	54	1	6.96669
56	234	1.46711	103.287	0.00766444	6.81157	30.6689	55	3	6.6806
57	250	17.0896	6544.17	0.00424267	308.689	31.1267	60	4	7.25837
58	231	1.66757	141.584	0.00144401	7.35306	32	52	1	7.23295
59	229	2.22325	265.224	0.00144401	13.388	33.5489	64	1	8.38351
60	248	2.60303	521.804	0.00144401	24.7018	35.2533	58	1	7.61506


S##.....
..#.....
...######.....##########....
........#.....#....
........#.....#.....
........#####.....######.....
............#.....#.....
............#.....#.....
............#.....#.....
......#######.....#.....
......#.....#.....
......#.....###.....
......#.....#.....
.......#.....#.....
........#.....#.....
.........#.....#.....#.....
.........#.....#.....
............#.....#.....
............#...#.....#.....
............#...#.....#.....
............#...#.....#.....
............#...#.....#.....
............#.....#.....
............#.....#....
...##..#####....#.....#....
.#.....#.....#...
.#.....#.....#...
.#.....#######.....#....
.#.....#.....#.....
.......#.....#.....
..####.....#.....#.....
.....######.....

gen	nevals	avg	std	min	max
0	400	1.4875	4.37491	0	62
1	231	4.285	7.56993	0	73
2	235	10.8925	14.8493	0	73
3	231	21.72	22.1239	0	73
4	238	29.9775	27.7861	0	76
5	224	37.6275	31.8698	0	76
6	231	42.845	33.0541	0	80
7	223	43.55	33.9369	0	83
8	234	44.0675	34.5201	0	83
9	231	49.2975	34.3065	0	83
10	249	47.075	36.4106	0	93
11	222	52.7925	36.2826	0	97
12	248	51.0725	37.2598	0	97
13	234	54.01	37.4614	0	97
14	229	59.615	37.7894	0	97
15	228	63.3	39.8205	0	97
16	220	64.605	40.3962	0	97
17	236	62.545	40.5607	0	97
18	233	67.99	38.9033	0	97
19	236	66.4025	39.6574	0	97
20	221	69.785	38.7117	0	97
21	244	65.705	39.0957	0	97
22	230	70.32	37.1206	0	97
23	241	67.3825	39.4028	0	97

26	214	71.505	36.964	0	97
27	246	72.72	37.1637	0	97
28	238	73.5975	36.5385	0	97
29	239	76.405	35.5696	0	97
30	246	78.6025	33.4281	0	97
31	240	74.83	36.5157	0	97
32	216	80.2625	32.6659	0	97
33	220	80.6425	33.0933	0	97
34	247	78.245	34.6022	0	97
35	241	81.22	32.1885	0	97
36	234	83.6375	29.0002	0	97
37	228	82.485	31.7354	0	97
38	219	83.4625	30.0592	0	97
39	212	88.64	24.2702	0	97
40	231	86.7275	27.0879	0	97
41	229	89.1825	23.8773	0	97
42	216	87.96	25.1649	0	97
43	218	86.85	27.1116	0	97
44	236	88.78	23.7278	0	97
45	225	89.115	23.4212	0	97
46	232	88.5425	24.187	0	97
47	245	87.7775	25.3909	0	97
48	231	87.78	26.3786	0	97
49	238	88.8525	24.5115	0	97
50	233	87.82	25.4164	1	97

Chapter 11: Probabilistic Reasoning for Sequential Data



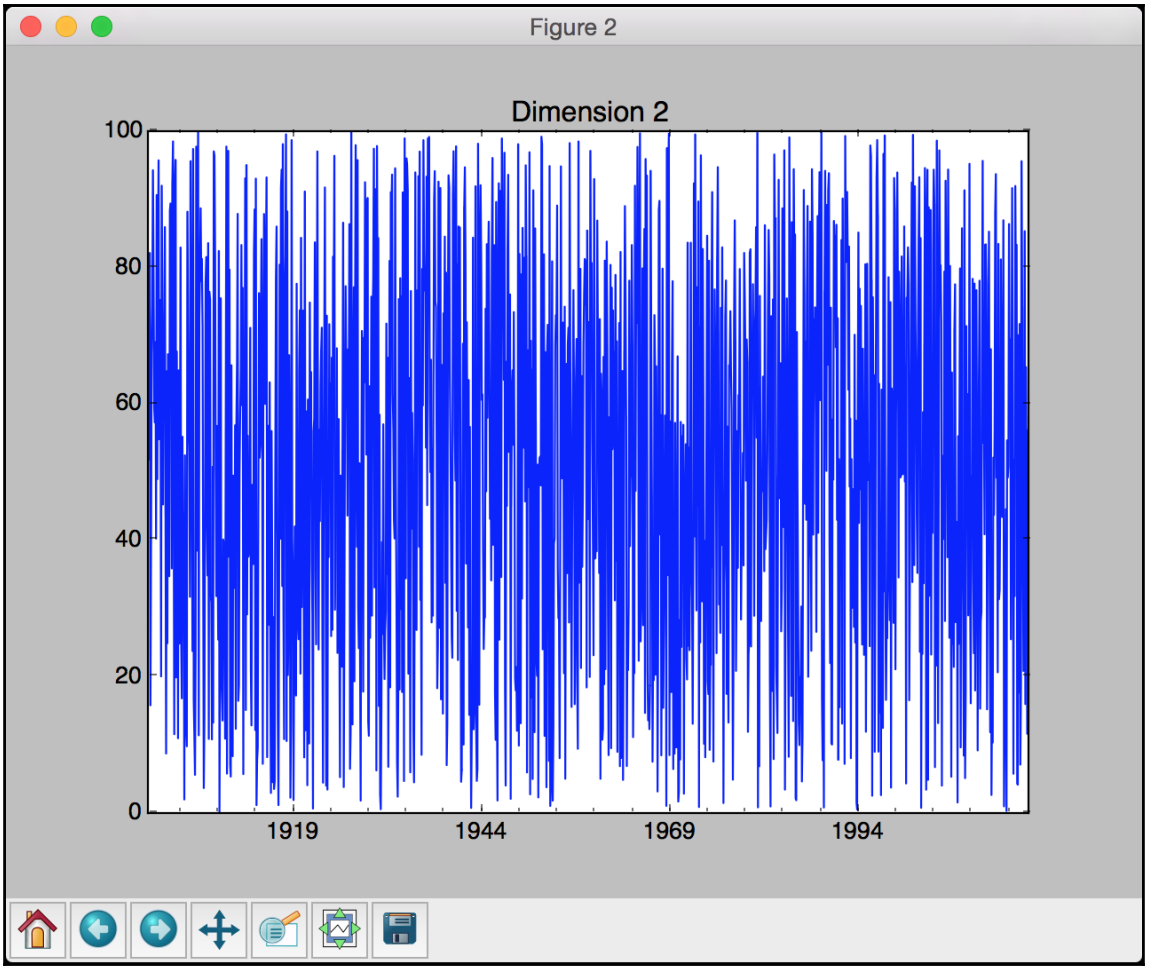


Figure 2
Dimension 2
100
80
60
40
20
0
1919
1944
1969
1994

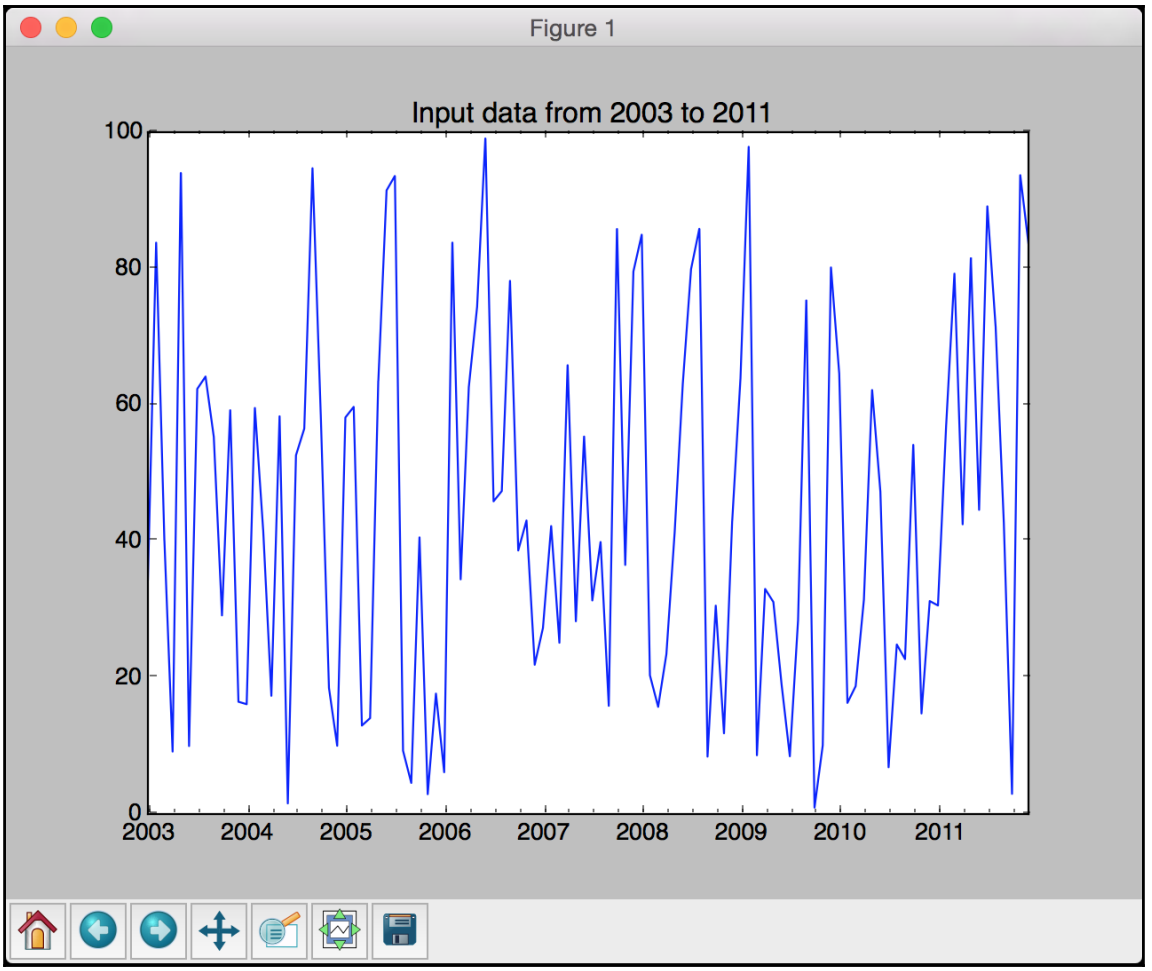


Figure 1
Input data from 2003 to 2011
100
80
60
40
20
0
2003
2004
2005
2006
2007
2008
2009
2010
2011

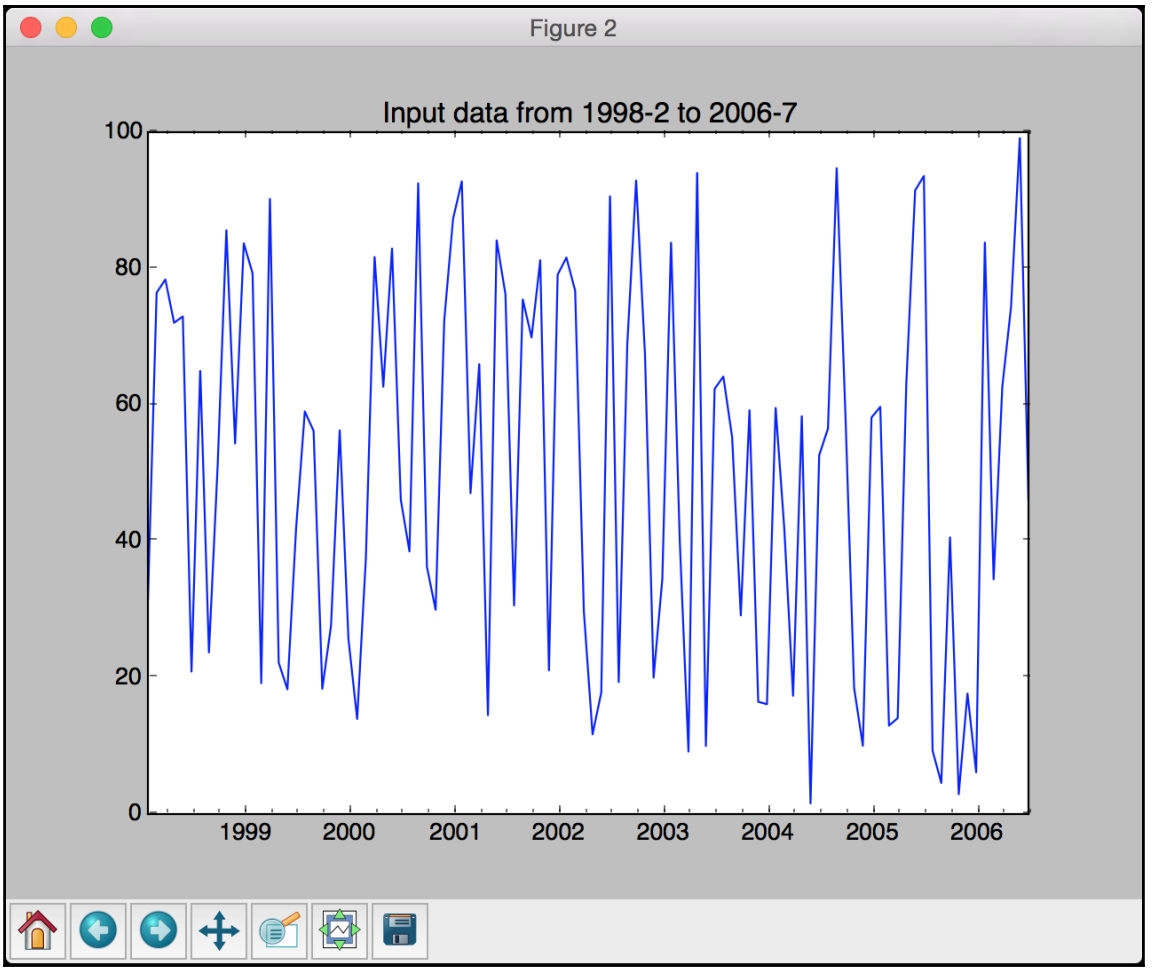


Figure 2
Input data from 1998-2 to 2006-7
100
80
60
40
20
0
1999
2000
2001
2002
2003
2004
2005
2006

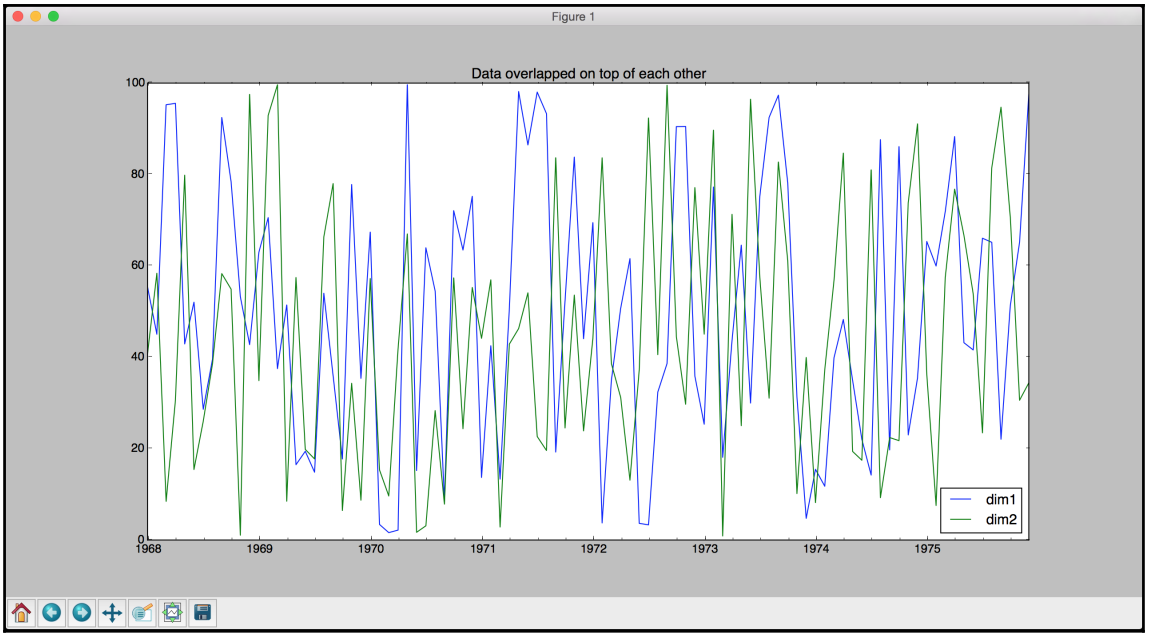


Figure 1
Data overlapped on top of each other
100
80
60
40
20
0
1968
1969
1970
1971
1972
1973
1974
1975
dim1
dim2

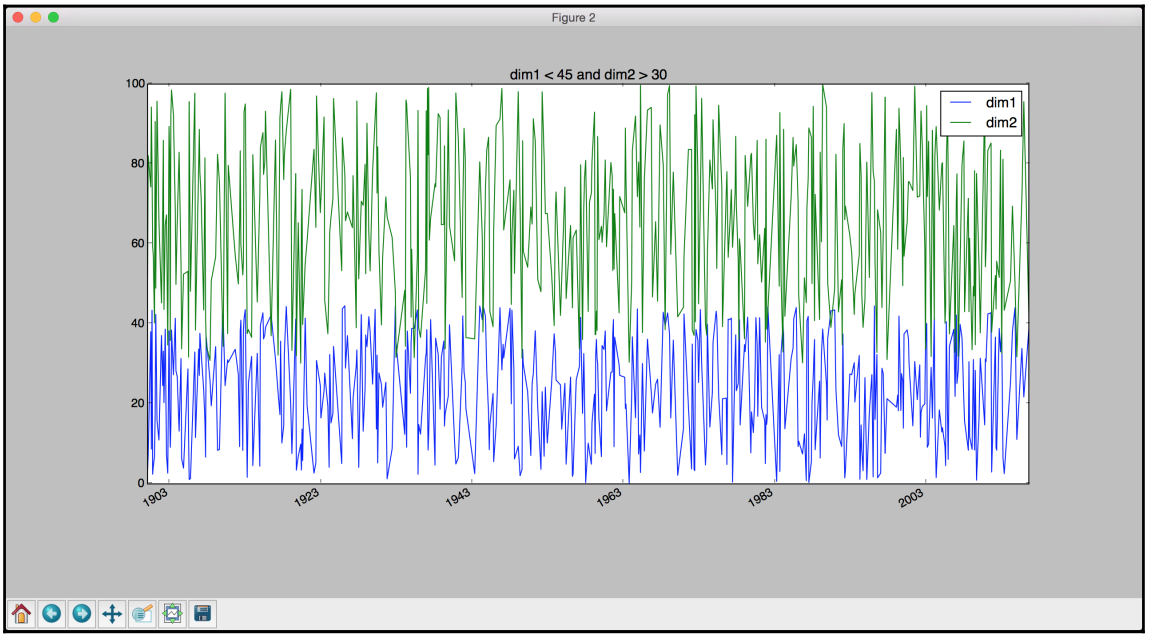


Figure 2
dim1 < 45 and dim2 > 30
100
80
60
40
20
0
1903
1923
1943
1963
1983
2003
dim1
dim2

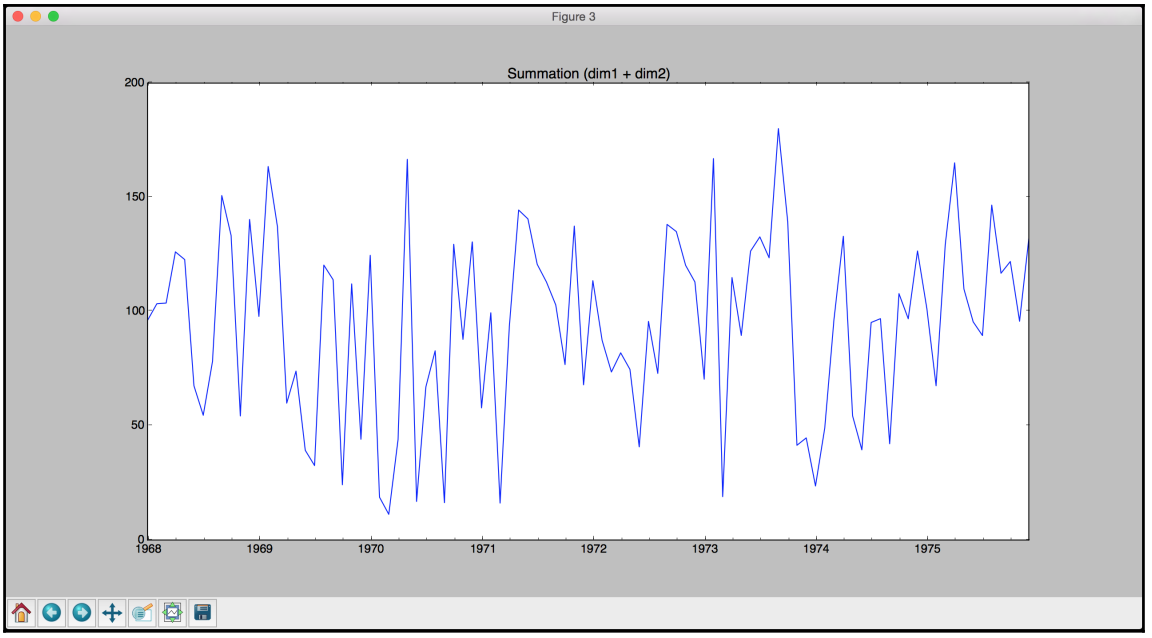


Figure 3
Summation (dim1 + dim2)
200
150
100
50
0
1968
1969
1970
1971
1972
1973
1974
1975

