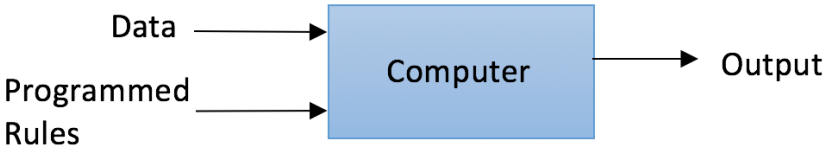
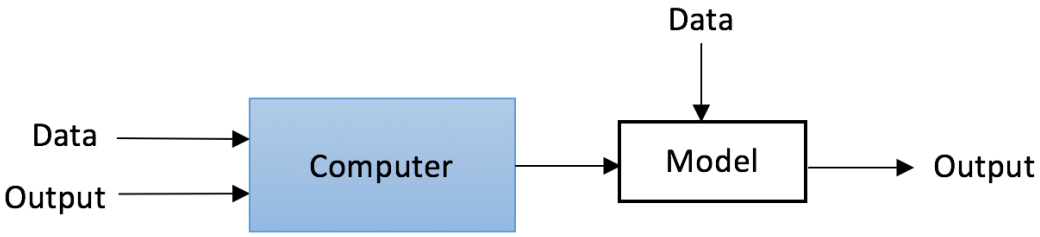
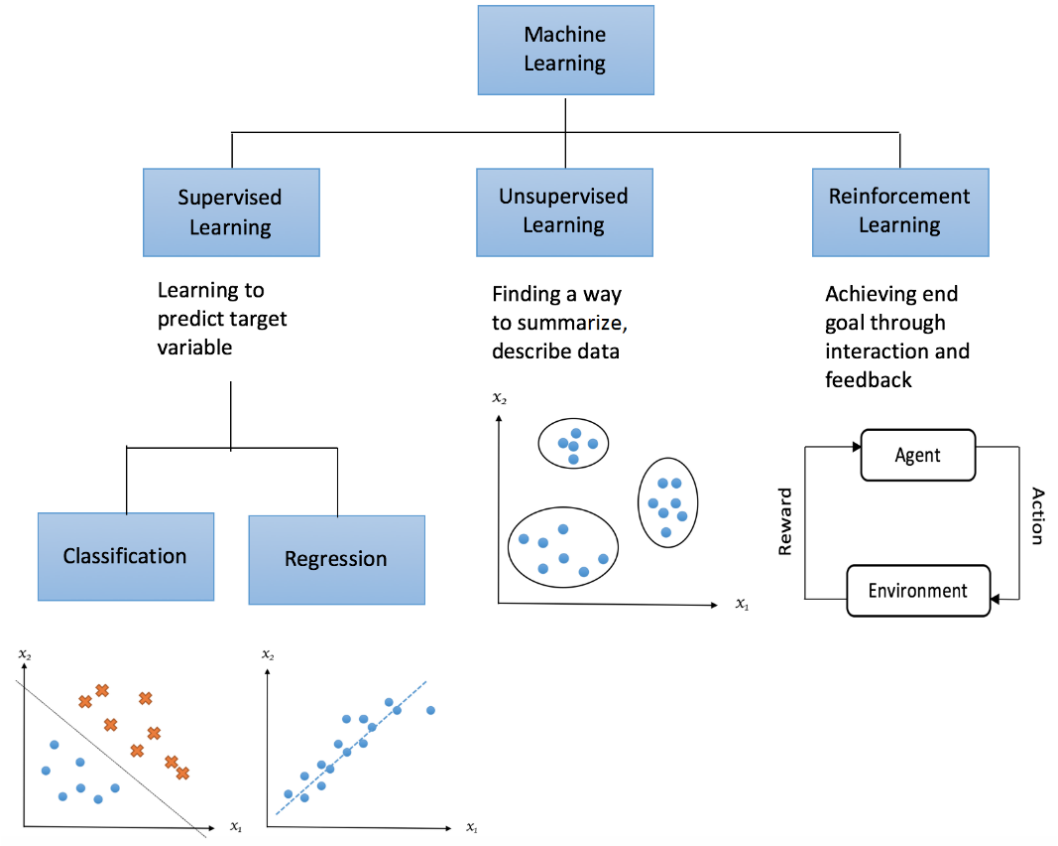
Chapter 1: Getting Started with Machine Learning and Python

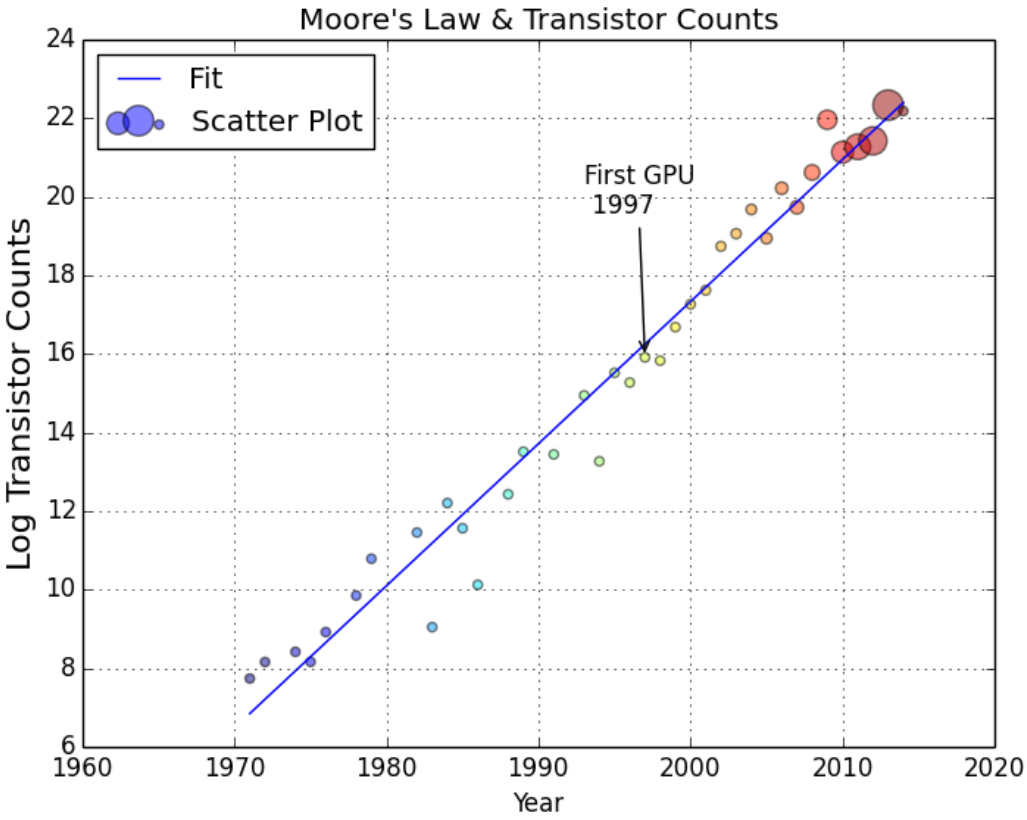


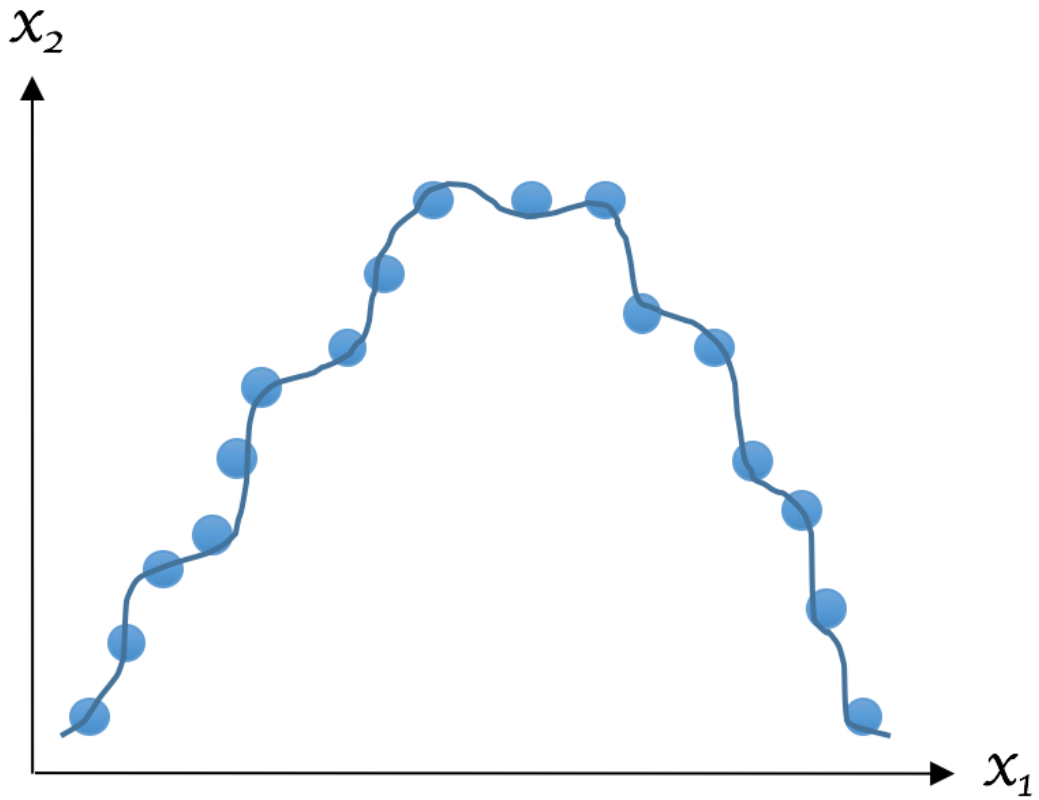
Traditional Programming



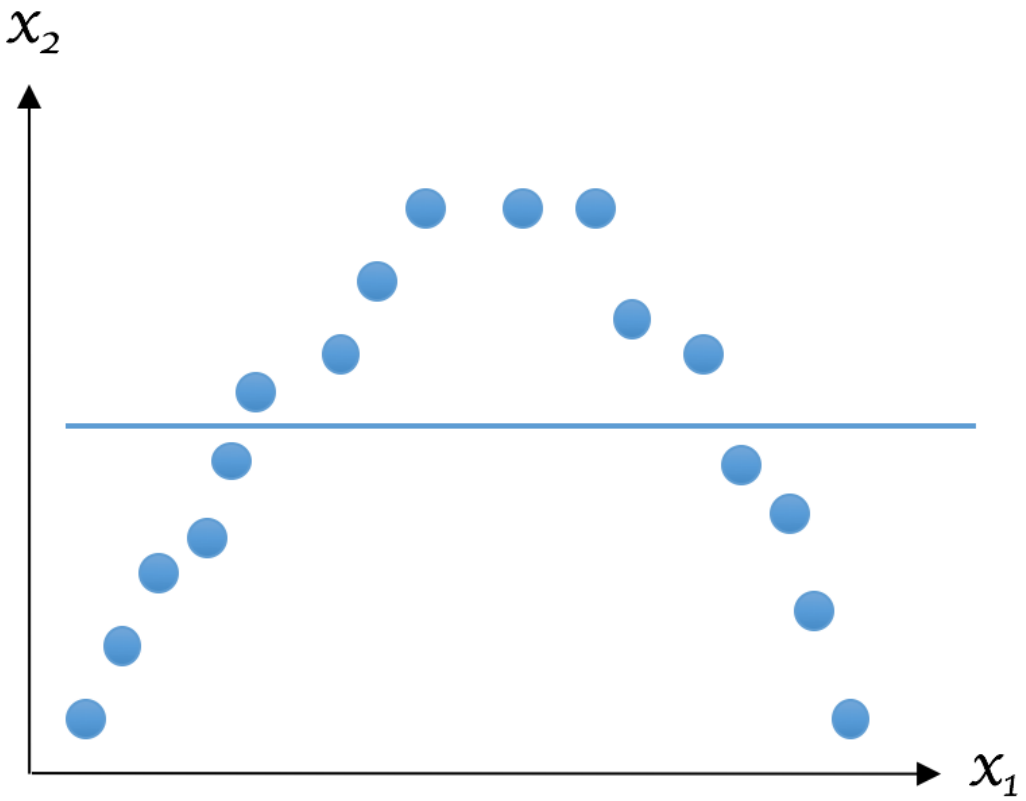
Machine Learning


Machine Learning
Supervised Learning
Unsupervised Learning
Reinforcement Learning
Learning to predict target variable
Finding a way to summarize, describe data
Achieving end goal through interaction and feedback
Classification
Regression
x_2
x_1
Agent
Reward
Action
Environment

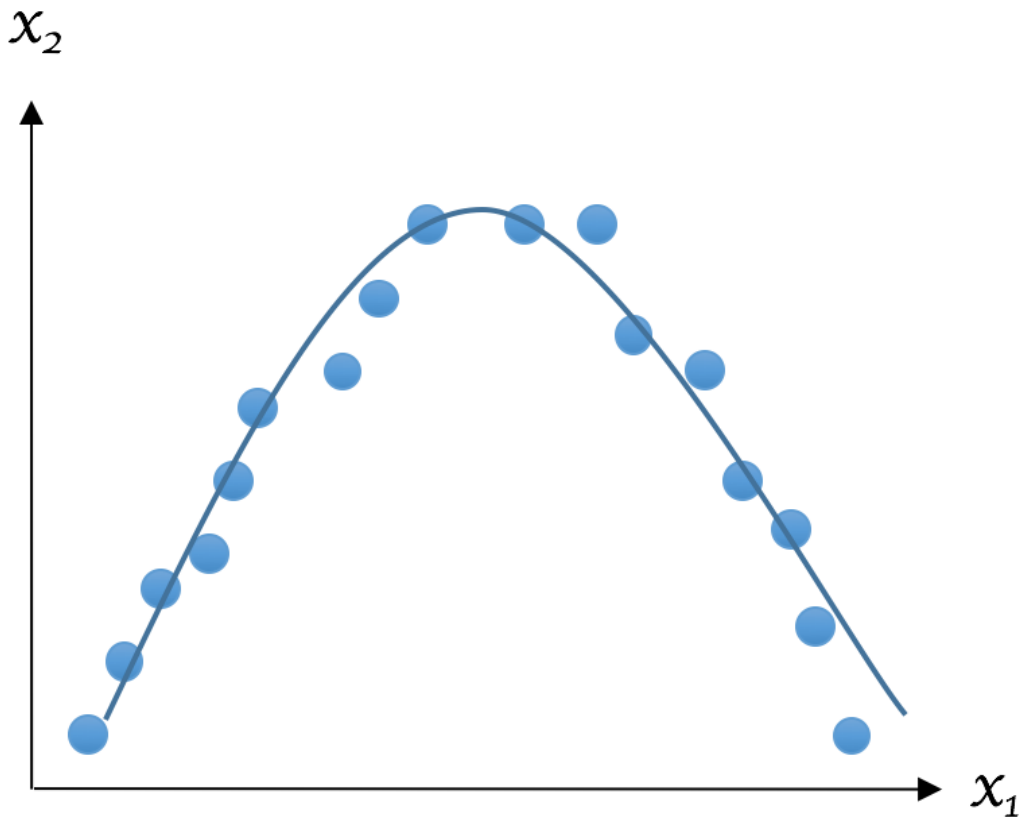

Moore's Law & Transistor Counts
Fit
Scatter Plot
First GPU
1997
Log Transistor Counts
Year
24
22
20
18
16
14
12
10
8
6
1960
1970
1980
1990
2000
2010
2020



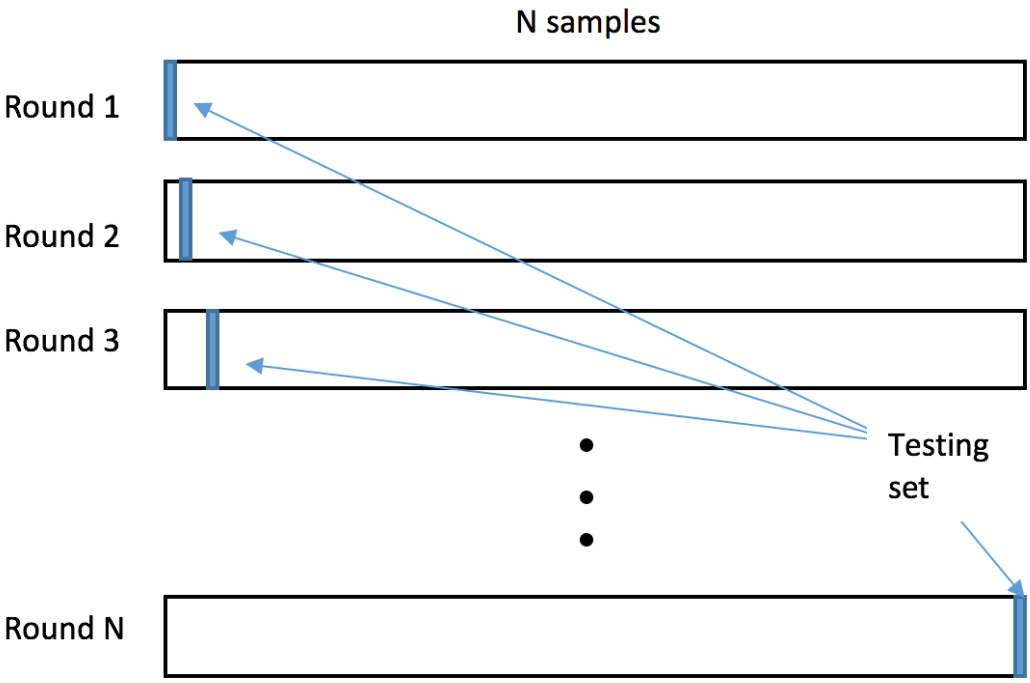
Overfitting



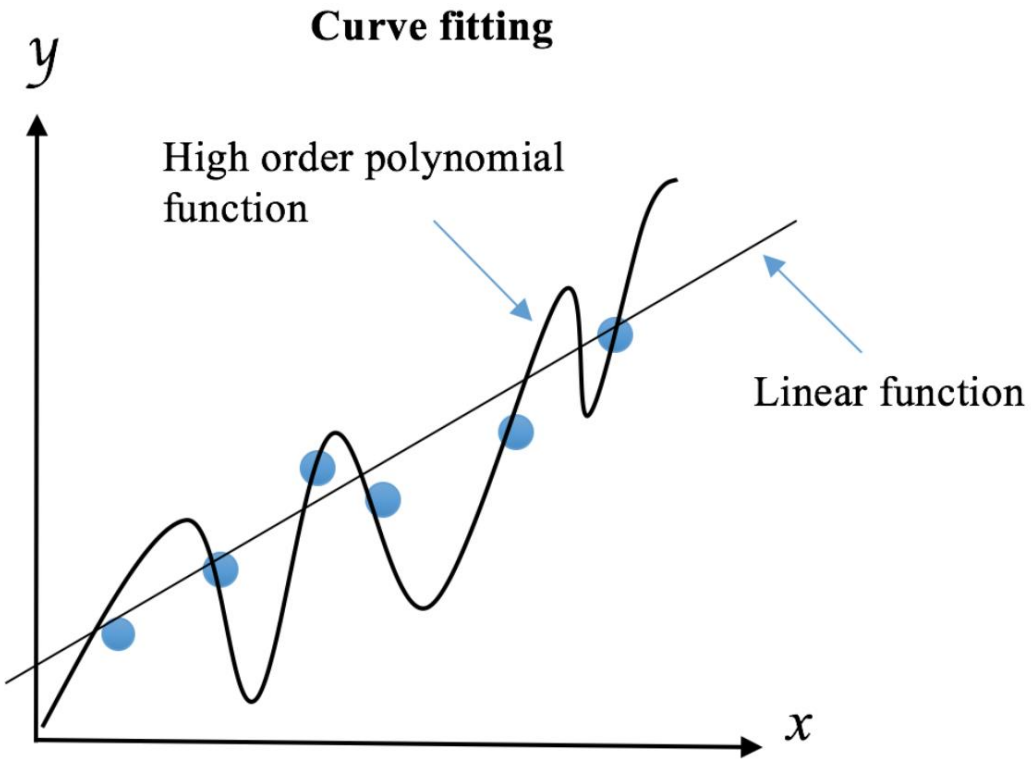
Underfitting



Desired



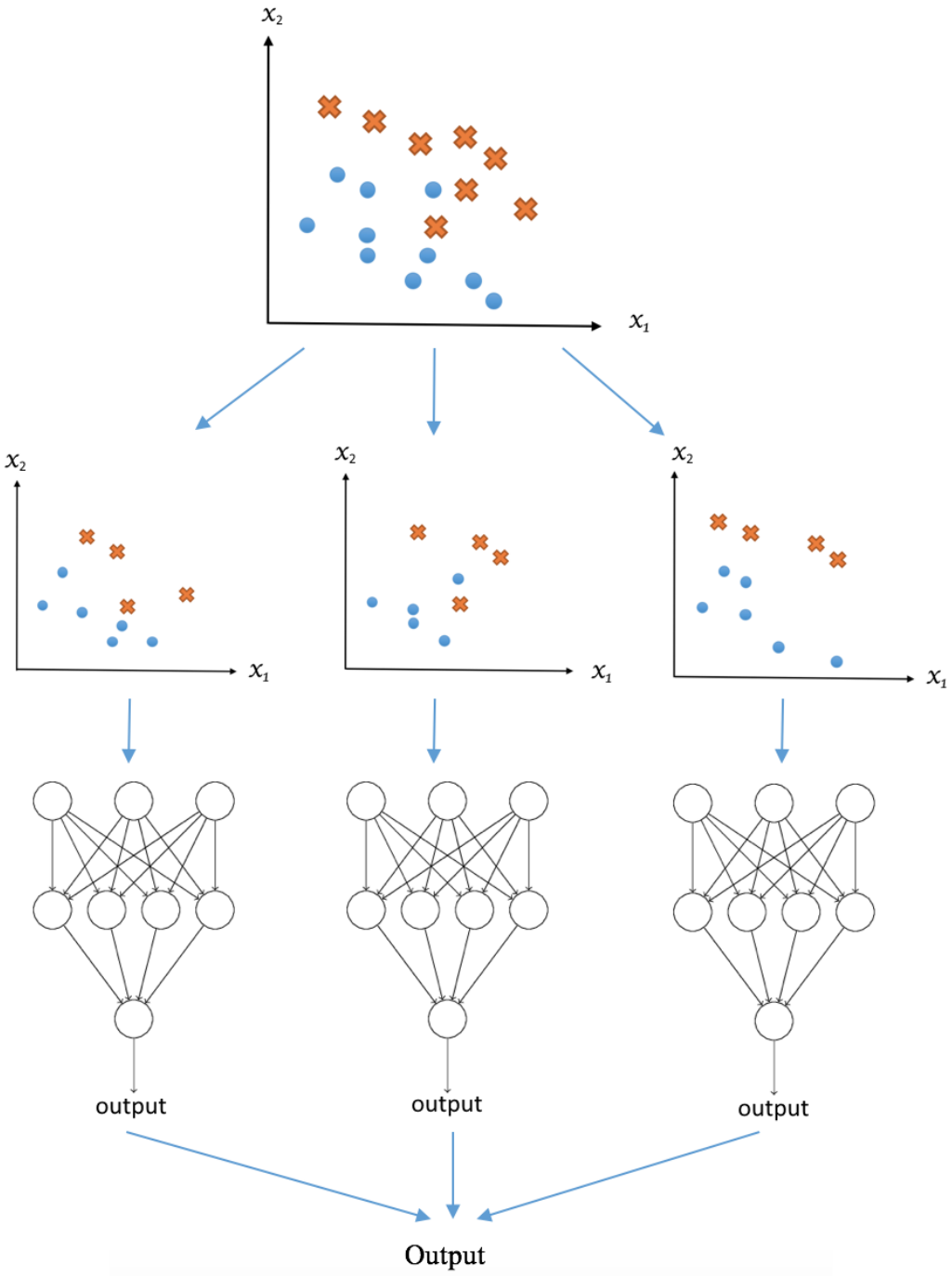
Round	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
1	**Testing**	Training	Training	Training	Training
2	Training	**Testing**	Training	Training	Training
3	Training	Training	**Testing**	Training	Training
4	Training	Training	Training	**Testing**	Training
5	Training	Training	Training	Training	**Testing**

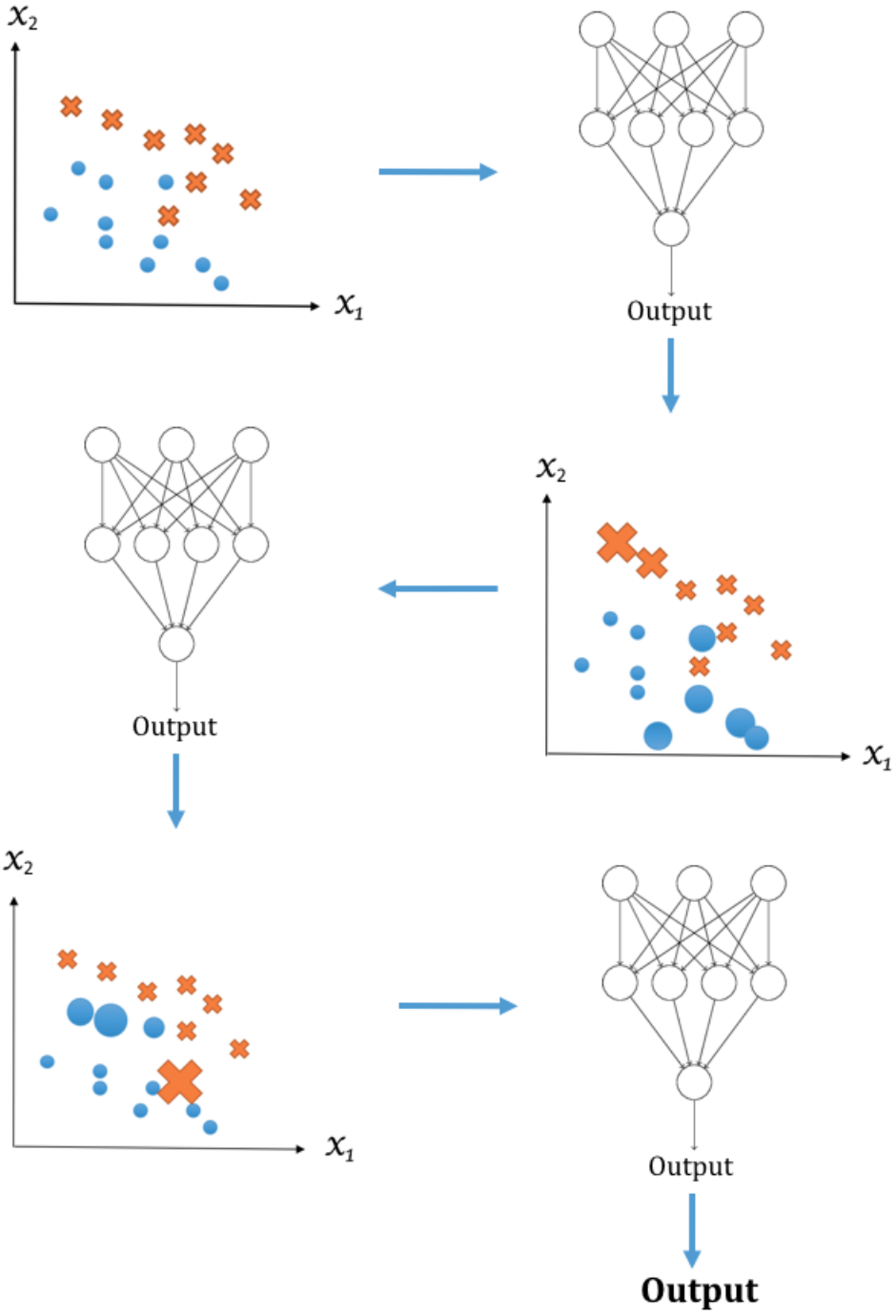


Male	Young	Tall	With glasses	In grey	**Friend**
Female	Middle	Average	Without glasses	In black	**Stranger**
Male	Young	Short	With glasses	In white	**Friend**
Male	Senior	Short	Without glasses	In black	**Stranger**
Female	Young	Average	With glasses	In white	**Friend**
Male	Young	Short	Without glasses	In red	**Friend**

Label	Encoded Label
Africa	1
Asia	2
Europe	3
South America	4
North America	5
Other	6

	is_africa	is_asia	is_europe	is_sam	is_nam
Africa	**1**	0	0	0	0
Asia	0	**1**	0	0	0
Europe	0	0	**1**	0	0
South America	0	0	0	**1**	0
North America	0	0	0	0	**1**
Other	0	0	0	0	0


x_2
x_1
x_2
x_1
x_2
x_1
x_2
x_1
output
output
output
Output

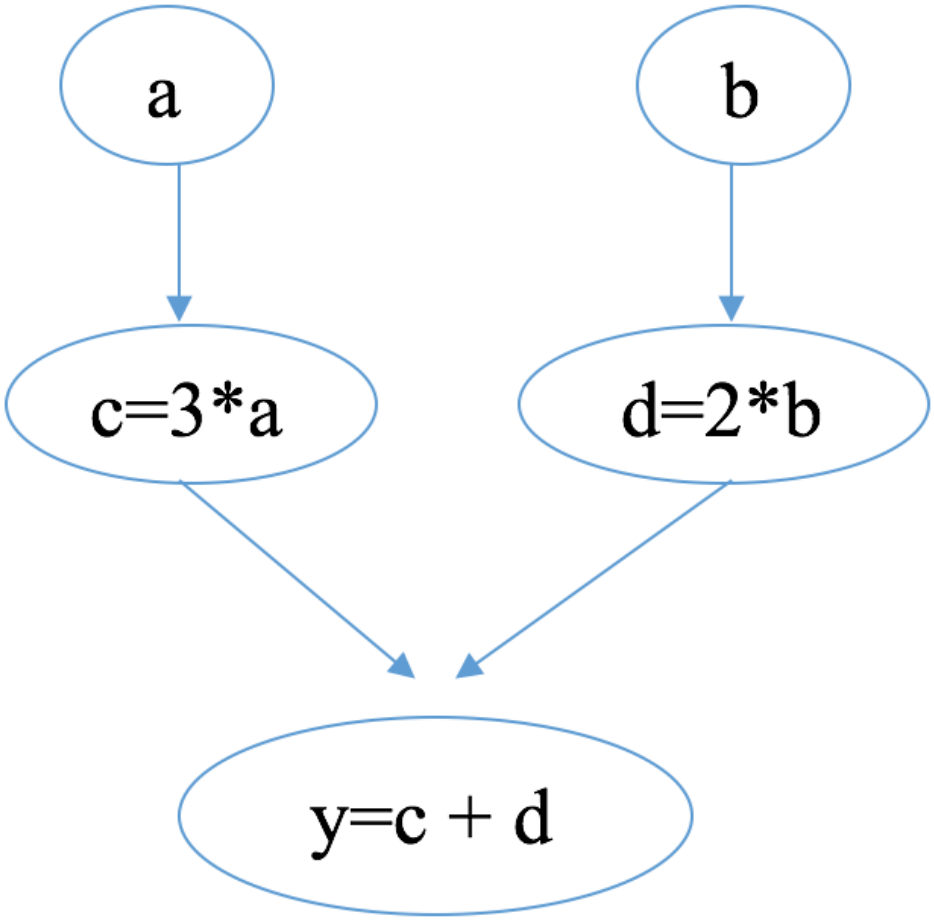

x_2
x_1
Output
Output
x_2
x_1
x_2
x_1
Output
Output

Regular installation

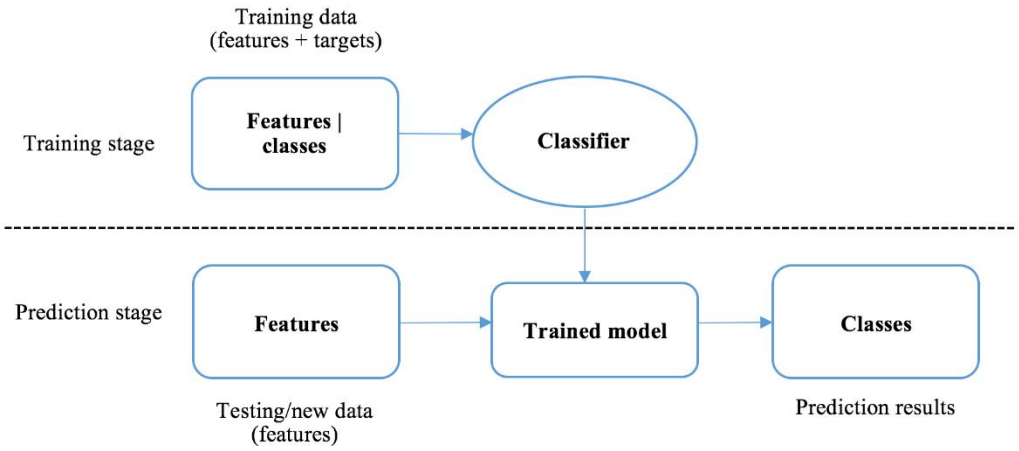
Follow the instructions for your operating system:

- Windows.
- macOS.
- Linux.