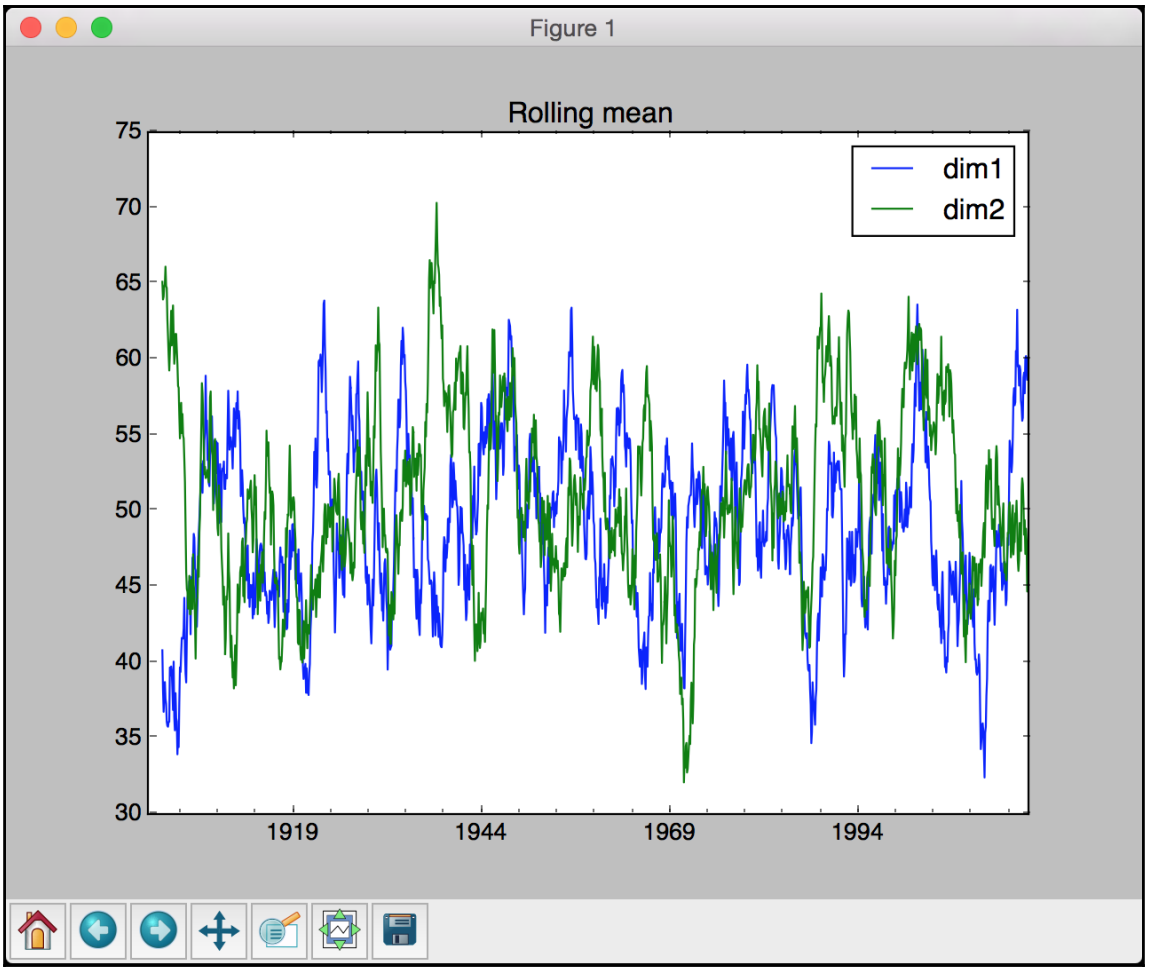


Figure 1
Rolling mean
dim1
dim2
75
70
65
60
55
50
45
40
35
30
1919
1944
1969
1994

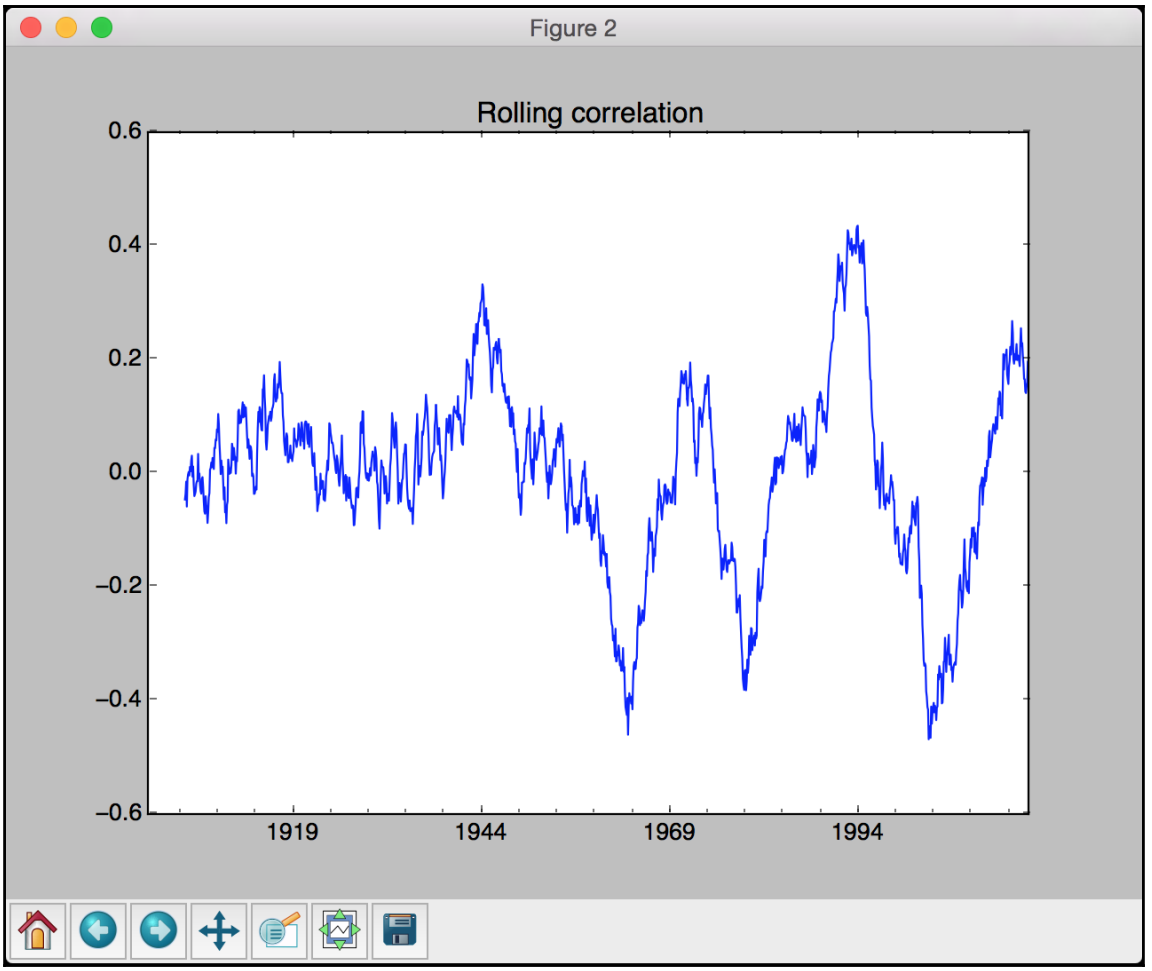


Figure 2
Rolling correlation
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
1919
1944
1969
1994

```
Maximum values for each dimension:
dim1    99.98
dim2    99.97
dtype: float64

Minimum values for each dimension:
dim1    0.18
dim2    0.16
dtype: float64

Overall mean:
dim1    49.030541
dim2    50.983291
dtype: float64
```

```
Row-wise mean:
1900-01-31    85.595
1900-02-28    75.310
1900-03-31    27.700
1900-04-30    44.675
1900-05-31    31.295
1900-06-30    44.160
1900-07-31    67.415
1900-08-31    56.160
1900-09-30    51.495
1900-10-31    61.260
1900-11-30    30.925
1900-12-31    30.785
Freq: M, dtype: float64

Correlation coefficients:
          dim1     dim2
dim1  1.00000  0.00627
dim2  0.00627  1.00000
```

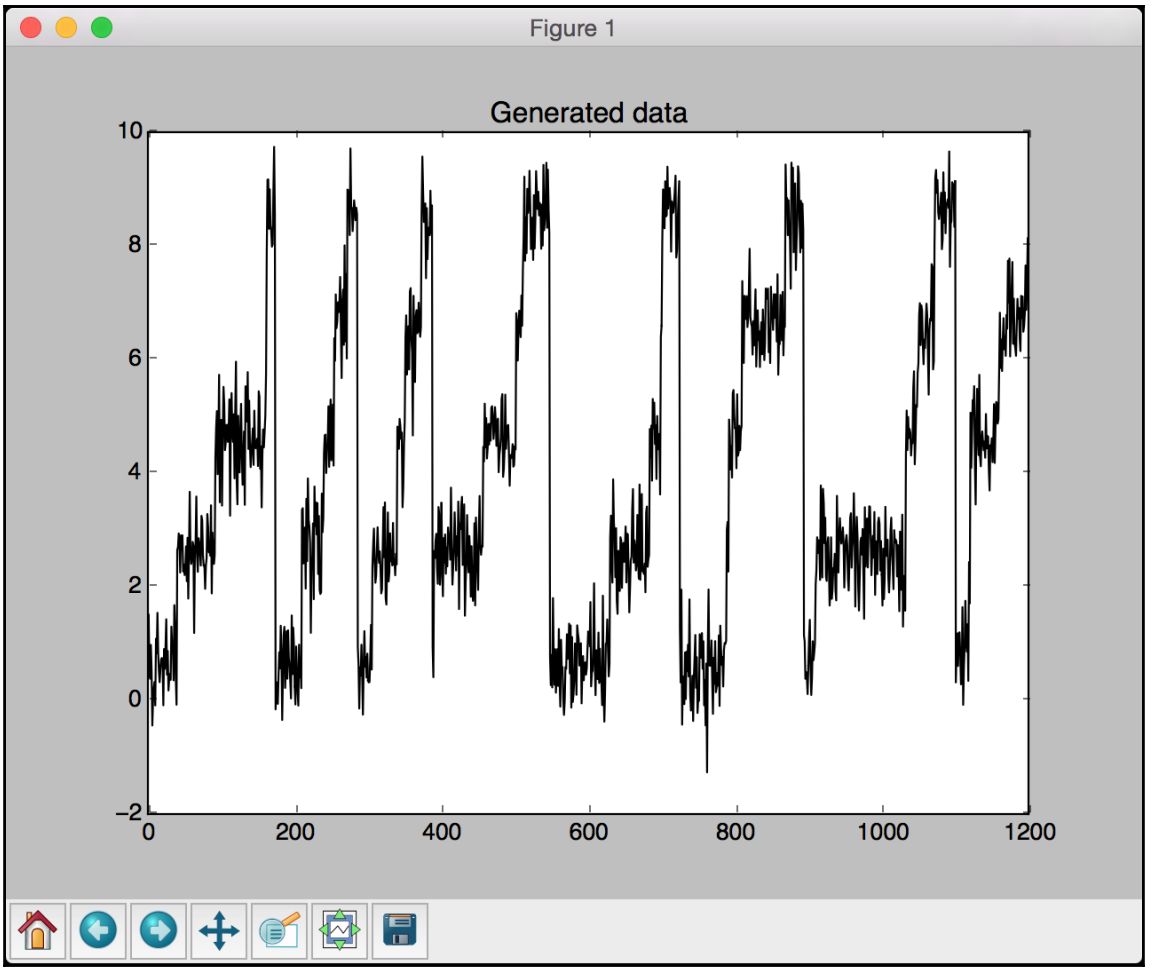


Figure 1
Generated data
10
8
6
4
2
0
−2
0
200
400
600
800
1000
1200

```
Training the Hidden Markov Model...

Means and variances:

Hidden state 1
Mean = 4.6
Variance = 0.25

Hidden state 2
Mean = 6.59
Variance = 0.25

Hidden state 3
Mean = 0.6
Variance = 0.25

Hidden state 4
Mean = 8.6
Variance = 0.26

Hidden state 5
Mean = 2.6
Variance = 0.26
```

```
Training the CRF model...

Accuracy score = 77.93%

Original  = rojections
Predicted = rojectiong

Original  = uff
Predicted = ufr

Original  = kiing
Predicted = kiing

Original  = ecompress
Predicted = ecomertig

Original  = uzz
Predicted = vex

Original  = poiling
Predicted = aciting
```

```
Original  = abulously
Predicted = abuloualy

Original  = ormalization
Predicted = ormalisation

Original  = ake
Predicted = aka

Original  = afeteria
Predicted = ateteria

Original  = obble
Predicted = obble

Original  = hadow
Predicted = habow

Original  = ndustrialized
Predicted = ndusqrialyled

Original  = ympathetically
Predicted = ympnshetically
```

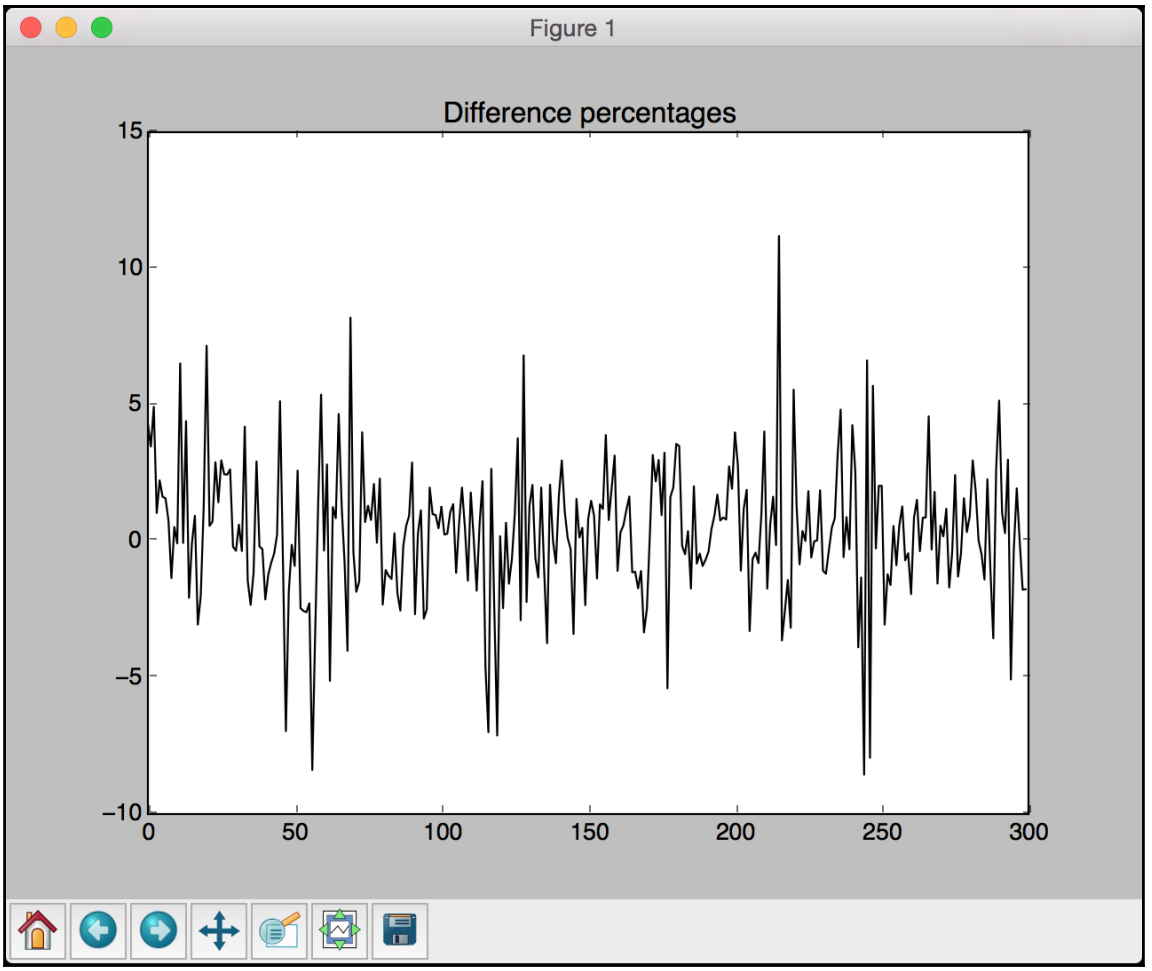
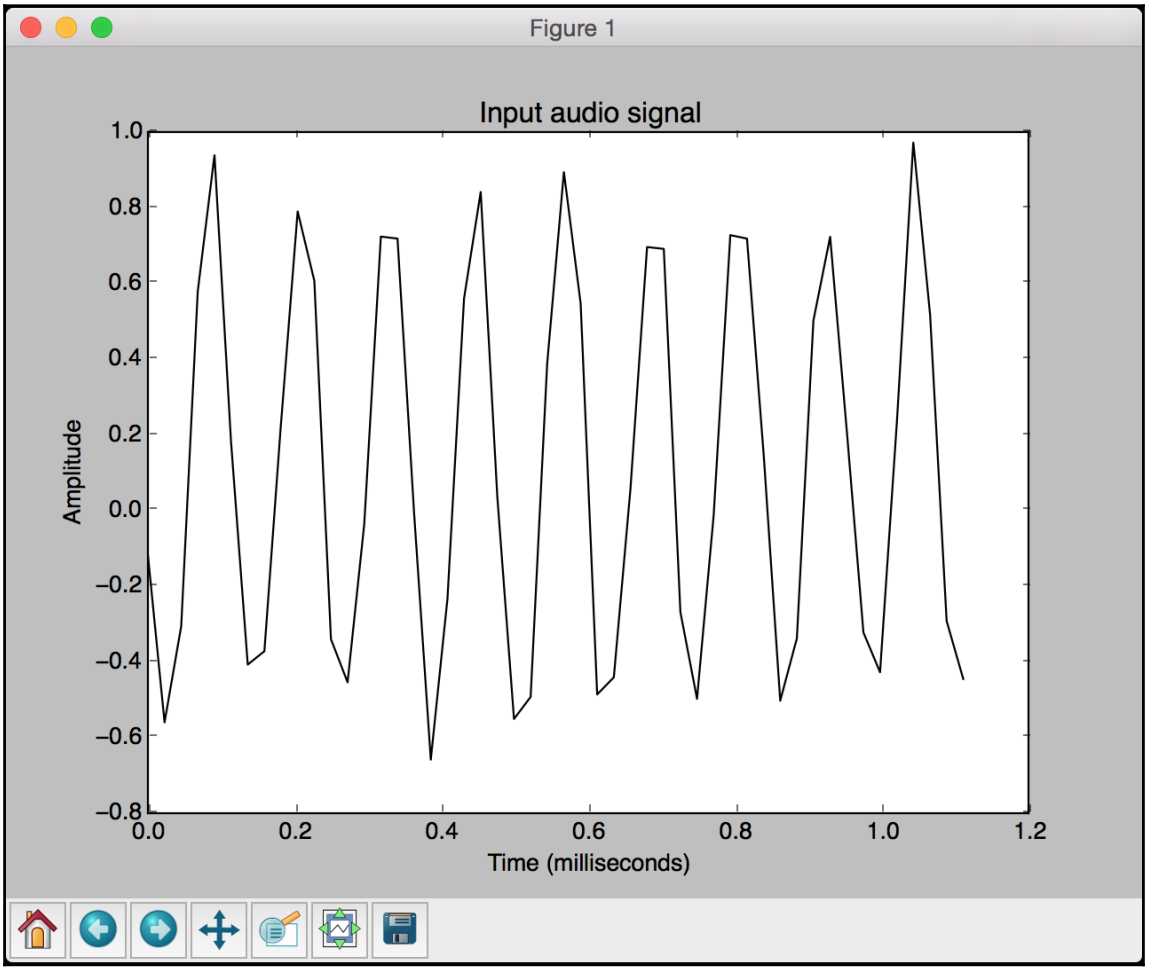


Figure 1
Difference percentages
15
10
5
0
−5
−10
0
50
100
150
200
250
300

Chapter 12: Building A Speech Recognizer




```
Signal shape: (132300,)
Datatype: int16
Signal duration: 3.0 seconds
```

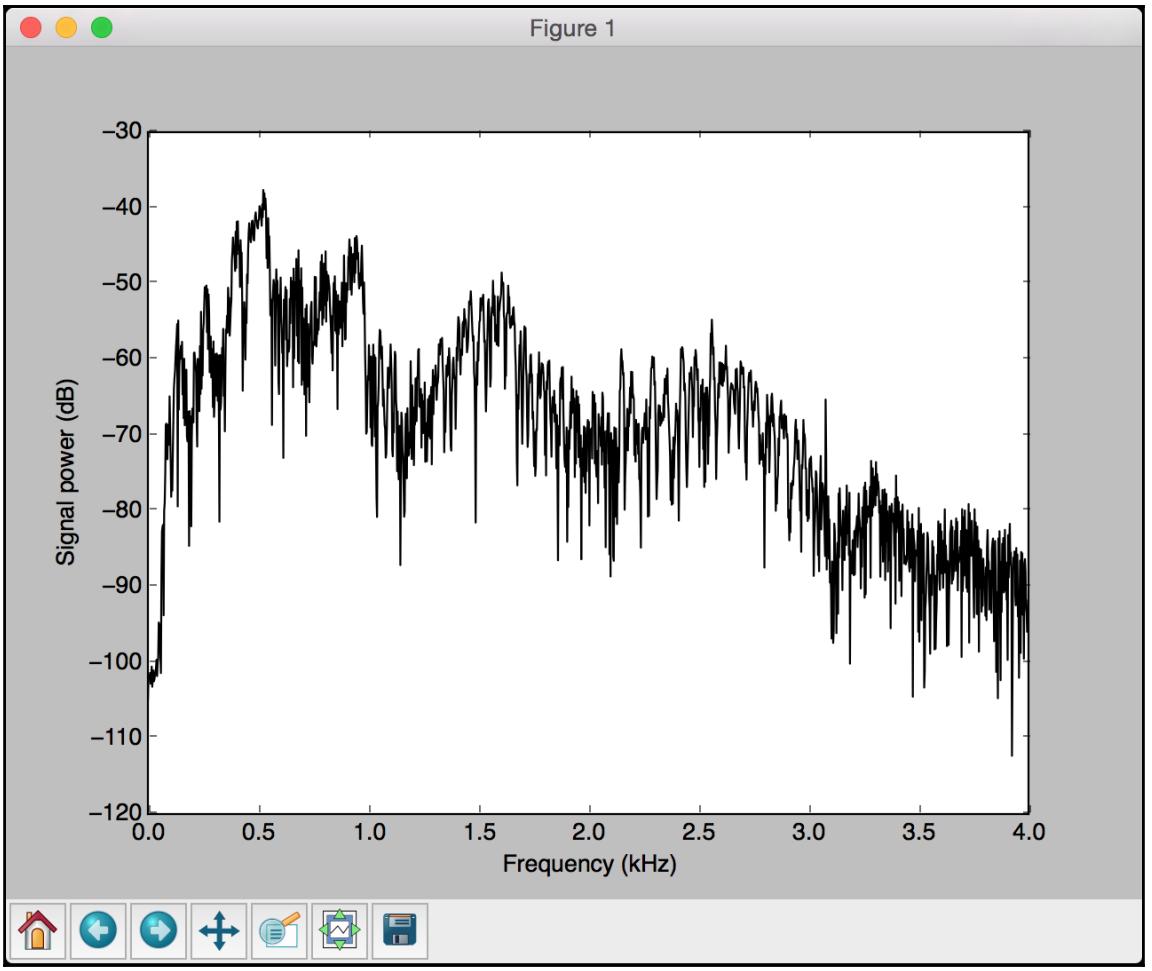


Figure 1
−30
−40
−50
−60
−70
−80
−90
−100
−110
−120
Signal power (dB)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Frequency (kHz)