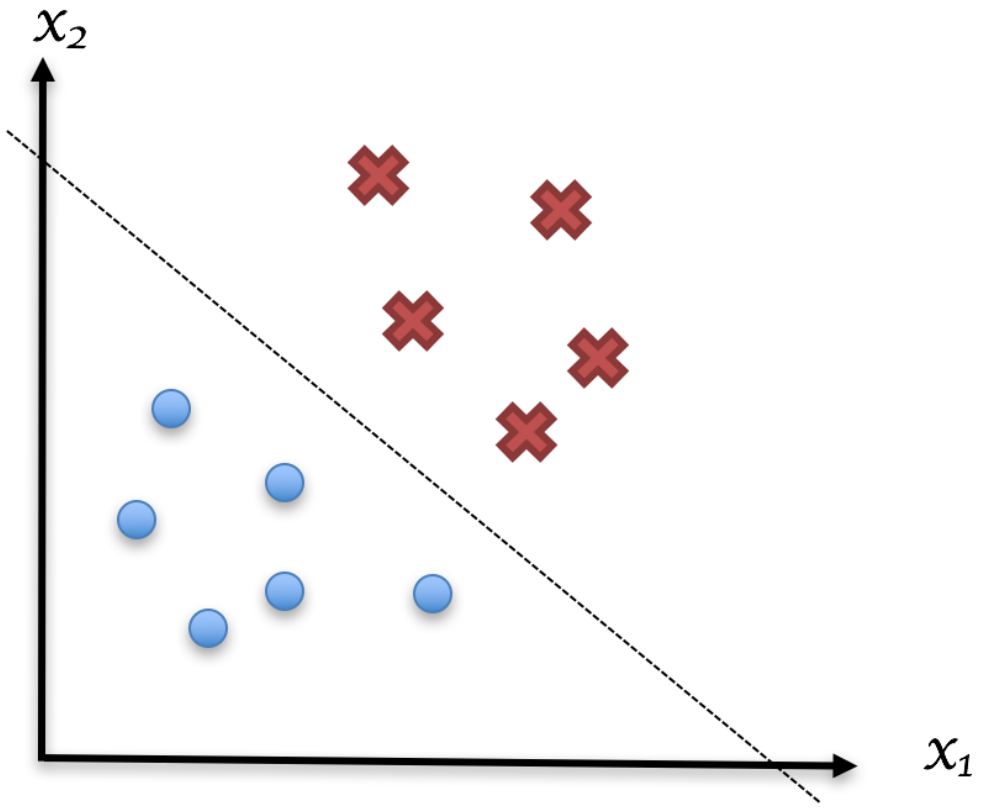
```
Python 3.7.2 (default, Dec 29 2018, 00:00:04)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda custom (64-bit) on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

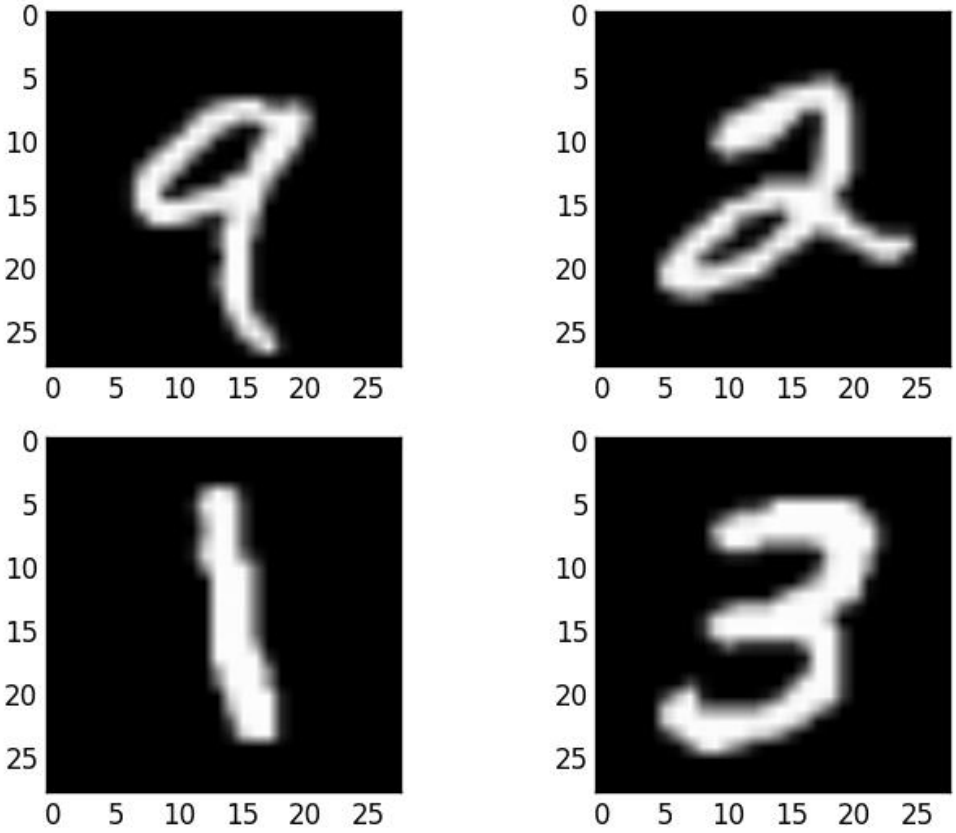

a
b
c=3*a
d=2*b
y=c + d

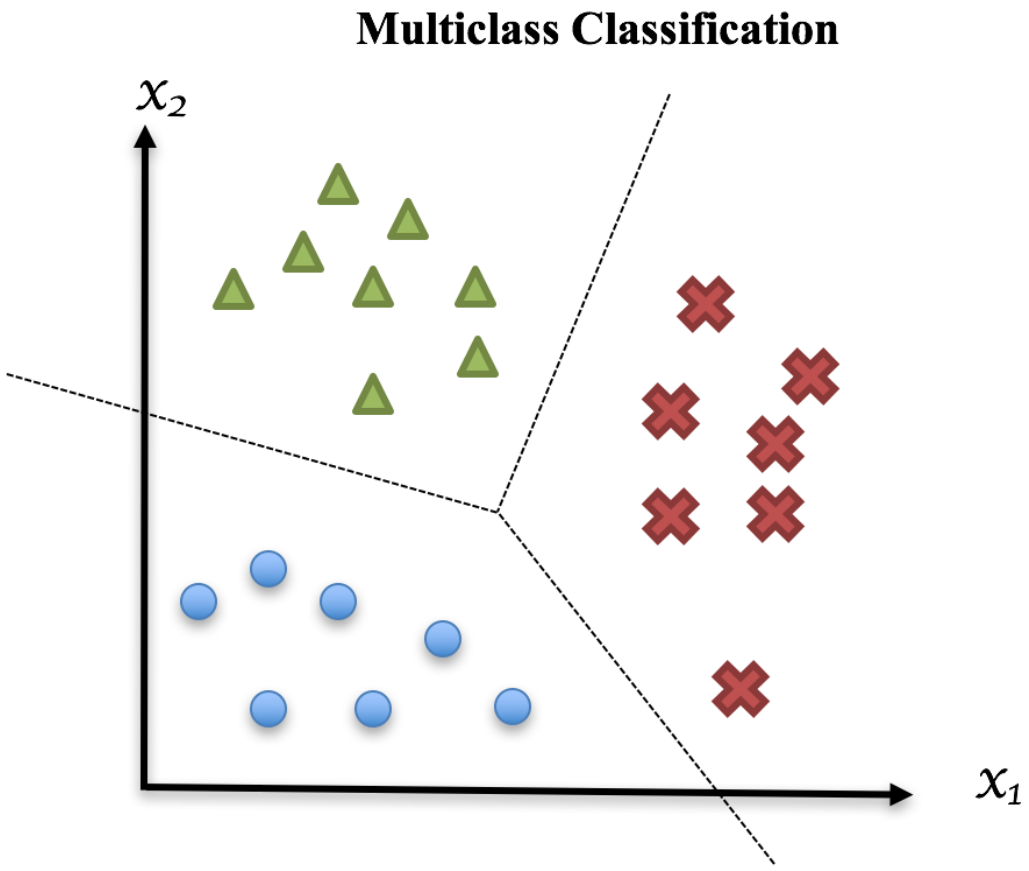
Chapter 2: Building a Movie Recommendation Engine with Naïve Bayes

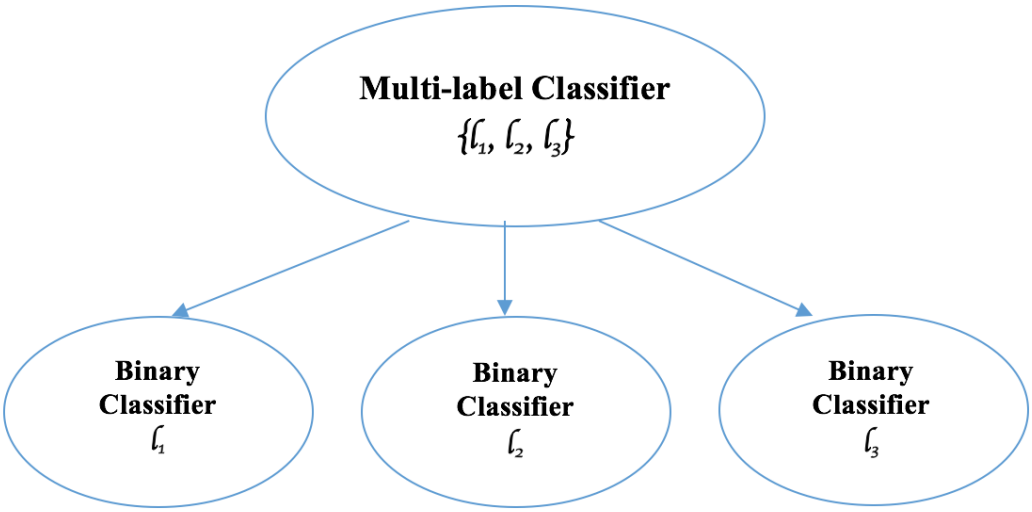


Binary Classification

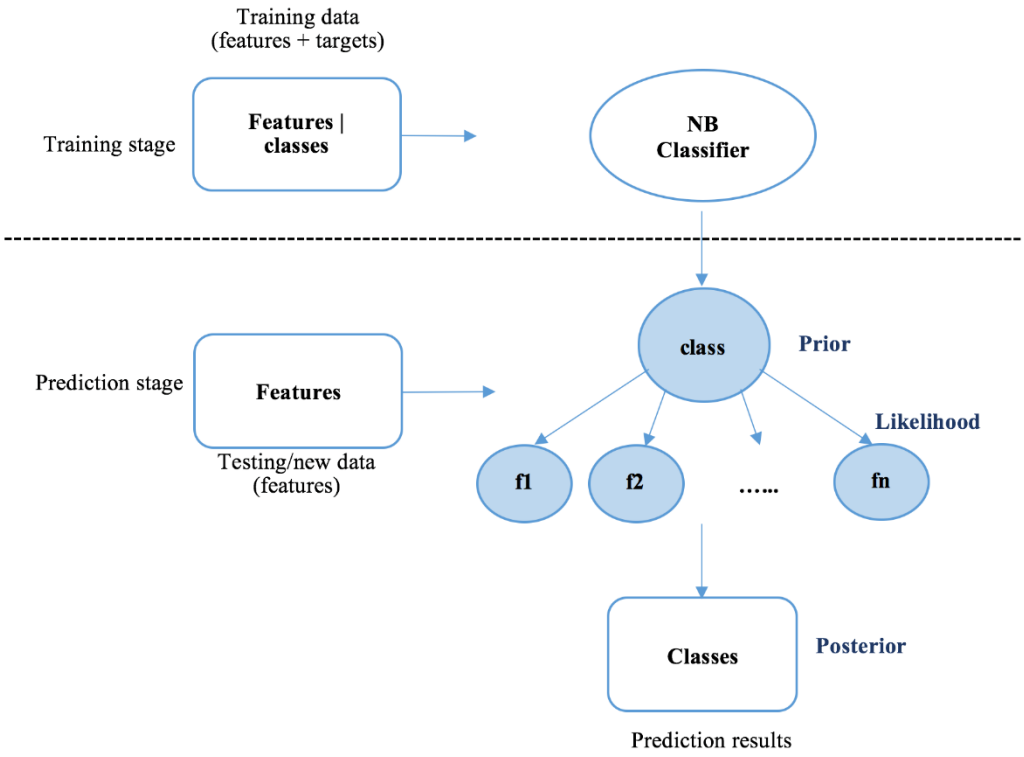



0
5
10
15
20
25
0 5 10 15 20 25

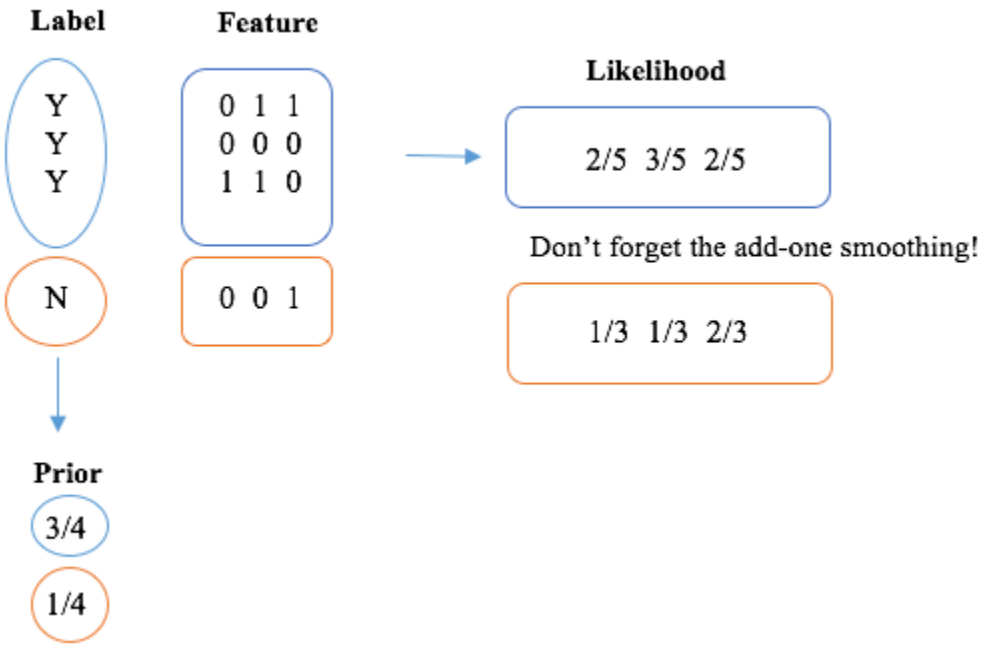

Multiclass Classification
x_2
x_1



	Cancer	**No Cancer**	Total
Test Positive	80	900	980
Test Negative	20	9000	9020
Total	100	9900	10000

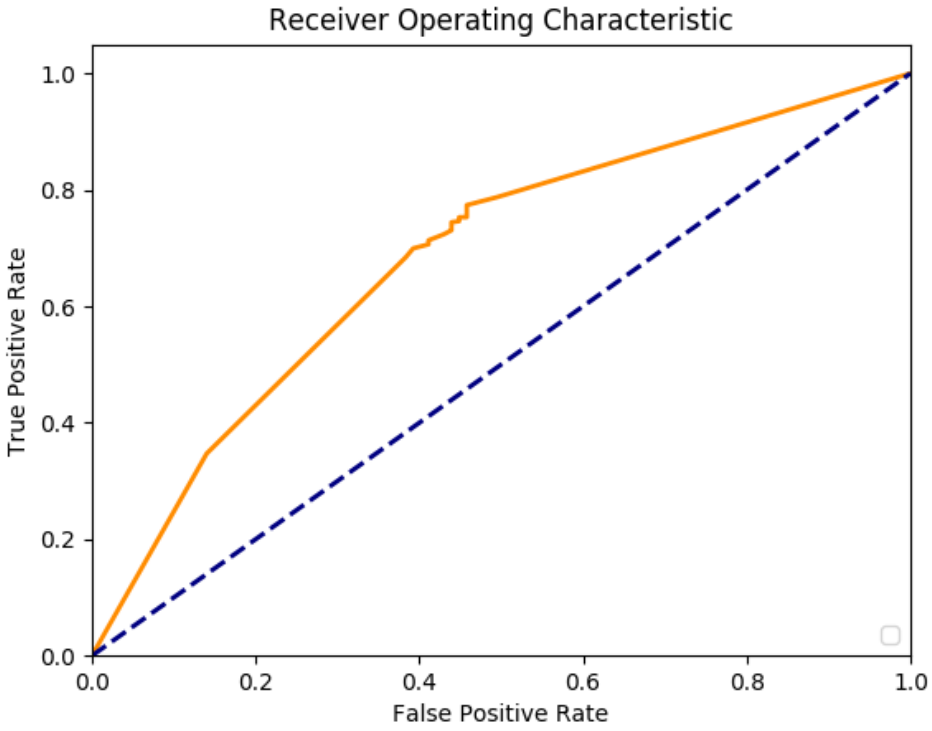


	ID	m_1	m_2	m_3	Whether the user likes the target movie
Training data	1	0	1	1	Y
	2	0	0	1	N
	3	0	0	0	Y
	4	1	1	0	Y
Testing case	5	1	1	0	?



		Predicted	
		Negative	Positive
Actual	Negative	TN	FP
	Positive	FN	TP

TN = True Negative
FP = False Positive
FN = False Negative
TP =True Positive



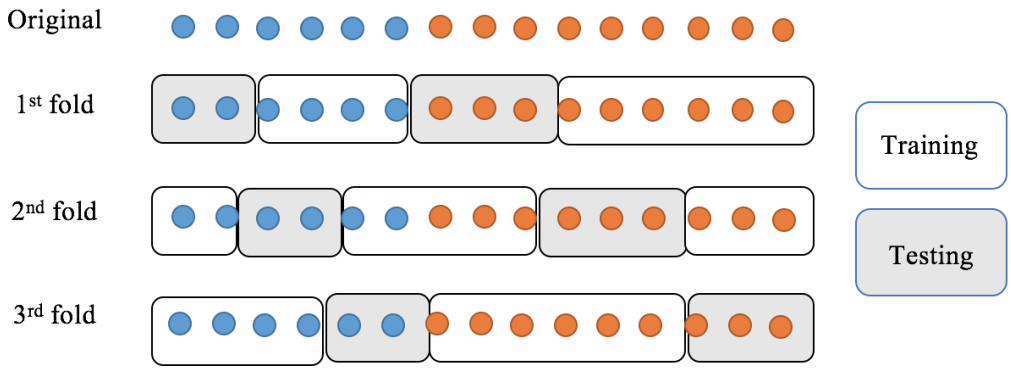
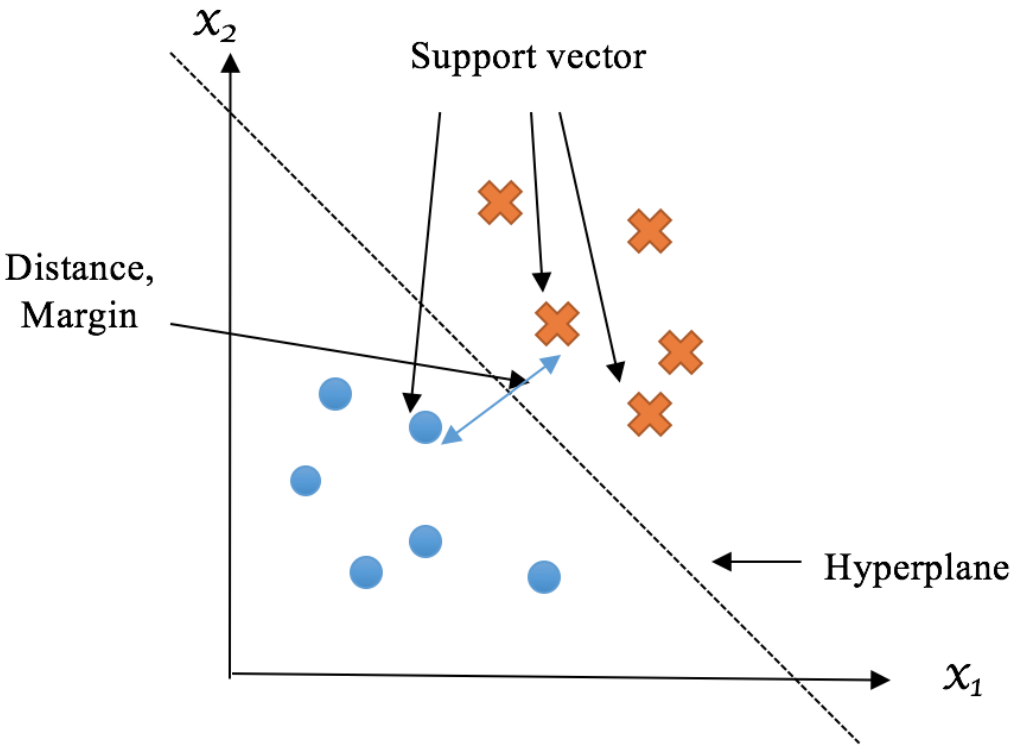
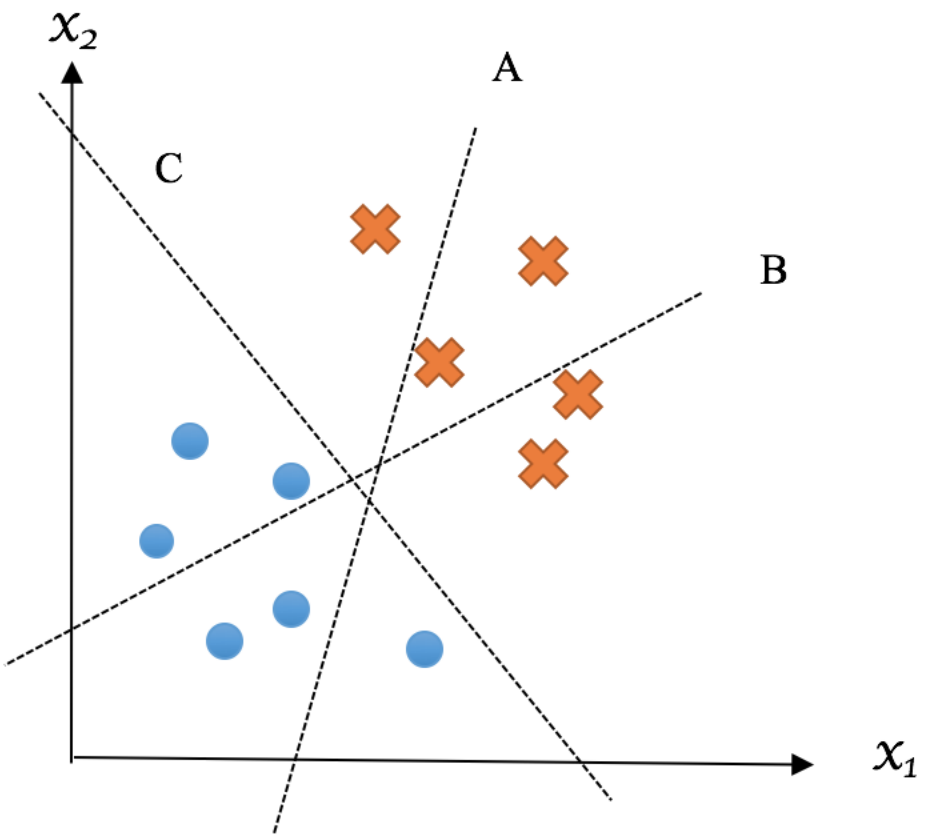
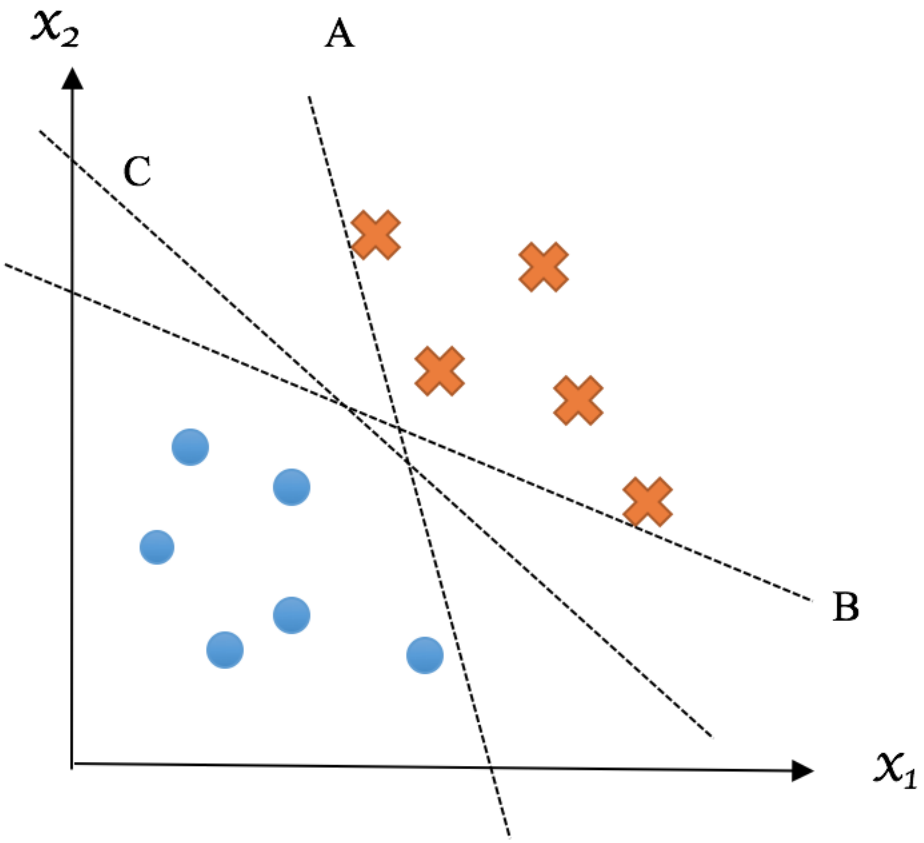


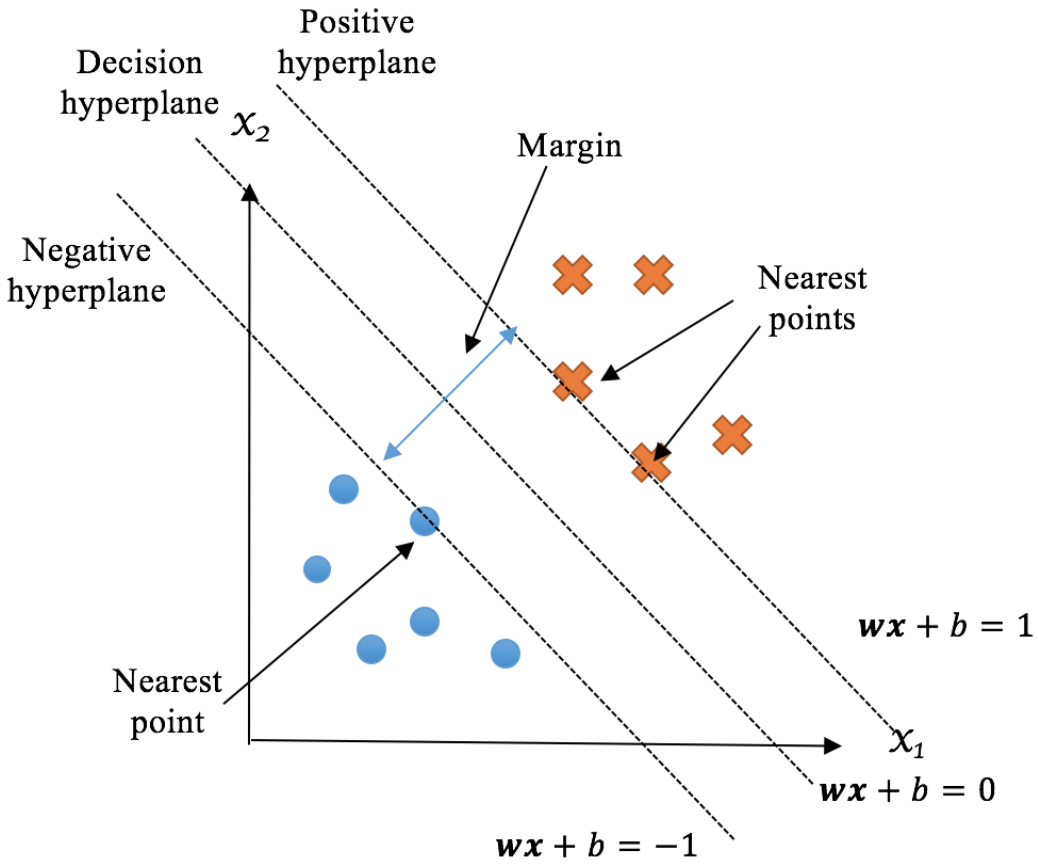
Diagram of 3-fold cross-validation

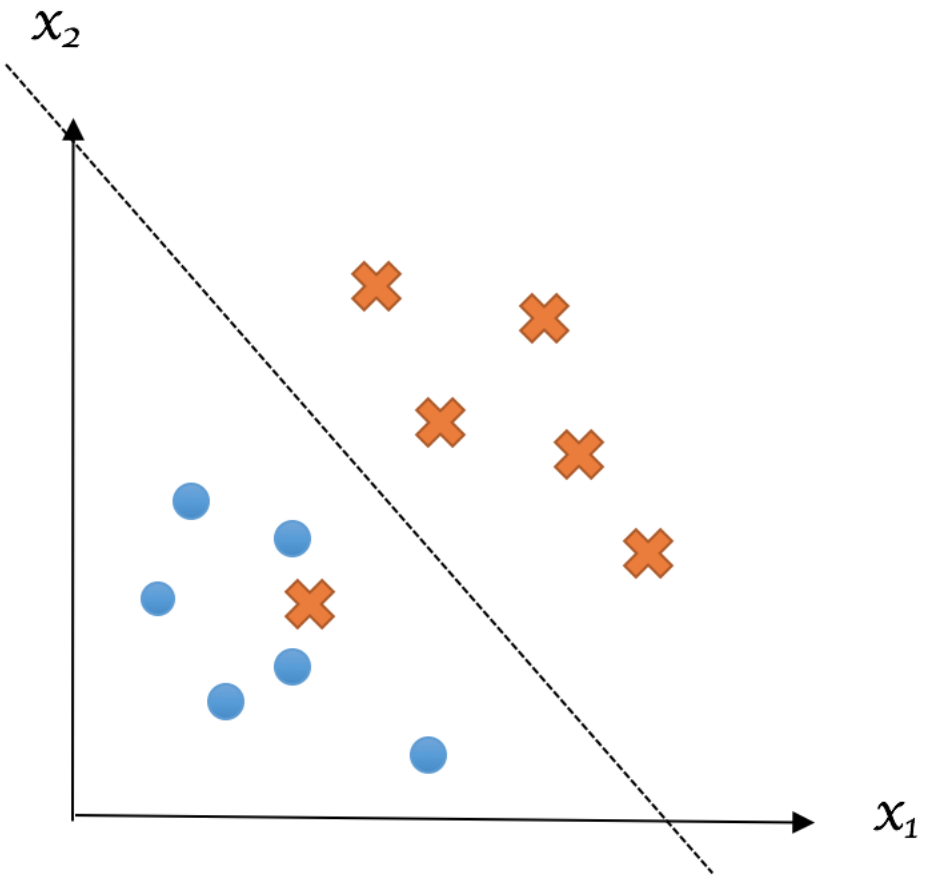
Chapter 3: Recognizing Faces with Support Vector Machine

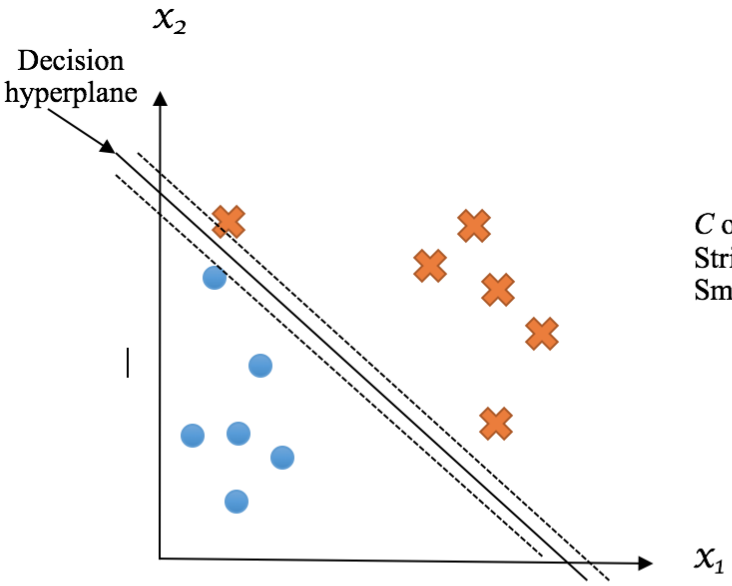



x_2
A
C
B
x_1

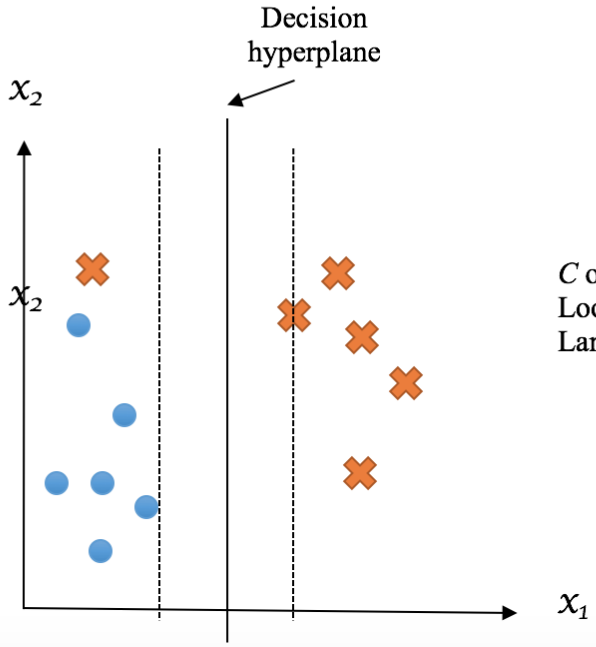

x_2
A
C
B
x_1


Positive
hyperplane
Decision
hyperplane
x_2
Margin
Negative
hyperplane
Nearest
points
Nearest
point
$wx + b = 1$
x_1
$wx + b = 0$
$wx + b = -1$

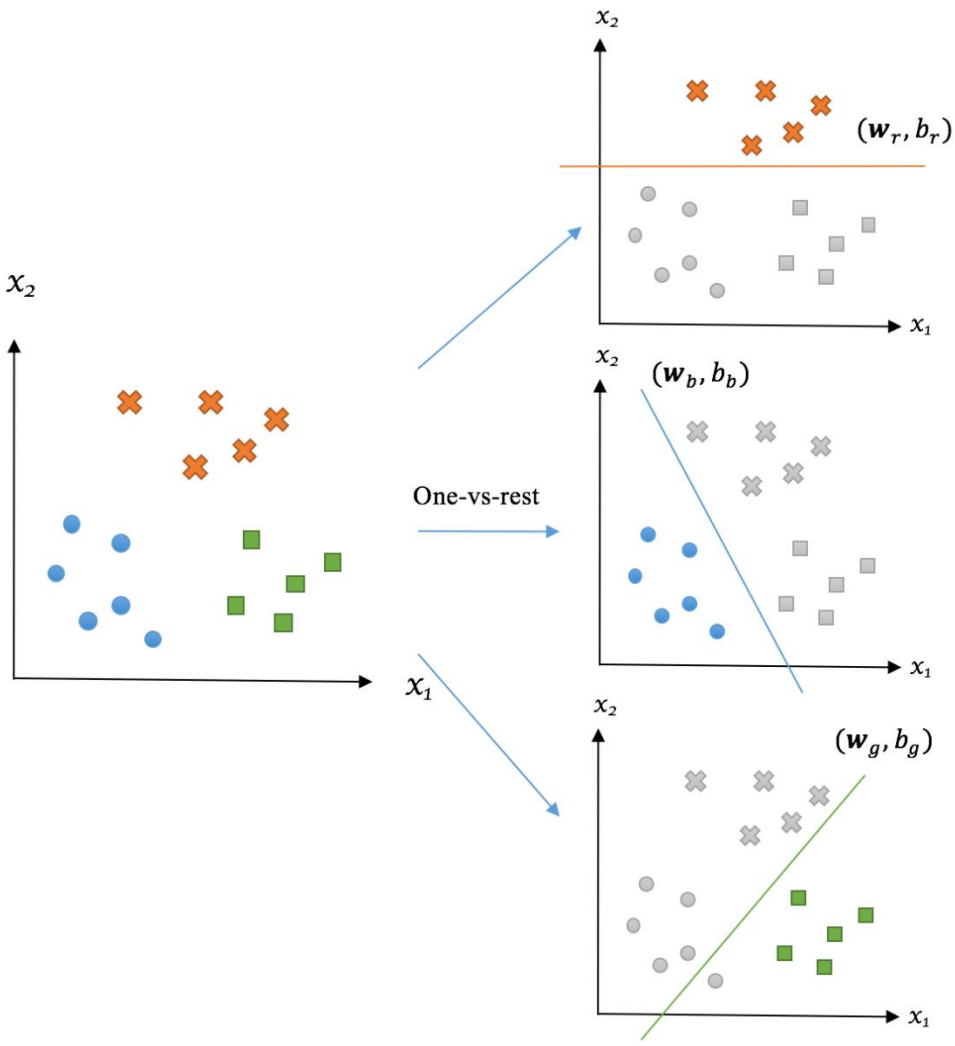

x_2
x_1

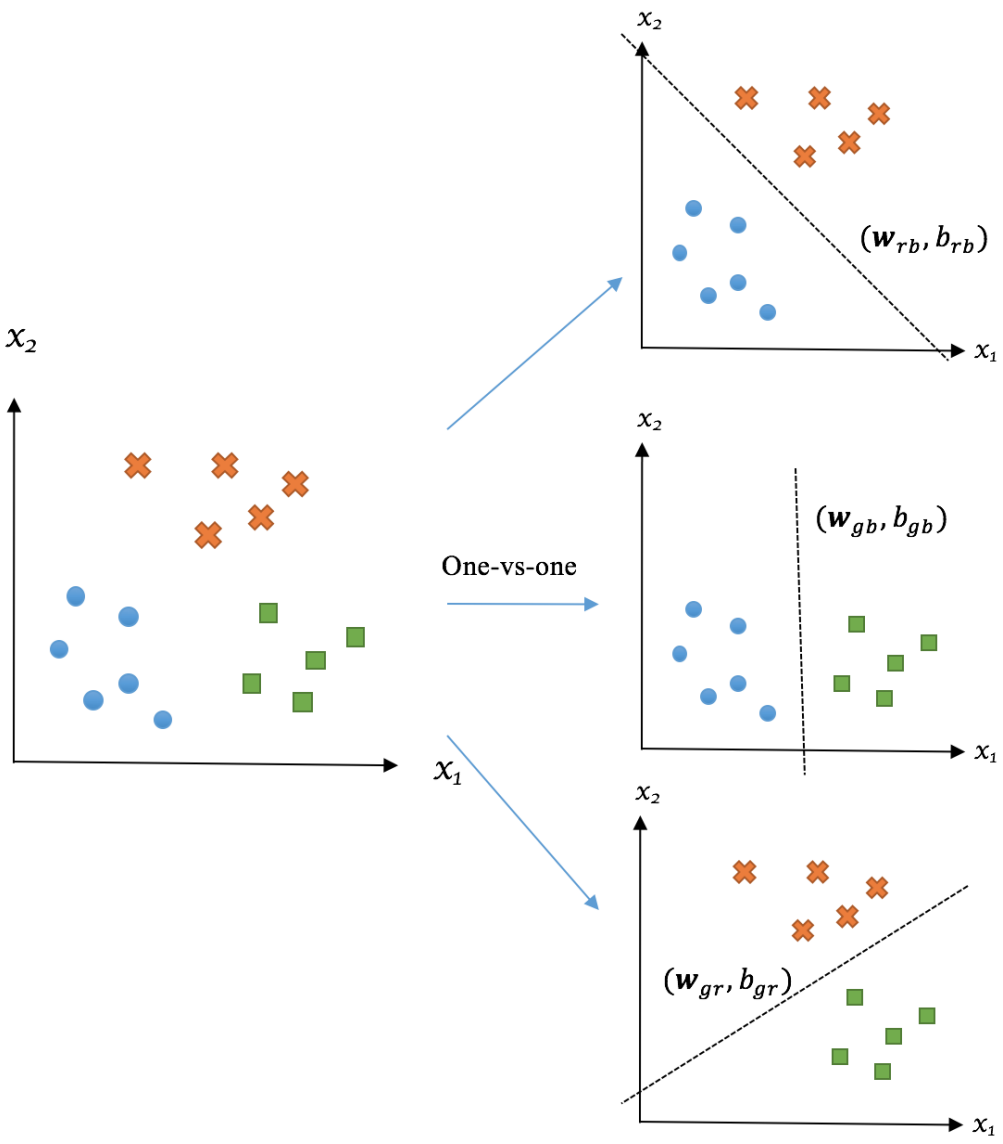


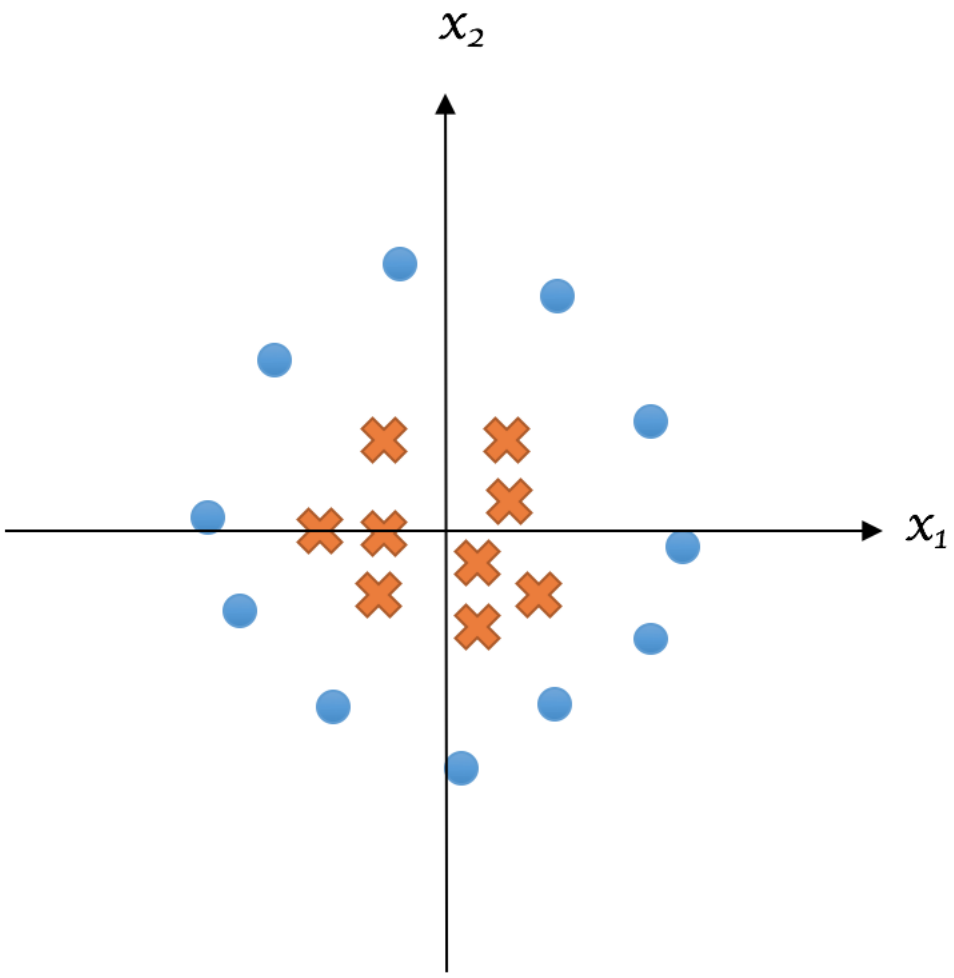
C of large values
Strict segregation
Small-margin hyperplane

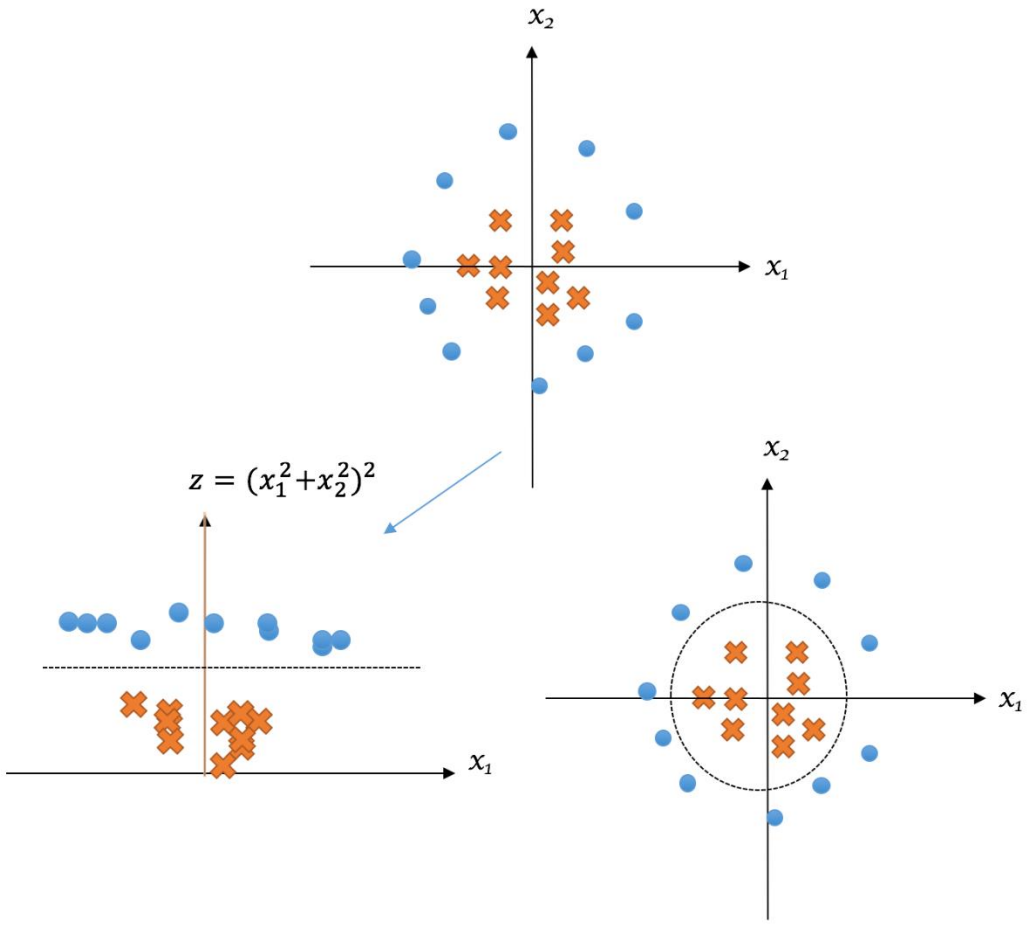


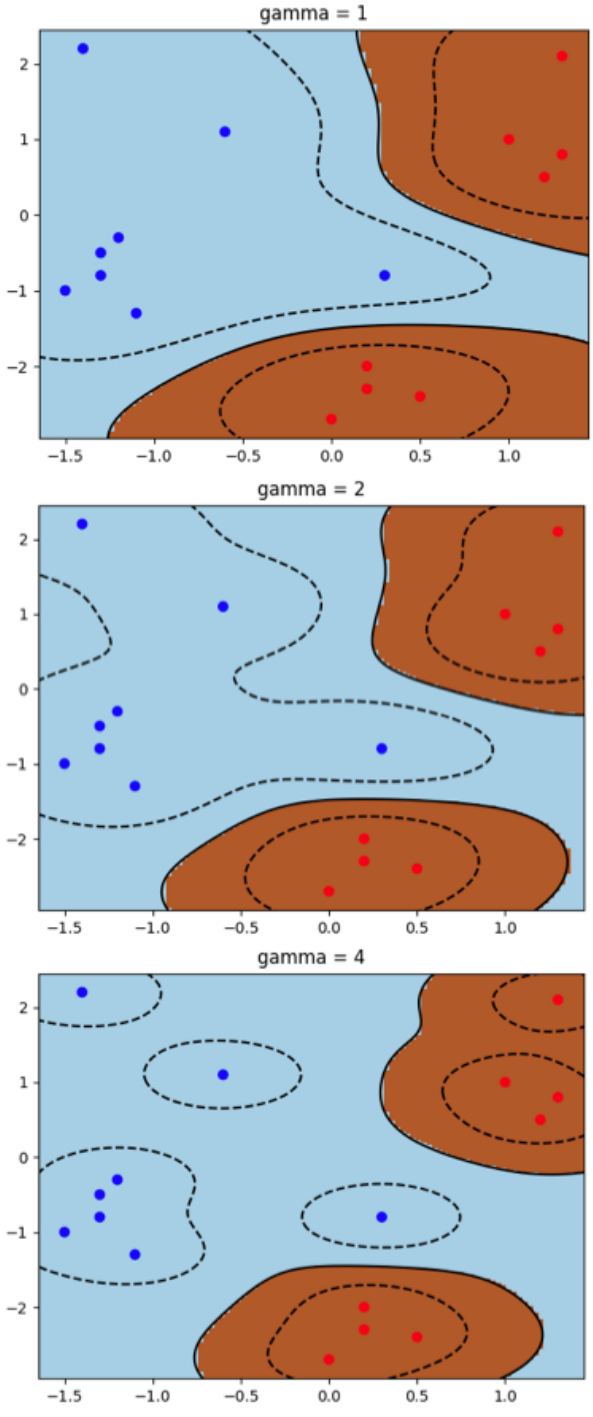
C of small values
Loose segregation
Large-margin hyperplane


x_2
x_1
One-vs-rest
(w_r, b_r)
(w_b, b_b)
(w_g, b_g)

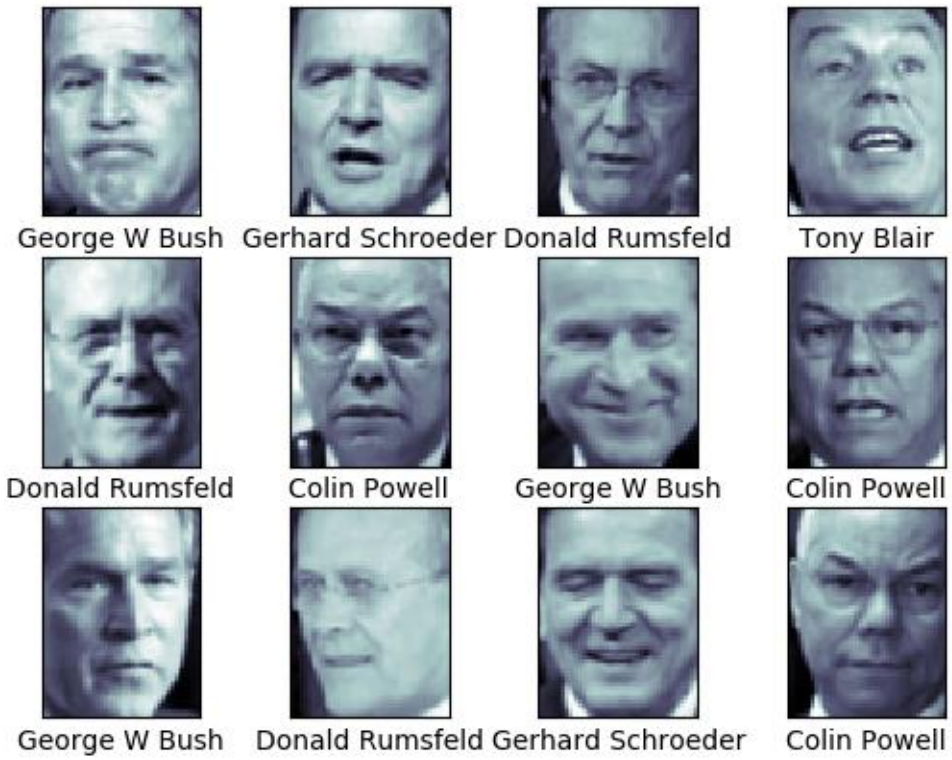

x_2
x_1
One-vs-one
$(\boldsymbol{w}_{rb}, b_{rb})$
$(\boldsymbol{w}_{gb}, b_{gb})$
$(\boldsymbol{w}_{gr}, b_{gr})$


x_2
x_1


x_2
x_1
$z = (x_1^2+x_2^2)^2$
x_1
x_2
x_1


gamma = 1
gamma = 2
gamma = 4
2
1
0
−1
−2
−1.5
−1.0
−0.5
0.0
0.5
1.0

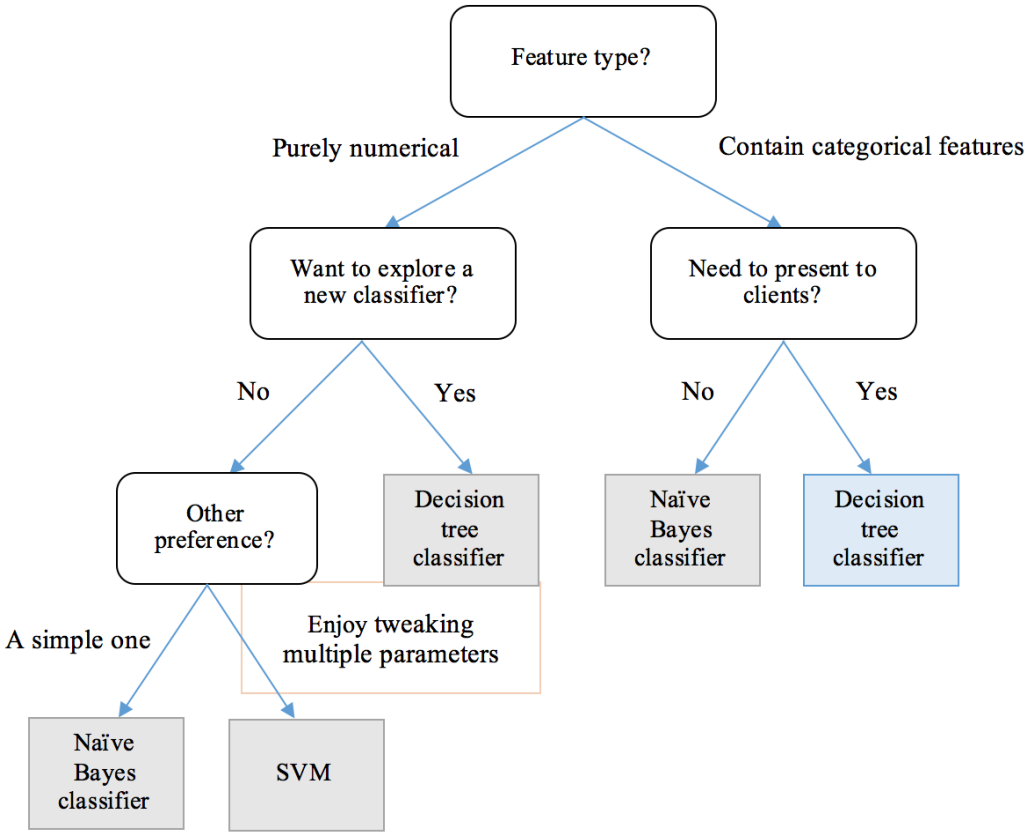
Scenario	Linear	RBF
Prior knowledge	If linearly separable	If nonlinearly separable
Visualizable data of 1 to 3 dimension(s)	If linearly separable	If nonlinearly separable
Both numbers of features and instances are large	First choice	
Features ≫ Instances	First choice	
Instances ≫ Features	First choice	
Others		First choice



Chapter 4: Predicting Online Ad Click-Through with Tree-Based Algorithms

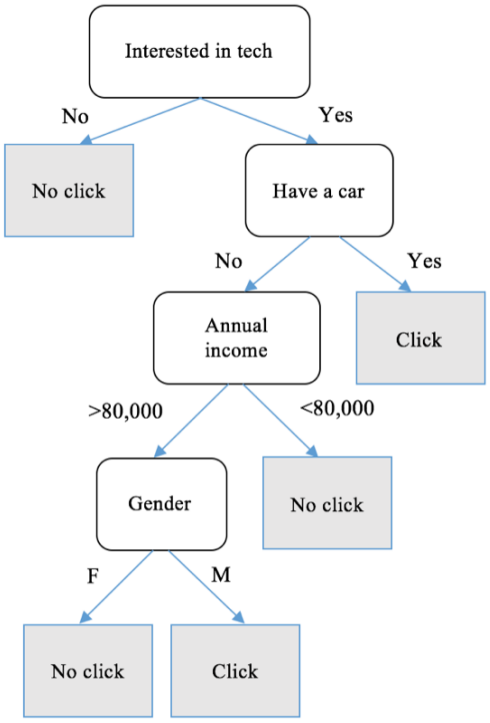
Ad category	Site category	Site domain	User age	User gender	User occupation	Interested in sports	Interested in tech	Click
Auto	News	cnn.com	25-34	M	Professional	True	True	1
Fashion	News	bbc.com	35-54	F	Professional	False	False	0
Auto	Edu	onlinestudy.com	17-24	F	Student	True	True	0
Food	Entertainment	movie.com	25-34	M	Clerk	True	False	1
Fashion	Sports	football.com	55+	M	Retired	True	False	0
…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…