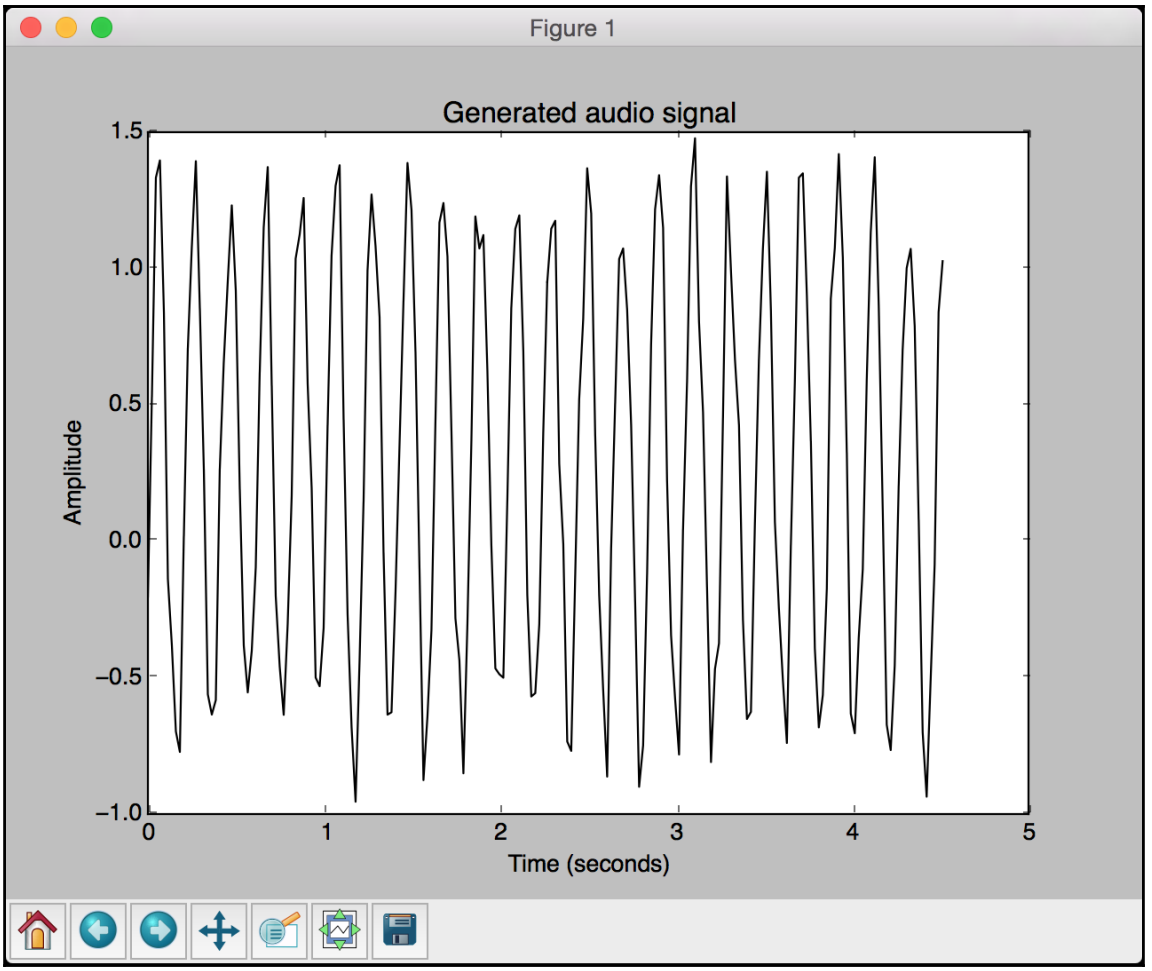


Figure 1
Generated audio signal
Amplitude
Time (seconds)
1.5
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5

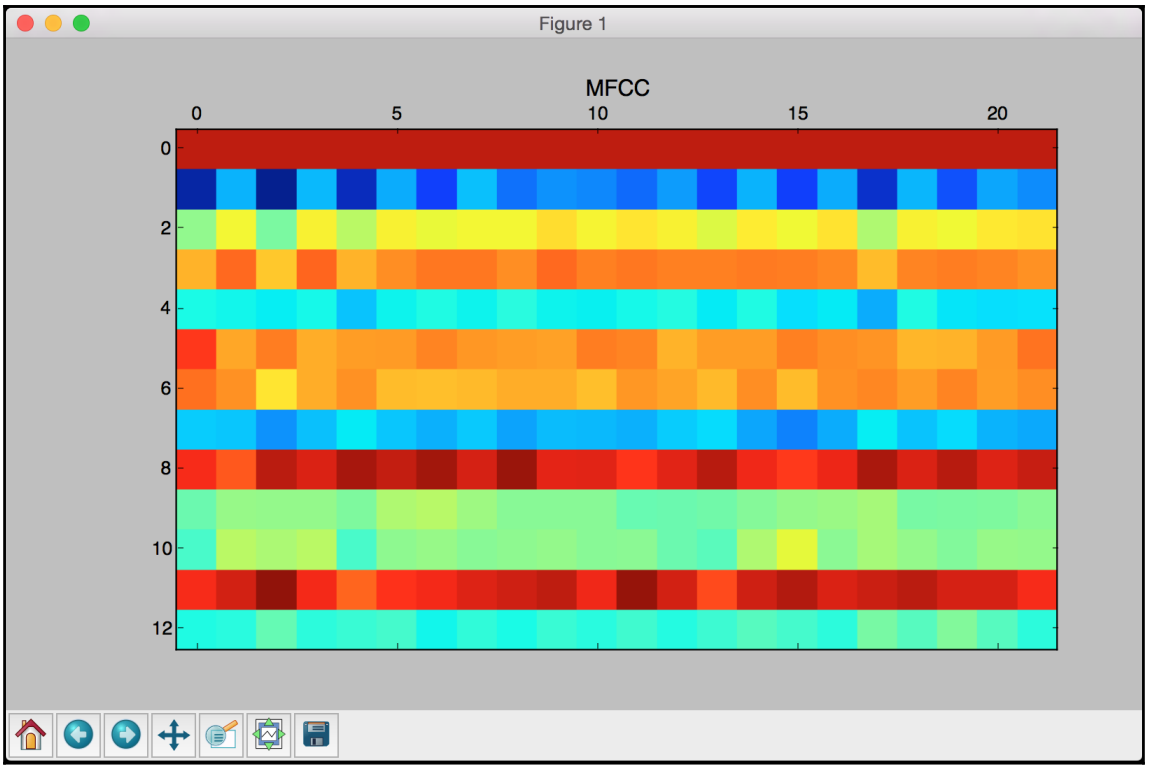


Figure 1
MFCC
0
5
10
15
20
0
2
4
6
8
10
12

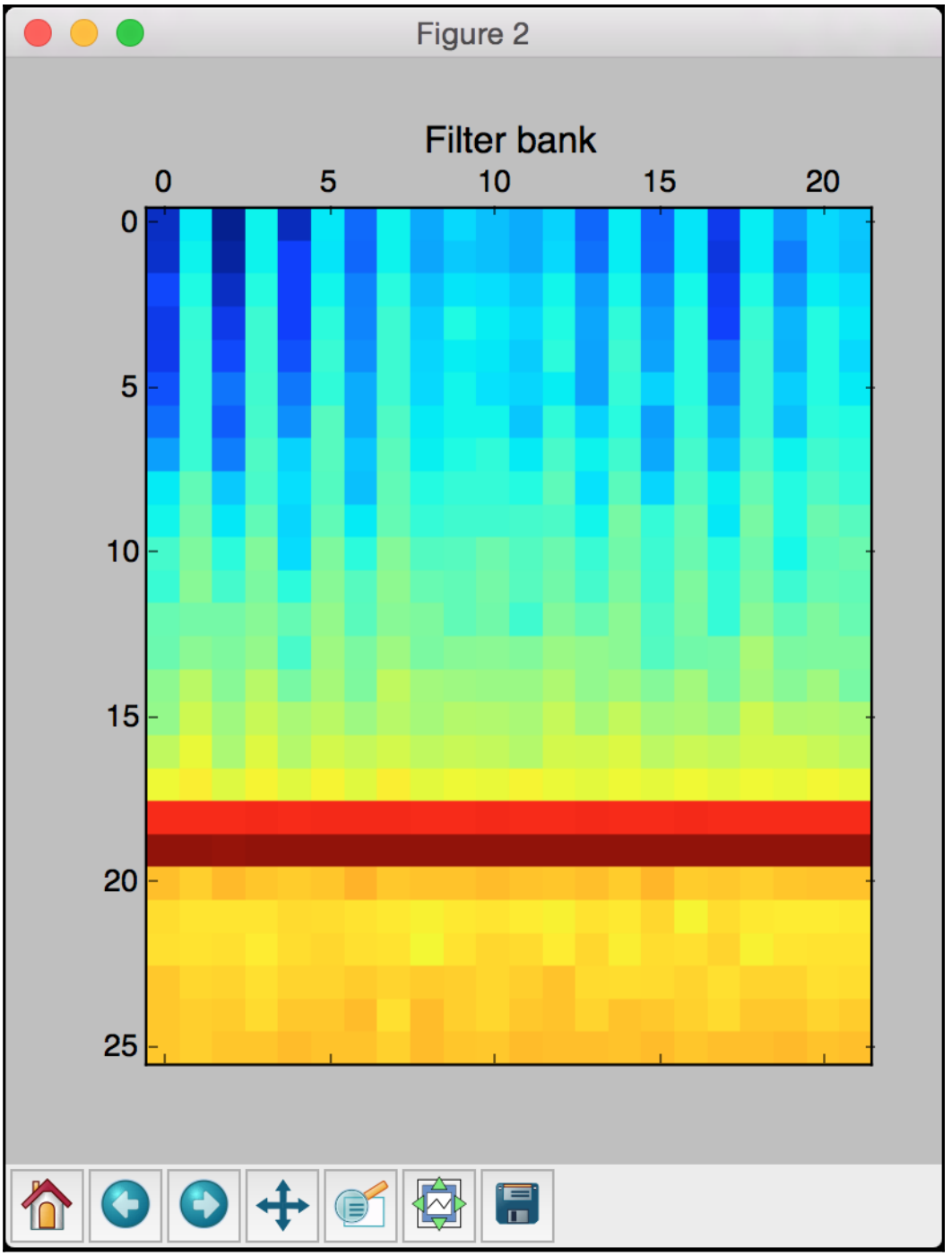


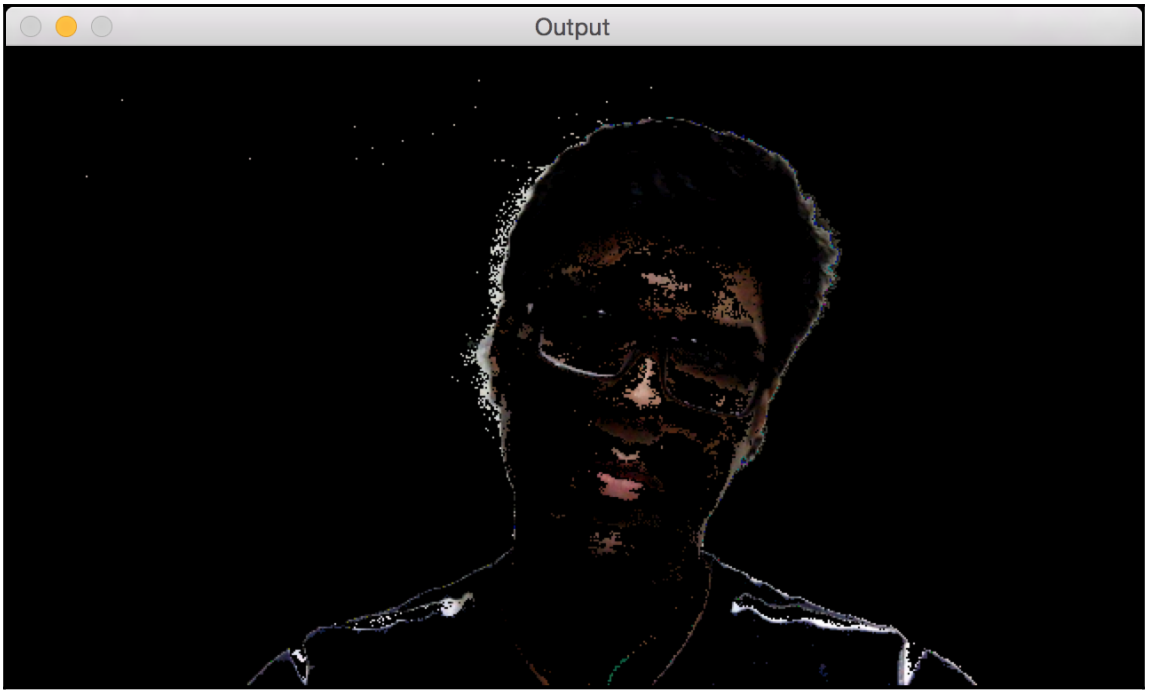
Figure 2
Filter bank
0
5
10
15
20
0
5
10
15
20
25

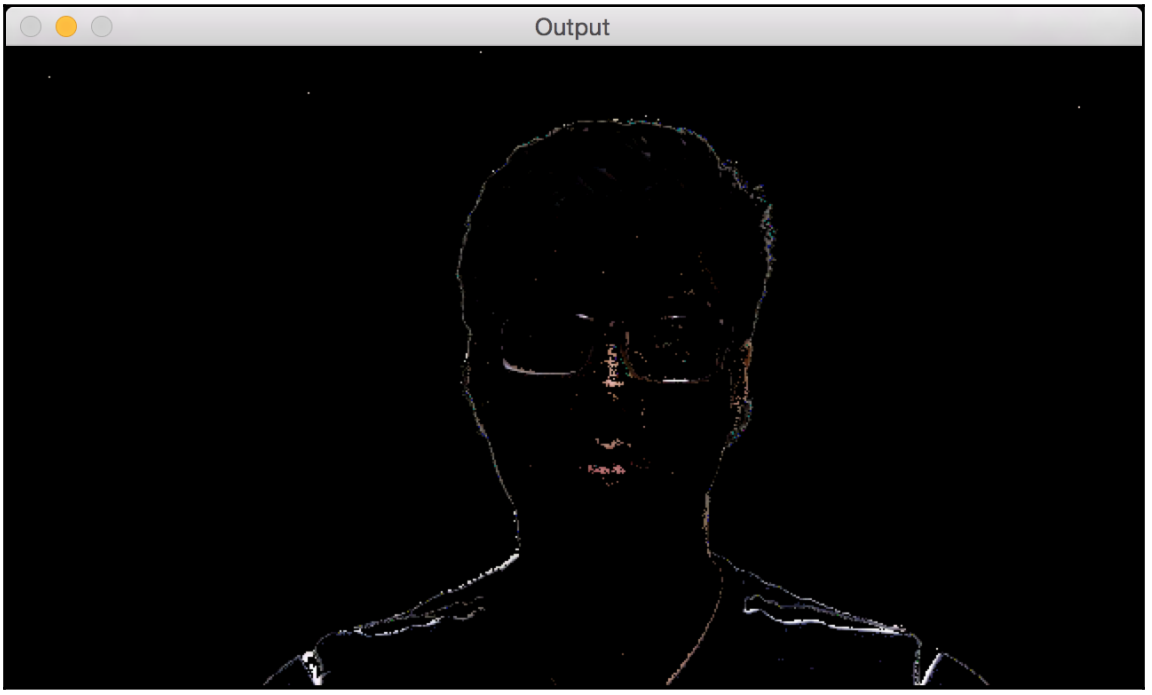
```
MFCC:
Number of windows = 22
Length of each feature = 13

Filter bank:
Number of windows = 22
Length of each feature = 26
```

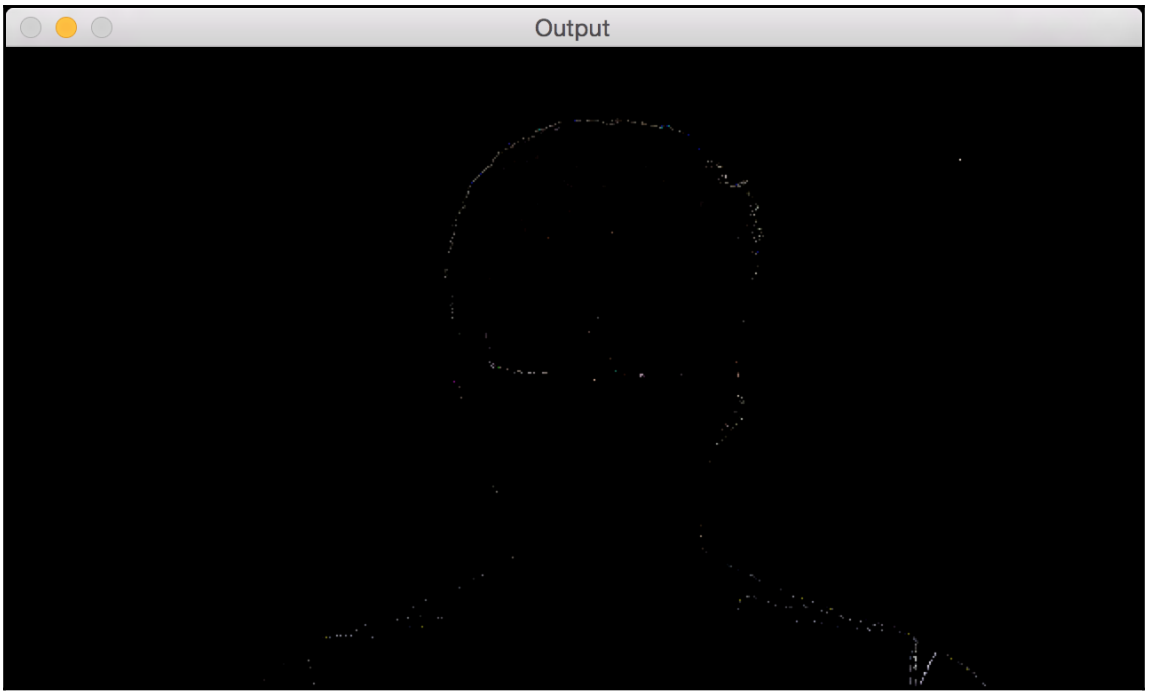
```
Original:  apple
Predicted: apple

Original:  banana
Predicted: banana

Original:  kiwi
Predicted: kiwi

Original:  lime
Predicted: lime

Original:  orange
Predicted: orange

Original:  peach
Predicted: peach

Original:  pineapple
Predicted: pineapple
```

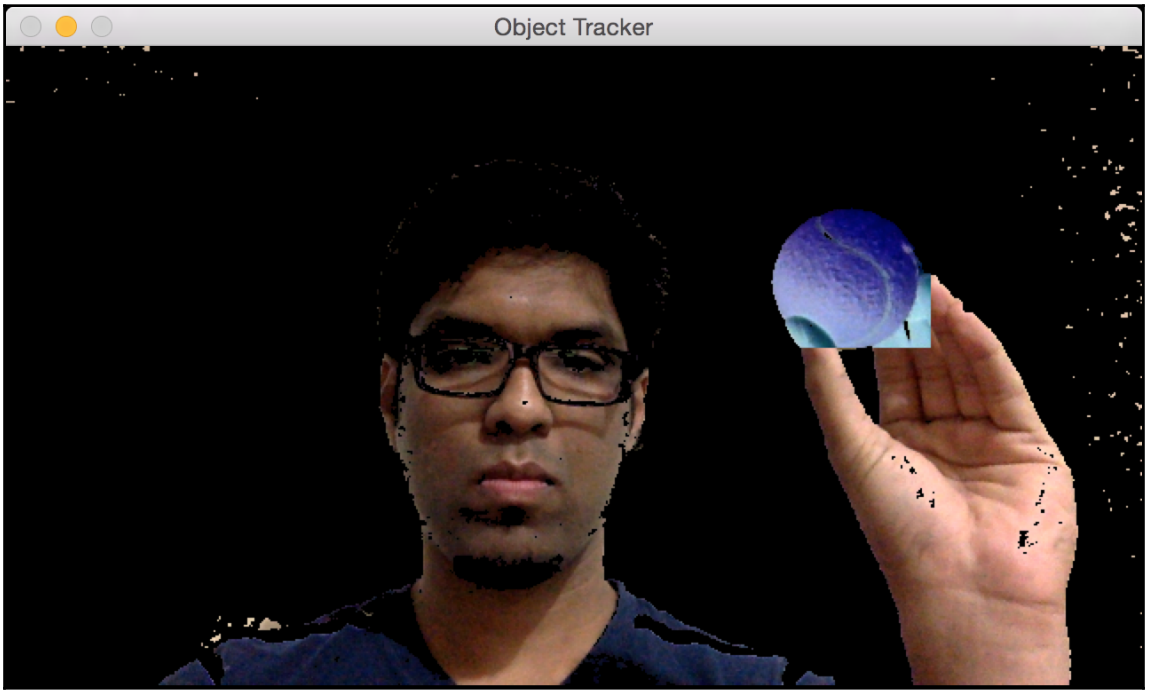
Chapter 13: Object Detection and Tracking

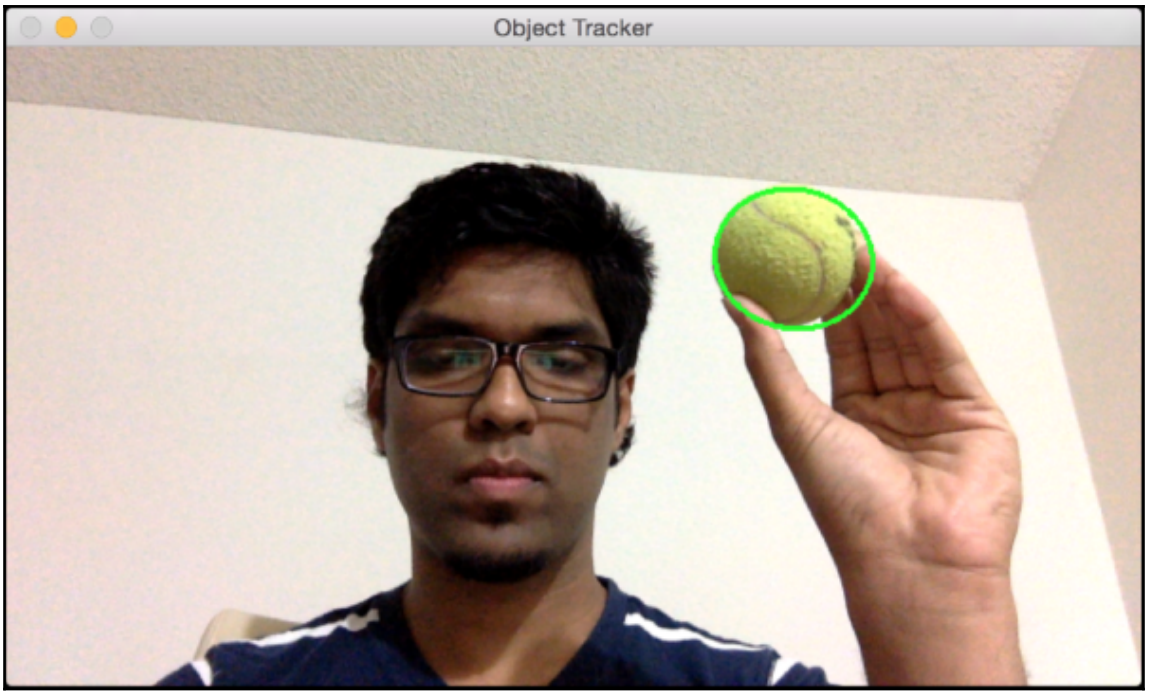

Input
Output

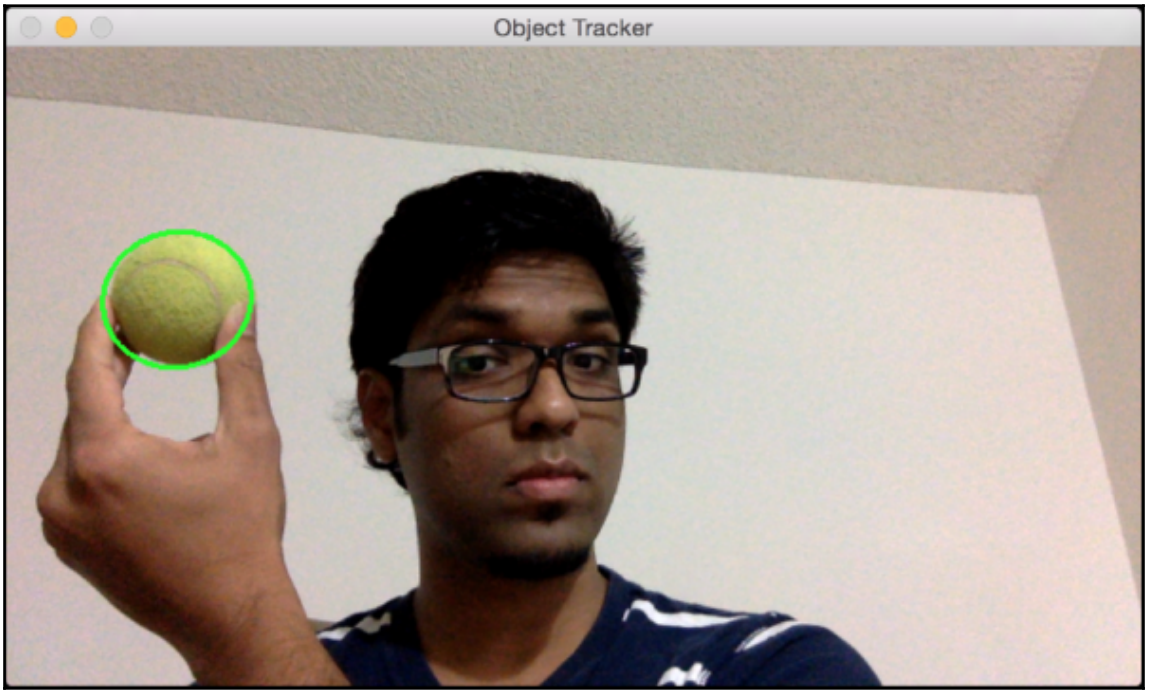

Output

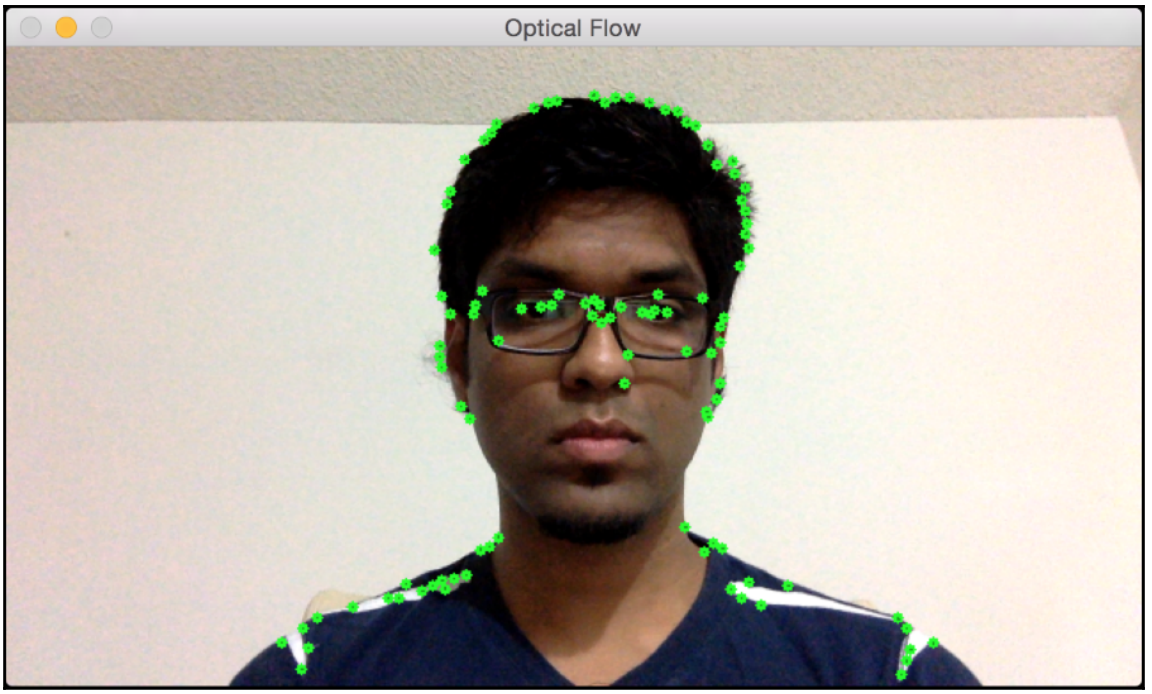

Output

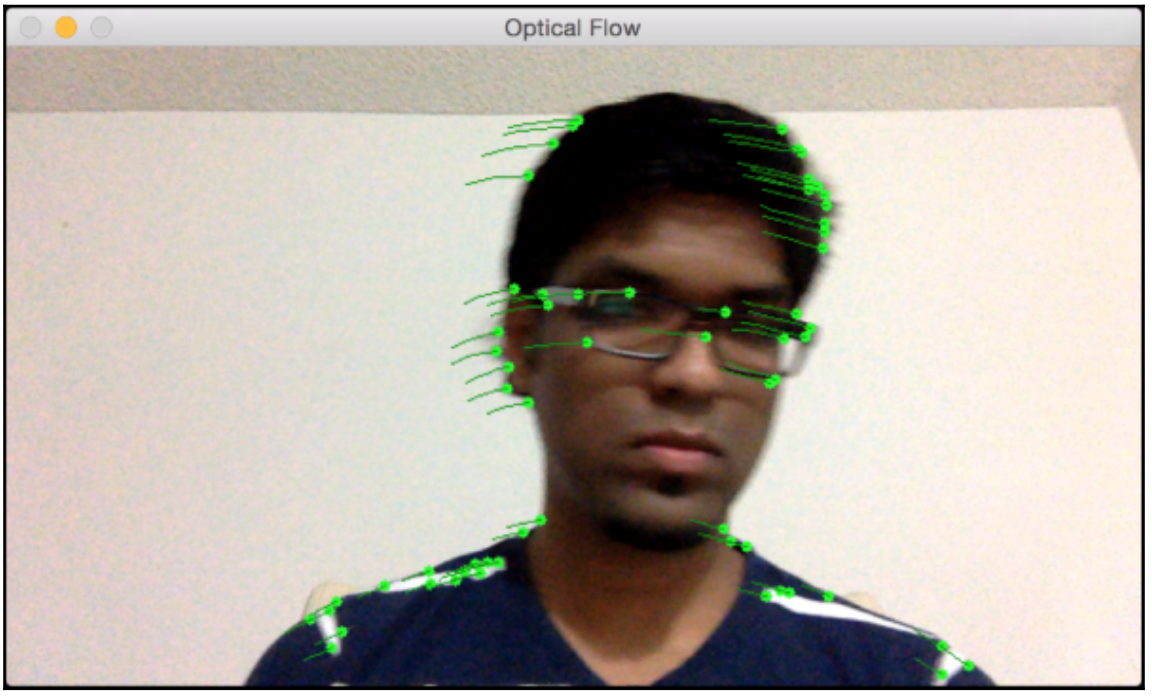

Output

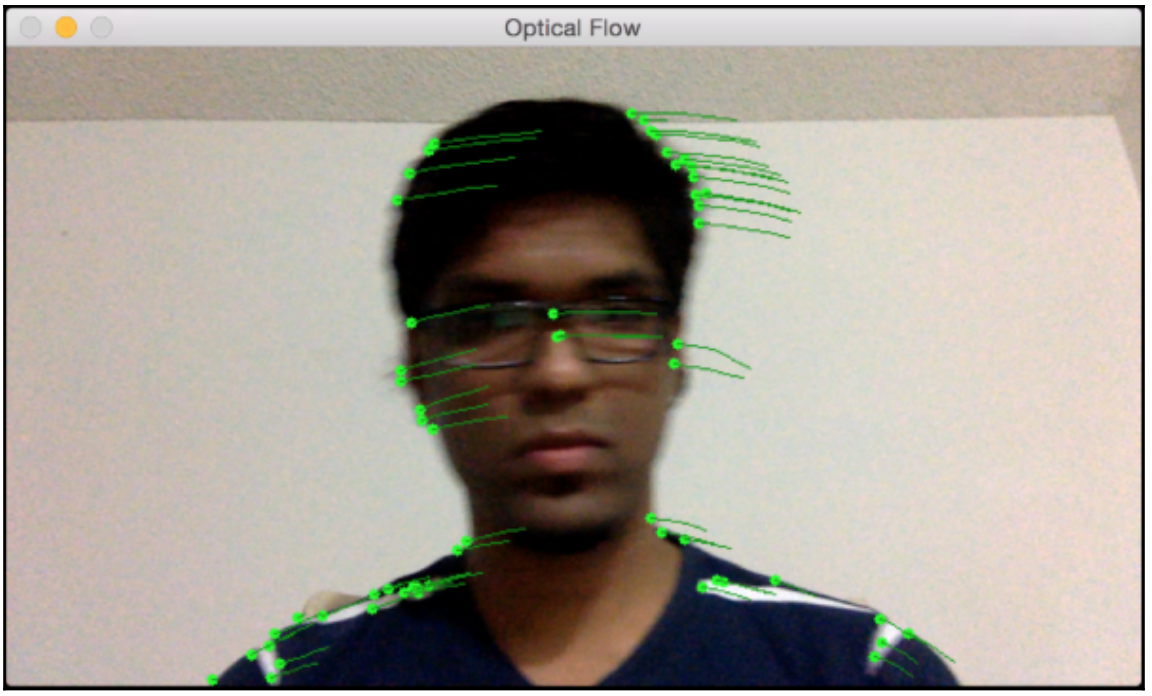

Object Tracker

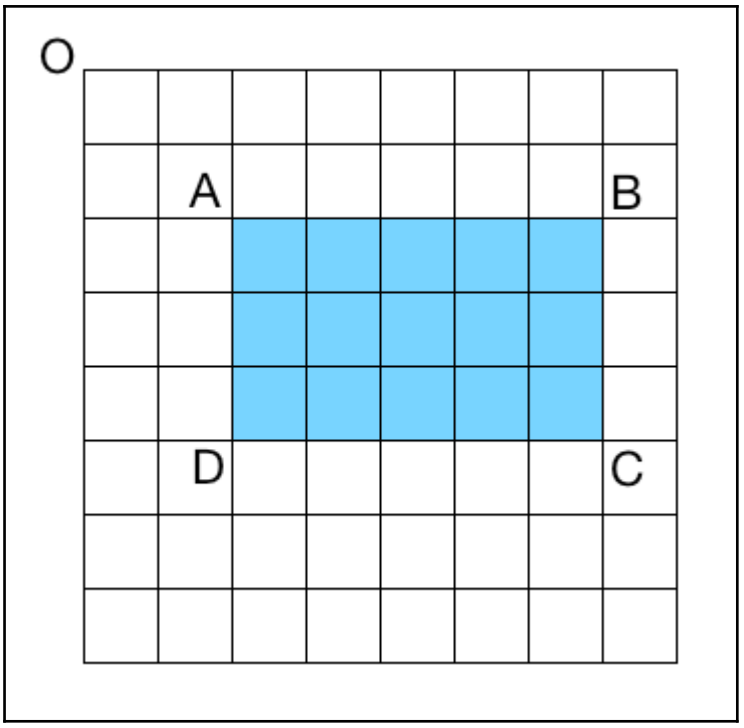

Object Tracker

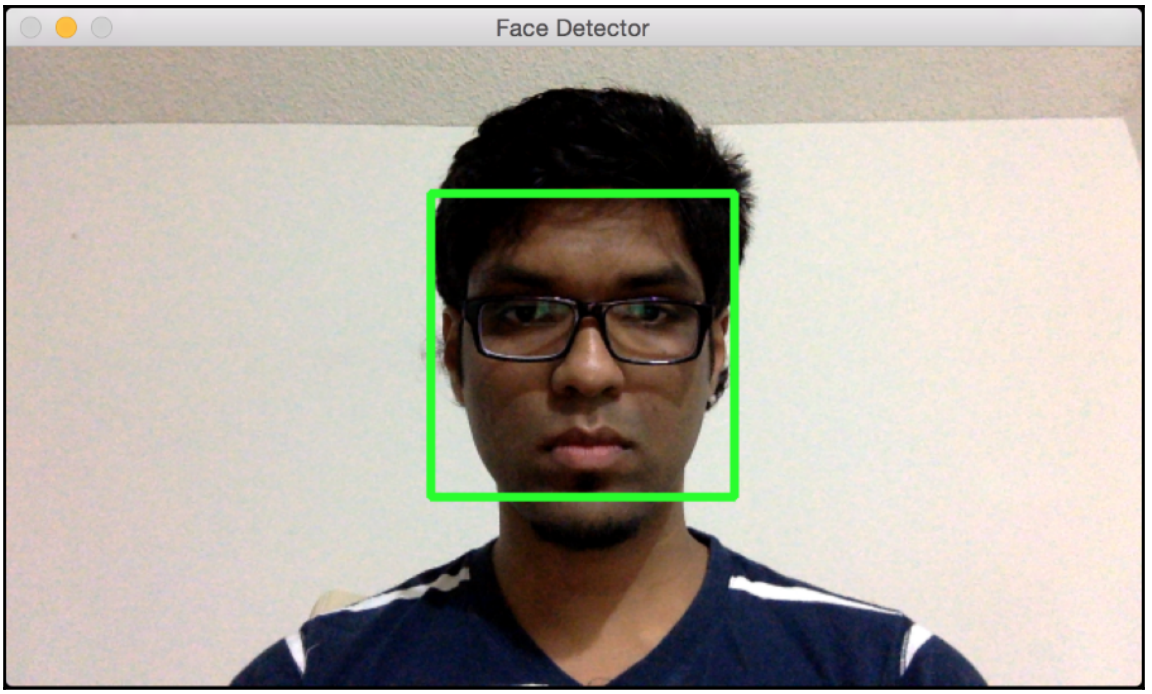

Object Tracker

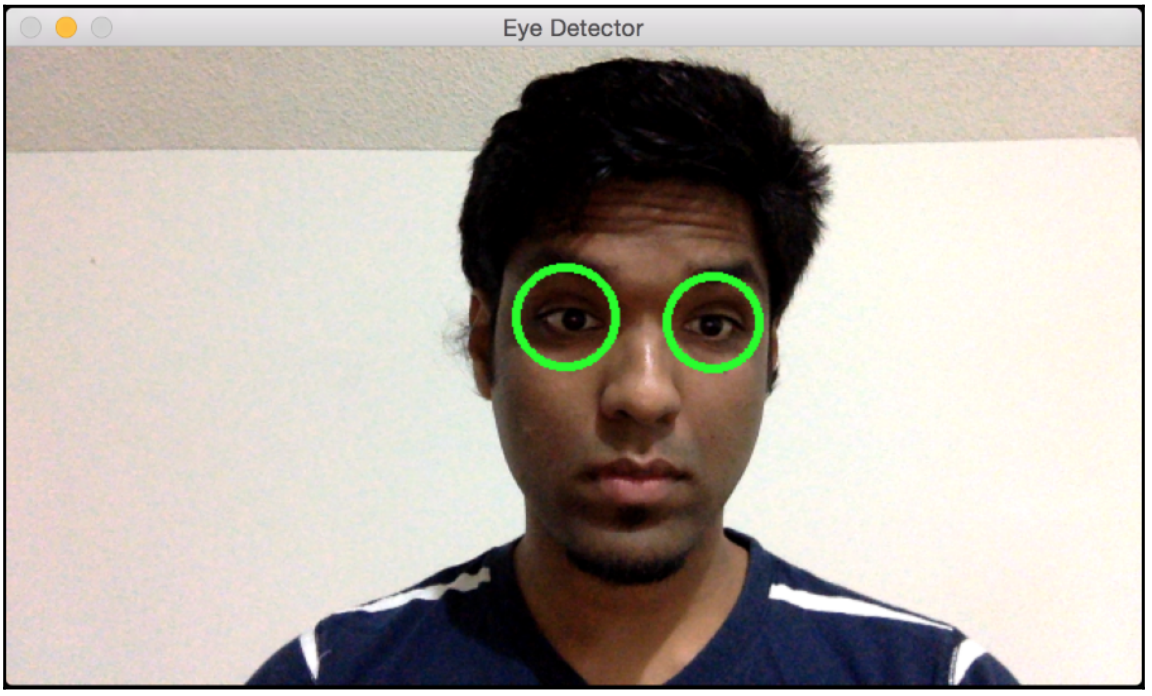

Optical Flow


Optical Flow

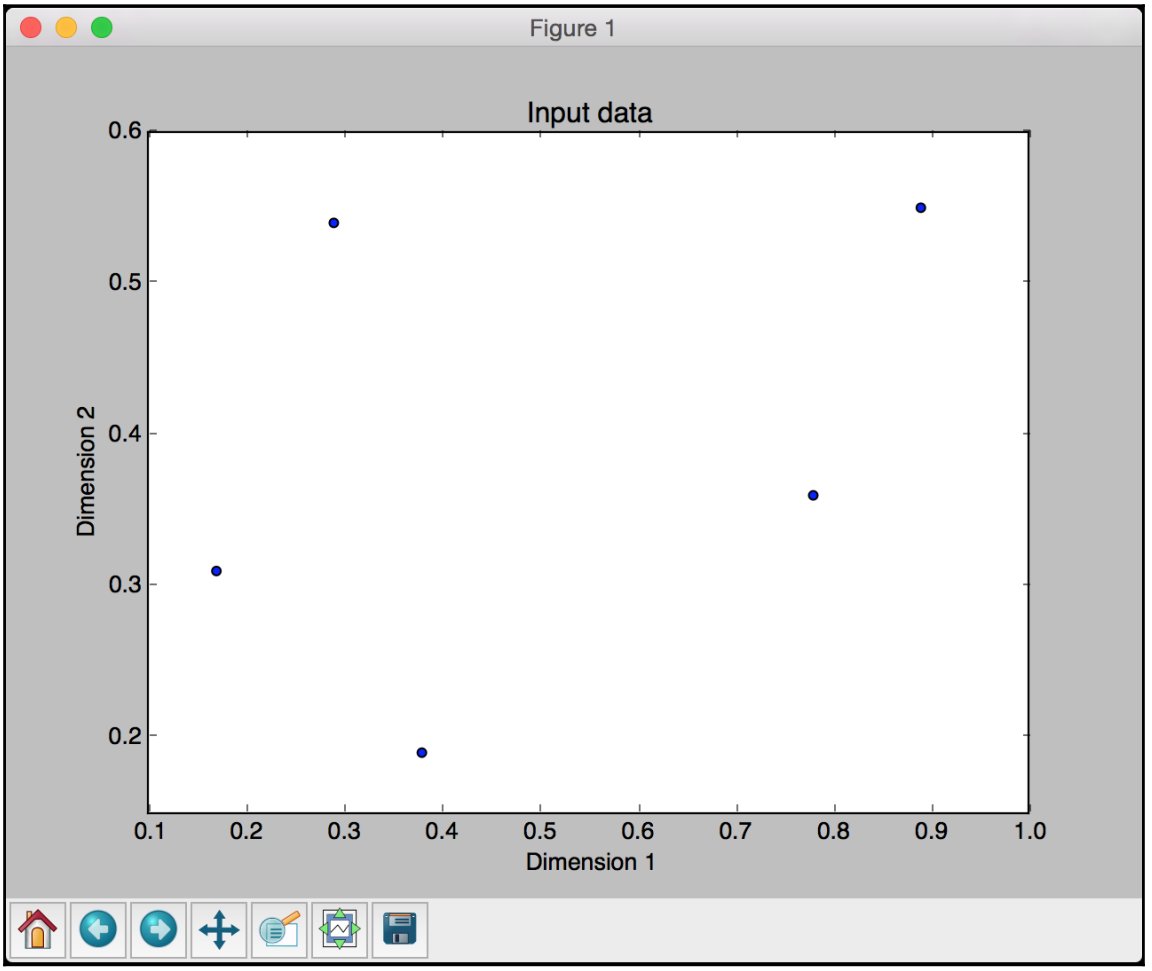

Optical Flow


O
A
B
D
C


Face Detector


Eye Detector

Chapter 14: Artificial Neural Networks



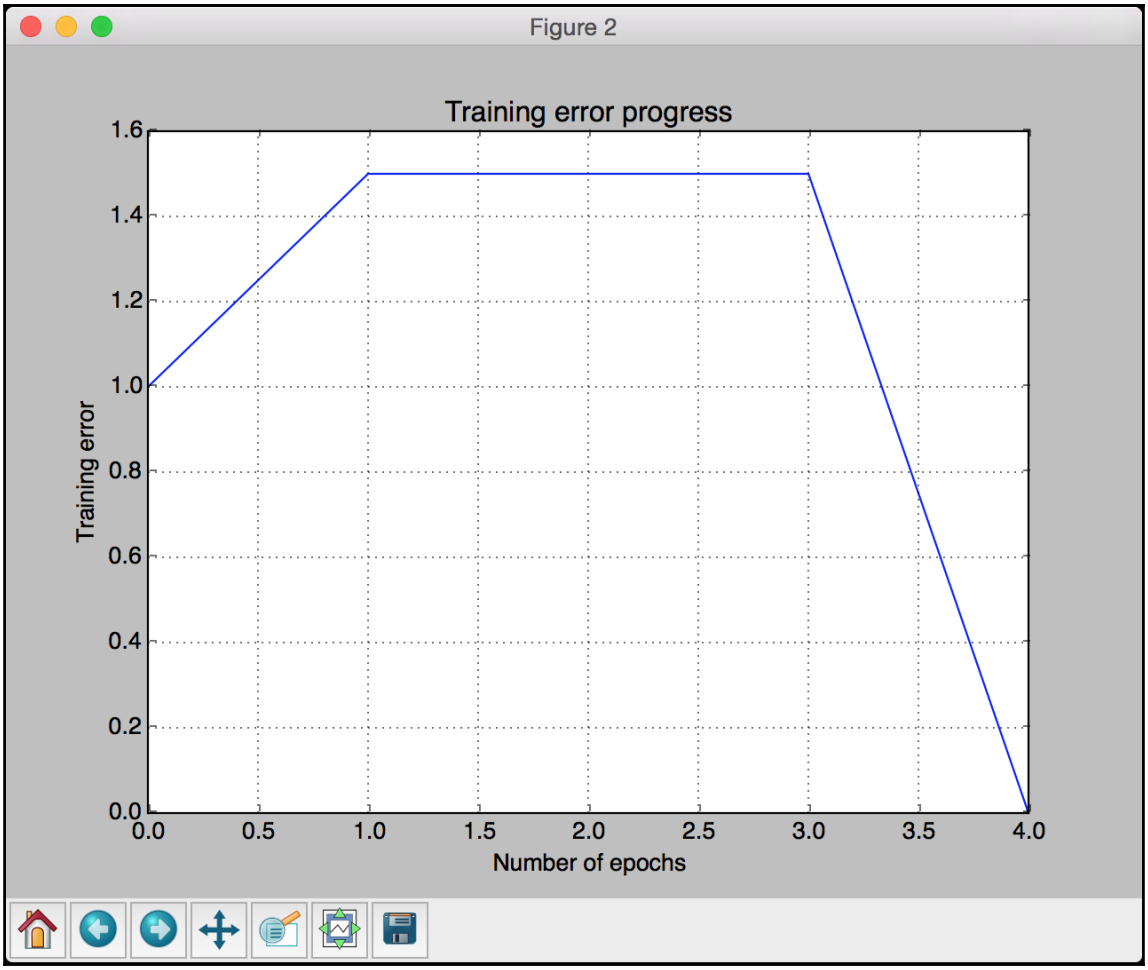


Figure 2
Training error progress
Training error
Number of epochs
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