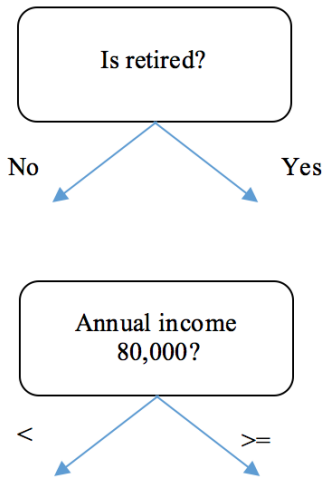
Food	News	abc.com	17-24	M	Student	True	True	**?**
Auto	Entertainment	movie.com	35-54	F	Professional	True	False	**?**


Feature type?
Purely numerical
Contain categorical features
Want to explore a new classifier?
Need to present to clients?
No
Yes
No
Yes
Other preference?
Decision tree classifier
Naïve Bayes classifier
Decision tree classifier
A simple one
Enjoy tweaking multiple parameters
Naïve Bayes classifier
SVM

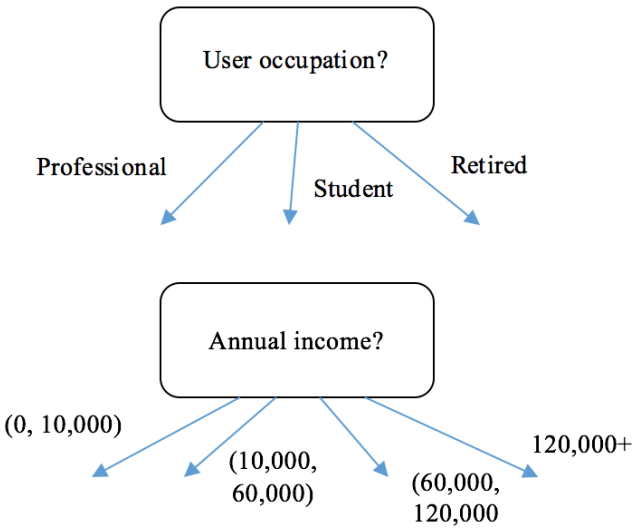
User gender	Annual income	Have a car	Interested in tech	Click
M	200,000	True	True	1
F	5,000	False	False	0
F	100,000	True	True	1
M	10,000	True	False	0
M	80,000	False	False	0
…	…	…	…	…
…	…	…	…	…

M	120,000	True	True	?
F	70,000	False	True	?

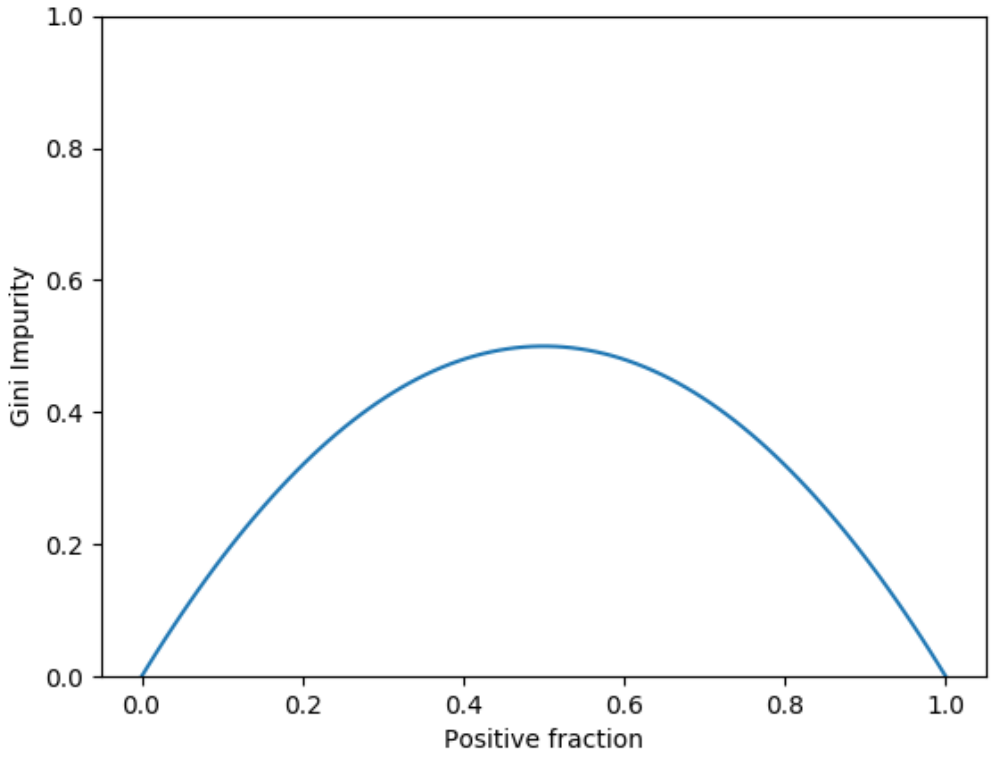




Binary splitting



Multiway splitting

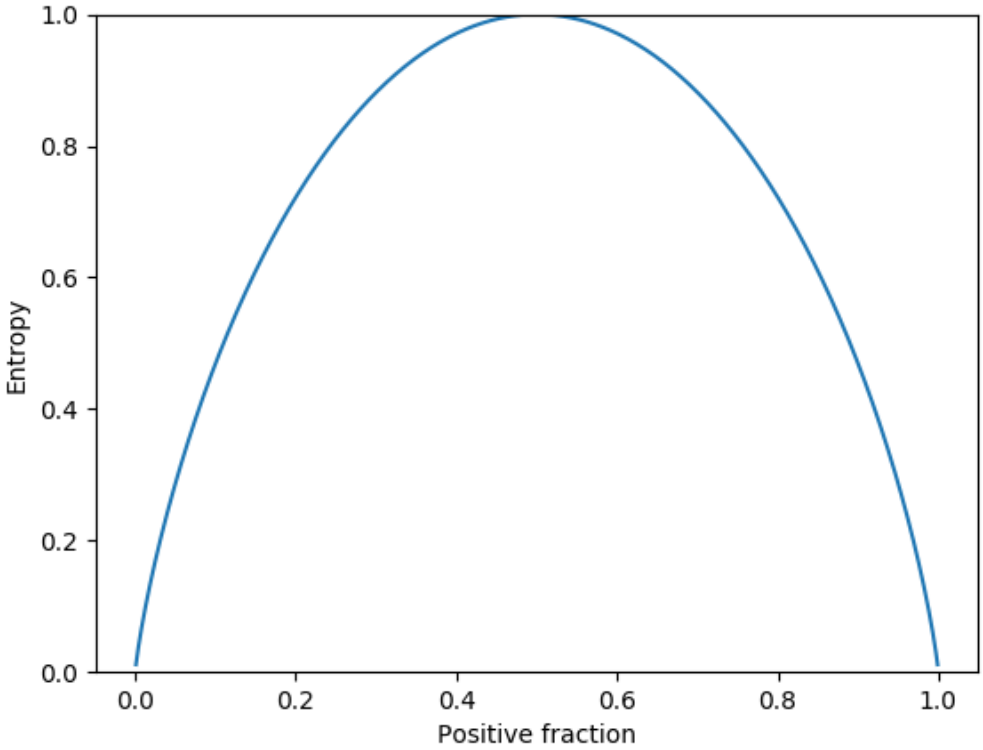


User gender	Interested in tech	Click	Group by gender
M	True	1	Group 1
F	False	0	Group 2
F	True	1	Group 2
M	False	0	Group 1
M	False	1	Group 1

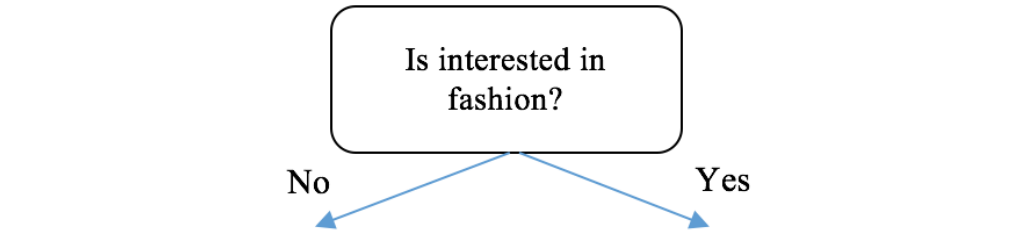
#1 split based on gender

User gender	Interested in tech	Click	Group by interest
M	True	1	Group 1
F	False	0	Group 2
F	True	1	Group 1
M	False	0	Group 2
M	False	1	Group 2

#2 split based on interest in tech


1.0
0.8
0.6
0.4
0.2
0.0
Entropy
0.0
0.2
0.4
0.6
0.8
1.0
Positive fraction

User interest	User occupation	Click
Tech	Professional	1
Fashion	Student	0
Fashion	Professional	0
Sports	Student	0
Tech	Student	1
Tech	Retired	0
Sports	Professional	1

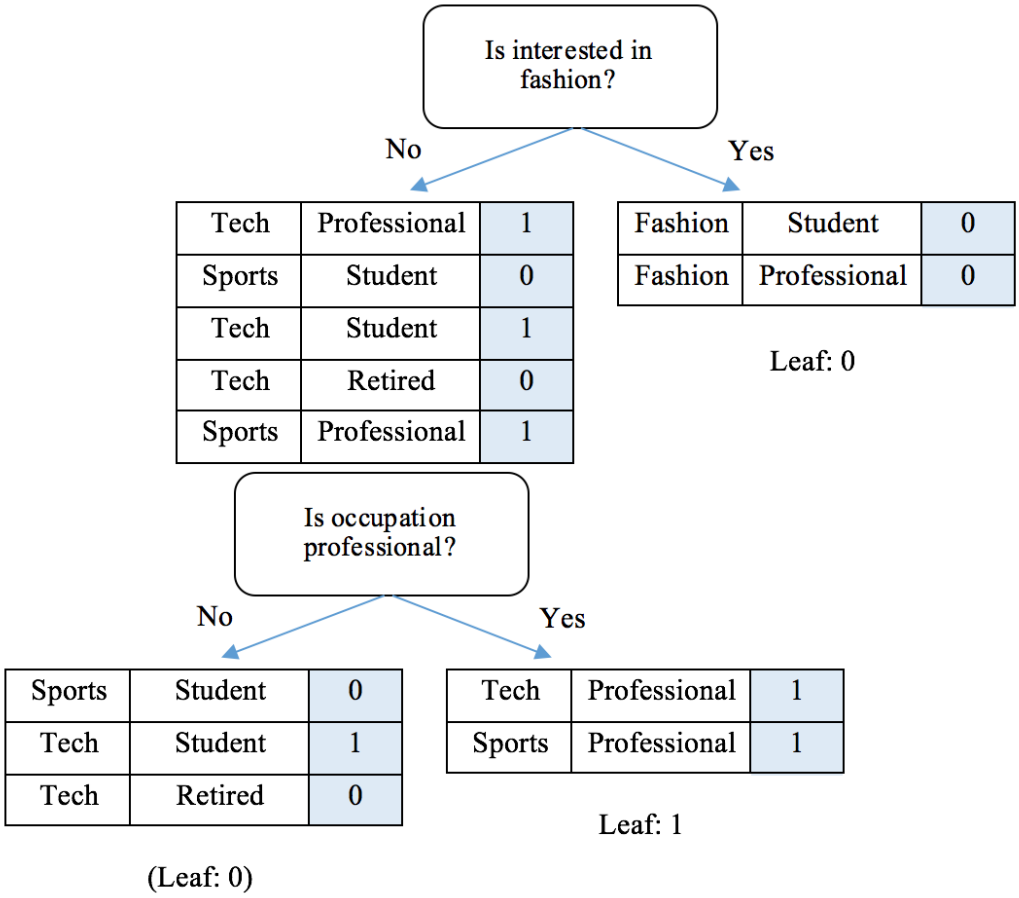


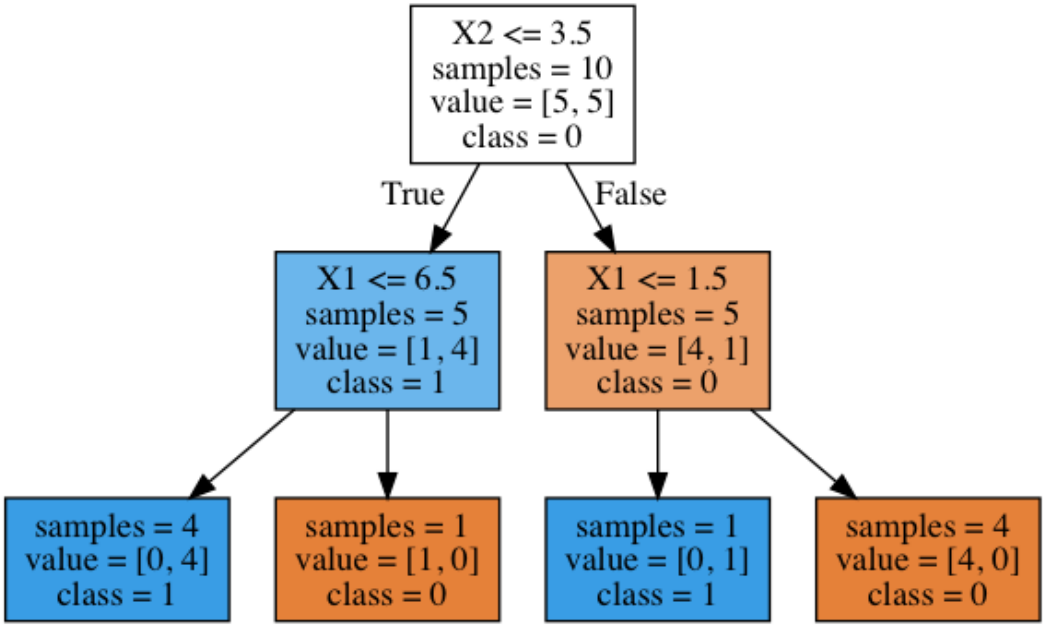
Tech	Professional	1
Sports	Student	0
Tech	Student	1
Tech	Retired	0
Sports	Professional	1

(Leaf: 1)

Fashion	Student	0
Fashion	Professional	0

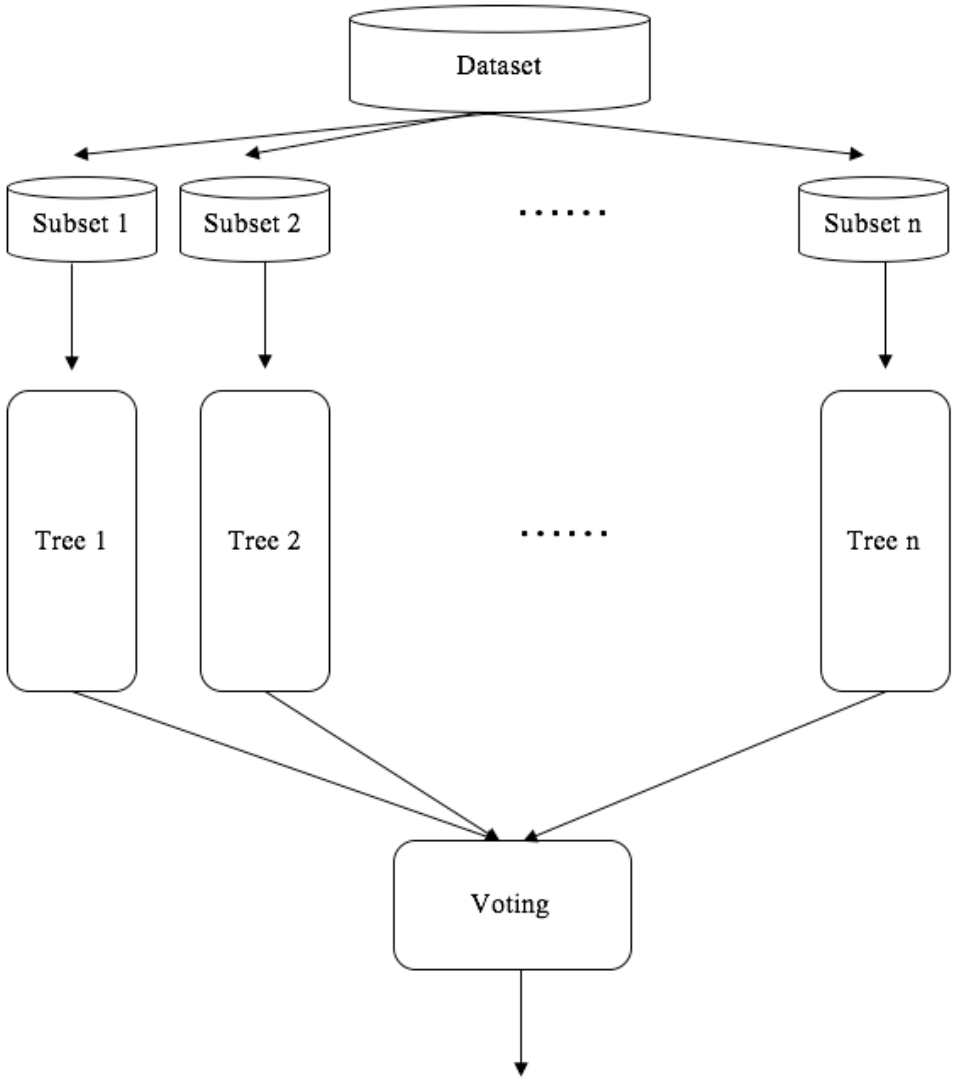
Leaf: 0


Is interested in fashion?
No
Yes
Tech Professional 1
Sports Student 0
Tech Student 1
Tech Retired 0
Sports Professional 1
Fashion Student 0
Fashion Professional 0
Leaf: 0
Is occupation professional?
No
Yes
Sports Student 0
Tech Student 1
Tech Retired 0
Tech Professional 1
Sports Professional 1
Leaf: 1
(Leaf: 0)

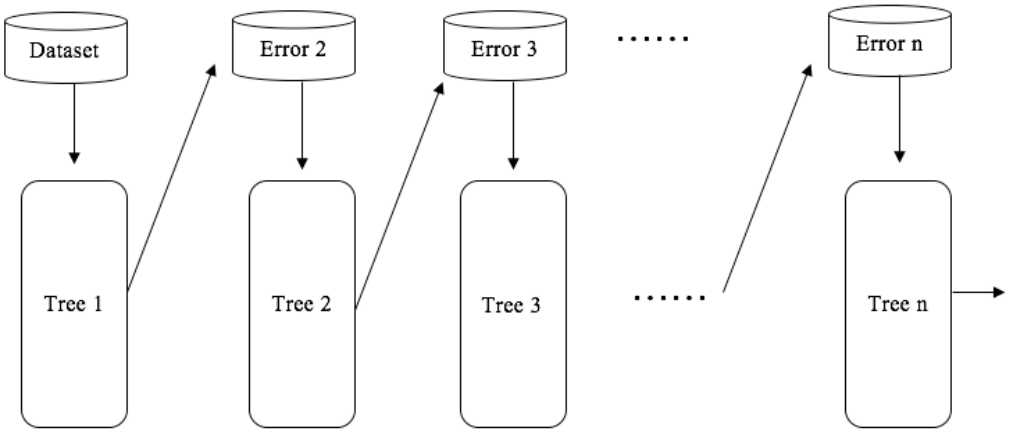

X2 <= 3.5
samples = 10
value = [5, 5]
class = 0
True
False
X1 <= 6.5
samples = 5
value = [1, 4]
class = 1
X1 <= 1.5
samples = 5
value = [4, 1]
class = 0
samples = 4
value = [0, 4]
class = 1
samples = 1
value = [1, 0]
class = 0
samples = 1
value = [0, 1]
class = 1
samples = 4
value = [4, 0]
class = 0

Field	Description	Example values
id	ad identifier	such as '1000009418151094273', '10000169349117863715'
click	'0' for non-click, '1' for click	0, 1
hour	in the format of YYMMDDHH	'14102100'
C1	anonymized categorical variable	'1005', '1002'
banner_pos	where banner is located	1, 0
site_id	site identifier	'1fbe01fe', 'fe8cc448', 'd6137915'
site_domain	hashed site domain	'bb1ef334', 'f3845767'
site_category	hashed site category	'28905ebd', '28905ebd'
app_id	mobile app identifier	'ecad2386'
app_domain	mobile app domain	'7801e8d9'
app_category	category of app	'07d7df22'
device_id	mobile device identifier	'a99f214a'
device_ip	IP address	'ddd2926e'
device_model	such as iphone 6, Samsung, hashed	'44956a24'
device_type	such as tablet, smartphone, hashed	1
device_conn_type	Wi-Fi or 3G for example, again hashed in the data	0, 2
C14-C21	anonymized categorical variables	