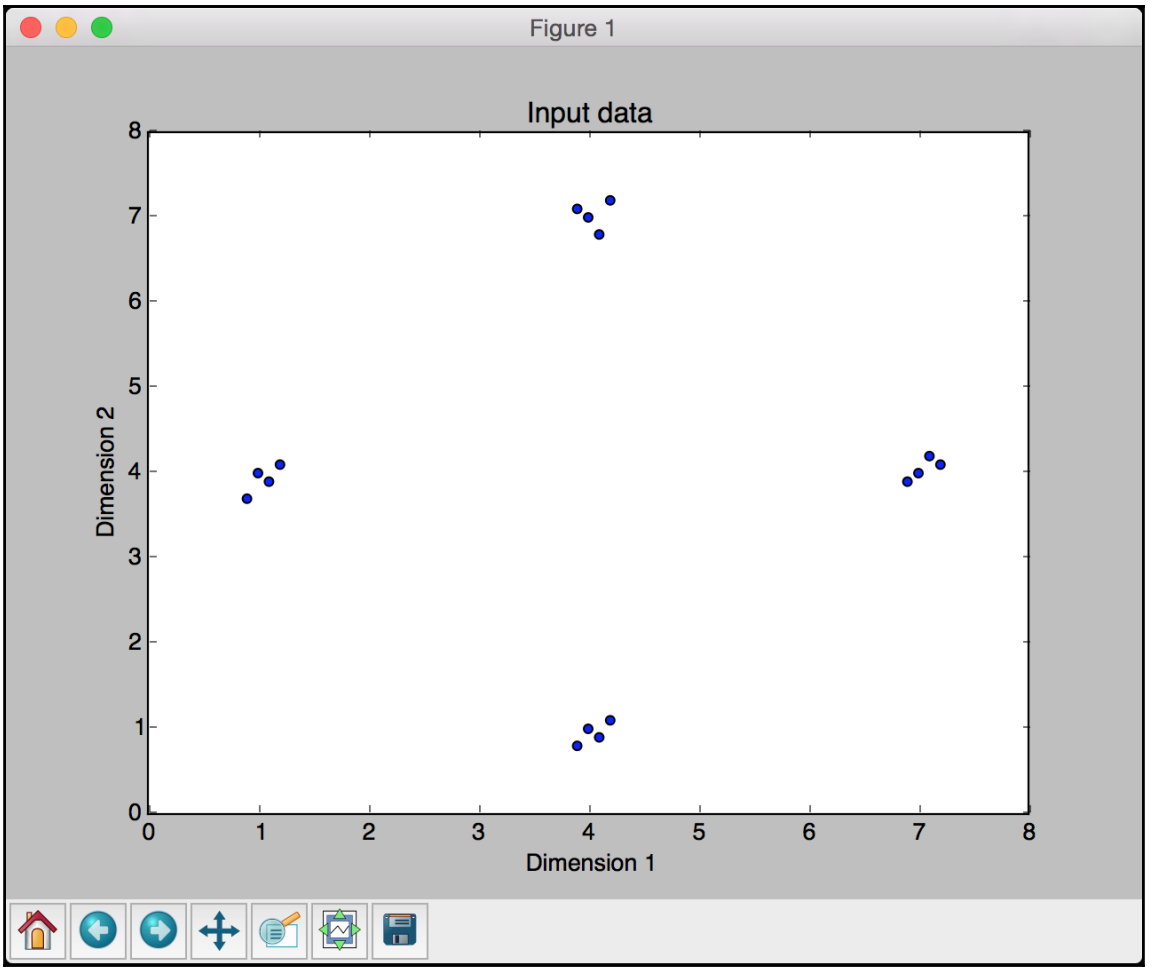


Figure 1
Input data
Dimension 2
Dimension 1
0
1
2
3
4
5
6
7
8

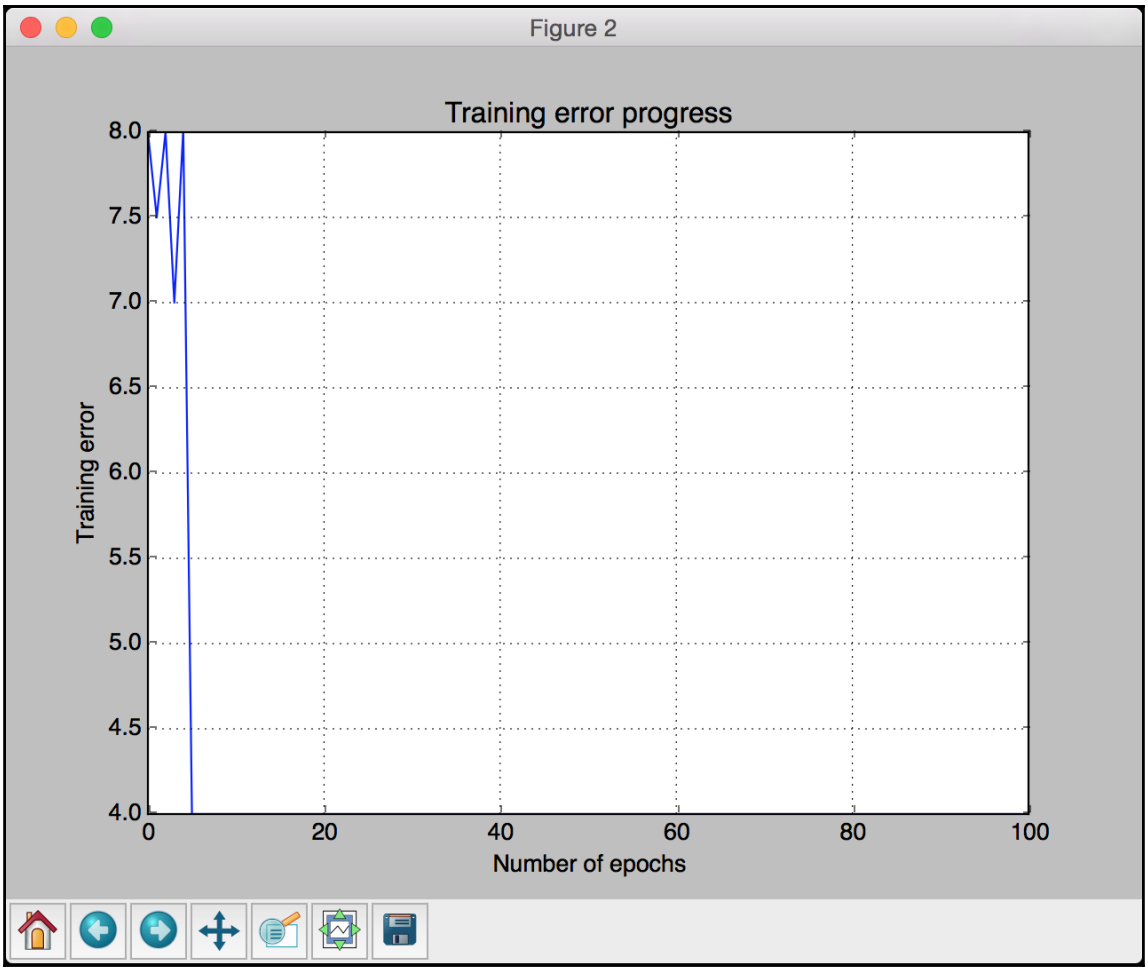


Figure 2
Training error progress
Training error
Number of epochs
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
4.0
0
20
40
60
80
100

```
Epoch: 20; Error: 4.0;
Epoch: 40; Error: 4.0;
Epoch: 60; Error: 4.0;
Epoch: 80; Error: 4.0;
Epoch: 100; Error: 4.0;
The maximum number of train epochs is reached

Test results:
[0.4, 4.3] --> [ 0.  0.]
[4.4, 0.6] --> [ 1.  0.]
[4.7, 8.1] --> [ 1.  1.]
```

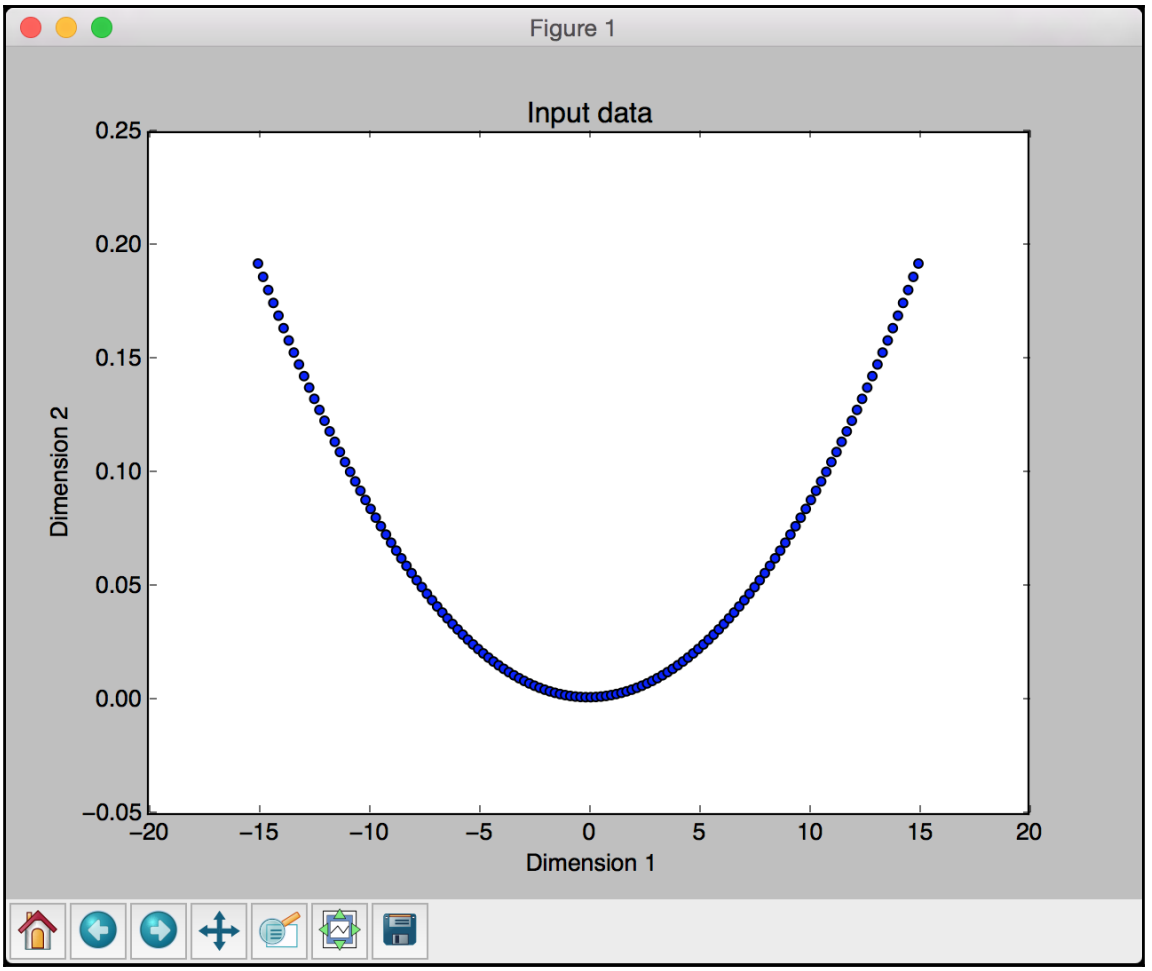


Figure 1
Input data
Dimension 2
Dimension 1
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
−20
−15
−10
−5
0
5
10
15
20

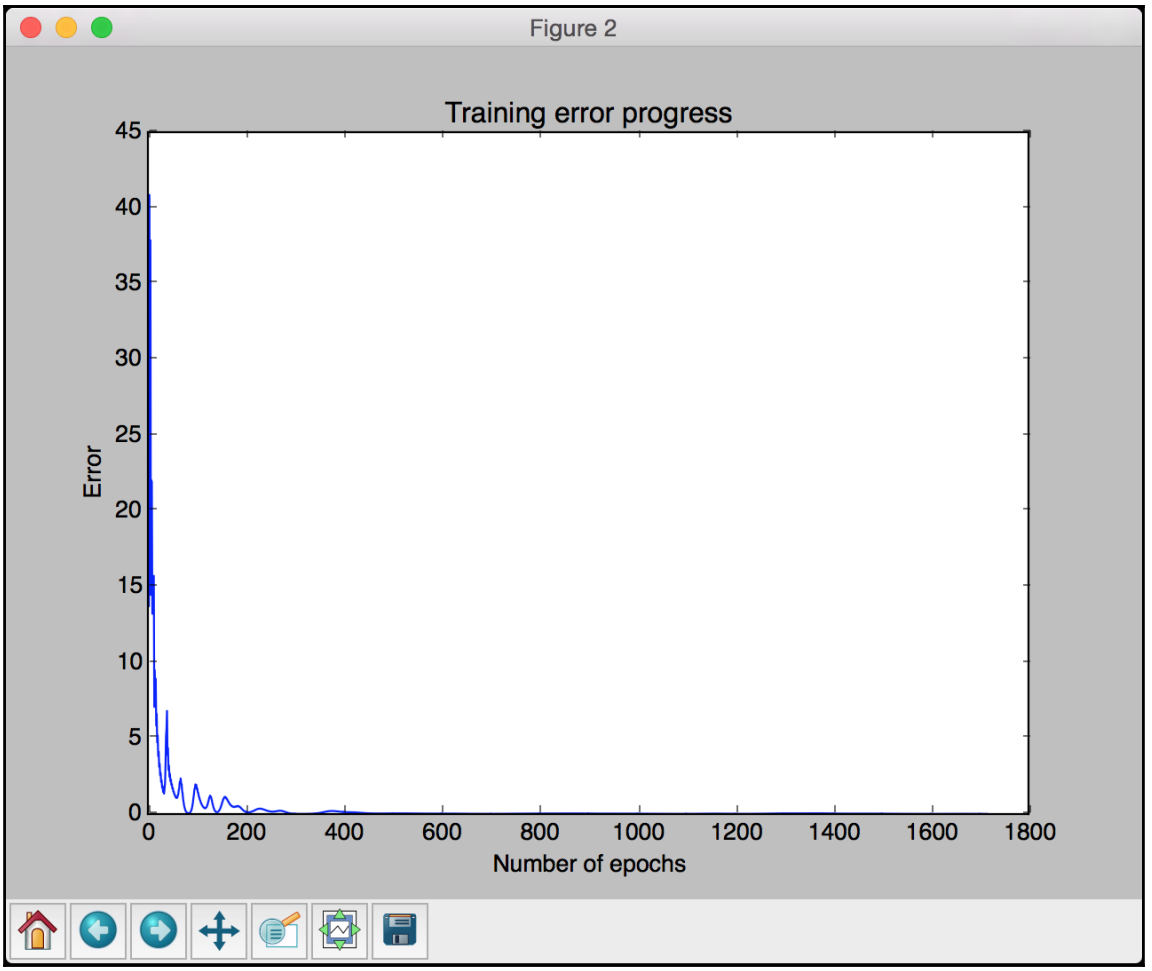


Figure 2
Training error progress
Error
Number of epochs
0
5
10
15
20
25
30
35
40
45
0
200
400
600
800
1000
1200
1400
1600
1800

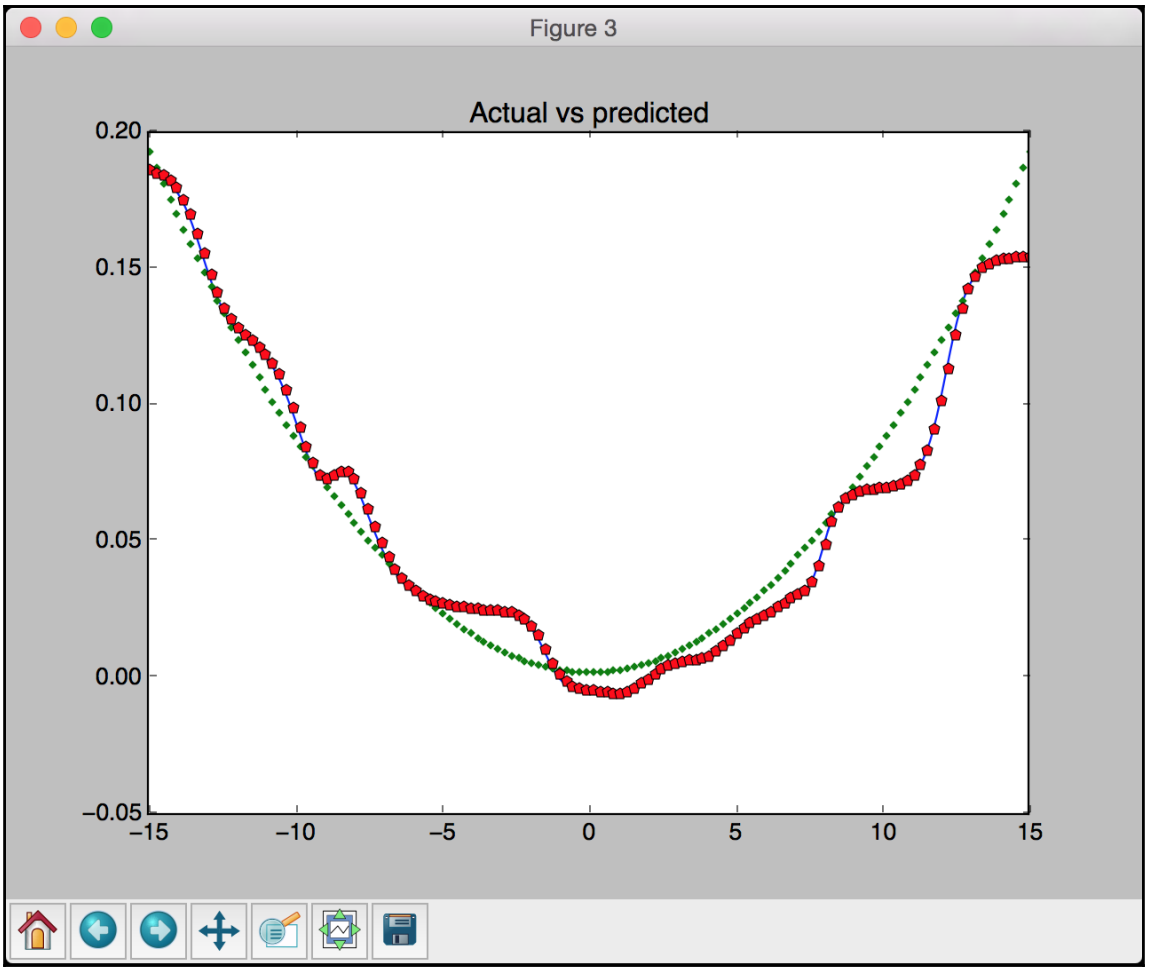
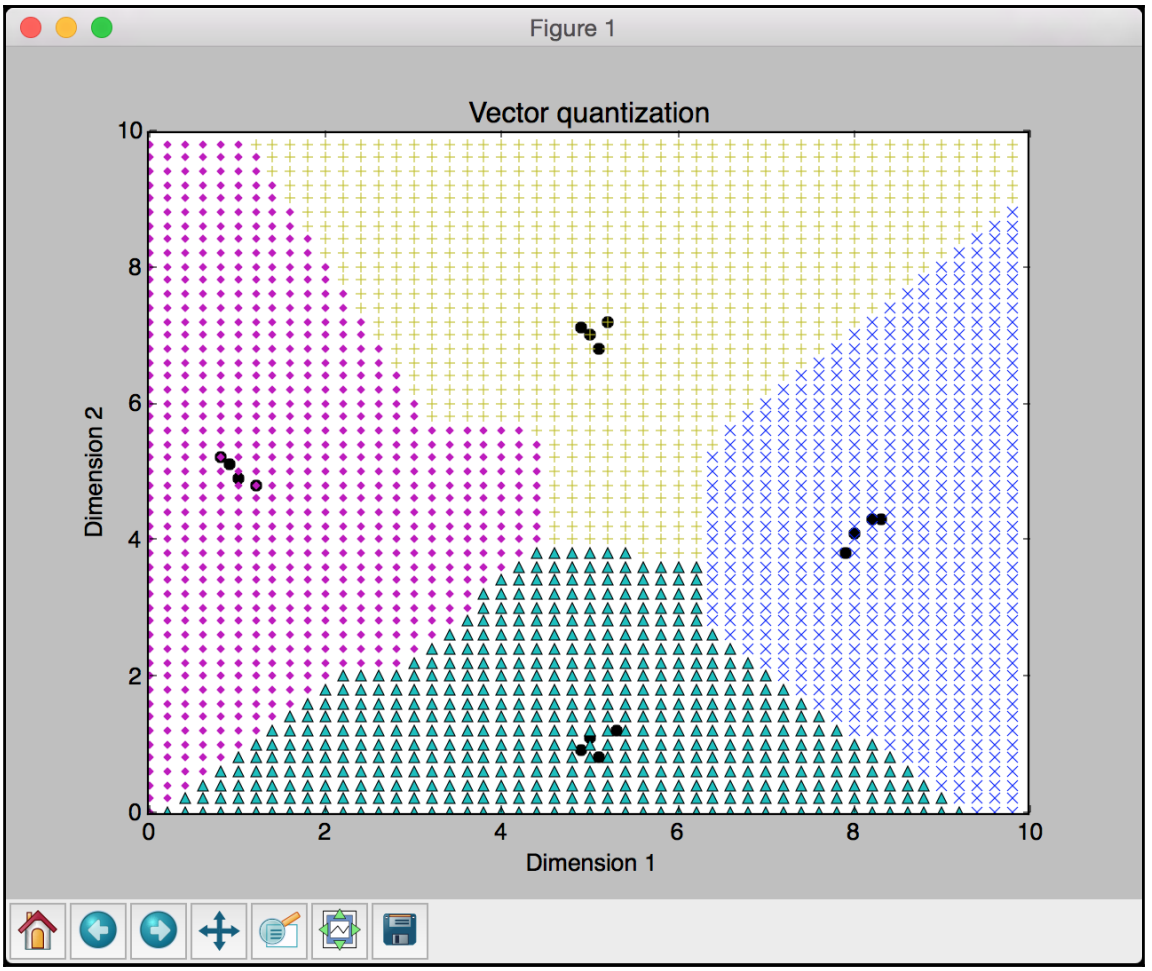


Figure 3
Actual vs predicted
0.20
0.15
0.10
0.05
0.00
−0.05
−15
−10
−5
0
5
10
15

```
Epoch: 100; Error: 1.9247718251621995;
Epoch: 200; Error: 0.15723294798079526;
Epoch: 300; Error: 0.021680213116912858;
Epoch: 400; Error: 0.1381761995539017;
Epoch: 500; Error: 0.043925533819487 37;
Epoch: 600; Error: 0.02975401597014979;
Epoch: 700; Error: 0.014228560930227126;
Epoch: 800; Error: 0.03460207842970052;
Epoch: 900; Error: 0.035934053149433196;
Epoch: 1000; Error: 0.025833284445815966;
Epoch: 1100; Error: 0.013672412879982398;
Epoch: 1200; Error: 0.01776586425692384;
Epoch: 1300; Error: 0.04310242610384976;
Epoch: 1400; Error: 0.03799681296096611;
Epoch: 1500; Error: 0.02467030041520845;
Epoch: 1600; Error: 0.010094873168855236;
Epoch: 1700; Error: 0.01210866043021068;
The goal of learning is reached
```




```
Epoch: 100; Error: 0.0;
Epoch: 200; Error: 0.0;
Epoch: 300; Error: 0.0;
Epoch: 400; Error: 0.0;
Epoch: 500; Error: 0.0;
The maximum number of train epochs is reached
```

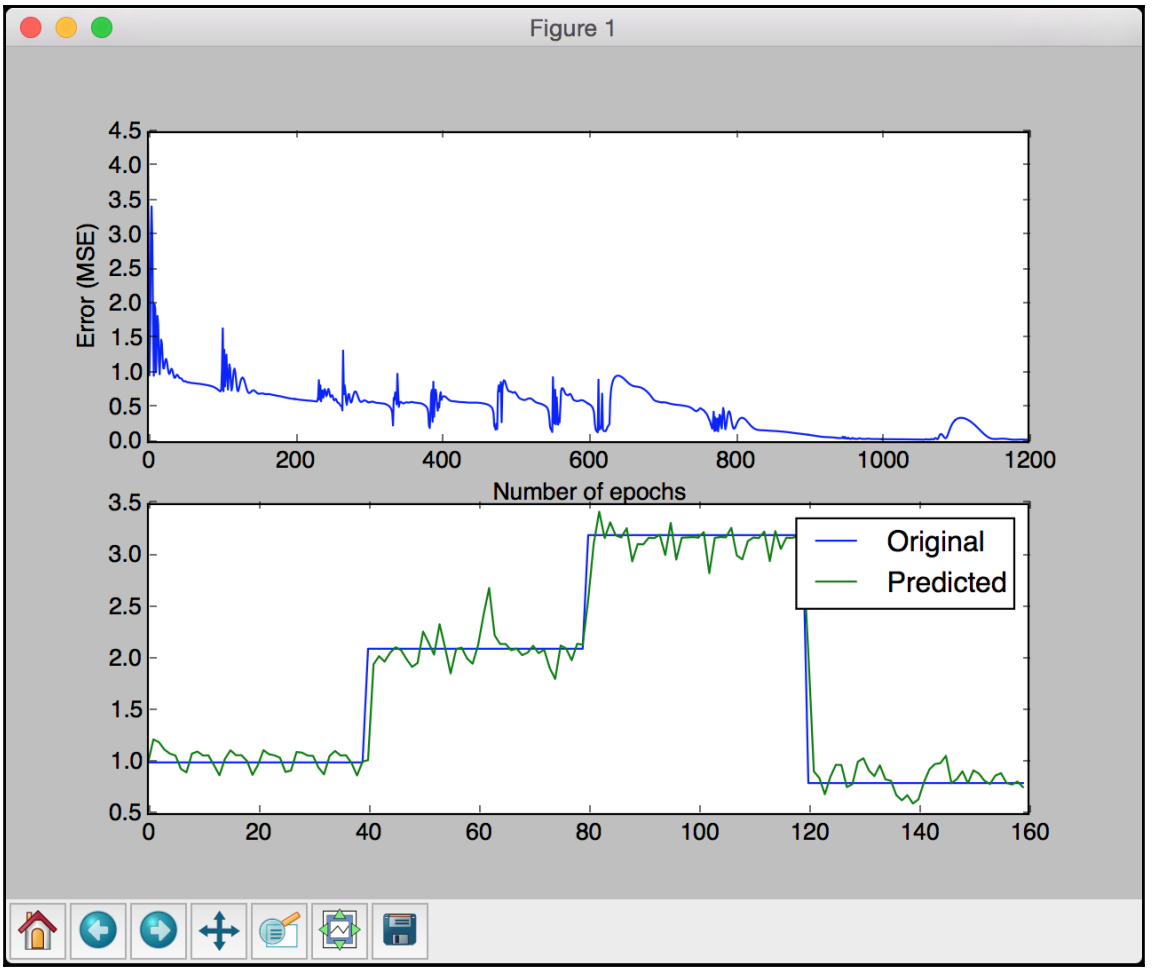


Figure 1
Error (MSE)
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
200
400
600
800
1000
1200
Number of epochs
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
20
40
60
80
100
120
140
160
Original
Predicted

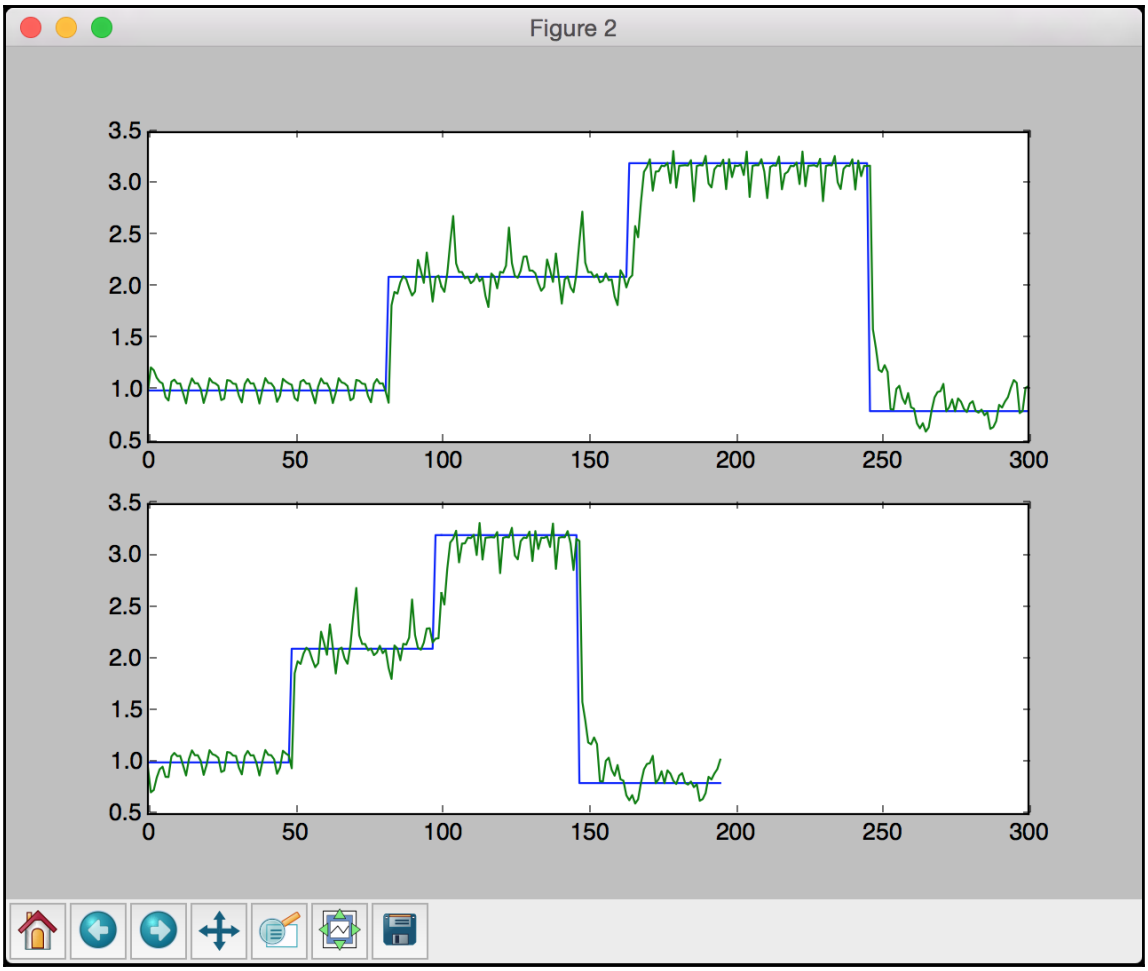
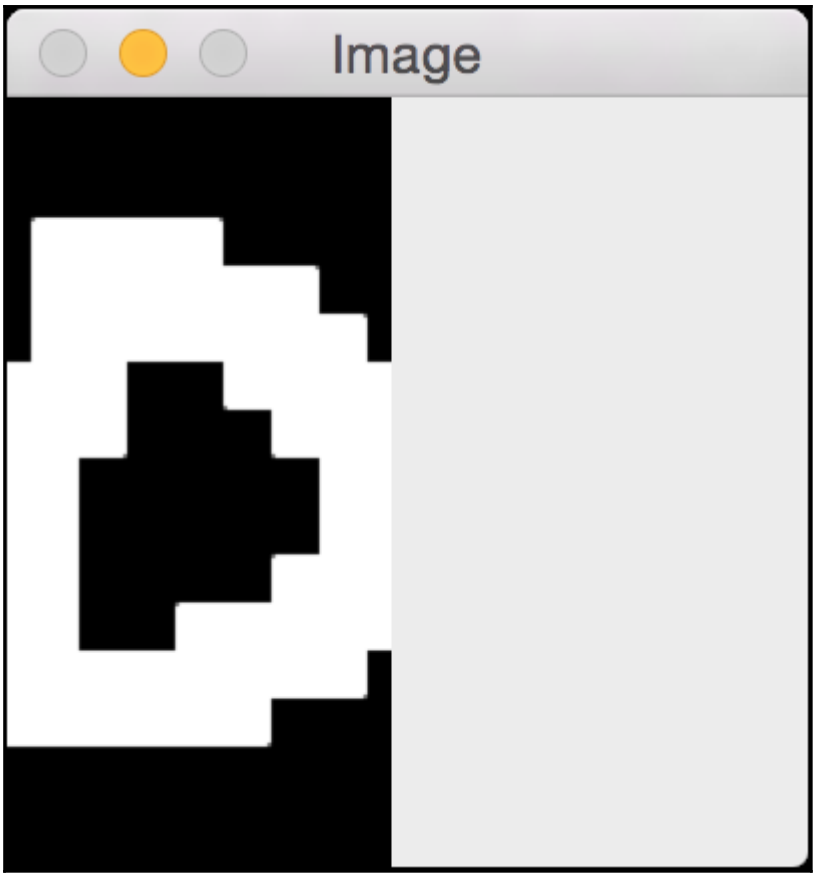
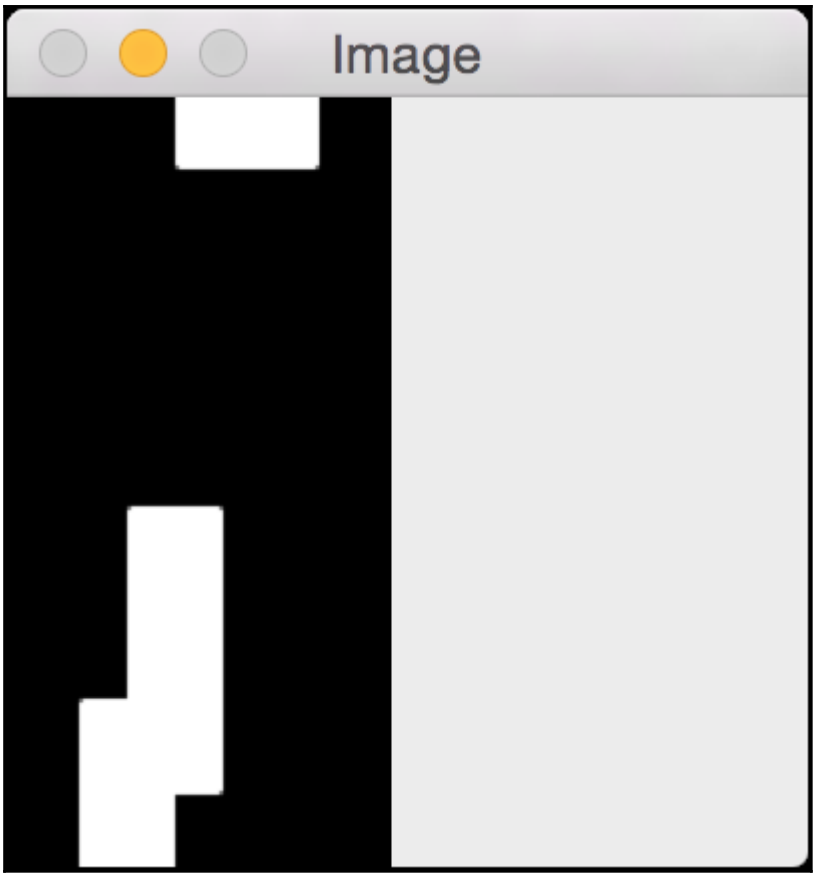


Figure 2
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
50
100
150
200
250
300

```
Epoch: 100; Error: 0.7378753203612153;
Epoch: 200; Error: 0.6276459886666788;
Epoch: 300; Error: 0.586316536629095;
Epoch: 400; Error: 0.7246461052491963;
Epoch: 500; Error: 0.7244266943409208;
Epoch: 600; Error: 0.5650581389122635;
Epoch: 700; Error: 0.5798180931911314;
Epoch: 800; Error: 0.195575666107898 26;
Epoch: 900; Error: 0.10837074465396046;
Epoch: 1000; Error: 0.04330852391940663;
Epoch: 1100; Error: 0.3073835343028226;
Epoch: 1200; Error: 0.0346852784161636 04;
The maximum number of train epochs is reached
```



Image

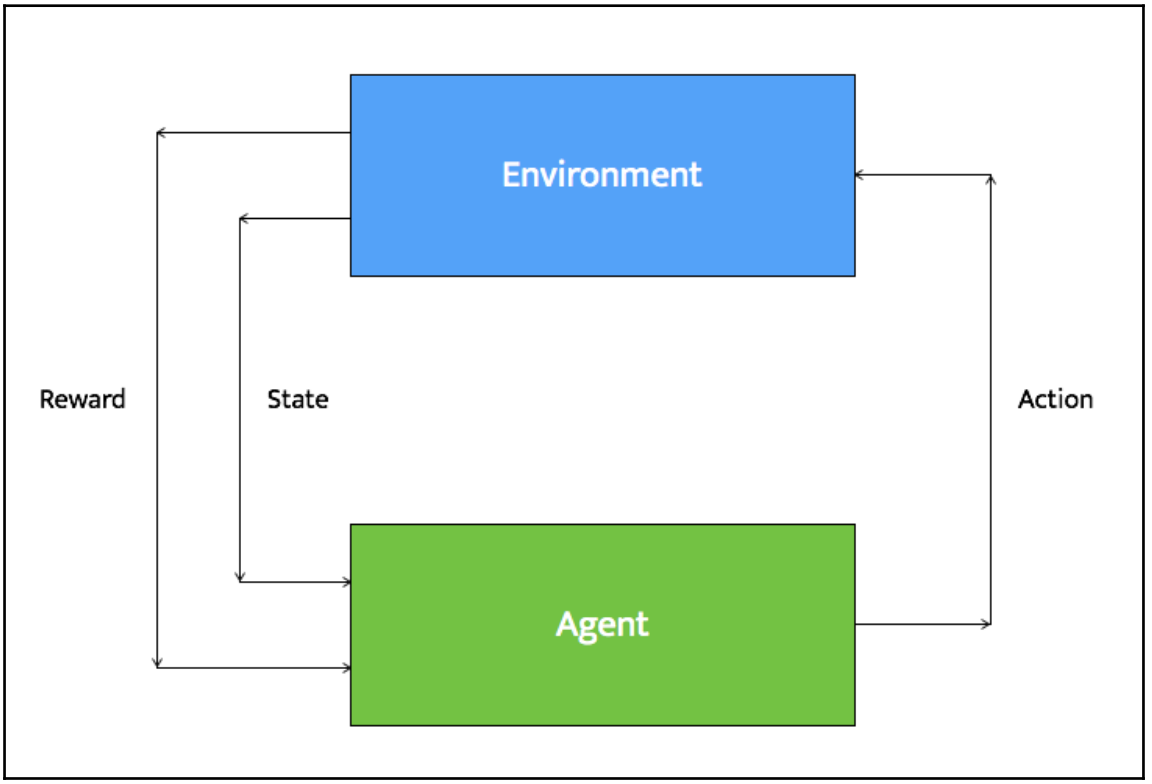


Image

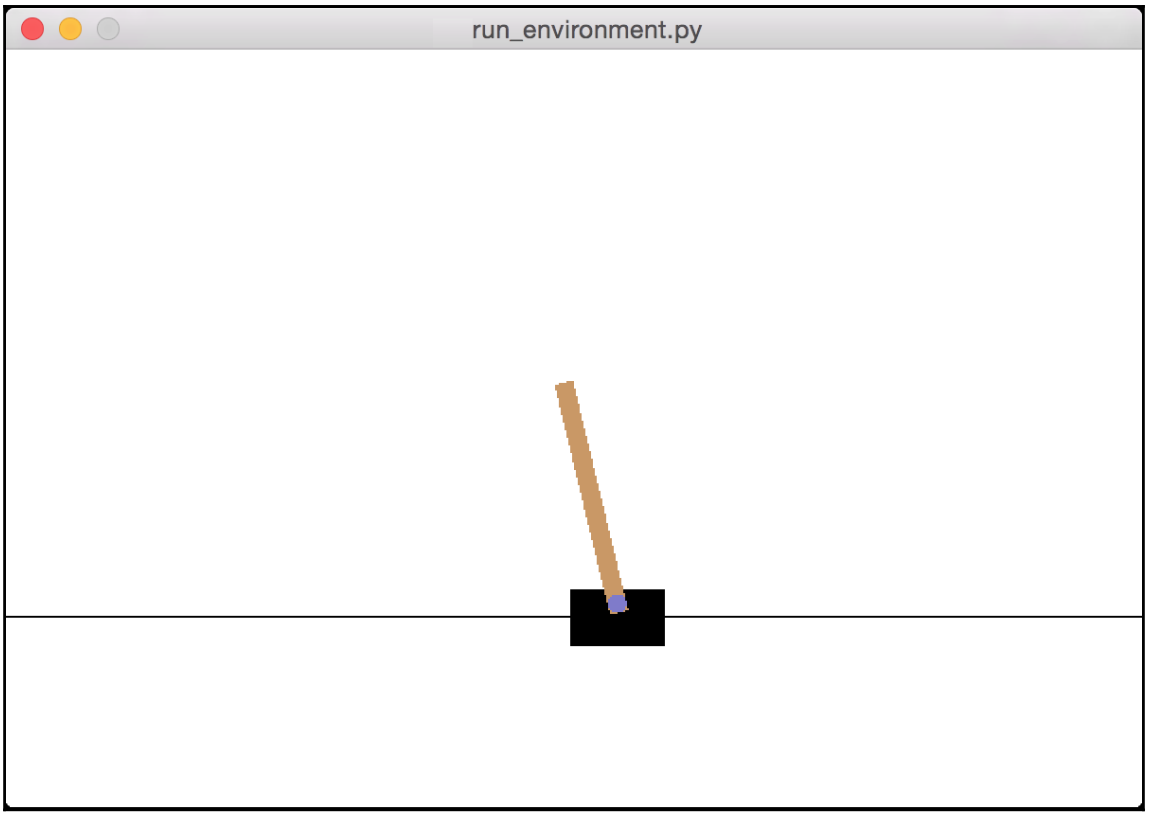
```
Epoch: 100; Error: 80.75182001223291;
Epoch: 200; Error: 49.823887961230206;
Epoch: 300; Error: 26.624261963923217;
Epoch: 400; Error: 31.131906412329677;
Epoch: 500; Error: 30.589610928772494;
Epoch: 600; Error: 23.129959531324324;
Epoch: 700; Error: 15.561849160600984;
Epoch: 800; Error: 9.52433563455828;
Epoch: 900; Error: 1.4032941634688987;
Epoch: 1000; Error: 1.1584148924740179;
Epoch: 1100; Error: 0.844934060039839;
Epoch: 1200; Error: 0.646187646028962;
Epoch: 1300; Error: 0.48881681329304894;
Epoch: 1400; Error: 0.4005475591737743;
Epoch: 1500; Error: 0.341458872835320 67;
Epoch: 1600; Error: 0.29871068426249625;
Epoch: 1700; Error: 0.2657577763744411;
Epoch: 1800; Error: 0.23921810237252988;
Epoch: 1900; Error: 0.2172060084455509;
Epoch: 2000; Error: 0.19856823374761018;
Epoch: 2100; Error: 0.18253521958793384;
Epoch: 2200; Error: 0.16855895648078095;
```