```
id                   click  hour      C1    banner_pos  site_id   site_domain  site_category  app_id
 app_domain  app_category  device_id  device_ip  device_model  device_type  device_conn_type  C14    C15
  C16  C17   C18  C19  C20     C21
10000094181510942 73   0      14102100  1005  0          1fbe01fe  f3845767     28905ebd       ecad2386
 7801e8d9    07d7df22      a99f214a   ddd2926e   44956a24      1            2                 15706  320
  50   1722  0    35   -1      79
10000169349117863715  0      14102100  1005  0          1fbe01fe  f3845767     28905ebd       ecad2386
 7801e8d9    07d7df22      a99f214a   96809ac8   711ee120      1            0                 15704  320
  50   1722  0    35   100084  79
10000371904215119486  0      14102100  1005  0          1fbe01fe  f3845767     28905ebd       ecad2386
 7801e8d9    07d7df22      a99f214a   b3cf8def   8a4875bd      1            0                 15704  320
  50   1722  0    35   100084  79
10000640724480838376  0      14102100  1005  0          1fbe01fe  f3845767     28905ebd       ecad2386
 7801e8d9    07d7df22      a99f214a   e8275b8f   6332421a      1            0                 15706  320
  50   1722  0    35   100084  79
10000679056417042096  0      14102100  1005  1          fe8cc448  9166c161     0569f928       ecad2386
 7801e8d9    07d7df22      a99f214a   9644d0bf   779d90c2      1            0                 18993  320
  50   2161  0    35   -1      157
10000720757801103869  0      14102100  1005  0          d6137915  bb1ef334     f028772b       ecad2386
 7801e8d9    07d7df22      a99f214a   05241af0   8a4875bd      1            0                 16920  320
  50   1899  0    431  100077  117
10000724729988544911  0      14102100  1005  0          8fda644b  25d4cfcd     f028772b       ecad2386
 7801e8d9    07d7df22      a99f214a   b264c159   be6db1d7      1            0                 20362  320
  50   2333  0    39   -1      157
10000918755742328737  0      14102100  1005  1          e151e245  7e091613     f028772b       ecad2386
 7801e8d9    07d7df22      a99f214a   e6f67278   be74e6fe      1            0                 20632  320
  50   2374  3    39   -1      23
10000949271186029916  1      14102100  1005  0          1fbe01fe  f3845767     28905ebd       ecad2386
 7801e8d9    07d7df22      a99f214a   37e8da74   5db079b5      1            2                 15707  320
  50   1722  0    35   -1      79
```



Random forest

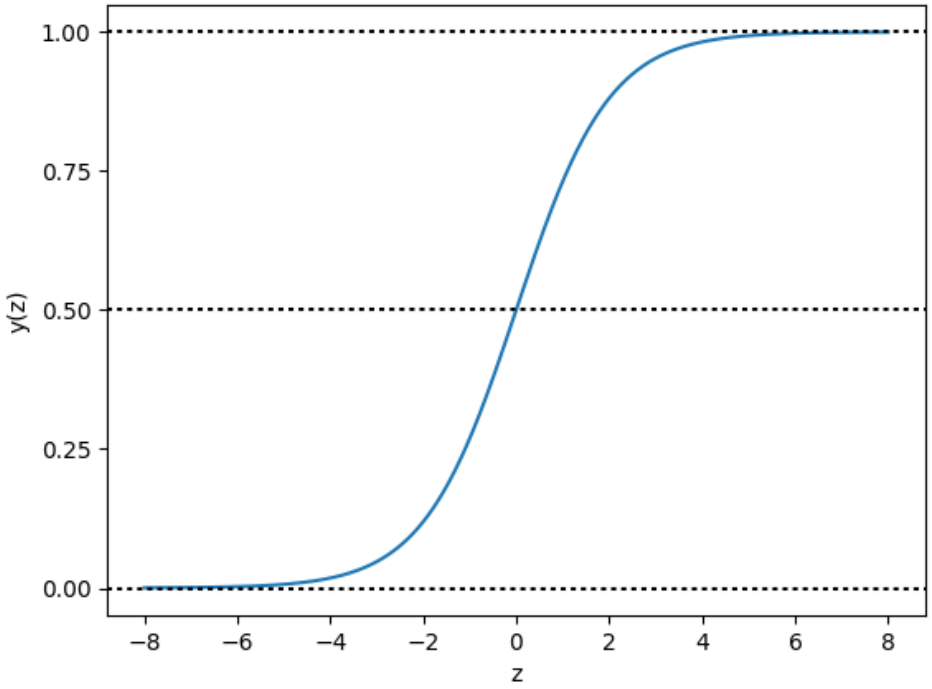


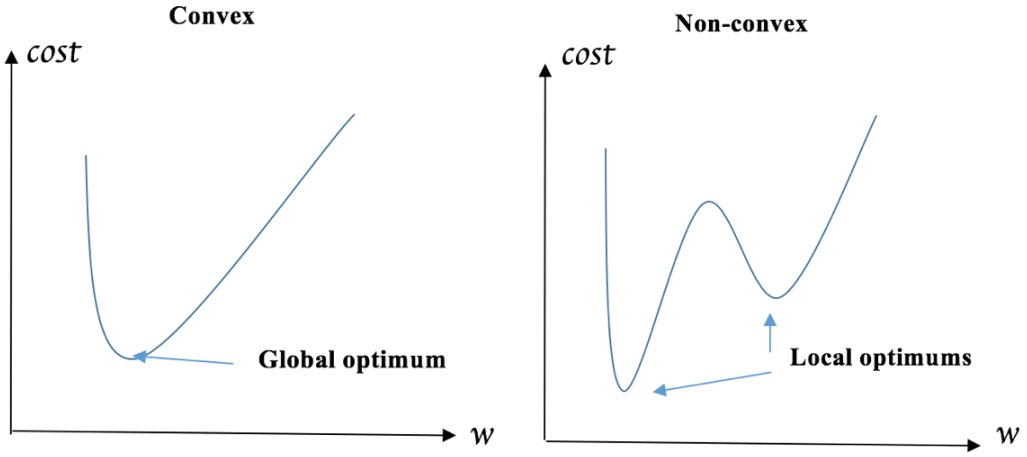
Gradient boosted trees

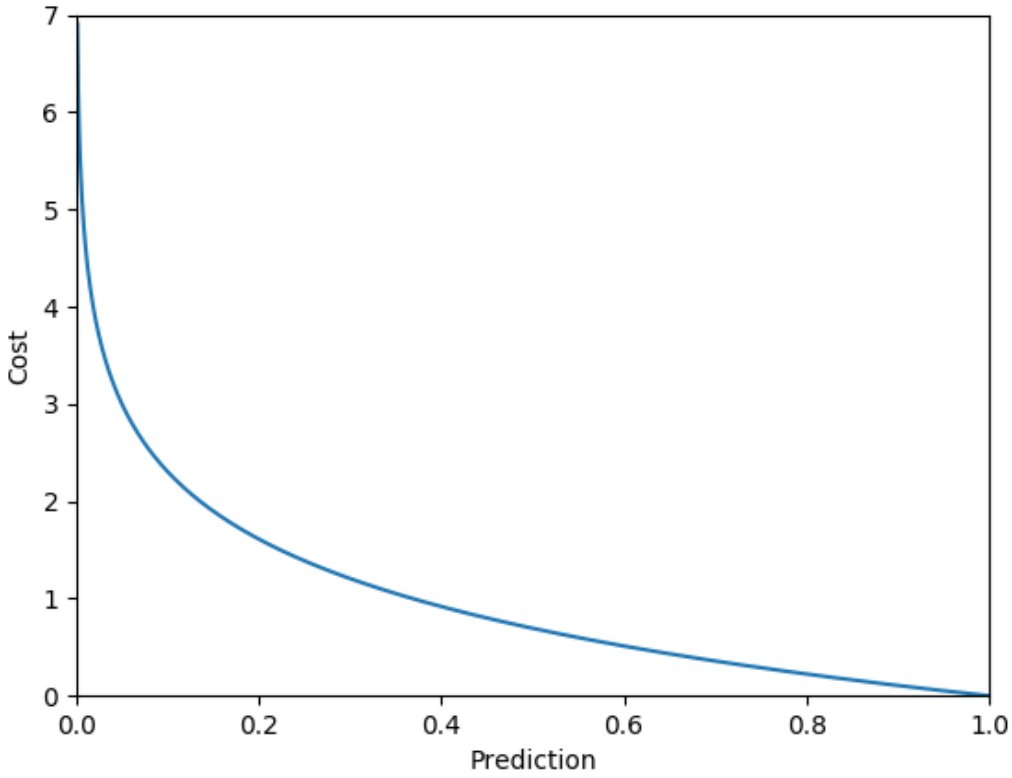
Chapter 5: Predicting Online Ads Click-Through with Logistic Regression

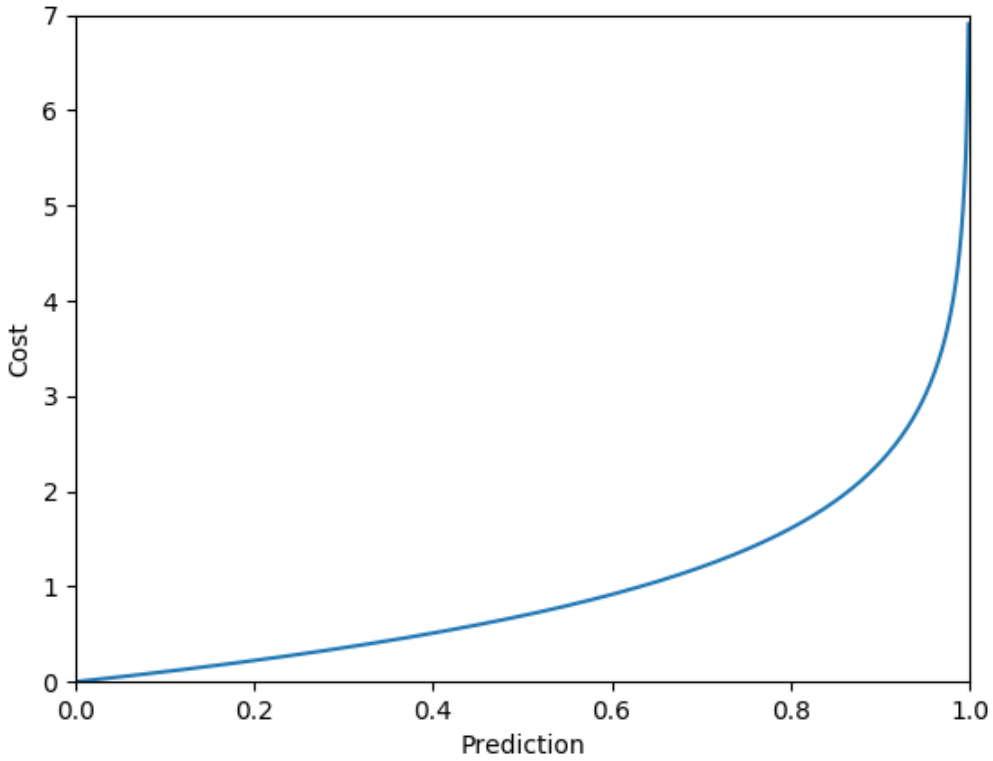
User interest
Tech
Fashion
Fashion
Sports
Tech
Tech
Sports

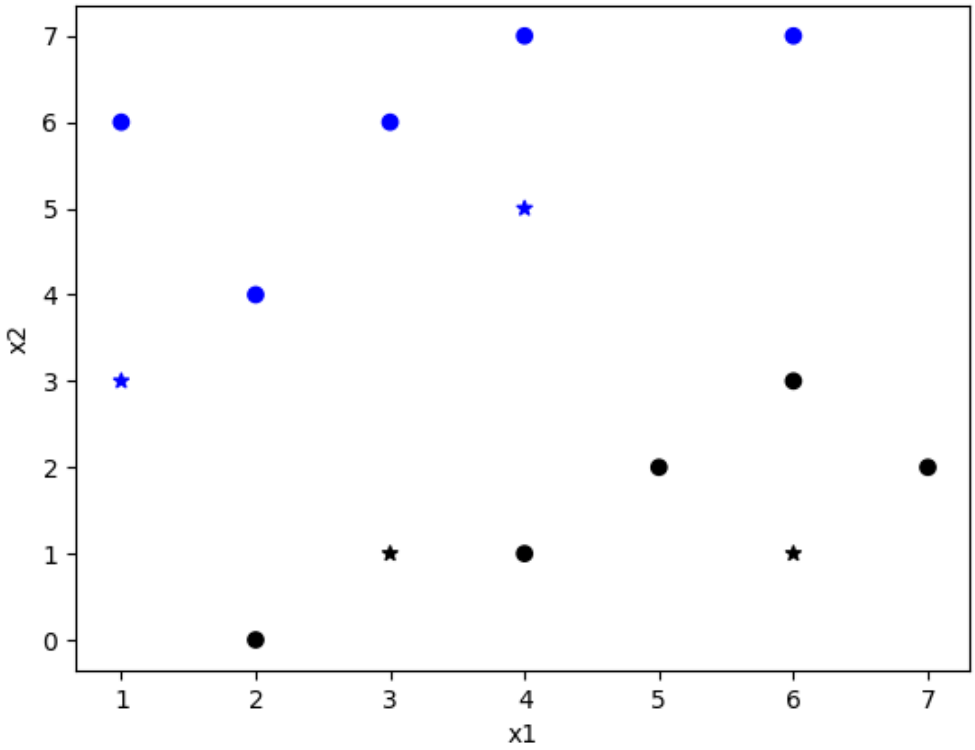
Interest: tech	Interest: fashion	Interest: sports
1	0	0
0	1	0
0	1	0
0	0	1
1	0	0
1	0	0
0	0	1

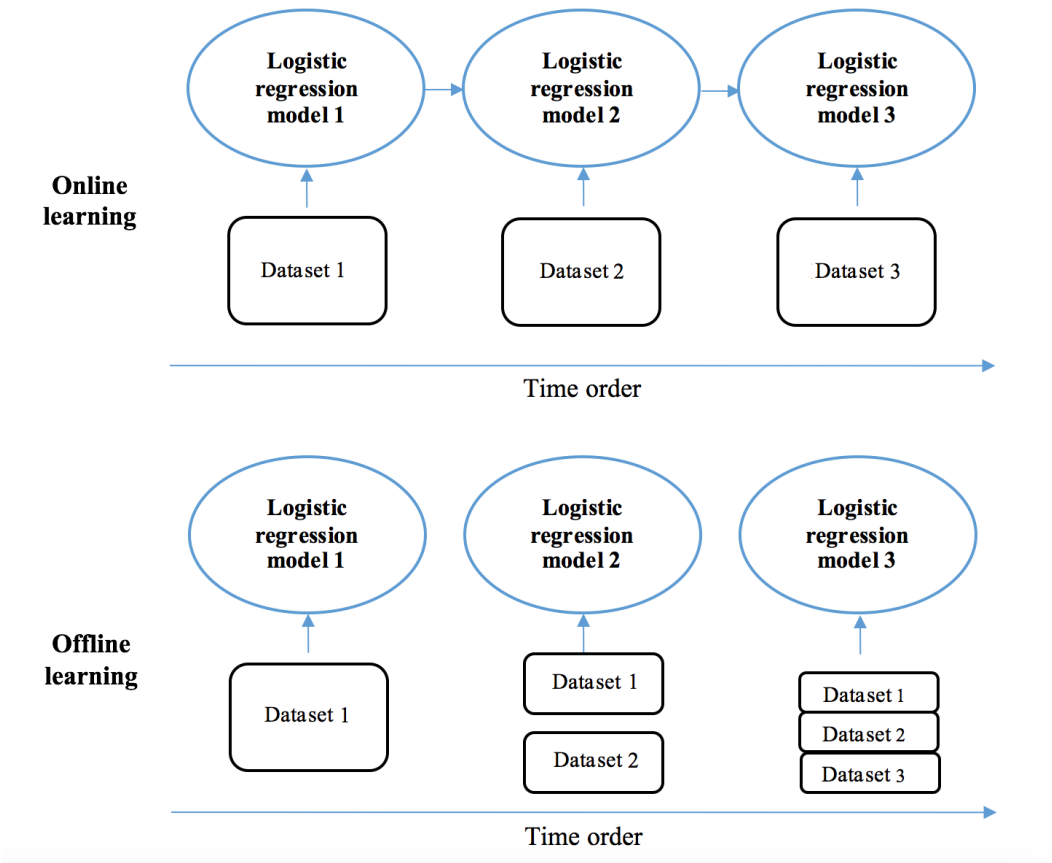

1.00
0.75
0.50
0.25
0.00
y(z)
−8
−6
−4
−2
0
2
4
6
8
z


Convex
cost
Global optimum
w
Non-convex
cost
Local optimums
w

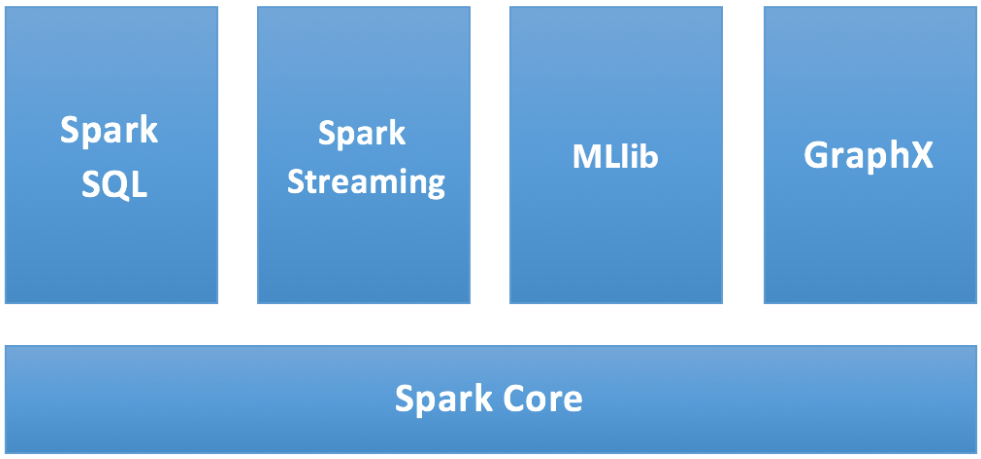

7
6
5
4
3
2
1
0
Cost
0.0
0.2
0.4
0.6
0.8
1.0
Prediction


7
6
5
4
3
2
1
0
Cost
0.0
0.2
0.4
0.6
0.8
1.0
Prediction


7
6
5
4
3
2
1
0
x2
1
2
3
4
5
6
7
x1

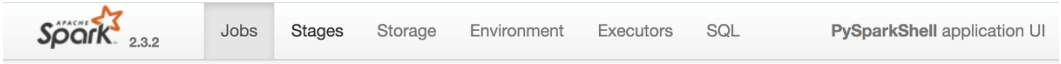

Online learning
Logistic regression model 1
Logistic regression model 2
Logistic regression model 3
Dataset 1
Dataset 2
Dataset 3
Time order
Offline learning
Logistic regression model 1
Logistic regression model 2
Logistic regression model 3
Dataset 1
Dataset 1
Dataset 2
Dataset 1
Dataset 2
Dataset 3
Time order

Chapter 6: Scaling Up Prediction to Terabyte Click Logs



Download Apache Spark™

1. Choose a Spark release: 2.4.5 (Feb 05 2020)
2. Choose a package type: Pre-built for Apache Hadoop 2.7
3. Download Spark: spark-2.4.5-bin-hadoop2.7.tgz
4. Verify this release using the 2.4.5 signatures, checksums and project release KEYS.



Spark Jobs (?)

User: hayden
Total Uptime: 2.0 h
Scheduling Mode: FIFO

▸ Event Timeline

Spark Jobs (?)

User: hayden
Total Uptime: 3.5 h
Scheduling Mode: FIFO
Completed Jobs: 1

▸ Event Timeline

Completed Jobs (1)

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2018/12/02 07:42:38	0.2 s	1/1	1/1

10	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:59:12	4 s	2/2	96/96
9	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:59:08	4 s	2/2	96/96
8	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:59:04	4 s	2/2	96/96
7	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:58:59	5 s	2/2	96/96
6	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:58:55	4 s	2/2	96/96
5	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:58:50	5 s	2/2	96/96
4	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:58:46	4 s	2/2	96/96
3	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:58:43	3 s	2/2	96/96
2	countByValue at StringIndexer.scala:140 countByValue at StringIndexer.scala:140	2018/12/03 18:58:28	15 s	2/2	96/96

33	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:49:50	20 s	2/2	54/54
32	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:49:28	21 s	2/2	54/54
31	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:49:07	20 s	2/2	54/54
30	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:48:48	19 s	2/2	54/54
29	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:48:24	23 s	2/2	54/54
28	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:48:01	23 s	2/2	54/54
27	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:47:38	23 s	2/2	54/54
26	treeAggregate at RDDLossFunction.scala:61 treeAggregate at RDDLossFunction.scala:61	2018/12/03 19:47:11	26 s	2/2	54/54
25	treeAggregate at LogisticRegression.scala:518 treeAggregate at LogisticRegression.scala:518	2018/12/03 19:28:25	19 min	2/2	54/54

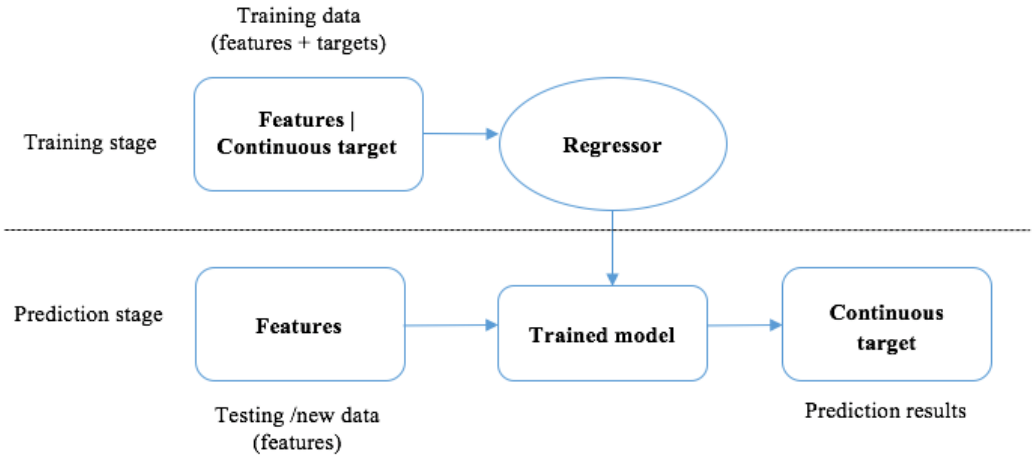
gender	site_domain	device_model
male	cnn	samsung
female	abc	iphone
male	nbc	huawei
male	facebook	xiaomi
female	abc	iphone

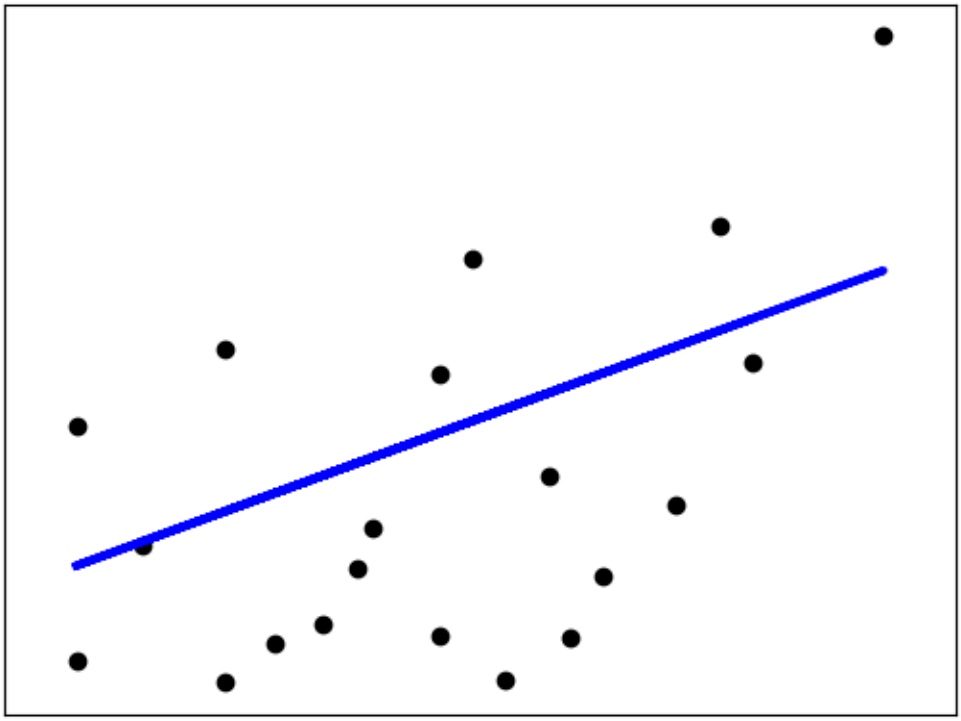
gender	site_domain	device_model	hash result
male	cnn	samsung	[1 0 0 0]
female	abc	iphone	[0 0 0 1]
male	nbc	huawei	[0 1 0 0]
male	facebook	xiaomi	[1 0 0 0]
female	abc	iphone	[0 0 0 1]

household income	household size	income/person
300,000	2	150,000
100,000	1	100,000
400,000	4	100,000
300,000	5	60,000
200,000	2	100,000

gender	site_domain	gender:site_domain
male	cnn	male:cnn
female	abc	female:abc
male	nbc	male:nbc
male	facebook	male:facebook
female	abc	female:abc

Chapter 7: Predicting Stock Prices with Regression Algorithms





Time Period: Apr 06, 2019 - Apr 06, 2020 Show: Historical Prices Frequency: Daily Apply

Currency in USD

Download Data

Date	Open	High	Low	Close*	Adj Close**	Volume
Apr 06, 2020	21,693.63	22,176.79	21,693.63	22,139.61	22,139.61	222,041,710
Apr 03, 2020	21,285.93	21,447.81	20,863.09	21,052.53	21,052.53	450,010,000
Apr 02, 2020	20,819.46	21,477.77	20,735.02	21,413.44	21,413.44	529,540,000
Apr 01, 2020	21,227.38	21,487.24	20,784.43	20,943.51	20,943.51	506,680,000
Mar 31, 2020	22,208.42	22,480.37	21,852.08	21,917.16	21,917.16	571,210,000
Mar 30, 2020	21,678.22	22,378.09	21,522.08	22,327.48	22,327.48	545,540,000
Mar 27, 2020	21,898.47	22,327.57	21,469.27	21,636.78	21,636.78	588,830,000
Mar 26, 2020	21,468.38	22,595.06	21,427.10	22,552.17	22,552.17	705,180,000
Mar 25, 2020	21,050.34	22,019.93	20,538.34	21,200.55	21,200.55	796,320,000
Mar 24, 2020	19,722.19	20,737.70	19,649.25	20,704.91	20,704.91	799,340,000
Mar 23, 2020	19,028.36	19,121.01	18,213.65	18,591.93	18,591.93	787,970,000
Mar 20, 2020	20,253.15	20,531.26	19,094.27	19,173.98	19,173.98	872,290,000

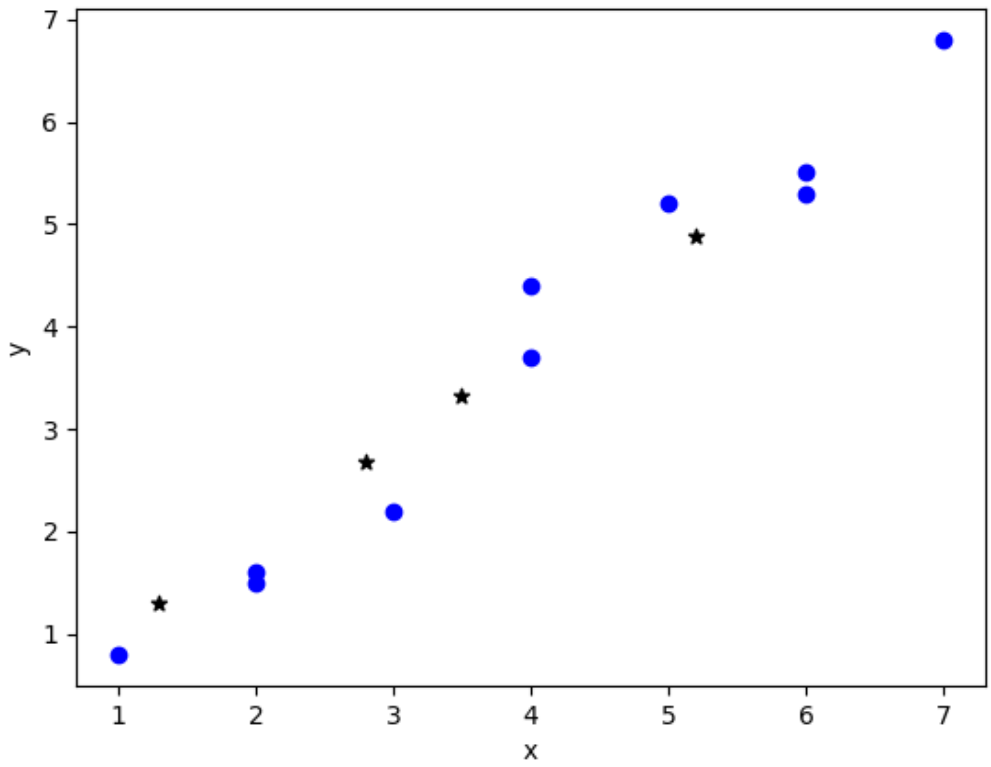
$Avg\text{Price}_5$	The average close price over the past five days
$Avg\text{Price}_{30}$	The average close price over the past month
$Avg\text{Price}_{365}$	The average close price over the past year
$\frac{Avg\text{Price}_5}{Avg\text{Price}_{30}}$	The ratio between the average price over the past week and that over the past month
$\frac{Avg\text{Price}_5}{Avg\text{Price}_{365}}$	The ratio between the average price over the past week and that over the past year
$\frac{Avg\text{Price}_{30}}{Avg\text{Price}_{365}}$	The ratio between the average price over the past month and that over the past year
$Avg\text{Volume}_5$	The average volume over the past five days
$Avg\text{Volume}_{30}$	The average volume over the past month
$Avg\text{Volume}_{365}$	The average volume over the past year
$\frac{Avg\text{Volume}_5}{Avg\text{Volume}_{30}}$	The ratio between the average volume over the past week and that over the past month
$\frac{Avg\text{Volume}_5}{Avg\text{Volume}_{365}}$	The ratio between the average volume over the past week and that over the past year
$\frac{Avg\text{Volume}_{30}}{Avg\text{Volume}_{365}}$	The ratio between the average volume over the past month and that over the past year
StdPrice_5	The standard deviation of the close prices over the past five days
StdPrice_{30}	The standard deviation of the close prices over the past month
StdPrice_{365}	The standard deviation of the close prices over the past year

$\frac{\text{StdPrice}_5}{\text{StdPrice}_{30}}$	The ratio between the standard deviation of the prices over the past week and that over the past month
$\frac{\text{StdPrice}_5}{\text{StdPrice}_{365}}$	The ratio between the standard deviation of the prices over the past week and that over the past year
$\frac{\text{StdPrice}_{30}}{\text{StdPrice}_{365}}$	The ratio between the standard deviation of the prices over the past month and that over the past year
StdVolume_5	The standard deviation of the volumes over the past five days
StdVolume_{30}	The standard deviation of the volumes over the past month
StdVolume_{365}	The standard deviation of the volumes over the past year
$\frac{\text{StdVolume}_5}{\text{StdVolume}_{30}}$	The ratio between the standard deviation of the volumes over the past week and that over the past month
$\frac{\text{StdVolume}_5}{\text{StdVolume}_{365}}$	The ratio between the standard deviation of the volumes over the past week and that over the past year
$\frac{\text{StdVolume}_{30}}{\text{StdVolume}_{365}}$	The ratio between the standard deviation of the volumes over the past month and that over the past year
$\text{return}_{i:i-1}$	Daily return of the past day
$\text{return}_{i:i-5}$	Weekly return of the past week
$\text{return}_{i:i-30}$	Monthly return of the past month
$\text{return}_{i:i-365}$	Yearly return of the past year
MovingAvg_{i_5}	Moving average of the daily returns over the past week
MovingAvg_{i_30}	Moving average of the daily returns over the past month
MovingAvg_{i_365}	Moving average of the daily returns over the past year

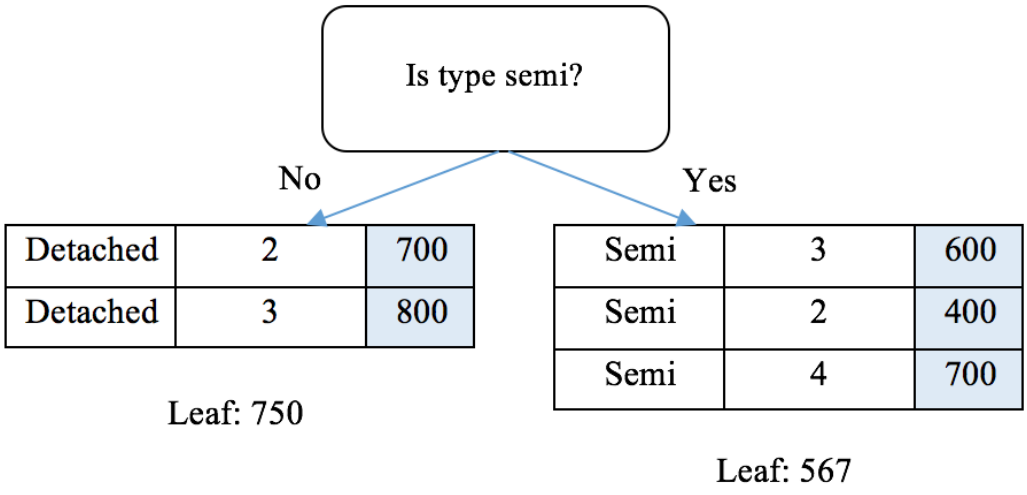
```
              open   open_1  close_1   high_1    low_1    volume_1  avg_price_5   ...     return_5
return_30  return_365  moving_avg_5  moving_avg_30  moving_avg_365    close
Date                                                                             ...