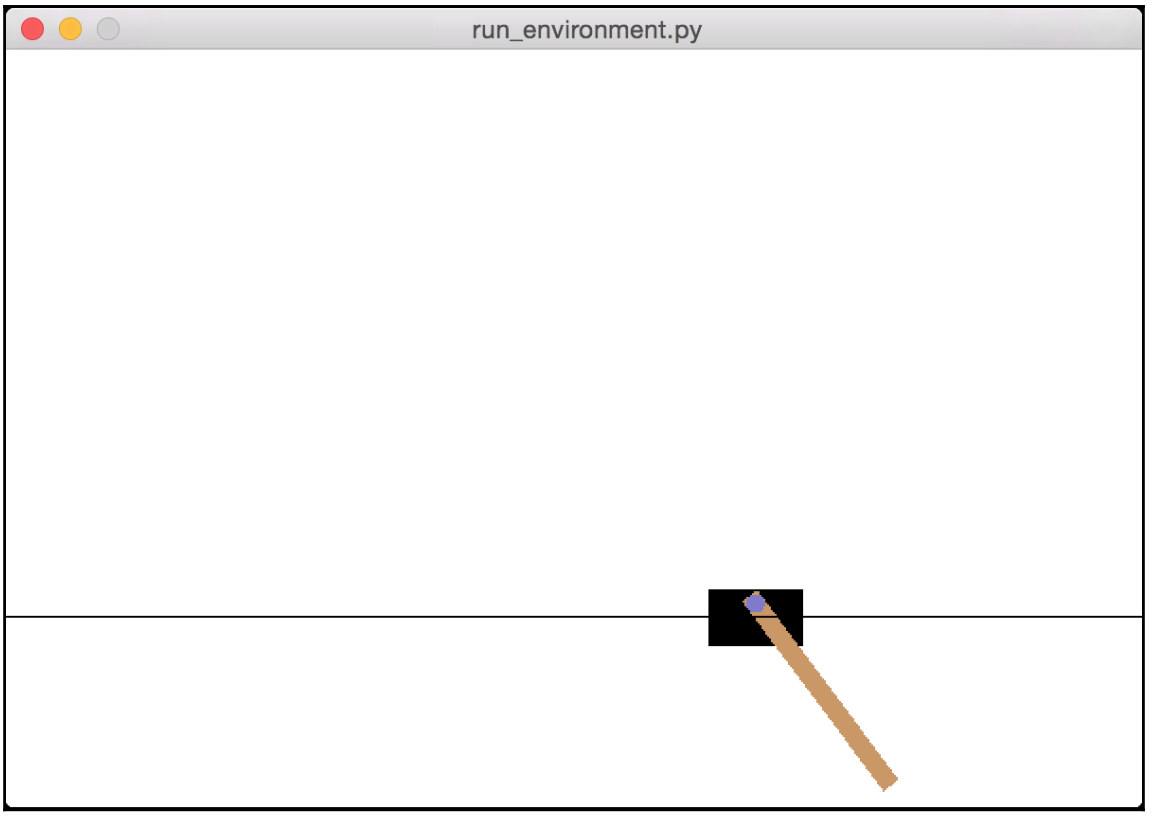
```
Epoch: 9500; Error: 0.032460181065798295;
Epoch: 9600; Error: 0.027044816600106478;
Epoch: 9700; Error: 0.022026328910164213;
Epoch: 9800; Error: 0.018353324233938713;
Epoch: 9900; Error: 0.01578969259136868;
Epoch: 10000; Error: 0.014064205770213847;
The maximum number of train epochs is reached

Testing on unknown data:

Original: o
Predicted: o

Original: m
Predicted: n

Original: m
Predicted: m

Original: a
Predicted: d

Original: n
Predicted: n
```

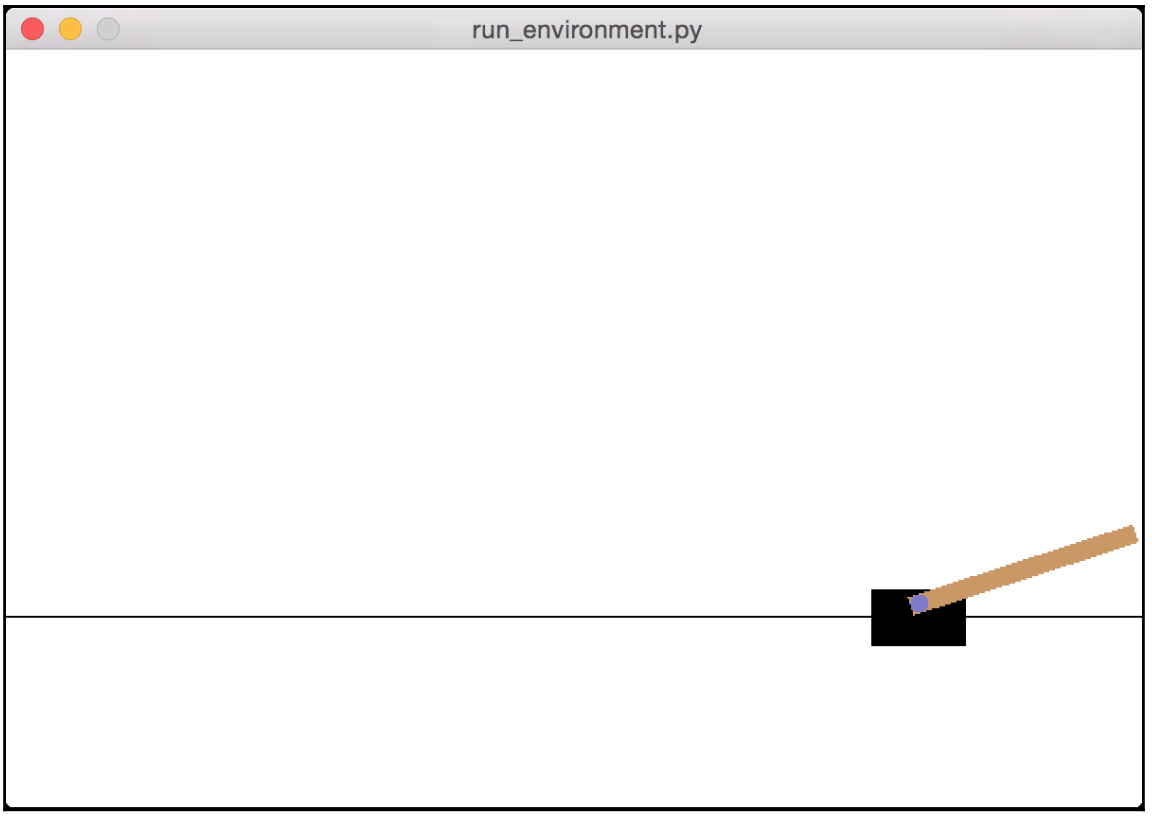
Chapter 15: Reinforcement Learning

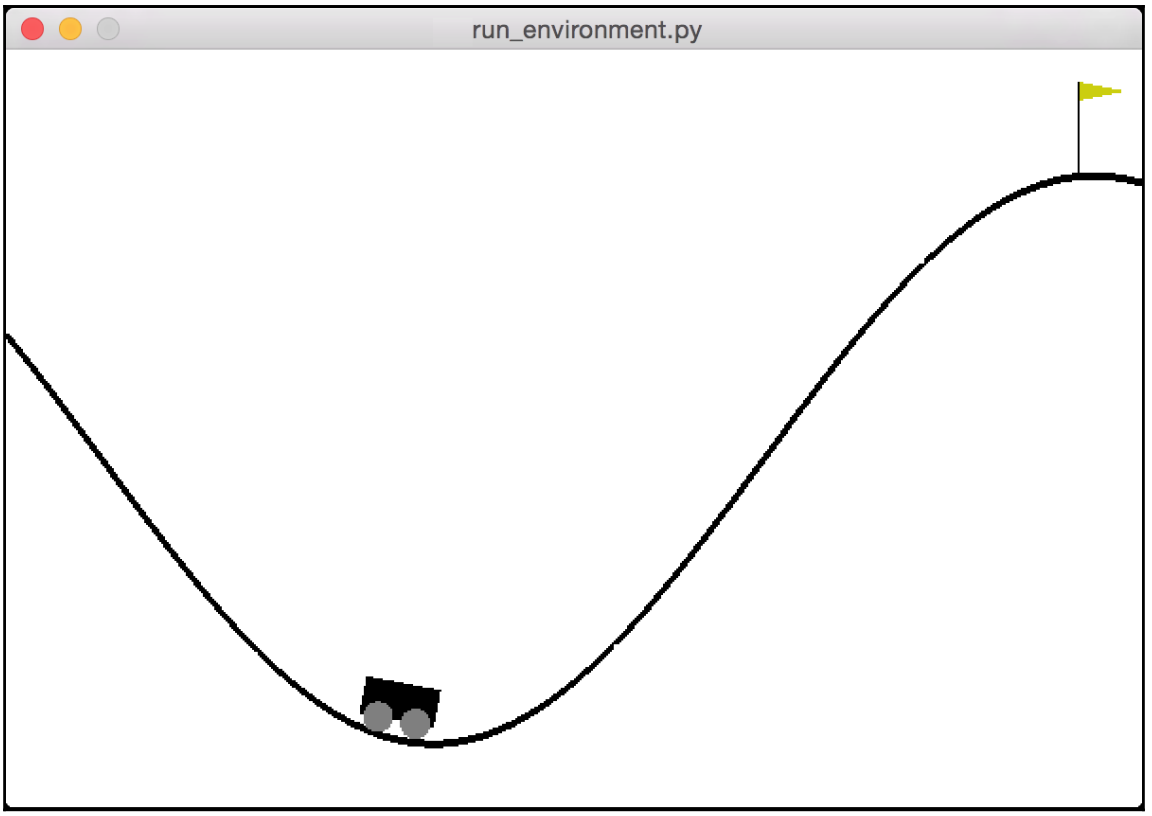


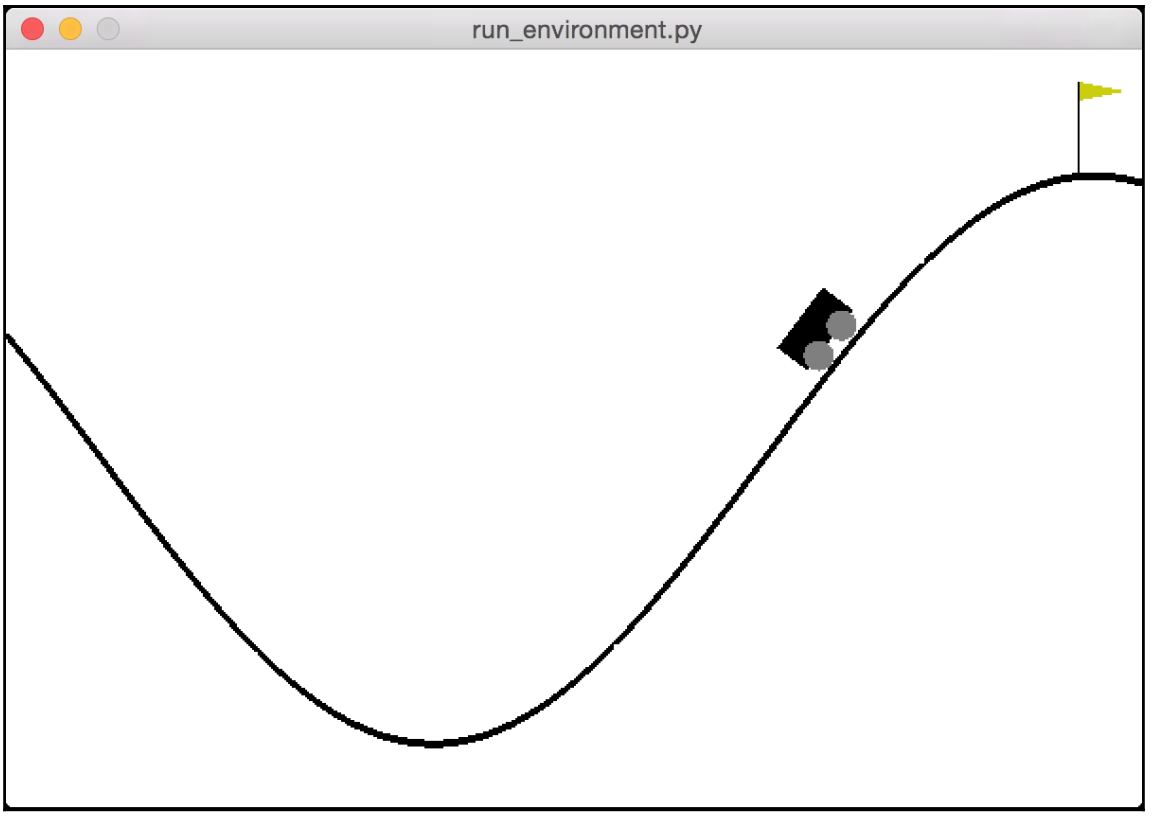

```
$ python3 run_environment.py --help
usage: run_environment.py [-h] --input-env
                          {cartpole,mountaincar,pendulum,taxi,lake}

Run an environment

optional arguments:
  -h, --help            show this help message and exit
  --input-env {cartpole,mountaincar,pendulum,taxi,lake}
                        Specify the name of the environment
```