1989-01-04  2153.75  2163.21  2144.64  2168.39  2127.14  17310000.0     2165.000   ...       -0.011
    0.020       0.056         0.001          0.001           0.000  2177.68
1989-01-05  2184.29  2153.75  2177.68  2183.39  2146.61  15710000.0     2168.000   ...        0.007
    0.041       0.069        -0.002          0.001           0.000  2190.54
1989-01-06  2195.89  2184.29  2190.54  2205.18  2173.04  20310000.0     2172.822   ...        0.011
    0.031       0.068         0.001          0.002           0.000  2194.29
1989-01-09  2194.82  2195.89  2194.29  2213.75  2182.32  16500000.0     2175.144   ...        0.005
    0.021       0.148         0.002          0.001           0.000  2199.46
1989-01-10  2205.36  2194.82  2199.46  2209.11  2185.00  18420000.0     2181.322   ...        0.014
    0.021       0.131         0.001          0.001           0.001  2193.21

[5 rows x 38 columns]
```



Type	Number of bedrooms	Price (thousand)
Semi	3	600
Detached	2	700
Detached	3	800
Semi	2	400
Semi	4	700

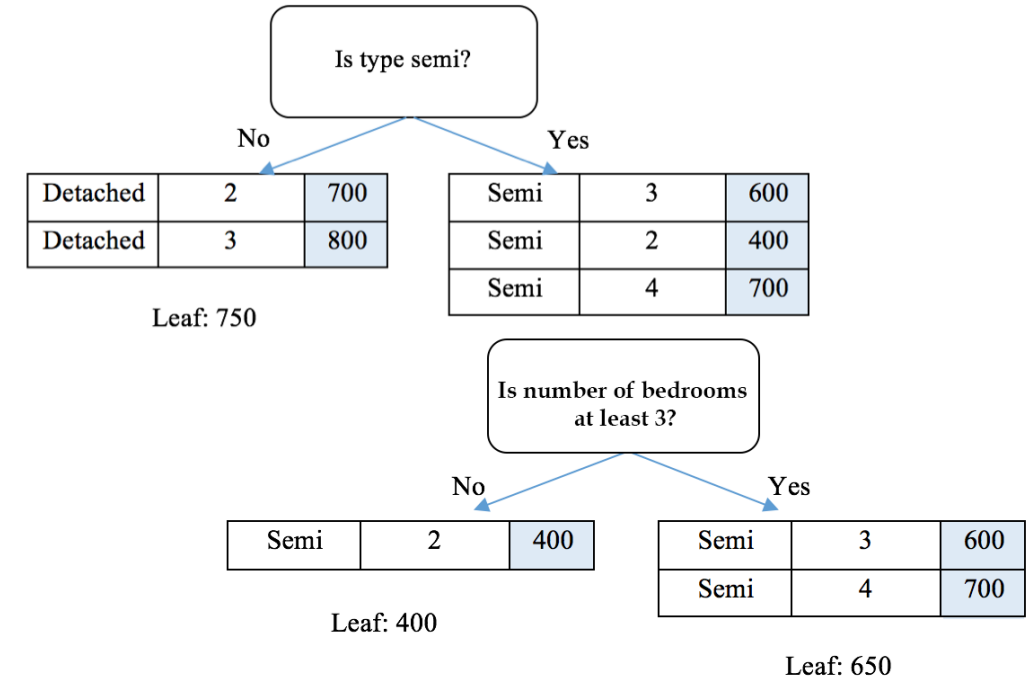


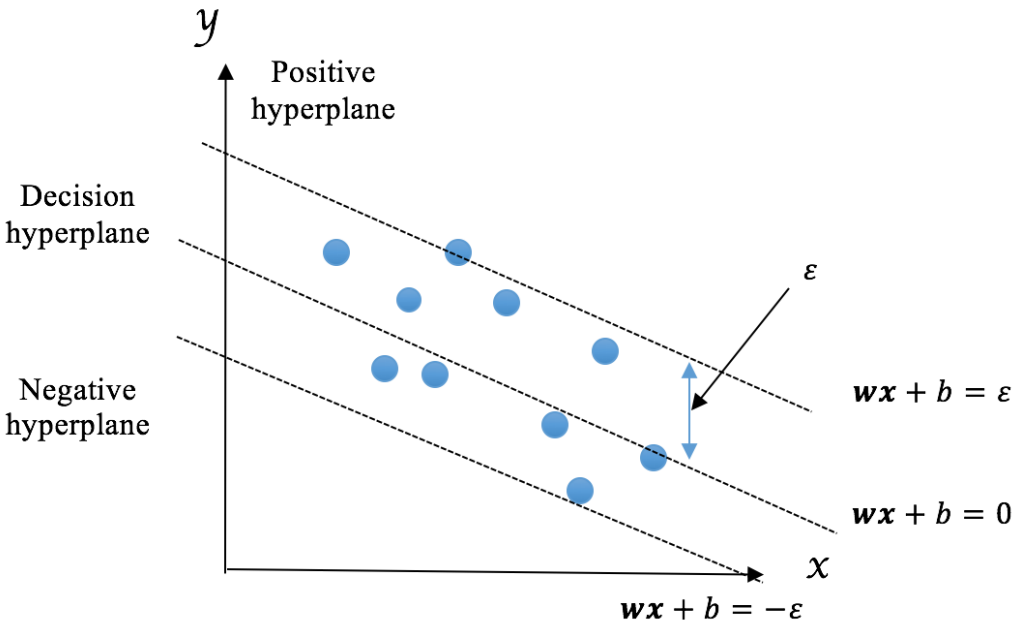
Detached	2	700
Detached	3	800

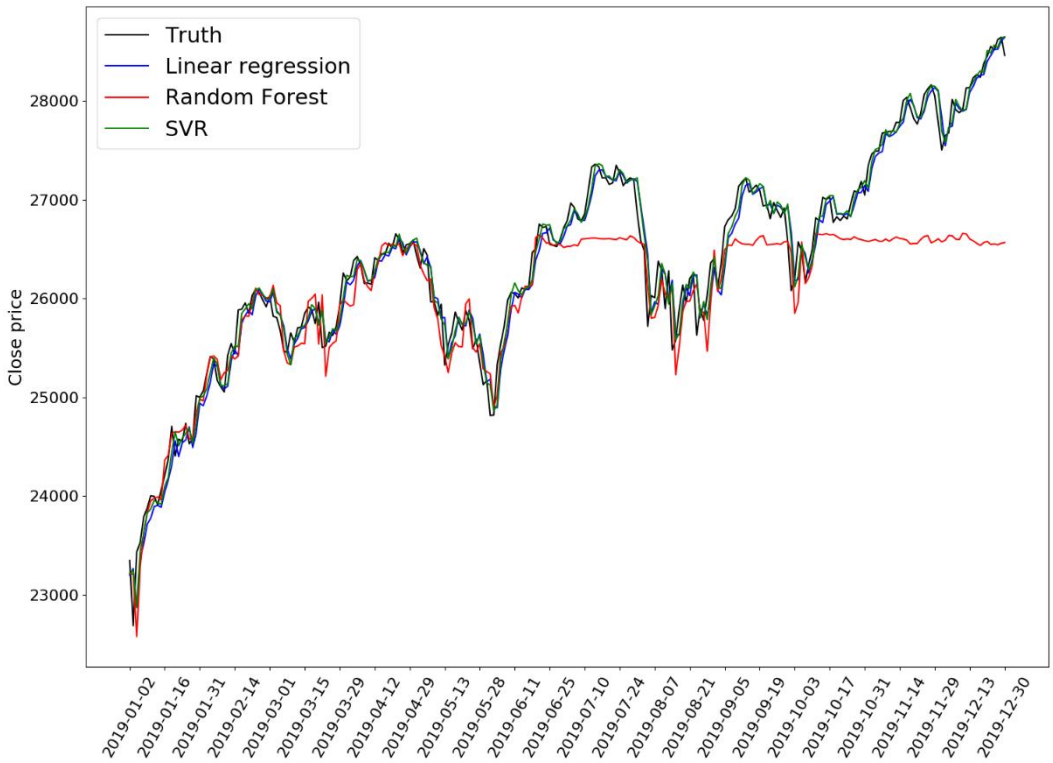
Leaf: 750

Semi	3	600
Semi	2	400
Semi	4	700

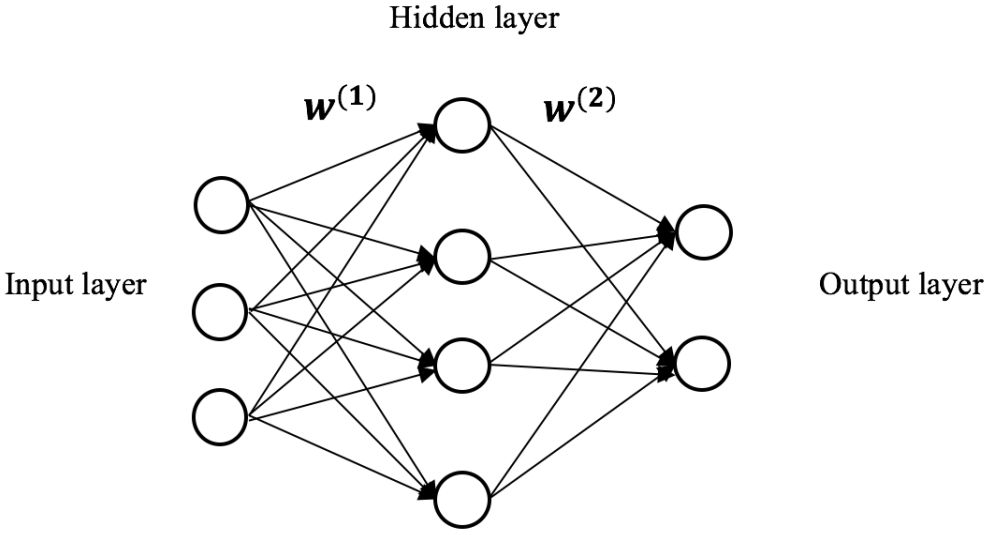
Leaf: 567

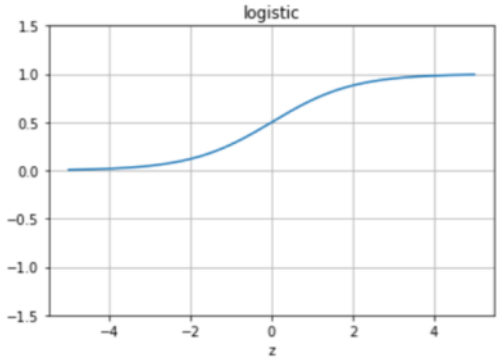

Is type semi?
No
Yes
Detached 2 700
Detached 3 800
Leaf: 750
Semi 3 600
Semi 2 400
Semi 4 700
Is number of bedrooms at least 3?
No
Yes
Semi 2 400
Leaf: 400
Semi 3 600
Semi 4 700
Leaf: 650

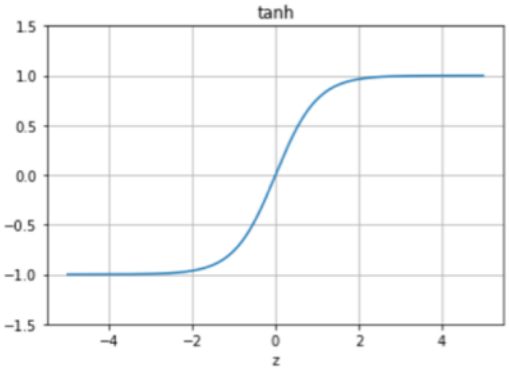

y
Positive hyperplane
Decision hyperplane
Negative hyperplane
ε
wx + b = ε
wx + b = 0
x
wx + b = −ε

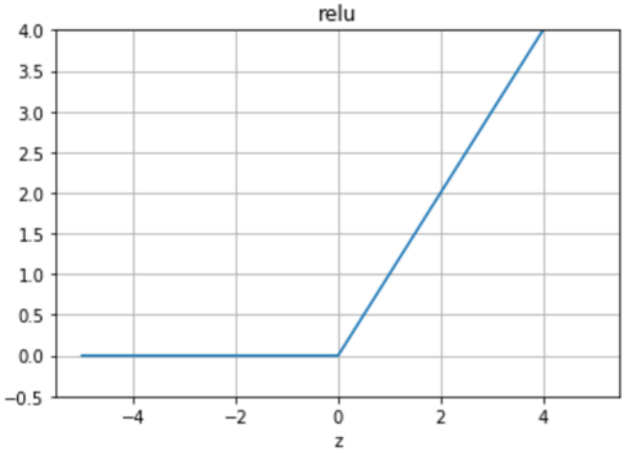

Truth
Linear regression
Random Forest
SVR
Close price
28000
27000
26000
25000
24000
23000
2019-01-02
2019-01-16
2019-01-31
2019-02-14
2019-03-01
2019-03-15
2019-03-29
2019-04-12
2019-04-29
2019-05-13
2019-05-28
2019-06-11
2019-06-25
2019-07-10
2019-07-24
2019-08-07
2019-08-21
2019-09-05
2019-09-19
2019-10-03
2019-10-17
2019-10-31
2019-11-14
2019-11-29
2019-12-13
2019-12-30

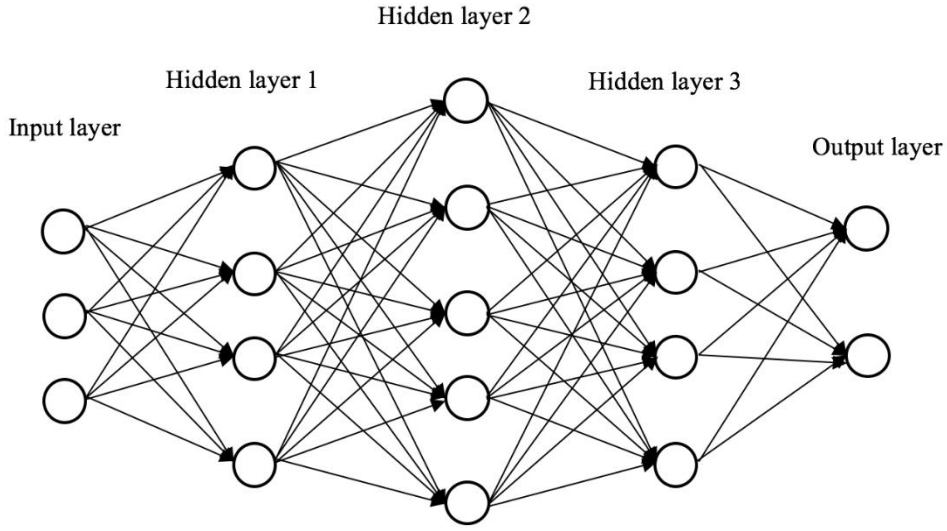
Chapter 8: Predicting Stock Prices with Artificial Neural Networks



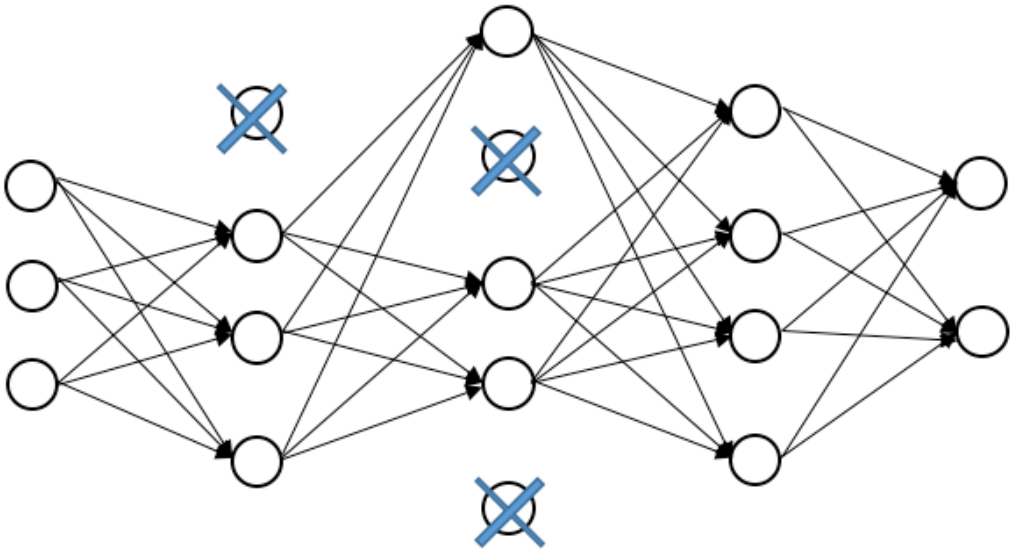



tanh
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−4
−2
0
2
4
z

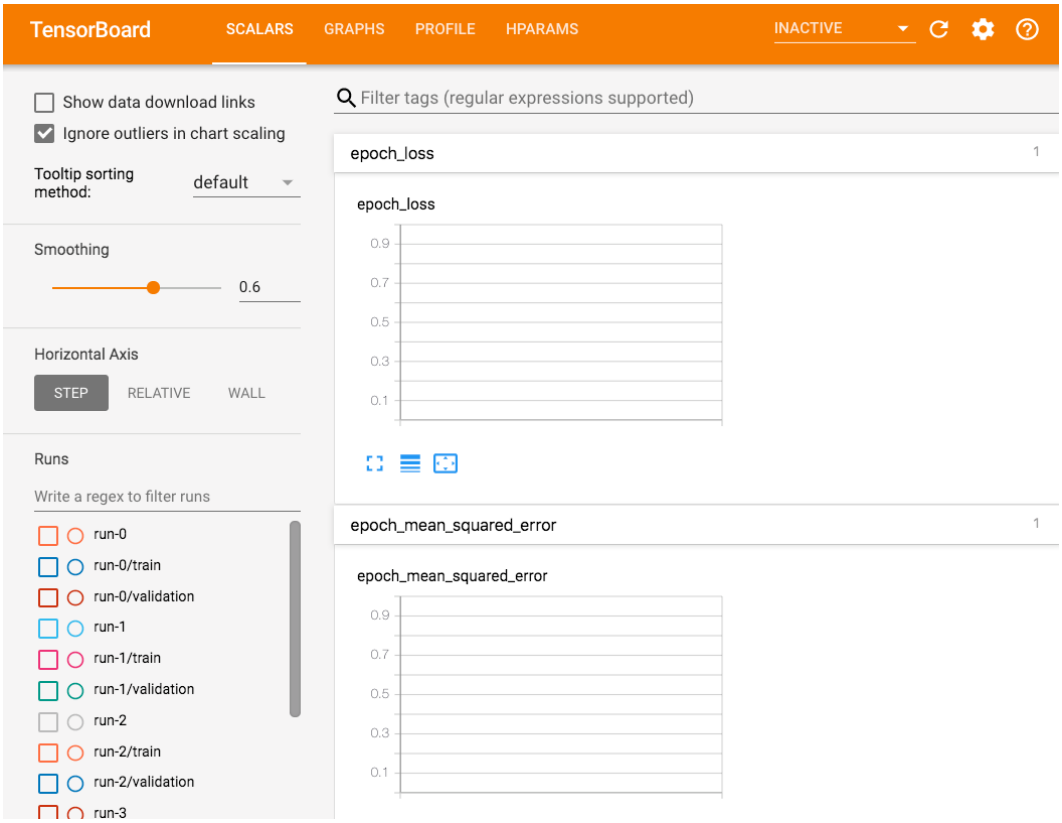

relu
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−4
−2
0
2
4
z

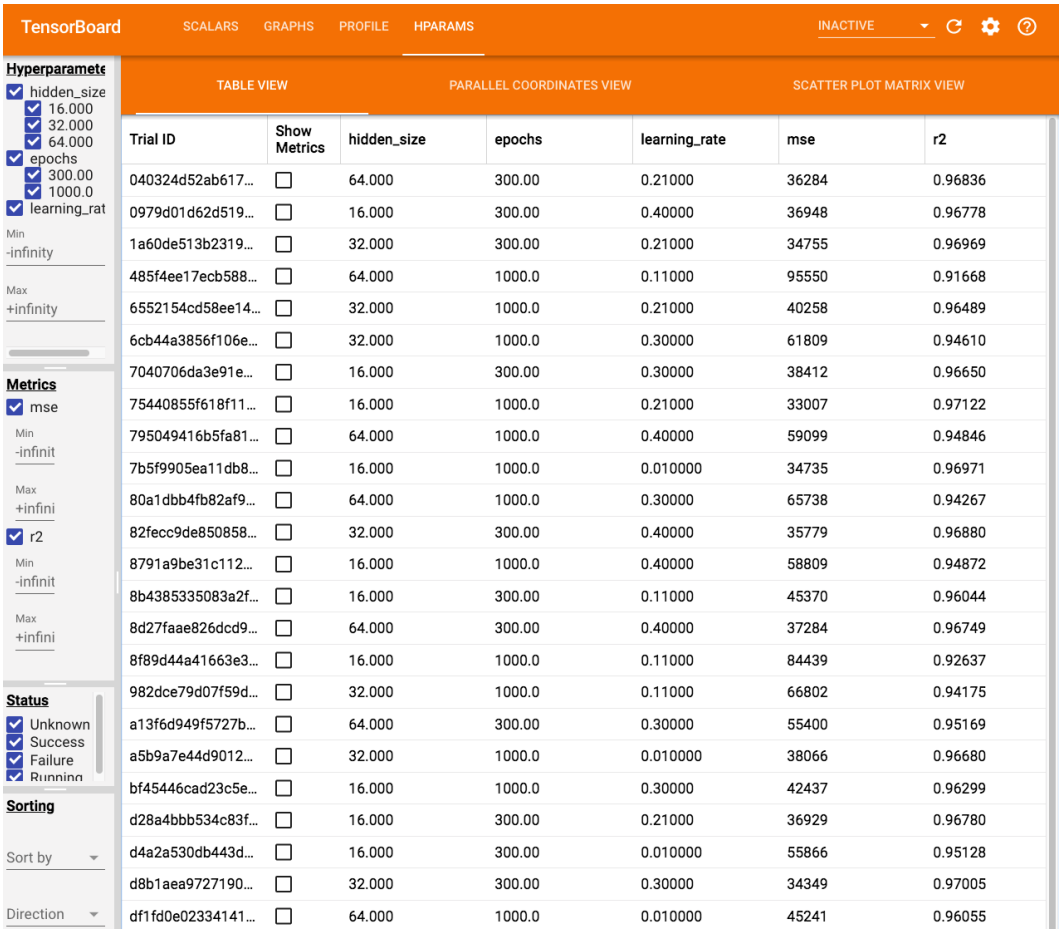


A Deep Learning Model

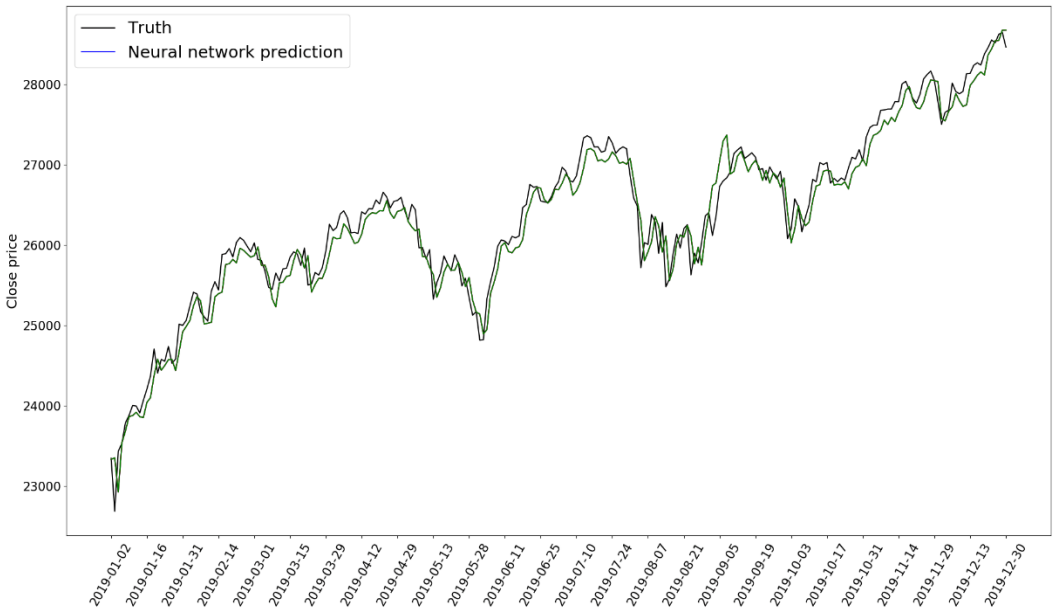


A neural network with dropout

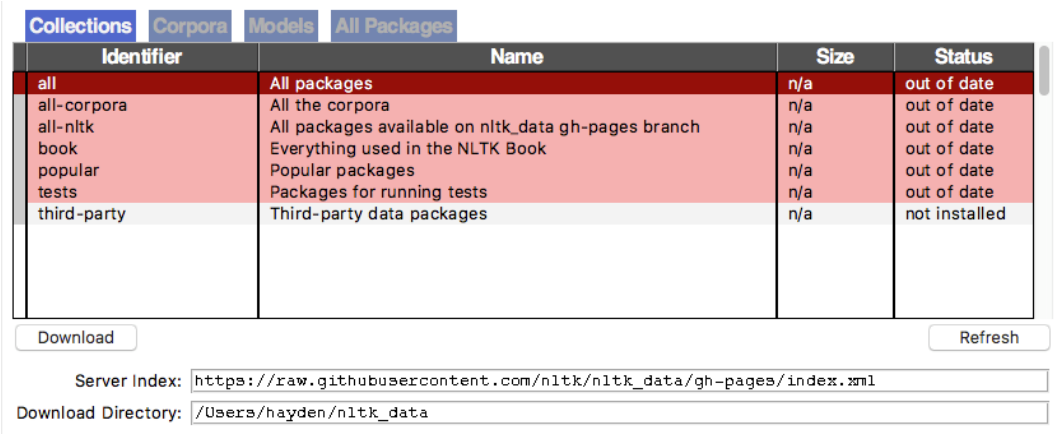

TensorBoard
SCALARS
GRAPHS
PROFILE
HPARAMS
INACTIVE
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method:
default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
run-0
run-0/train
run-0/validation
run-1
run-1/train
run-1/validation
run-2
run-2/train
run-2/validation
run-3
Filter tags (regular expressions supported)
epoch_loss
1
epoch_loss
0.9
0.7
0.5
0.3
0.1
epoch_mean_squared_error
1
epoch_mean_squared_error
0.9
0.7
0.5
0.3
0.1

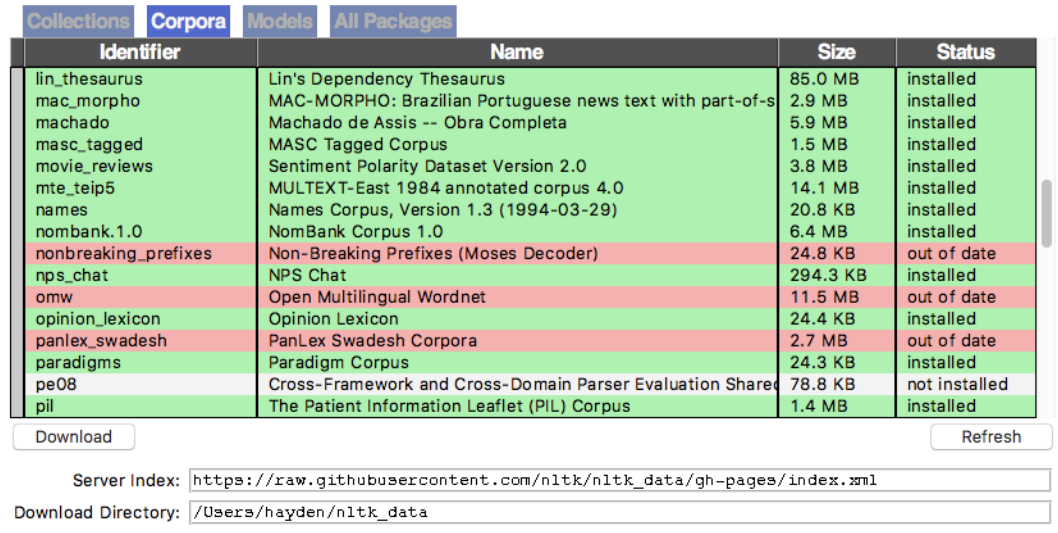


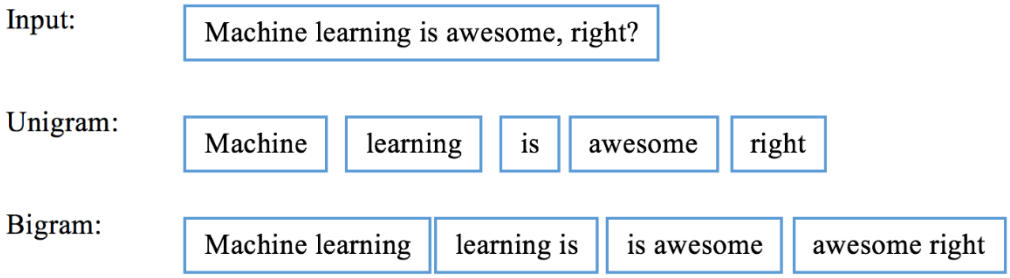
Trial ID	Show Metrics	hidden_size	epochs	learning_rate	mse	r2
040324d52ab617…	☐	64.000	300.00	0.21000	36284	0.96836
0979d01d62d519…	☐	16.000	300.00	0.40000	36948	0.96778
1a60de513b2319…	☐	32.000	300.00	0.21000	34755	0.96969
485f4ee17ecb588…	☐	64.000	1000.0	0.11000	95550	0.91668
6552154cd58ee14…	☐	32.000	1000.0	0.21000	40258	0.96489
6cb44a3856f106e…	☐	32.000	1000.0	0.30000	61809	0.94610
7040706da3e91e…	☐	16.000	300.00	0.30000	38412	0.96650
75440855f618f11…	☐	16.000	1000.0	0.21000	33007	0.97122
795049416b5fa81…	☐	64.000	1000.0	0.40000	59099	0.94846
7b5f9905ea11db8…	☐	16.000	1000.0	0.010000	34735	0.96971
80a1dbb4fb82af9…	☐	64.000	1000.0	0.30000	65738	0.94267
82fecc9de850858…	☐	32.000	300.00	0.40000	35779	0.96880
8791a9be31c112…	☐	16.000	1000.0	0.40000	58809	0.94872
8b4385335083a2f…	☐	16.000	300.00	0.11000	45370	0.96044
8d27faae826dcd9…	☐	64.000	300.00	0.40000	37284	0.96749
8f89d44a41663e3…	☐	16.000	1000.0	0.11000	84439	0.92637
982dce79d07f59d…	☐	32.000	1000.0	0.11000	66802	0.94175
a13f6d949f5727b…	☐	64.000	300.00	0.30000	55400	0.95169
a5b9a7e44d9012…	☐	32.000	1000.0	0.010000	38066	0.96680
bf45446cad23c5e…	☐	16.000	1000.0	0.30000	42437	0.96299
d28a4bbb534c83f…	☐	16.000	300.00	0.21000	36929	0.96780
d4a2a530db443d…	☐	16.000	300.00	0.010000	55866	0.95128
d8b1aea9727190…	☐	32.000	300.00	0.30000	34349	0.97005
df1fd0e02334141…	☐	64.000	1000.0	0.010000	45241	0.96055


Truth
Neural network prediction
Close price
28000
27000
26000
25000
24000
23000
2019-01-02
2019-01-16
2019-01-31
2019-02-14
2019-03-01
2019-03-15
2019-03-29
2019-04-12
2019-04-29
2019-05-13
2019-05-28
2019-06-11
2019-06-25
2019-07-10
2019-07-24
2019-08-07
2019-08-21
2019-09-05
2019-09-19
2019-10-03
2019-10-17
2019-10-31
2019-11-14
2019-11-29
2019-12-13
2019-12-30

Chapter 9: Mining the 20 Newsgroups Dataset with Text Analysis Techniques

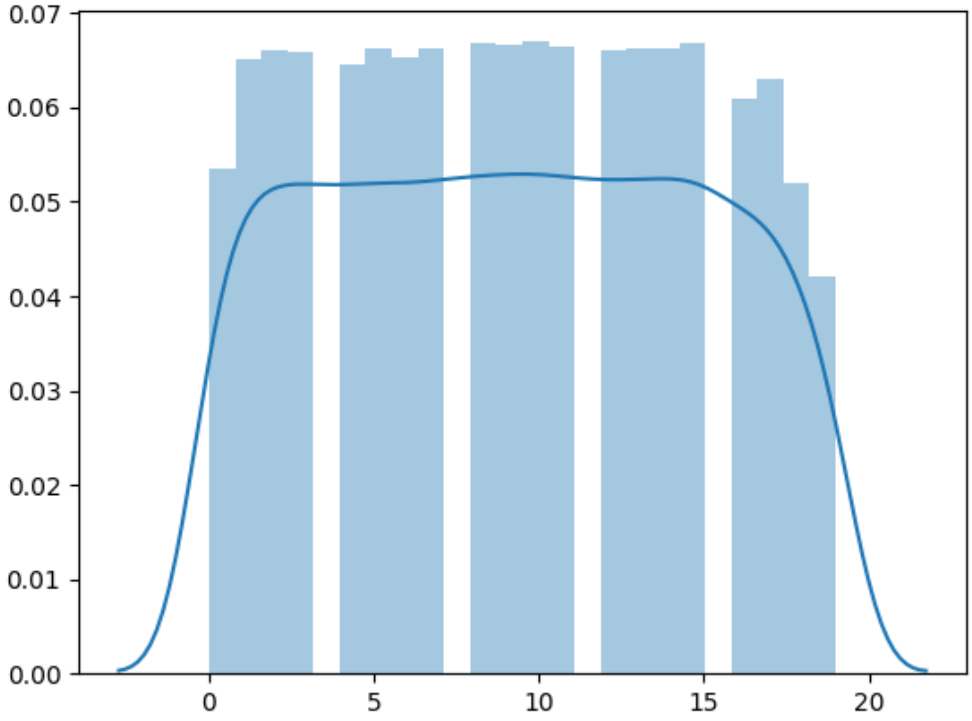





Input:
Machine learning is awesome, right?
Unigram:
Machine
learning
is
awesome
right
Bigram:
Machine learning
learning is
is awesome
awesome right

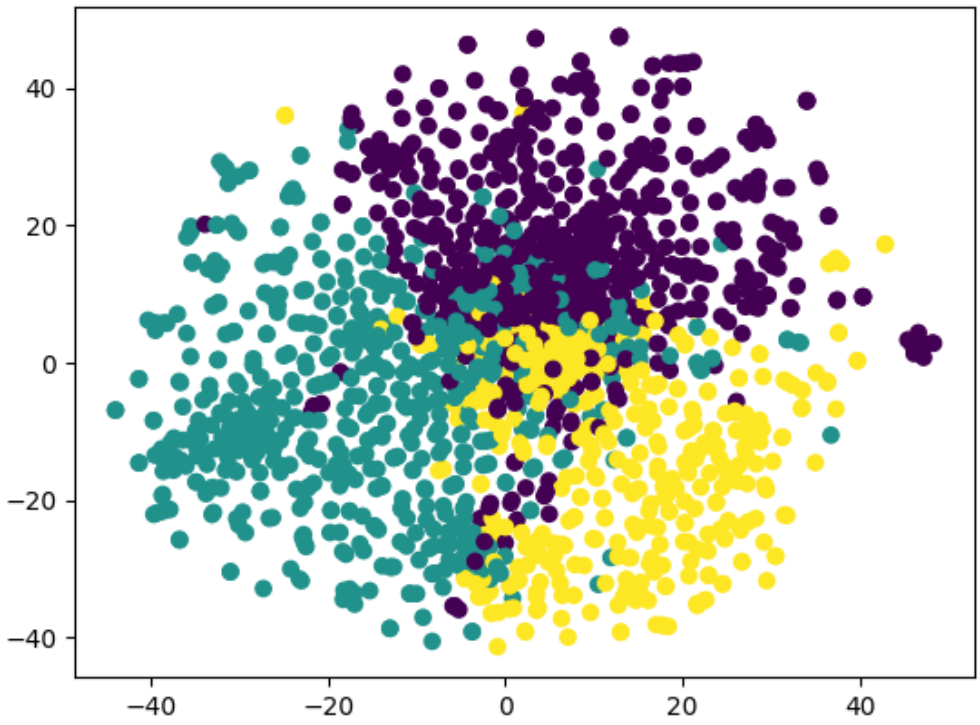
Parameter	Default value	Example values	Description
subset	'train'	'train', 'test', 'all'	The dataset to load: the training set, the testing set or both.
data_home	~/scikit_learn_data	~/myfolder	Directory where the files are stored and cached
categories	None	['sci.space", alt.atheism']	List of newsgroups to load. If None, all newsgroups will be loaded.
shuffle	True	True, False	Boolean indicating whether to shuffle the data
random_state	42	7, 43	Random seed integer used to shuffle the data
remove	()	('headers', 'footers', 'quotes')	Tuple indicating part(s) among header, footer and quote of each newsgroup post to omit. Nothing is removed by default.
download_if_missing	True	True, False	Boolean indicating whether to download the data if it is not found locally

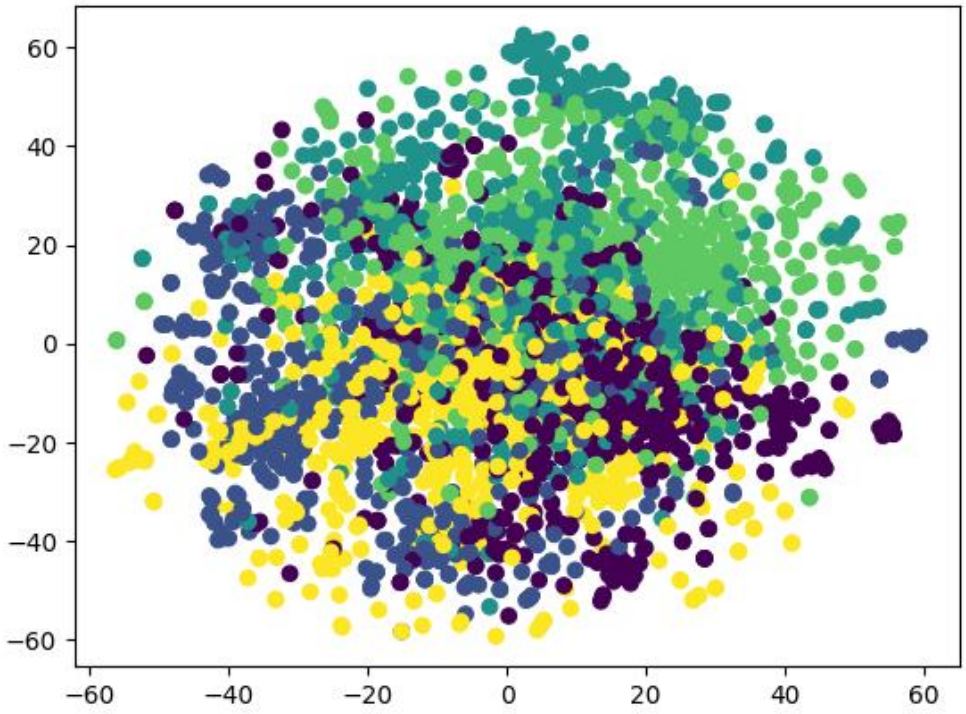
Parameter	Default value	Example values	Description
subset	'train'	'train', 'test', 'all'	The dataset to load: the training set, the testing set or both.
data_home	~/scikit_learn_data	~/myfolder	Directory where the files are stored and cached
categories	None	['sci.space", alt.atheism']	List of newsgroups to load. If None, all newsgroups will be loaded.
shuffle	True	True, False	Boolean indicating whether to shuffle the data
random_state	42	7, 43	Random seed integer used to shuffle the data
remove	()	('headers', 'footers', 'quotes')	Tuple indicating the part(s) among header, footer, and quote of each newsgroup post to omit. Nothing is removed by default.
download_if_missing	True	True, False	Boolean indicating whether to download the data if it is not found locally


0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
5
10
15
20

Constructor parameter	**Default value**	**Example values**	**Description**
ngram_range	(1,1)	(1, 2), (2, 2)	Lower and upper bound of the n-grams to be extracted in the input text, for example (1, 1) means unigram, (1, 2) means unigram and bigram
stop_words	None	'english', or list [‘a’, ‘the’, ‘of’] or None	Which stop word list to use, can be “english” referring to the built-in list, or a customized input list. If None, no word will be removed.
lowercase	True	True, False	Whether or not converting all characters to lowercase
max_features	None	None, 200, 500	The number of top (most frequent) tokens to consider, or all tokens if None
binary	False	True, False	If True, all non-zero counts become 1s.

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0], dtype=int64)
```

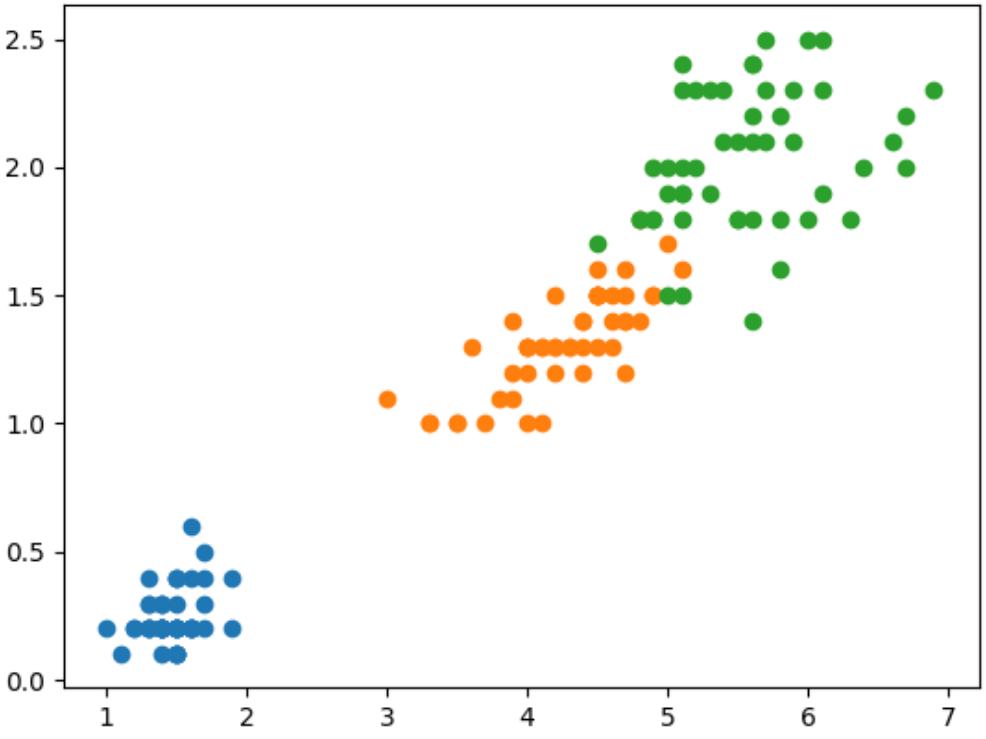

40
20
0
−20
−40
−40
−20
0
20
40

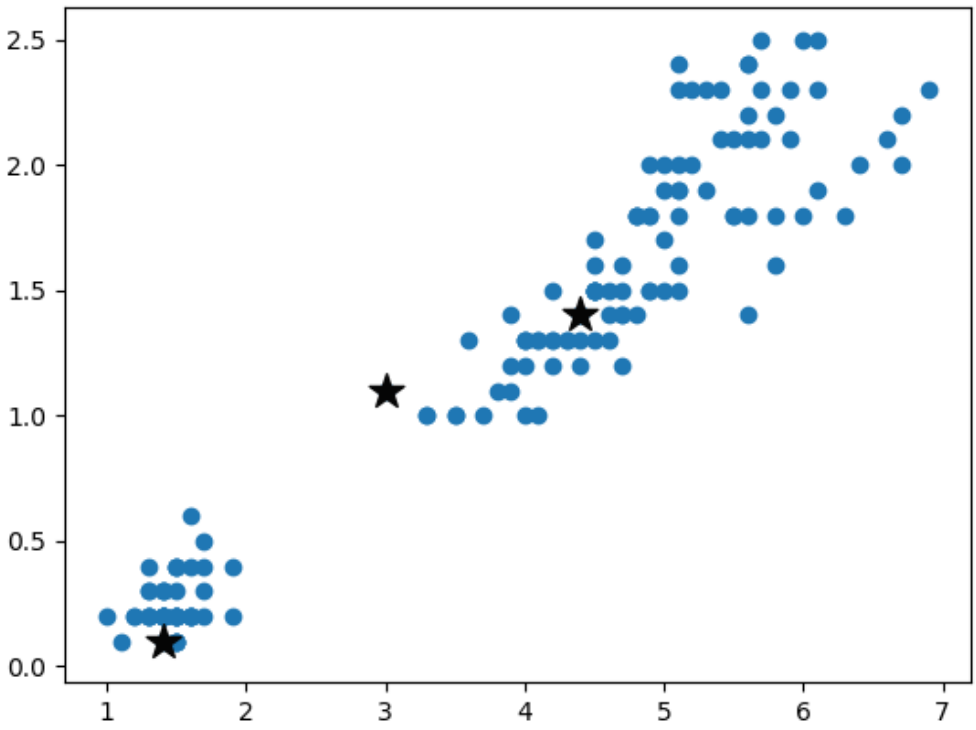

60
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60

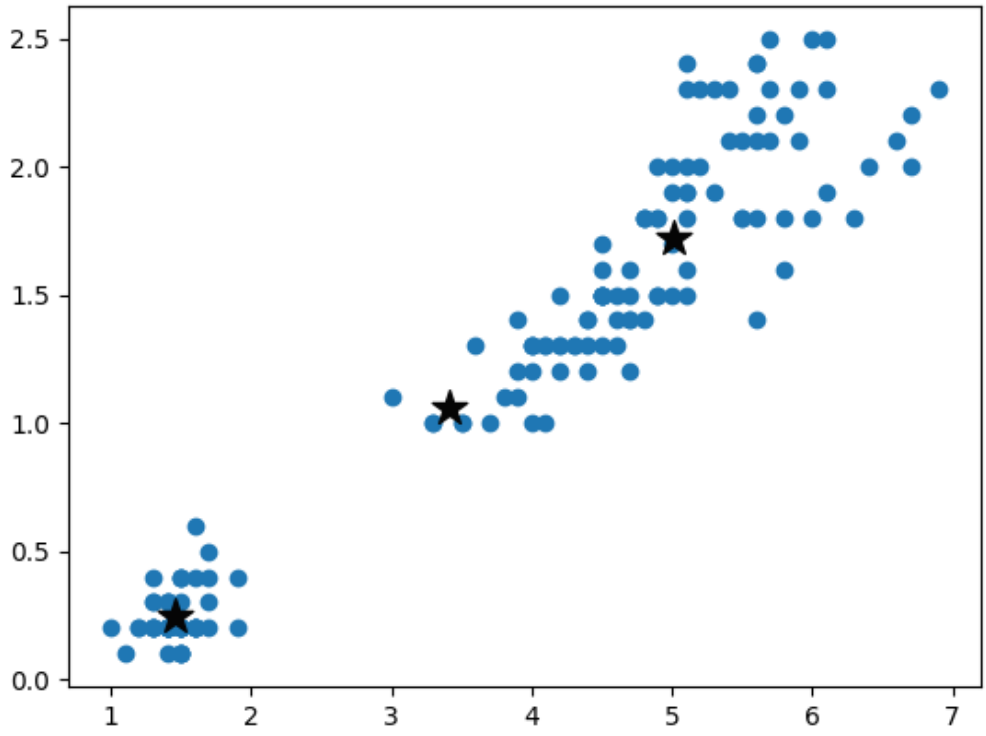
Chapter 10: Discovering Underlying Topics in the Newsgroups Dataset with Clustering and Topic Modeling

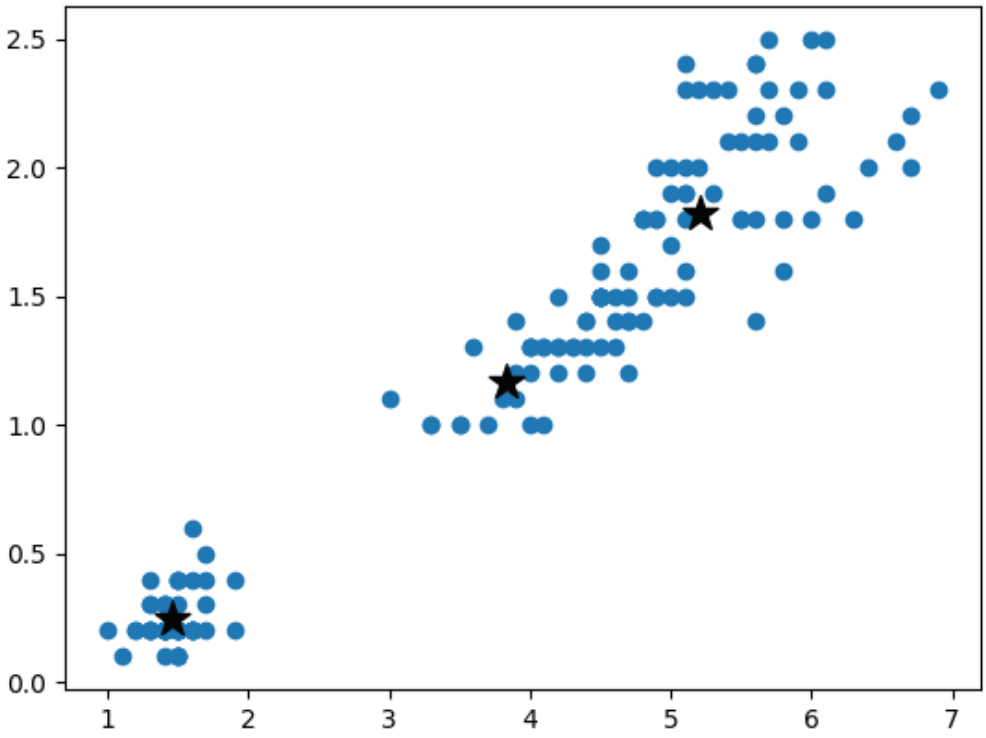
Given two 2-dimension data points (x_1, y_1) and (x_2, y_2)

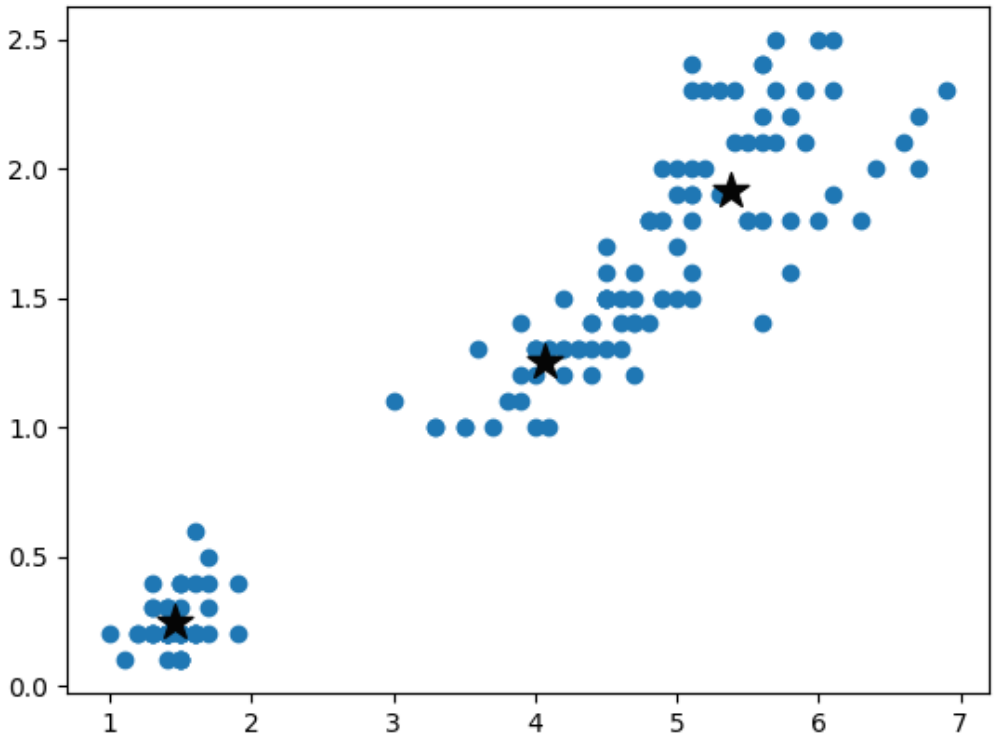
Distance metric	Calculation
Euclidean distance	$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$
Manhattan distance	$\lvert x_1 - x_2 \rvert + \lvert y_1 - y_2 \rvert$
Chebyshev distance	$\max(\lvert x_1 - x_2 \rvert, \lvert y_1 - y_2 \rvert)$