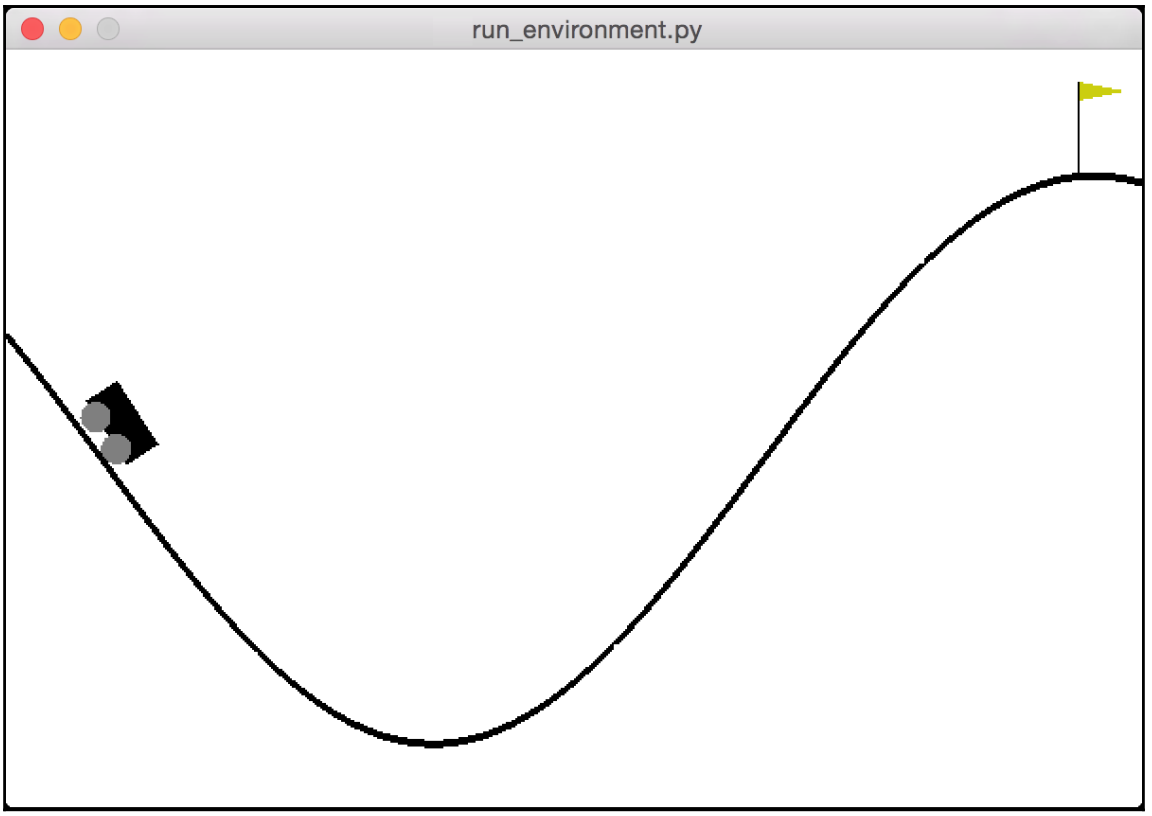

run_environment.py


run_environment.py

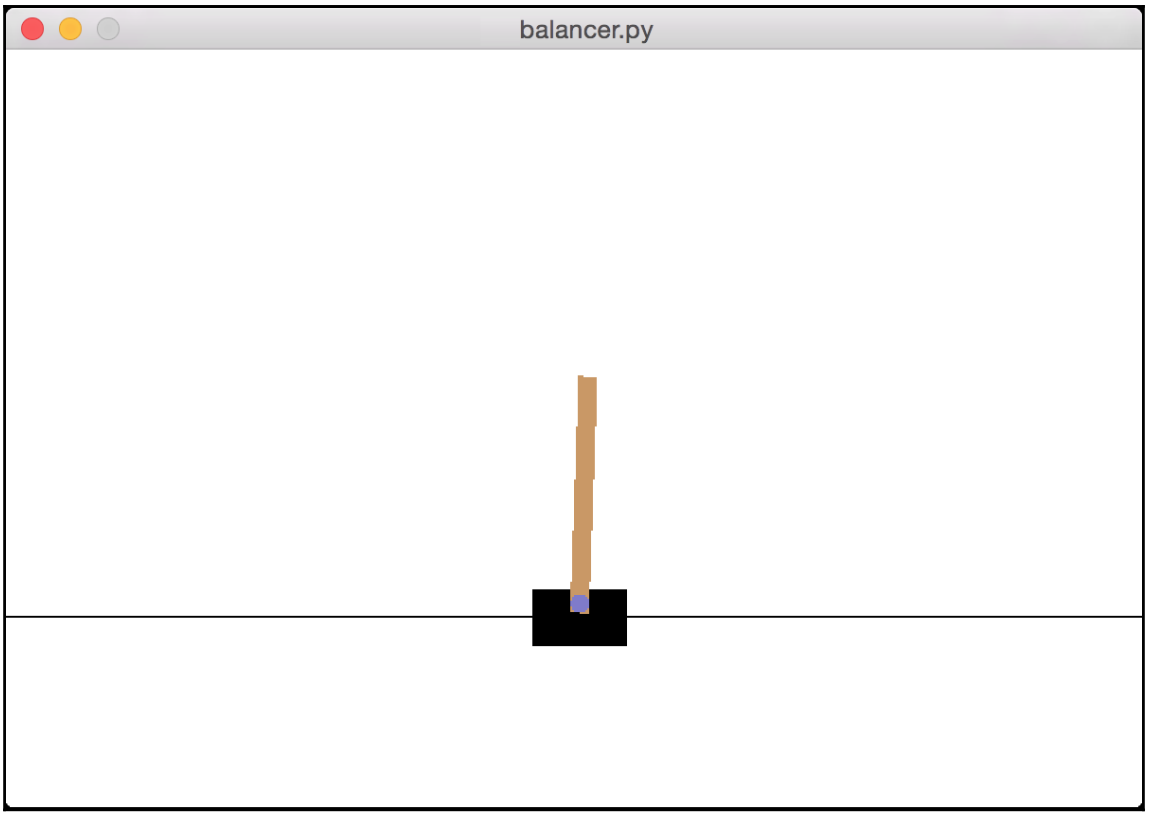

run_environment.py

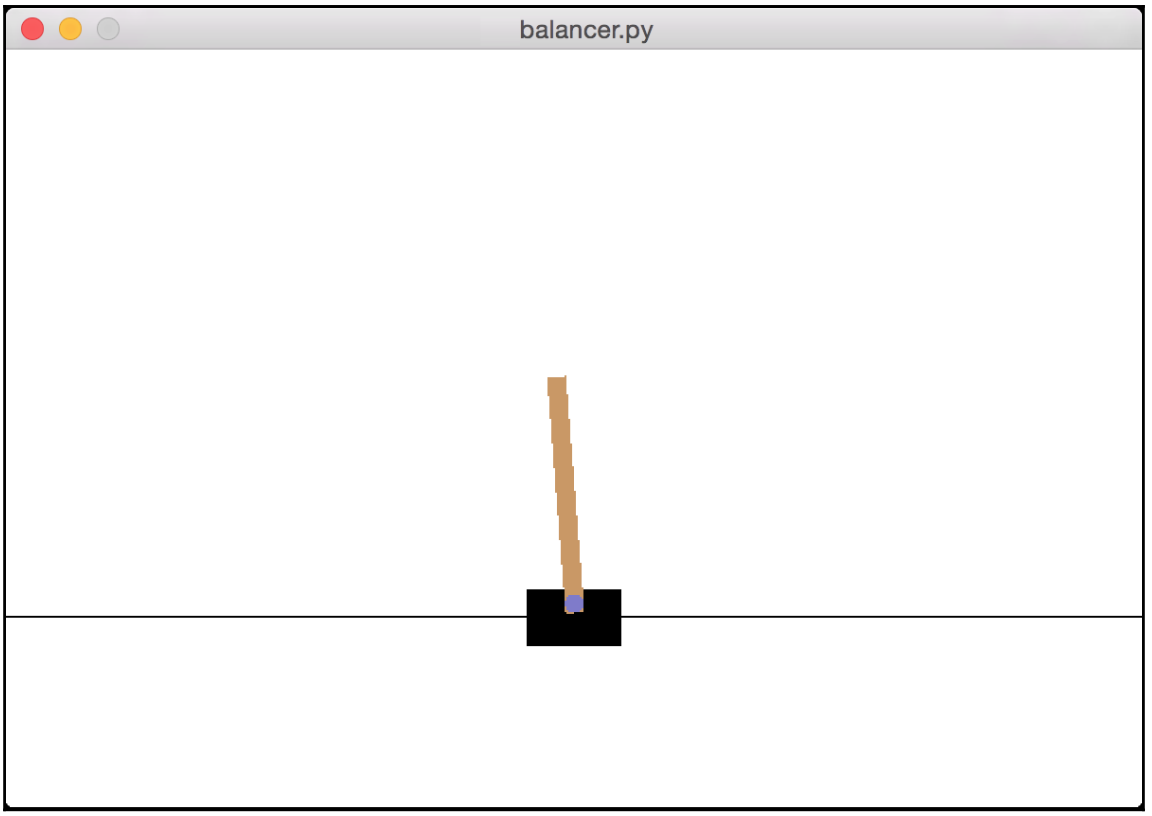

run_environment.py


run_environment.py



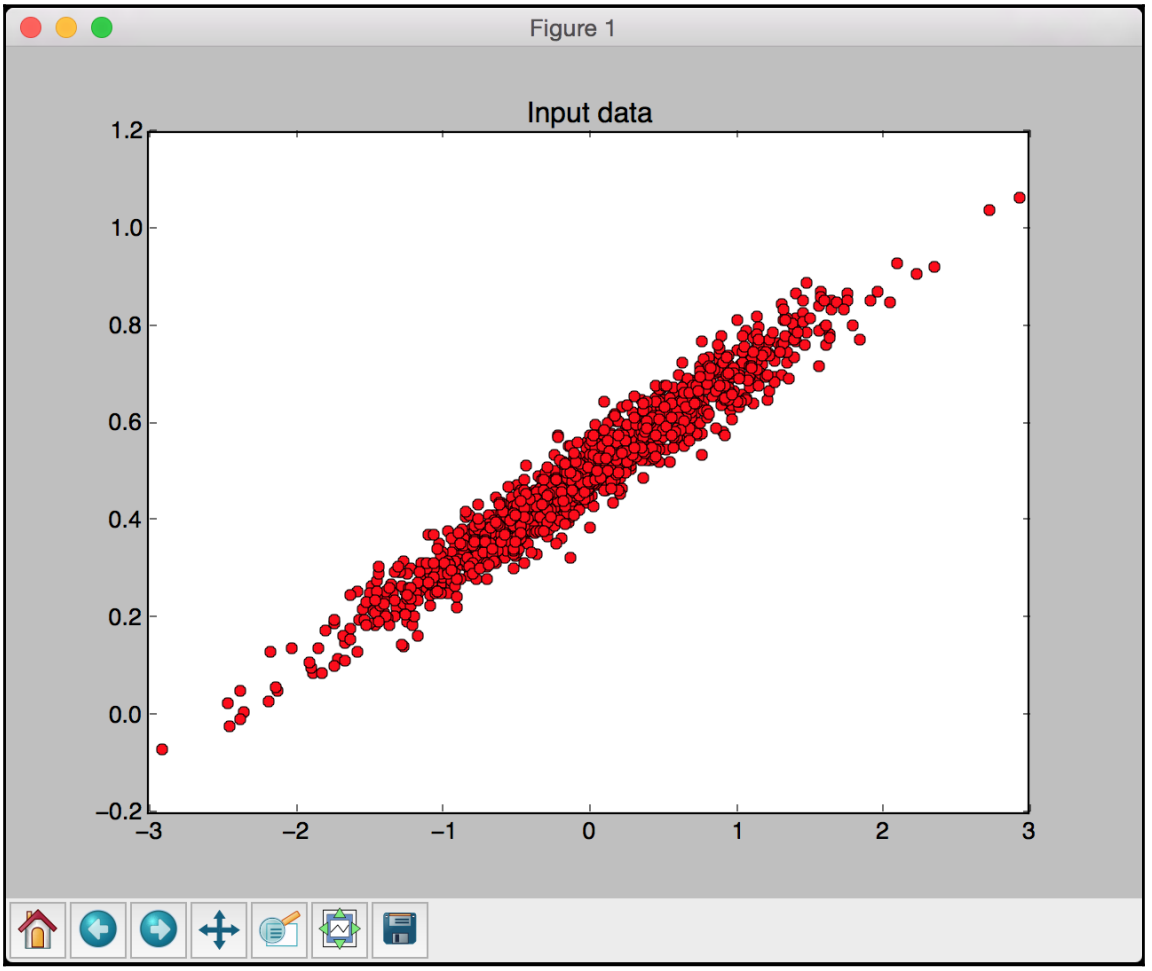

```
$ python3 balancer.py --help
usage: balancer.py [-h] --input-env {cartpole,mountaincar,pendulum}

Run an environment

optional arguments:
  -h, --help            show this help message and exit
  --input-env {cartpole,mountaincar,pendulum}
                        Specify the name of the environment
```


balancer.py


balancer.py

```
[ 0.01704777  0.03379922 -0.01628054  0.02868271]
[ 0.01772375 -0.16108552 -0.01570689  0.31618481]
[ 0.01450204  0.03425659 -0.00938319  0.01859014]
[ 0.01518717 -0.16072954 -0.00901139  0.30829785]
[ 0.01197258 -0.35572194 -0.00284543  0.59812526]
[ 0.00485814 -0.16056029  0.00911707  0.30454742]
[ 0.00164694 -0.35581098  0.01520802  0.60009165]
[-0.00546928 -0.16090505  0.02720986  0.31223756]
[-0.00868738 -0.35640386  0.03345461  0.61337594]
[-0.01581546 -0.55197696  0.04572213  0.91640525]
[-0.026855   -0.3575021   0.06405023  0.63843544]
[-0.03400504 -0.16332896  0.07681894  0.36659087]
[-0.03727162 -0.3594537   0.08415076  0.68247294]
[-0.04446069 -0.5556372   0.09780022  1.00041801]
[-0.05557344 -0.75192055  0.11780858  1.32214352]
[-0.07061185 -0.55846765  0.14425145  1.06853119]
[-0.0817812  -0.36551752  0.16562207  0.82437502]
[-0.08909155 -0.56247052  0.18210957  1.16423244]
[-0.10034096 -0.75943464  0.20539422  1.50803784]
Episode finished after 19 timesteps
```

Chapter 16 : Deep Learning with Convolutional Neural Networks



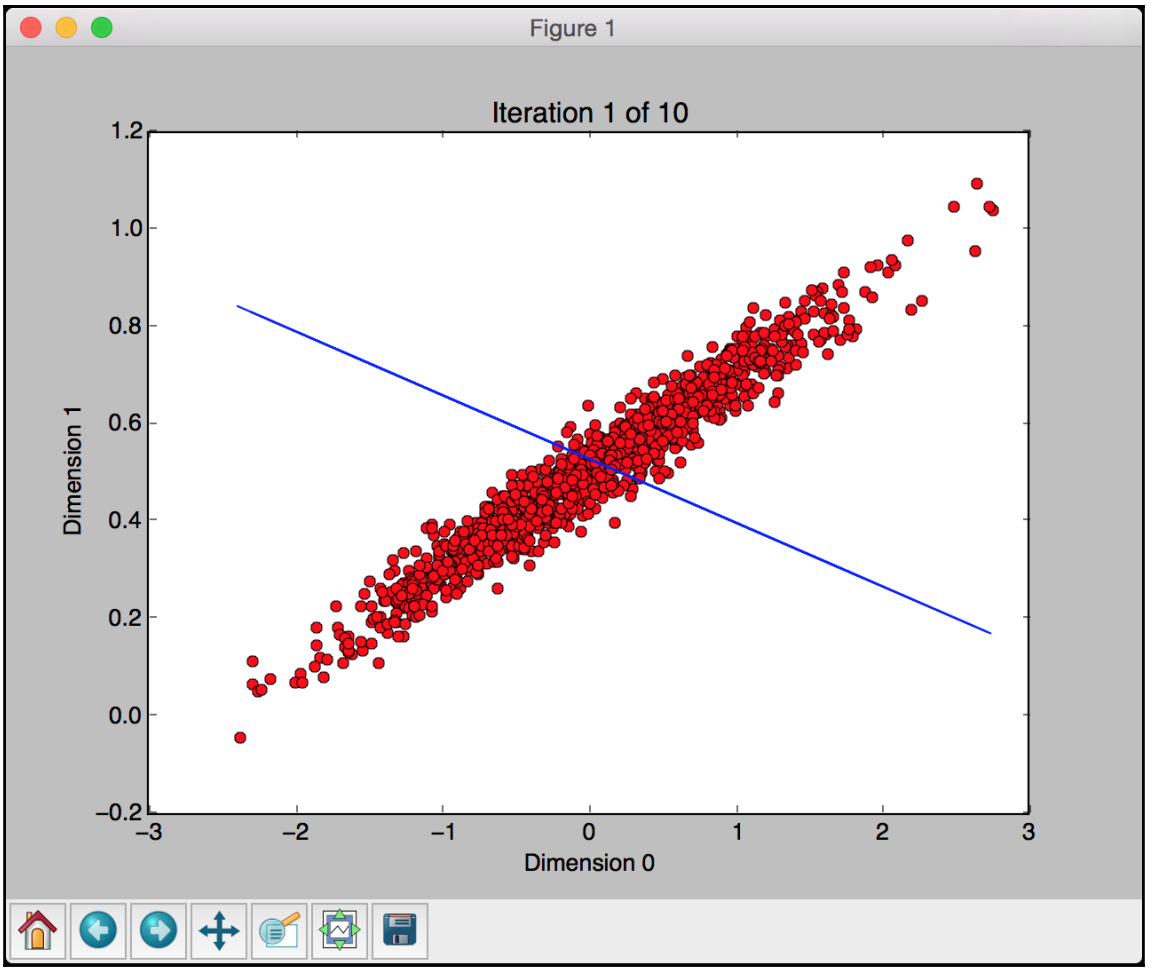


Figure 1
Iteration 1 of 10
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Dimension 1
−3
−2
−1
0
1
2
3
Dimension 0

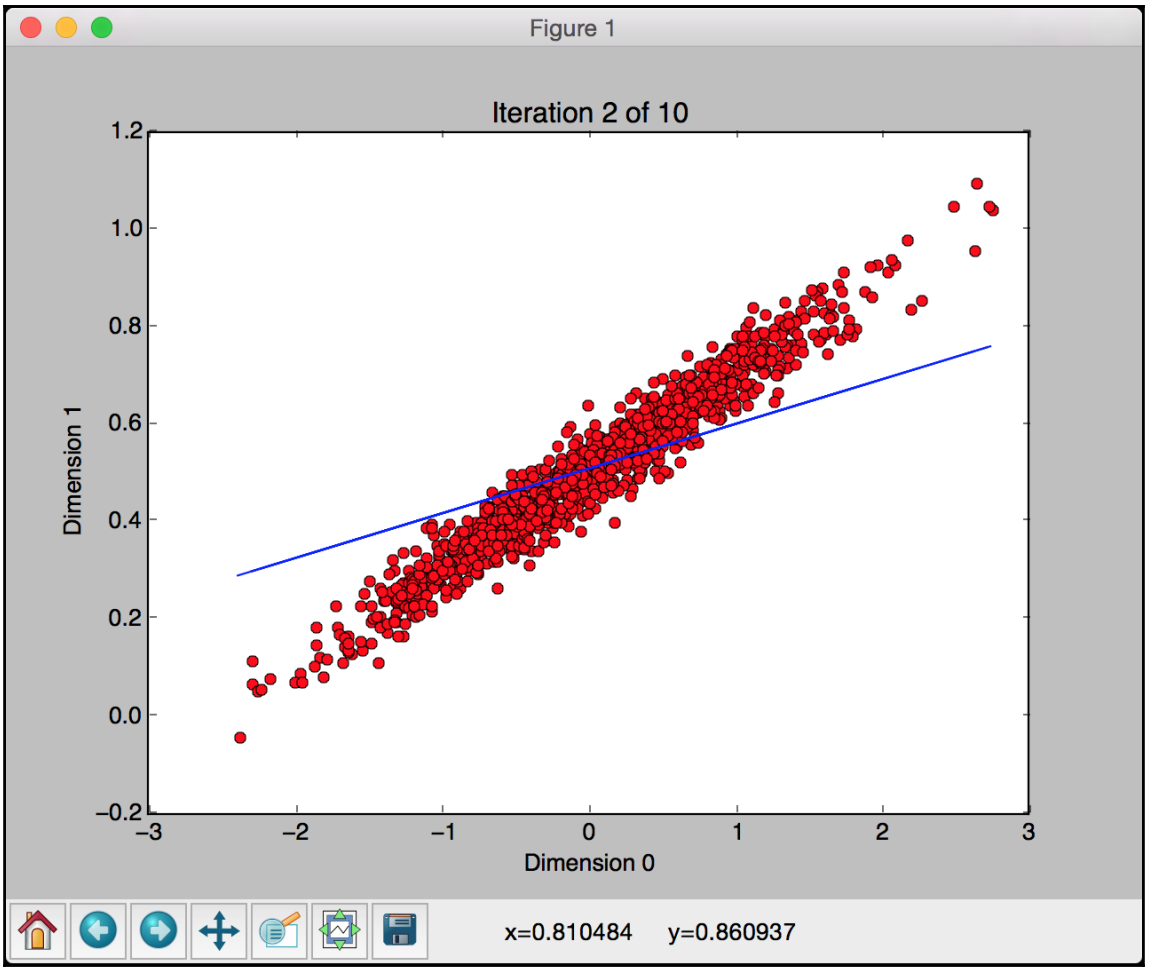


Figure 1
Iteration 2 of 10
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Dimension 1
−3
−2
−1
0
1
2
3
Dimension 0
x=0.810484
y=0.860937

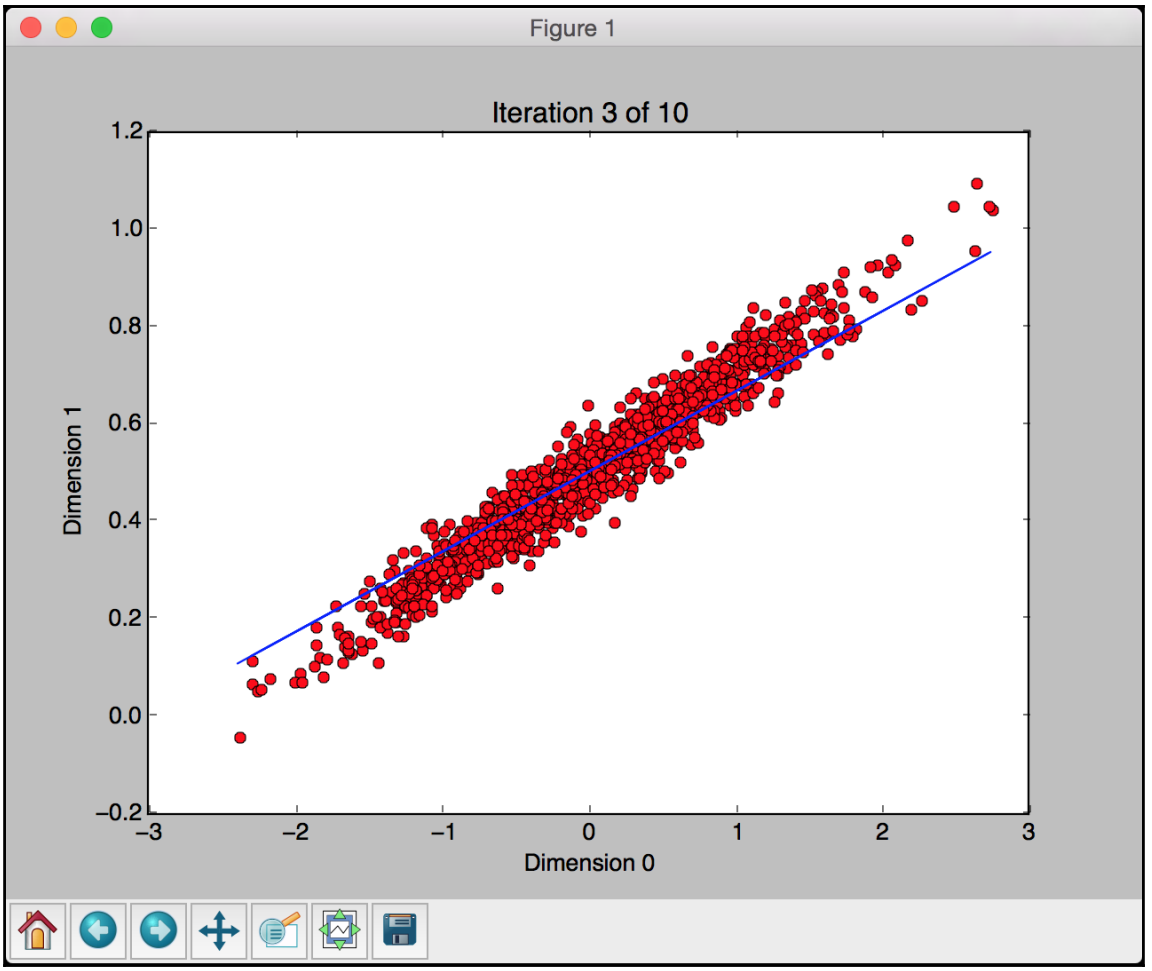


Figure 1
Iteration 3 of 10
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Dimension 1
−3
−2
−1
0
1
2
3
Dimension 0

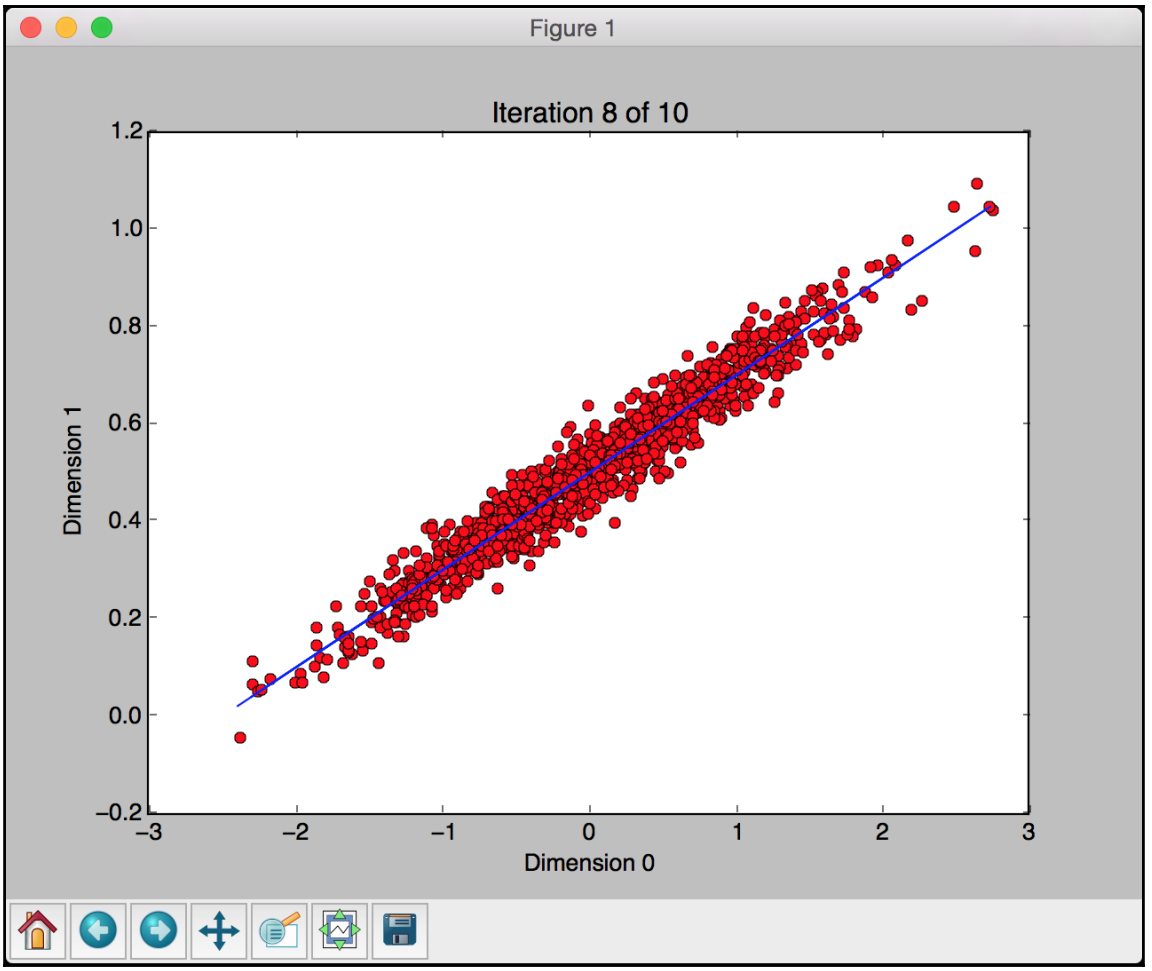


Figure 1
Iteration 8 of 10
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
Dimension 1
−3
−2
−1
0
1
2
3
Dimension 0

```
ITERATION 1
W = -0.130961
b = 0.53005
loss = 0.0760343

ITERATION 2
W = 0.0917911
b = 0.508959
loss = 0.00960302

ITERATION 3
W = 0.164665
b = 0.502555
loss = 0.00250165

ITERATION 4
W = 0.188492
b = 0.500459
loss = 0.0017425
```

```
ITERATION 7
W = 0.199662
b = 0.499477
loss = 0.00165175

ITERATION 8
W = 0.199934
b = 0.499453
loss = 0.00165165

ITERATION 9
W = 0.200023
b = 0.499445
loss = 0.00165164

ITERATION 10
W = 0.200052
b = 0.499443
loss = 0.00165164
```

```
Extracting ./mnist_data/train-images-idx3-ubyte.gz
Extracting ./mnist_data/train-labels-idx1-ubyte.gz
Extracting ./mnist_data/t10k-images-idx3-ubyte.gz
Extracting ./mnist_data/t10k-labels-idx1-ubyte.gz

Accuracy = 0.921
```

```
Extracting ./mnist_data/train-images-idx3-ubyte.gz
Extracting ./mnist_data/train-labels-idx1-ubyte.gz
Extracting ./mnist_data/t10k-images-idx3-ubyte.gz
Extracting ./mnist_data/t10k-labels-idx1-ubyte.gz

Training the model....
Iteration 0 , Accuracy = 0.0533333
Iteration 50 , Accuracy = 0.813333
Iteration 100 , Accuracy = 0.8
Iteration 150 , Accuracy = 0.906667
Iteration 200 , Accuracy = 0.84
Iteration 250 , Accuracy = 0.92
Iteration 300 , Accuracy = 0.933333
Iteration 350 , Accuracy = 0.866667
Iteration 400 , Accuracy = 0.973333
Iteration 450 , Accuracy = 0.933333
Iteration 500 , Accuracy = 0.906667
Iteration 550 , Accuracy = 0.853333
Iteration 600 , Accuracy = 0.973333
Iteration 650 , Accuracy = 0.973333
Iteration 700 , Accuracy = 0.96
Iteration 750 , Accuracy = 0.933333
```

```
Iteration 2900 , Accuracy = 0.973333
Iteration 2950 , Accuracy = 1.0
Iteration 3000 , Accuracy = 0.973333
Iteration 3050 , Accuracy = 1.0
Iteration 3100 , Accuracy = 0.986667
Iteration 3150 , Accuracy = 1.0
Iteration 3200 , Accuracy = 1.0
Iteration 3250 , Accuracy = 1.0
Iteration 3300 , Accuracy = 1.0
Iteration 3350 , Accuracy = 1.0
Iteration 3400 , Accuracy = 0.986667
Iteration 3450 , Accuracy = 0.946667
Iteration 3500 , Accuracy = 0.973333
Iteration 3550 , Accuracy = 0.973333
Iteration 3600 , Accuracy = 1.0
Iteration 3650 , Accuracy = 0.986667
Iteration 3700 , Accuracy = 1.0
Iteration 3750 , Accuracy = 1.0
Iteration 3800 , Accuracy = 0.986667
Iteration 3850 , Accuracy = 0.986667
Iteration 3900 , Accuracy = 1.0
```