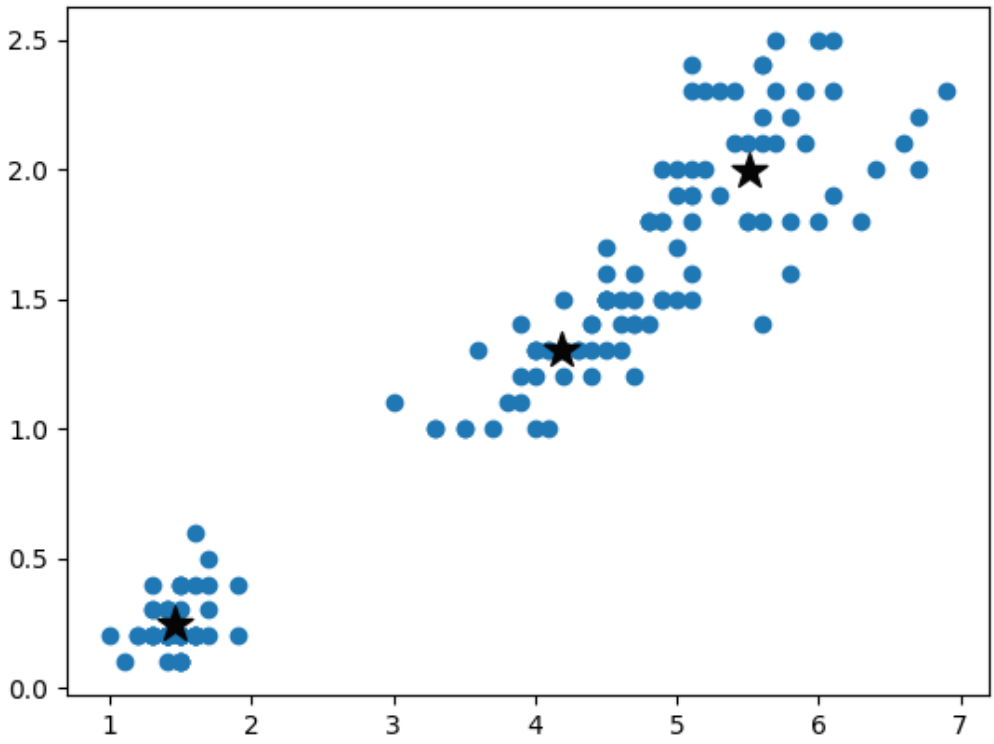


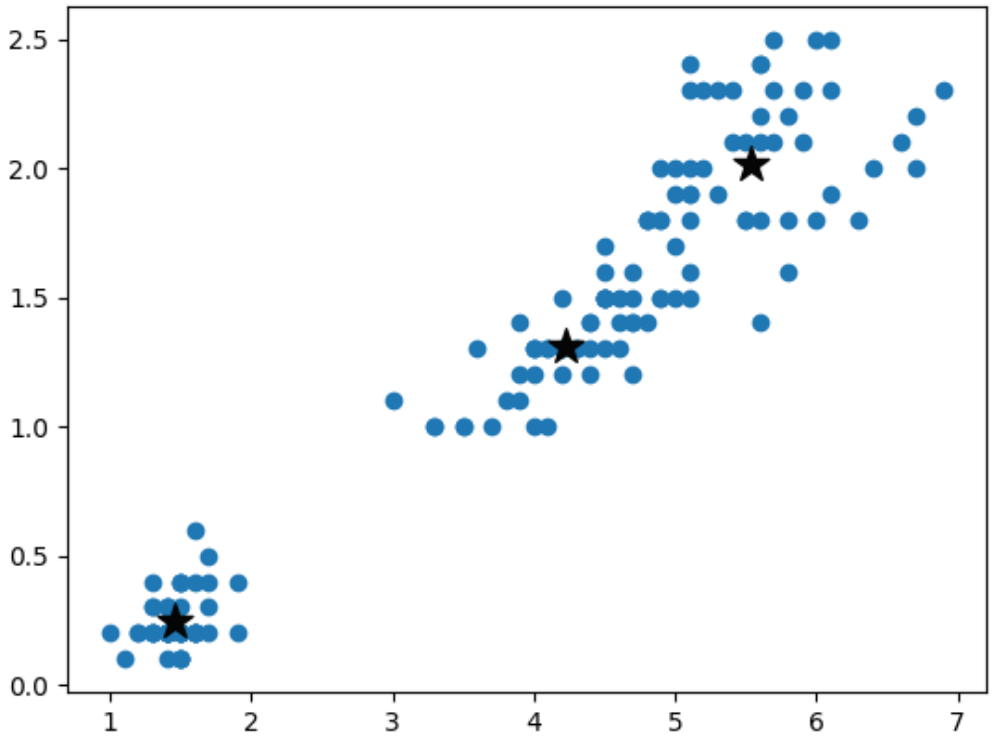

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

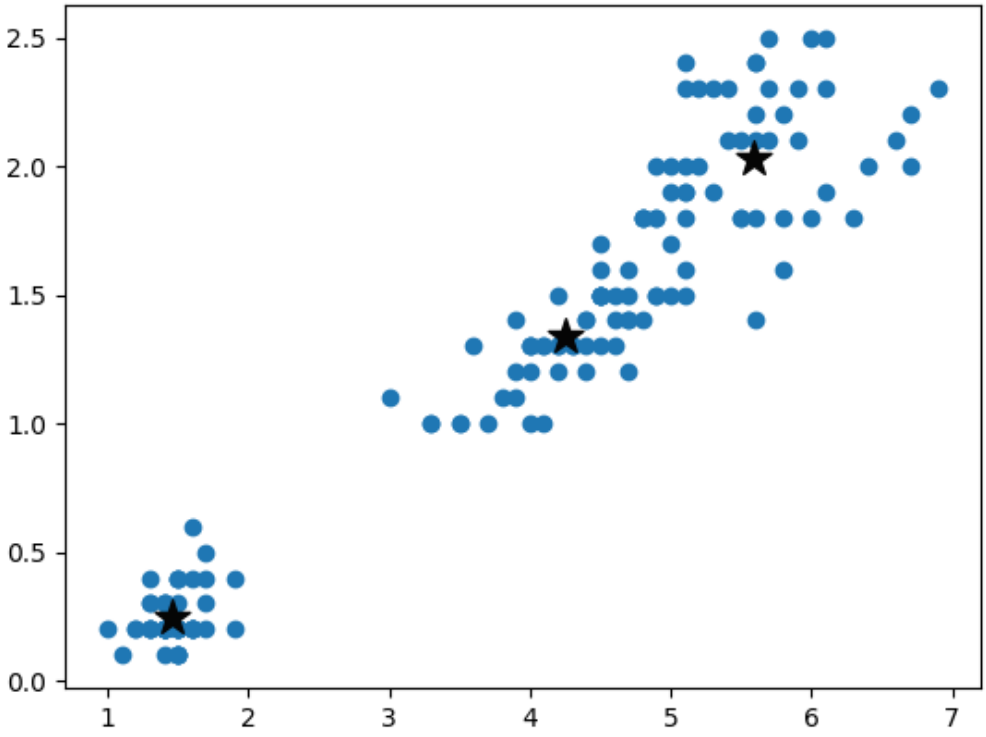

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

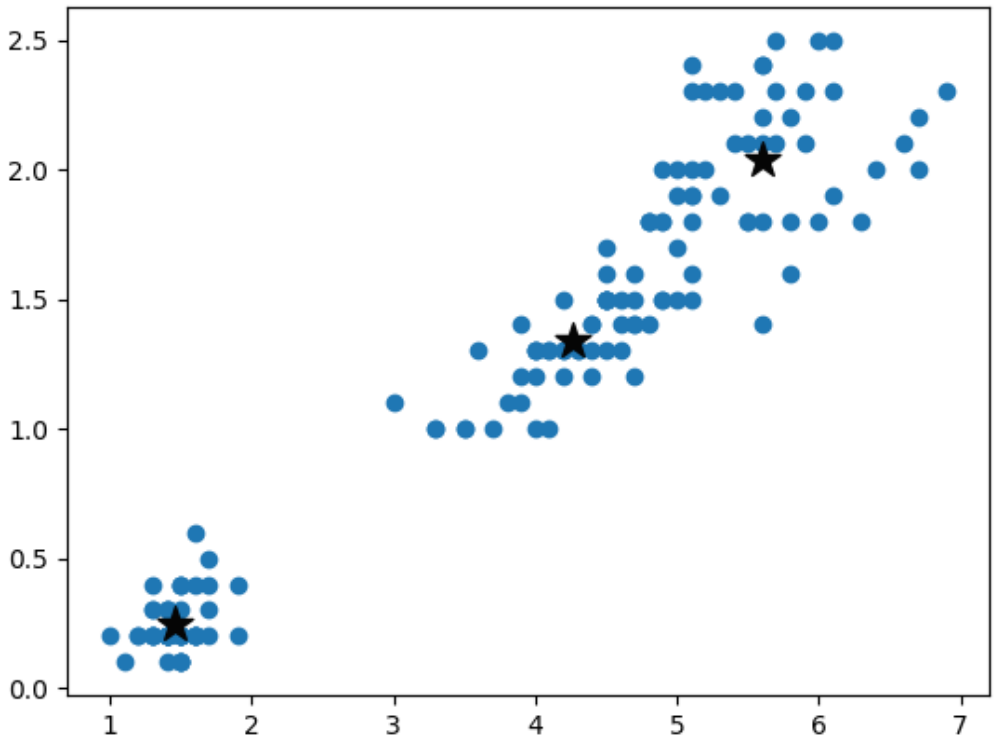

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

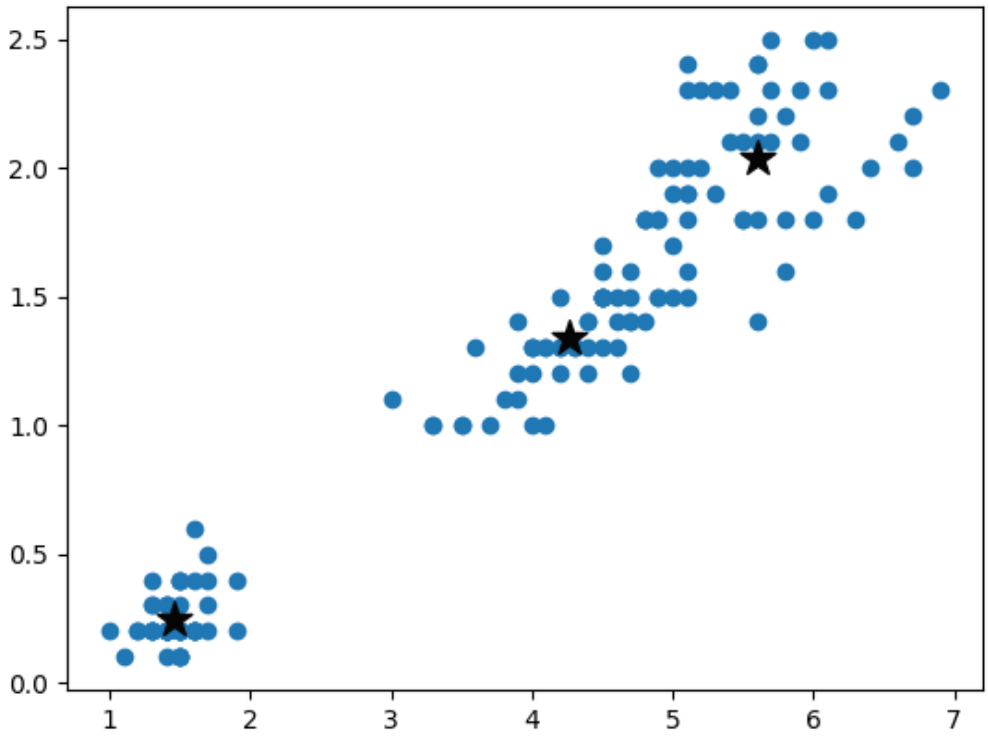

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

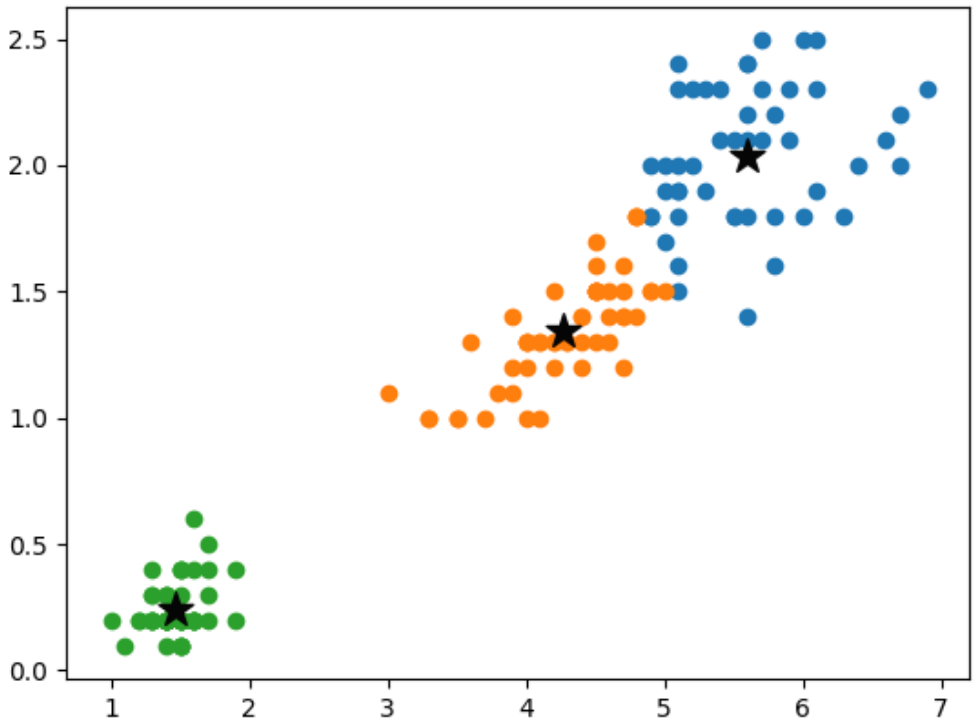

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7


2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

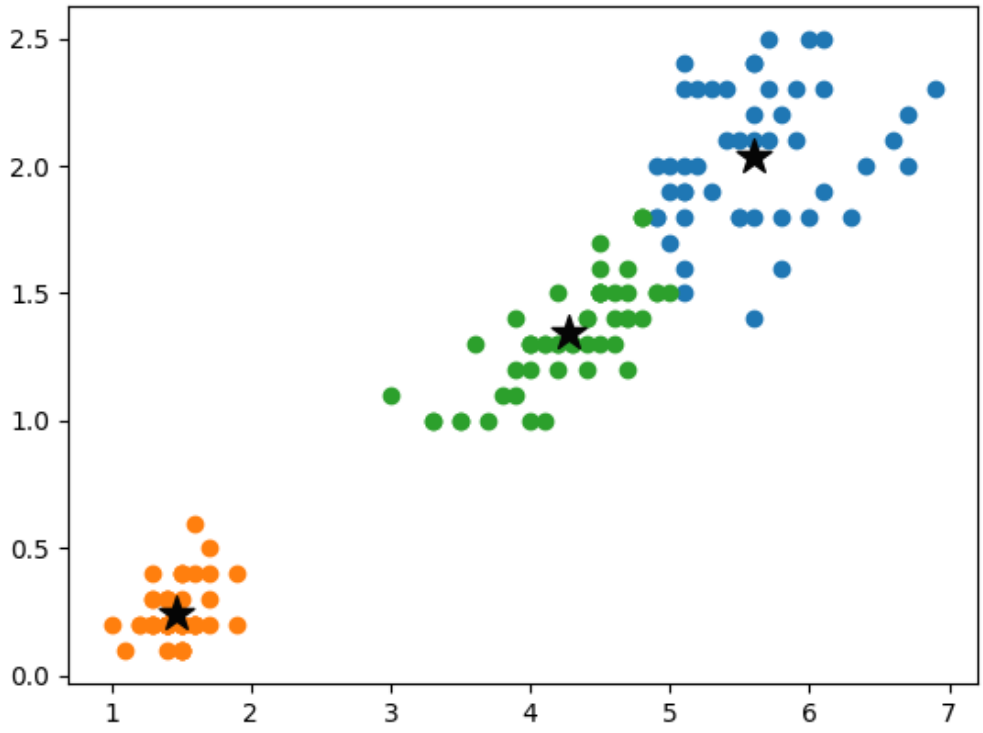

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7

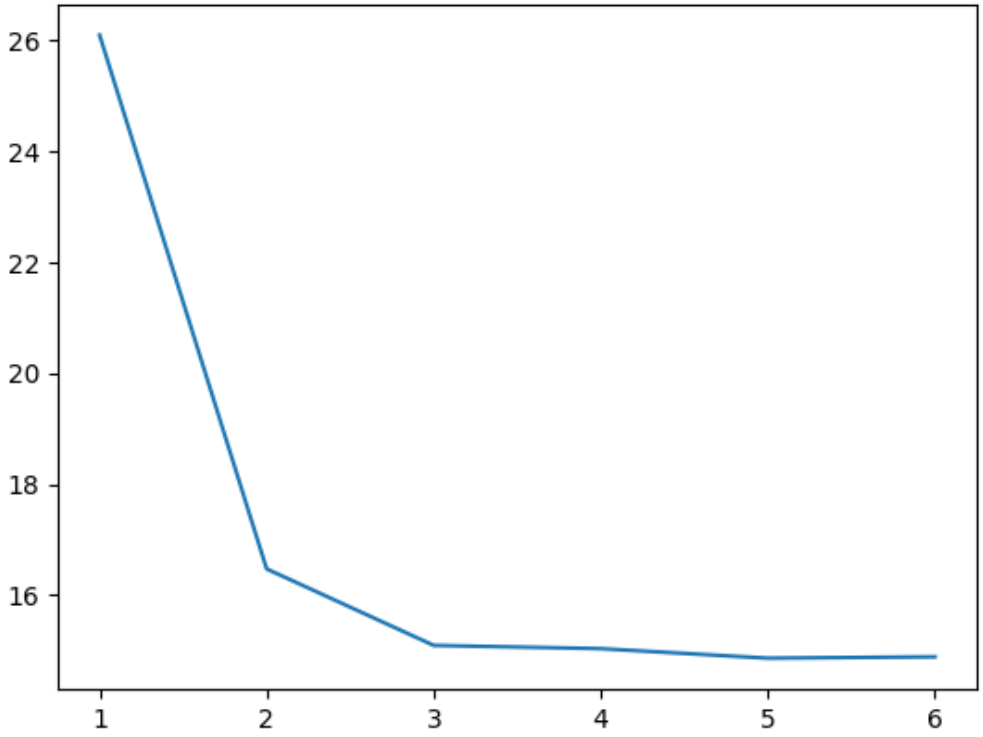

2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7


2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7



Constructor parameter	**Default value**	**Example values**	**Description**
n_clusters	8	3, 5, 10	K clusters
max_iter	300	10, 100, 500	Maximum number of iterations
tol	1e-4	1e-5, 1e-8	Tolerance to declare convergence
random_state	None	0, 42	Random seed for program reproducibility


2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
6
7



Input matrix V

	Term1	Term2	Term3	Term4	Term5	Term6
Document1	4	2	0	0	3	1
Document2	0	1	1	0	2	0
Document3	1	0	1	4	0	2
Document4	2	0	0	0	0	1

Feature matrix W

	Term1	Term2	Term3	Term4	Term5	Term6
Topic1	0.2	0	0.5	0	0	0
Topic2	0	1	0	0	0.5	0
Topic3	0	0	1	0	0	0.5

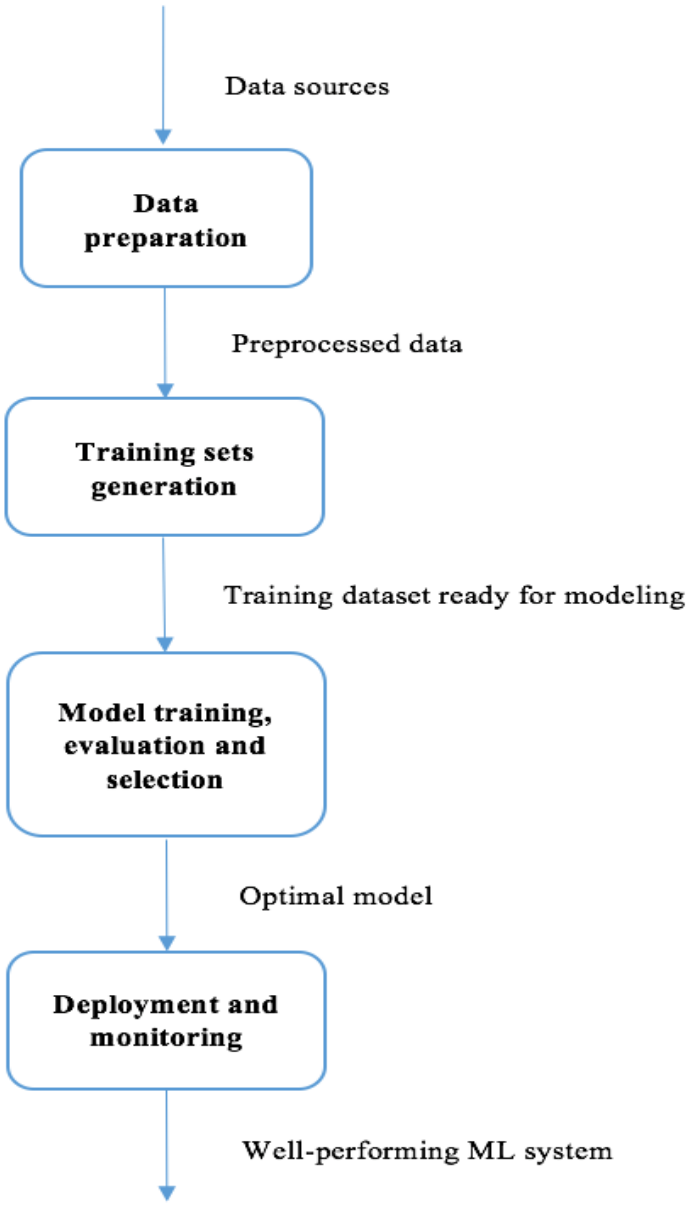
Coefficient matrix H

	Topic1	Topic2	Topic3
Document1	1	0	0
Document2	0	0.5	0.5
Document3	0.2	0	0.8
Document4	0	1	0

Constructor parameter	Default value	Example values	Description
n_components	None	5, 10, 20	Number of components – in the context of topic modeling, this corresponds to the number of topics. If None, it becomes the number of input features.
max_iter	200	100, 200	Maximum number of iterations
tol	1e-4	1e-5, 1e-8	Tolerance to declare convergence

Constructor parameter	Default value	Example values	Description
n_components	10	5, 10, 20	Number of components – in the context of topic modeling, this corresponds to the number of topics.
learning_method	“batch”	“online”, “batch”	In batch mode, all training data are used for each update. In online mode, mini-batch of training data is used for each update. In general, if the data size is large, the online mode is faster.
max_iter	10	10, 20	Maximum number of iterations
randome_state	None	0, 42	Seed used by the random number generator.

Chapter 11: Machine Learning Best Practices

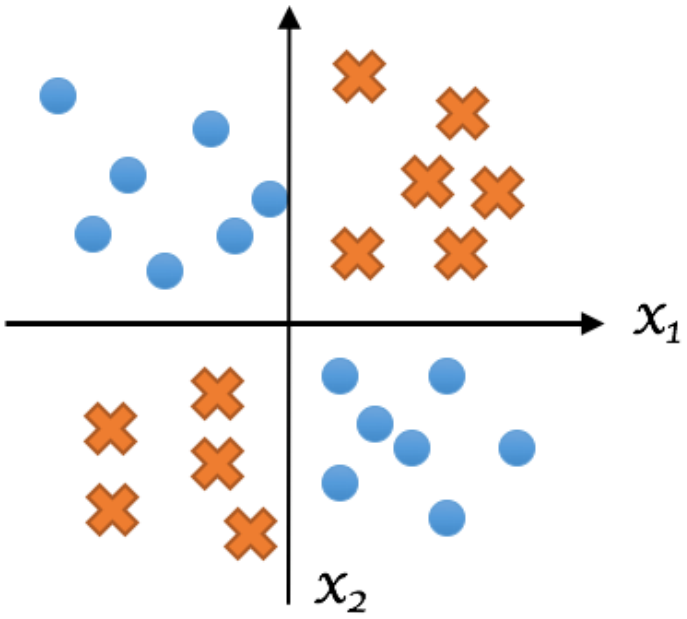


Input of neural network	**Output of neural network**
(I, love, python, machine)	(reading)
(love, reading, machine, learning)	(python)
(reading, python, learning, by)	(machine)
(python, machine, by, example)	(learning)

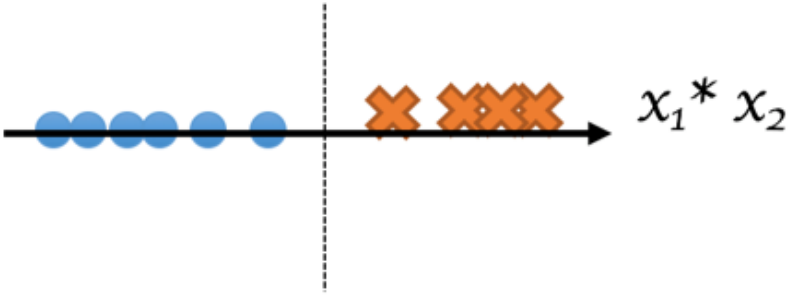
Input of neural network	Output of neural network
(reading)	(i)
(reading)	(love)
(reading)	(python)
(reading)	(machine)
(python)	(love)
(python)	(reading)
(python)	(machine)
(python)	(learning)
(machine)	(reading)
(machine)	(python)
(machine)	(learning)
(machine)	(by)
(learning)	(python)
(learning)	(machine)
(learning)	(by)
(learning)	(example)

Name	**fasttext-wiki-news-subwords-300**	
Corpus	Wikipedia 2017	
Vector size	300	
Vocabulary size	1 million	
File size	958 MB	
More information	https://fasttext.cc/docs/en/english-vectors.html	
Name	**glove-twitter-100**	**glove-twitter-25**
Corpus	Twitter (2 billion tweets)	
Vector size	100	25
Vocabulary size	1.2 million	
File size	387 MB	104 MB
More information	https://nlp.stanford.edu/projects/glove/	
Name	**word2vec-google-news-300**	
Corpus	Google News (about 100 billion words)	
Vector size	300	
Vocabulary size	3 million	
File size	1662 MB	
More information	https://code.google.com/archive/p/word2vec/	

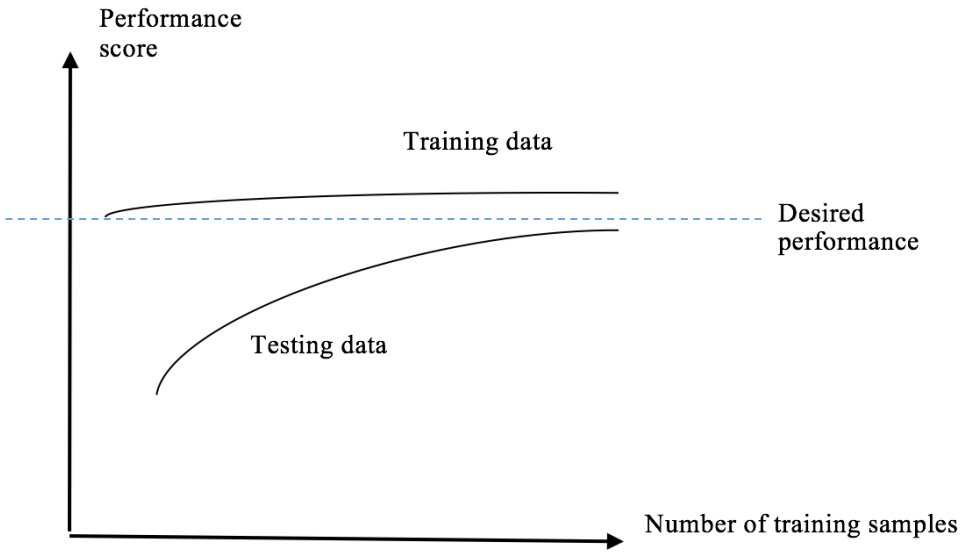
Linearly non-separable



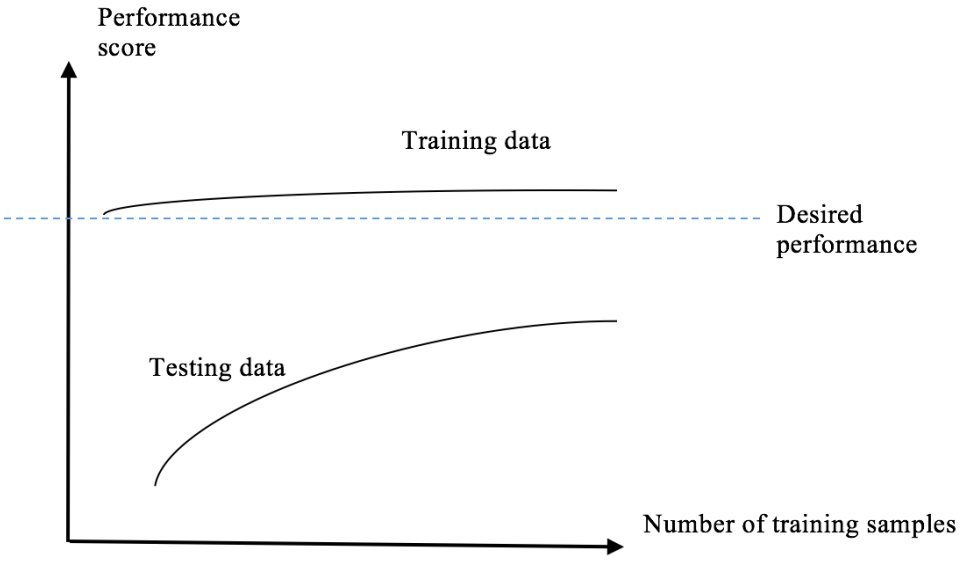
Linearly separable



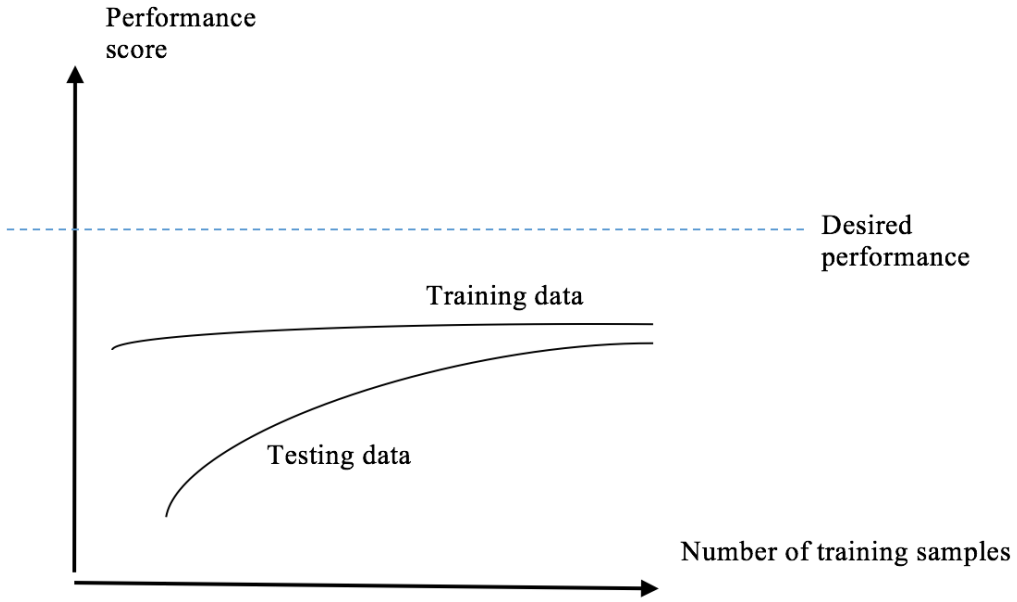
Ideal



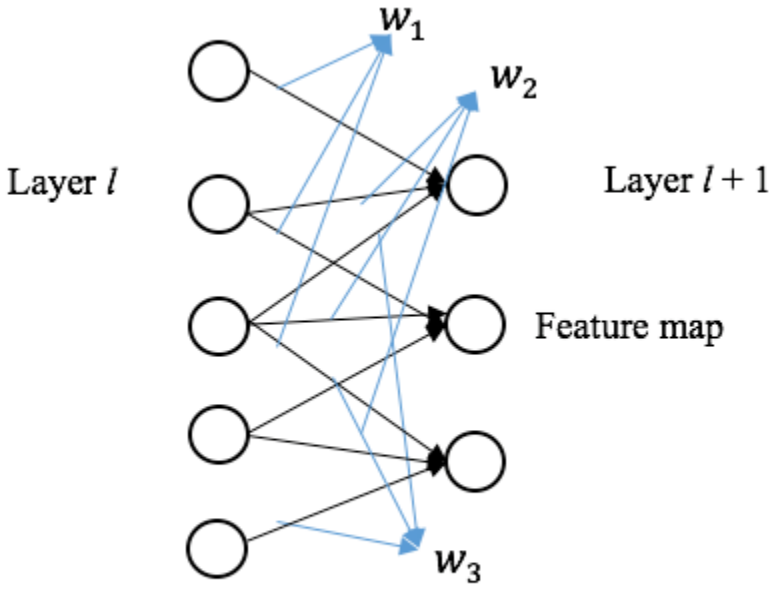
Overfitting



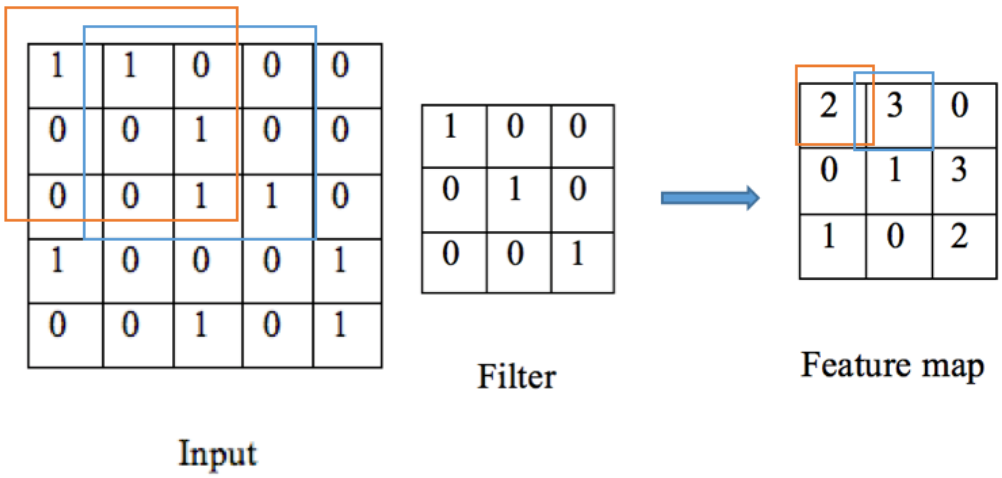
Underfitting

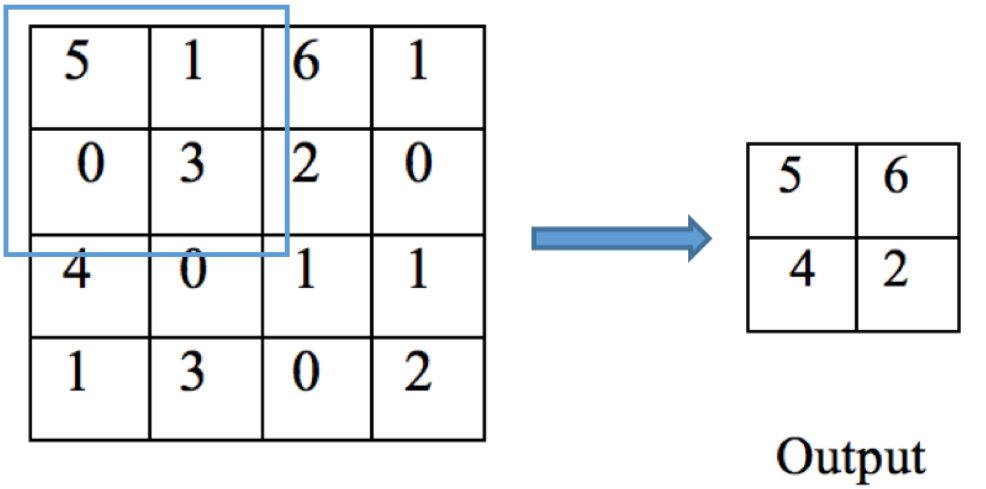


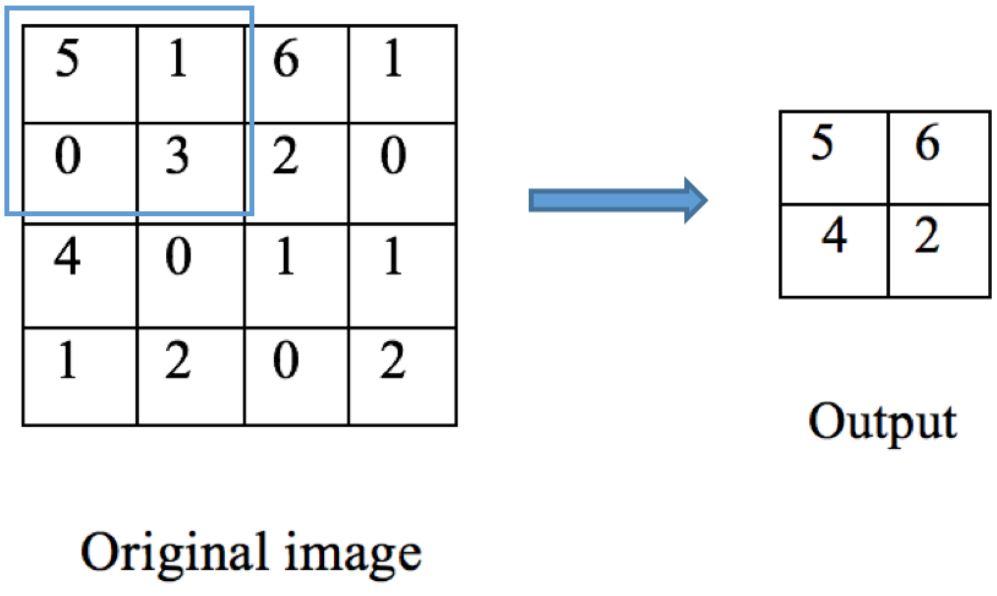
Chapter 12: Categorizing Images of Clothing with Convolutional Neural Networks



Filter $\boldsymbol{w} = [w_1, w_2, w_3]$.




5 1 6 1
0 3 2 0
4 0 1 1
1 3 0 2
5 6
4 2
Output
Input


5 1 6 1
0 3 2 0
4 0 1 1
1 2 0 2
5 6
4 2
Output
Original image

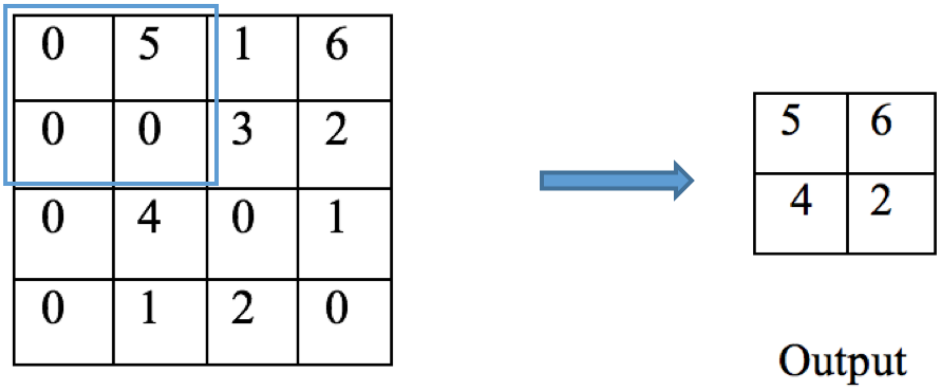
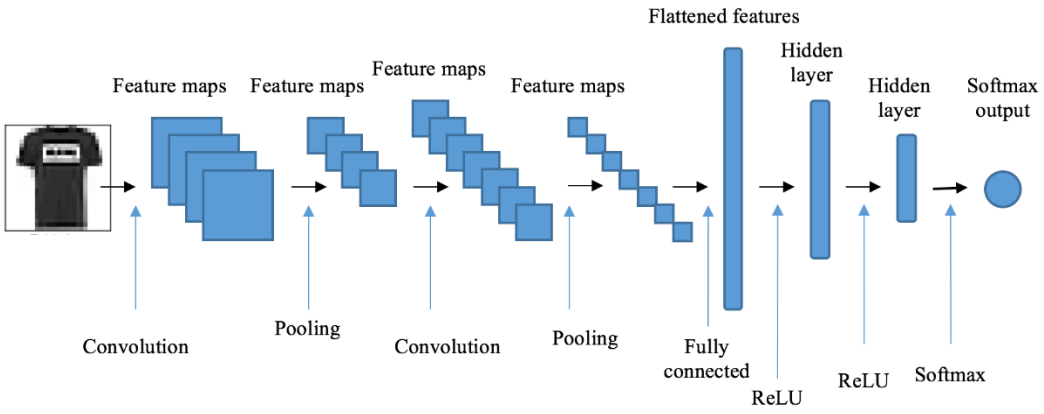
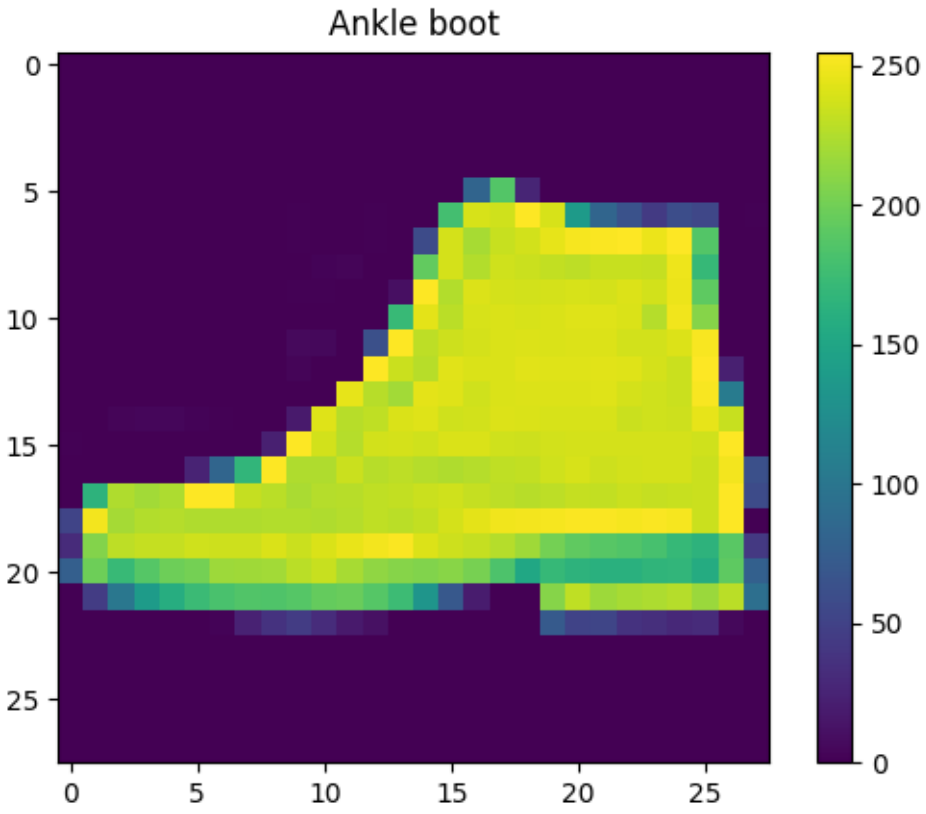
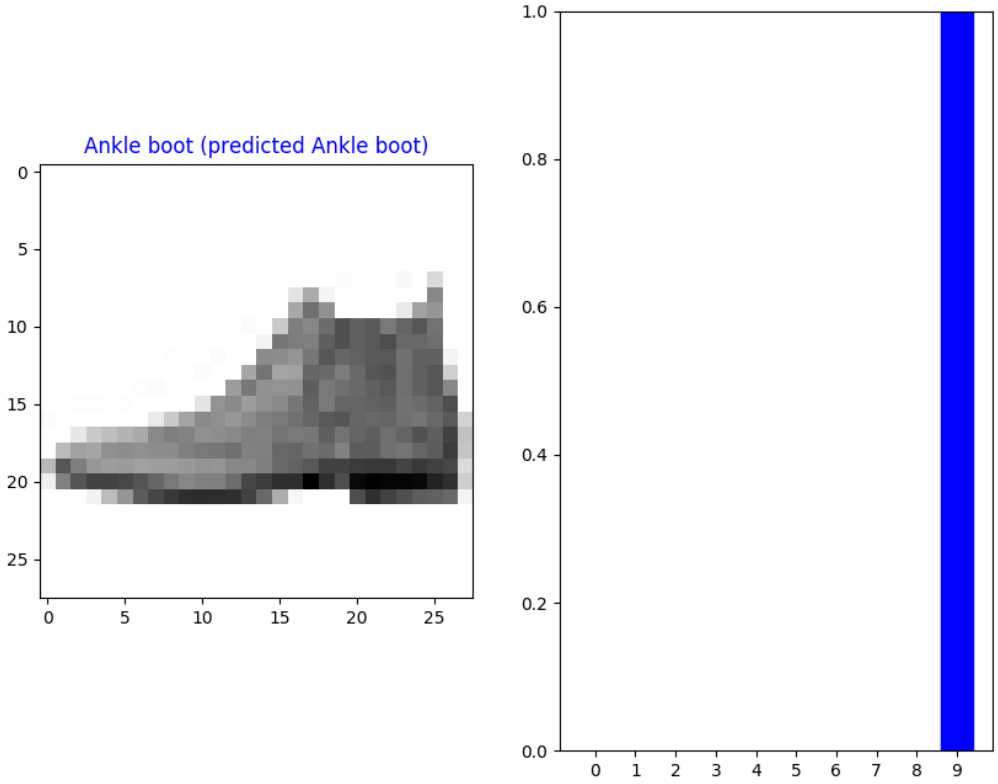


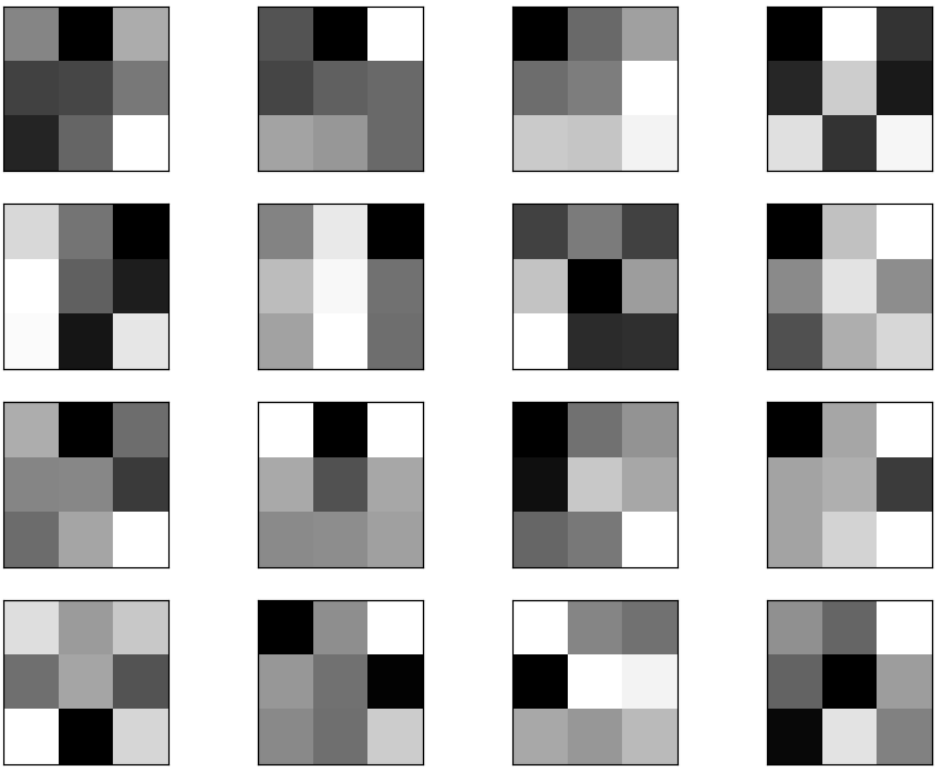
Image shifted 1-pixel right

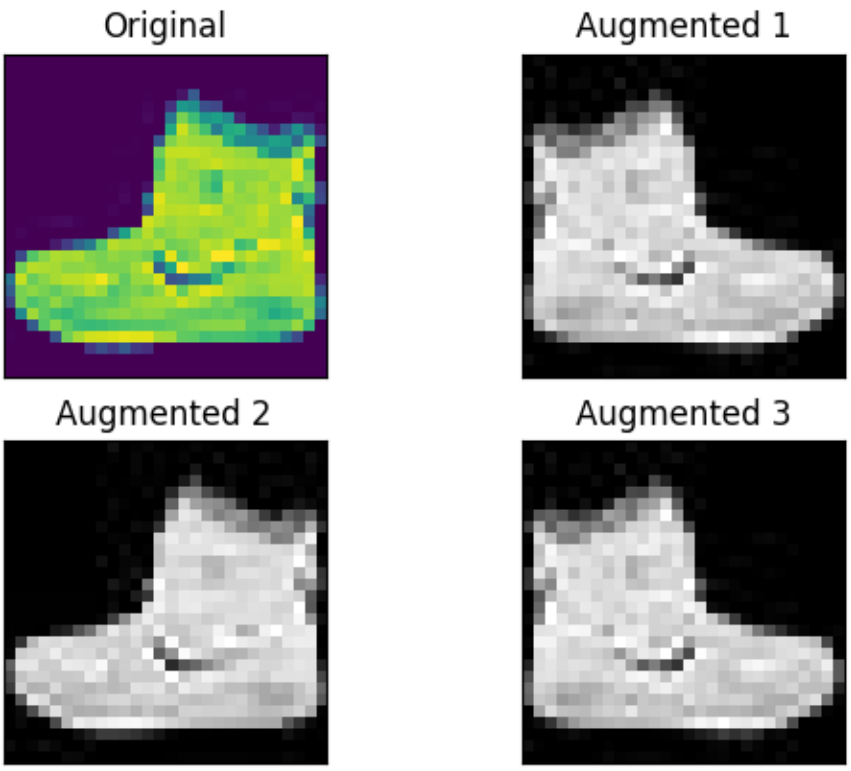


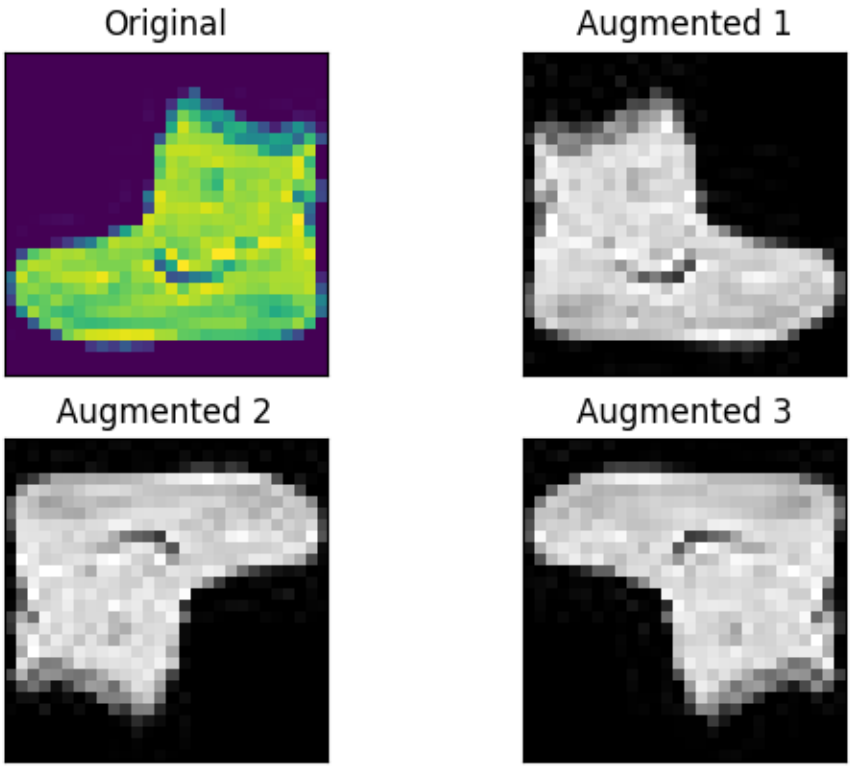

Ankle boot
0
5
10
15
20
25
0
5
10
15
20
25
250
200
150
100
50
0

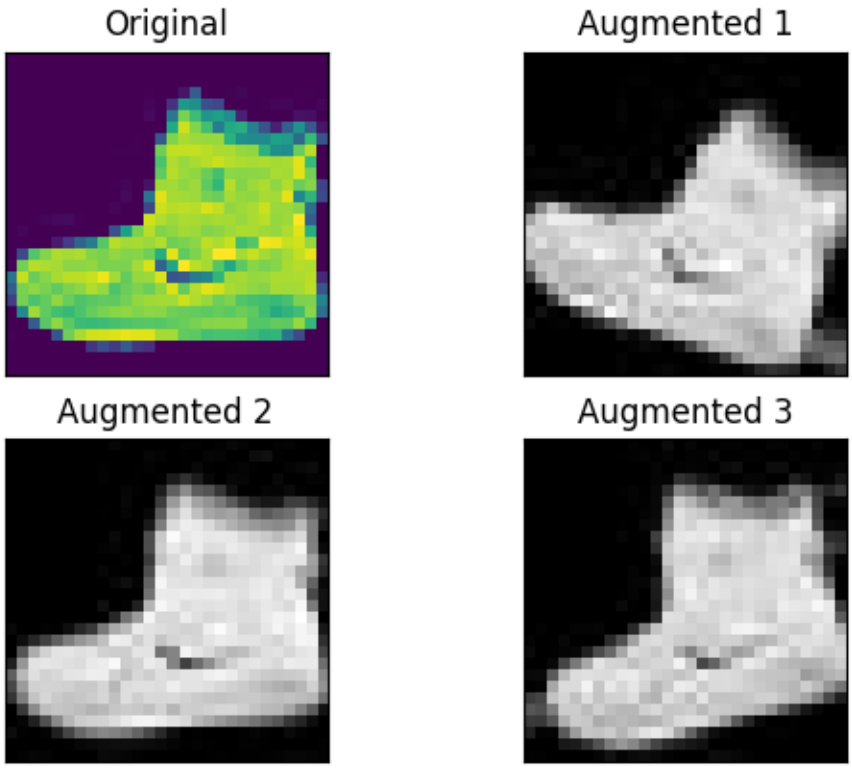

Ankle boot
T-shirt/top
T-shirt/top
Dress
T-shirt/top
Pullover
Sneaker
Pullover
Sandal
Sandal
T-shirt/top
Ankle boot
Sandal
Sandal
Sneaker
Ankle boot

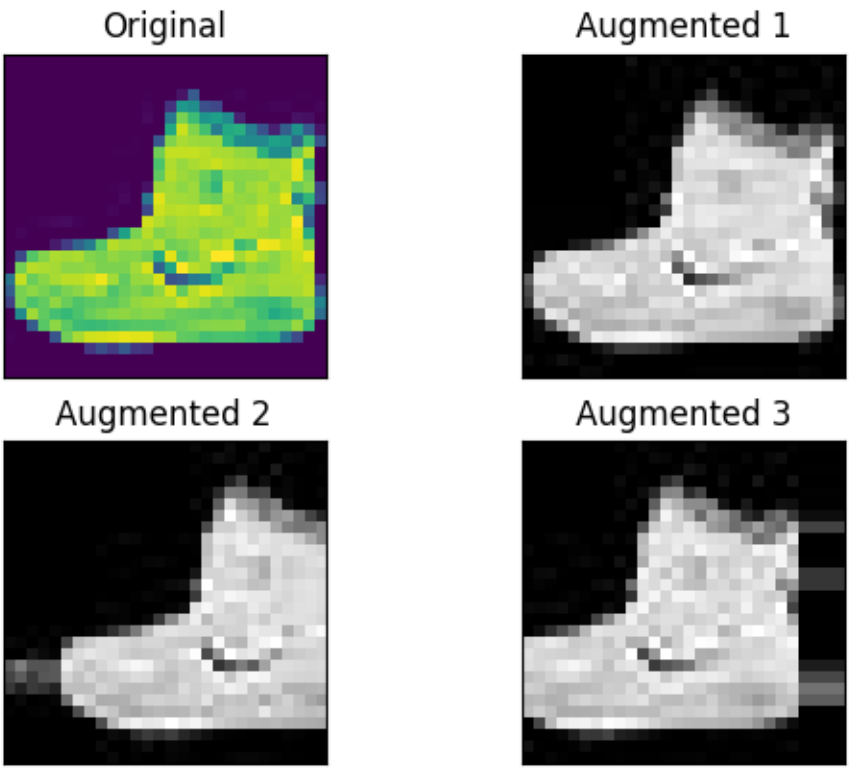

Ankle boot (predicted Ankle boot)
0
5
10
15
20
25
0
5
10
15
20
25
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
9

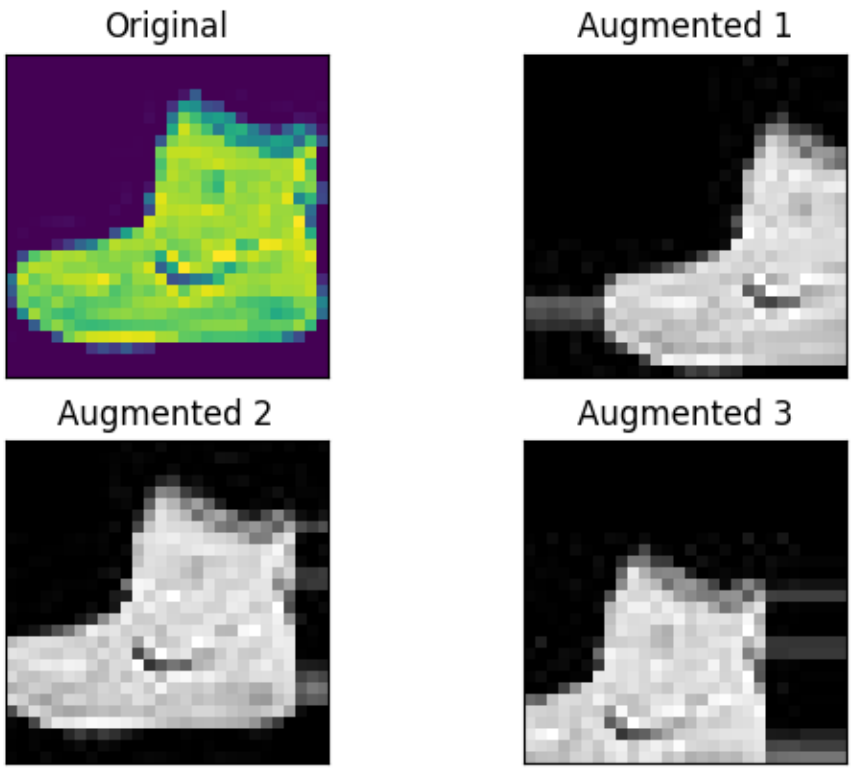



Original
Augmented 1
Augmented 2
Augmented 3

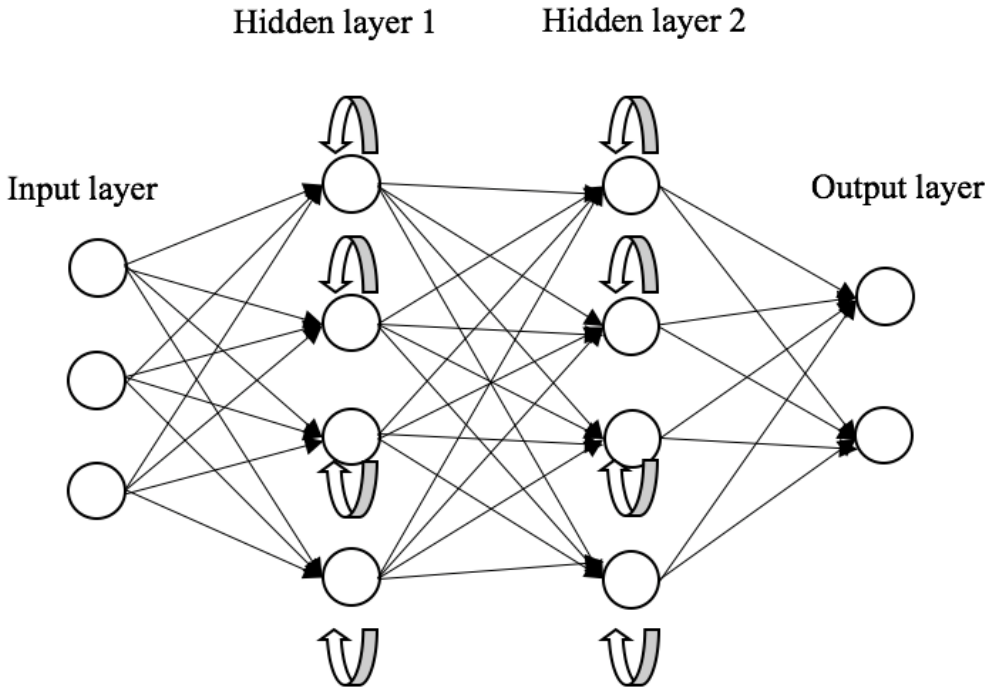

Original
Augmented 1
Augmented 2
Augmented 3


Original
Augmented 1
Augmented 2
Augmented 3

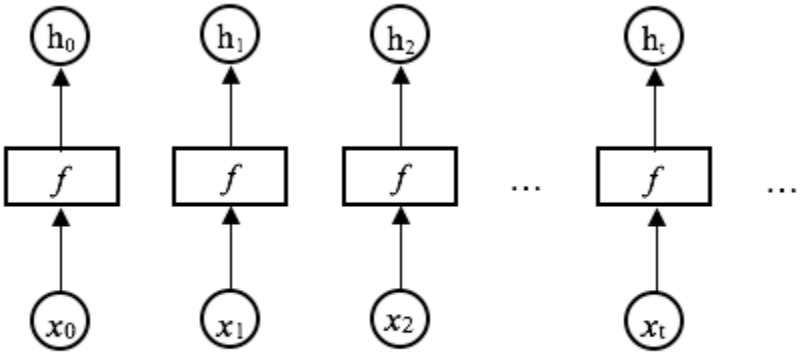

Original
Augmented 1
Augmented 2
Augmented 3


Original
Augmented 1
Augmented 2
Augmented 3

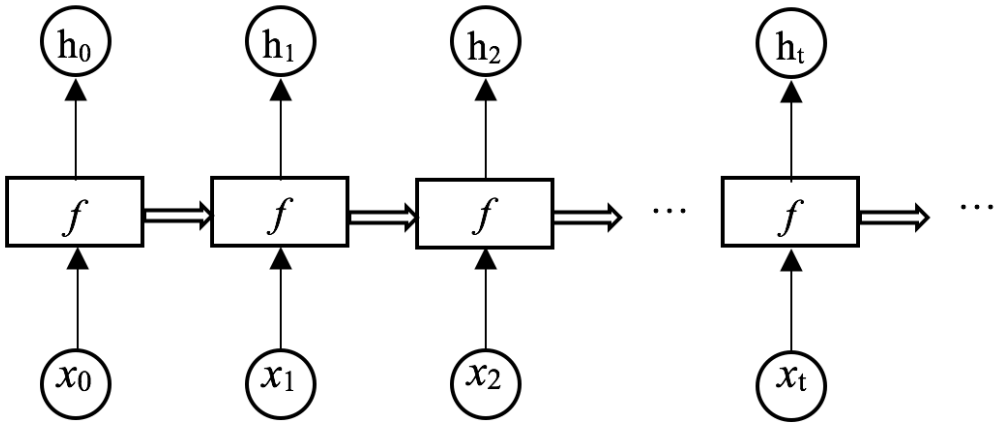
Chapter 13: Making Predictions with Sequences Using Recurrent Neural Networks



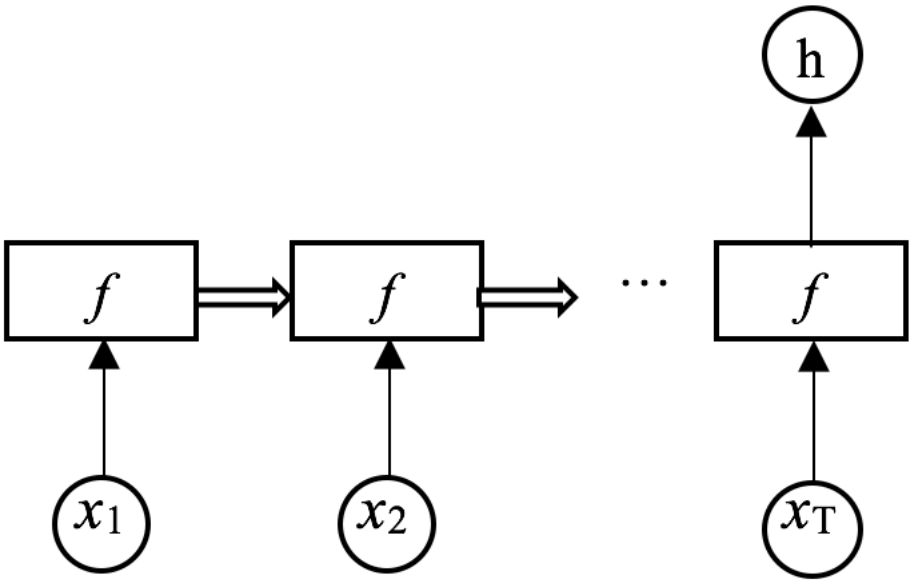
General Form of RNNs



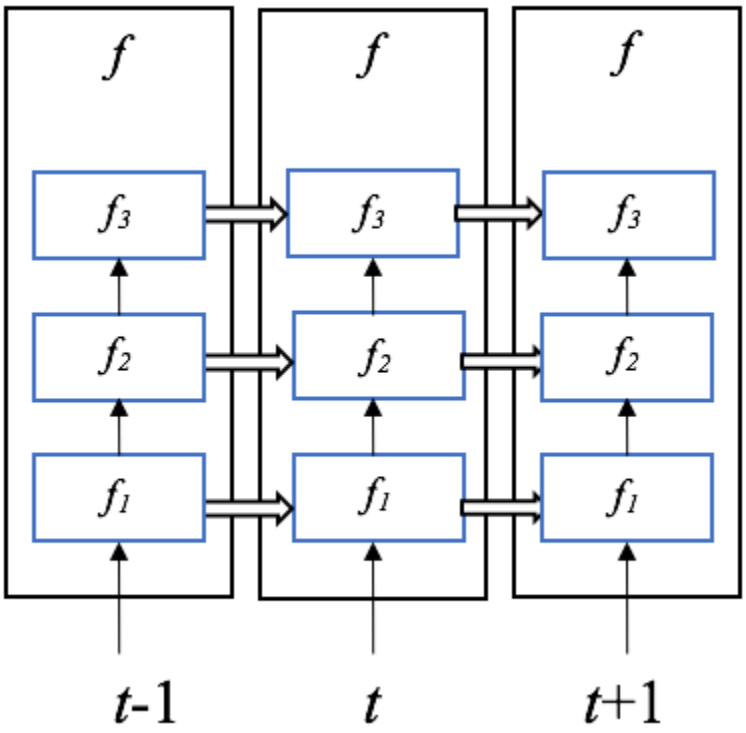
Feedforward NN

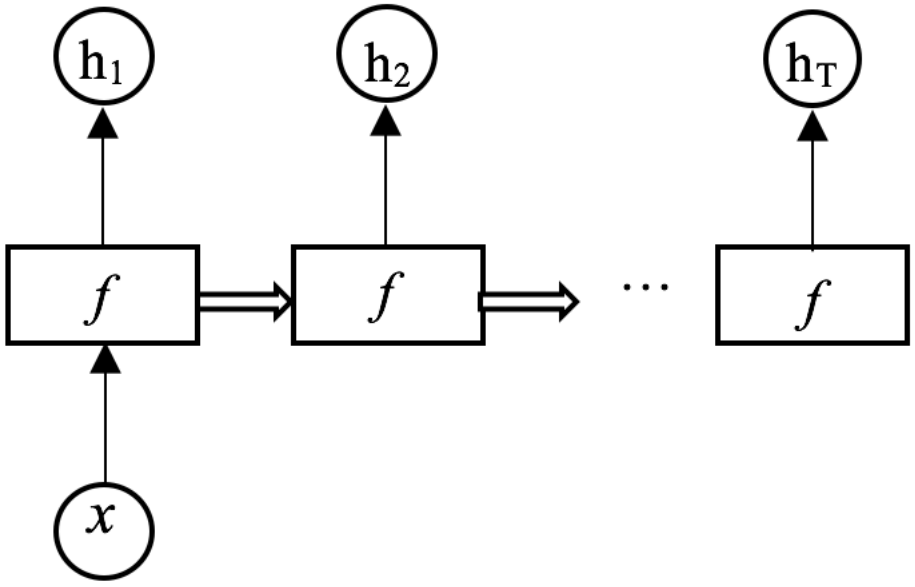


RNN

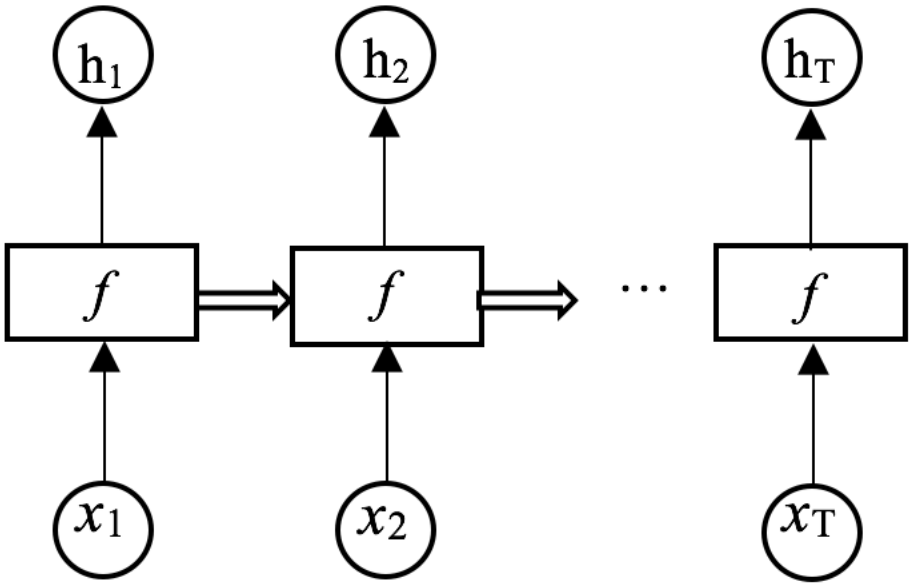


Many-to-one RNN

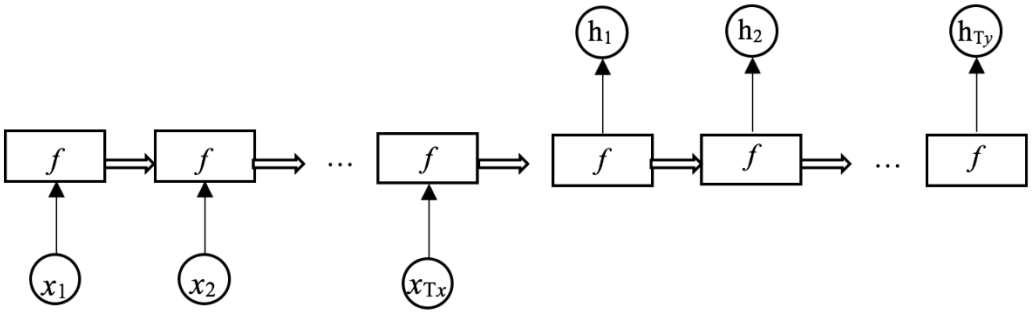

f
f
f
f3
f3
f3
f2
f2
f2
f1
f1
f1
t-1
t
t+1



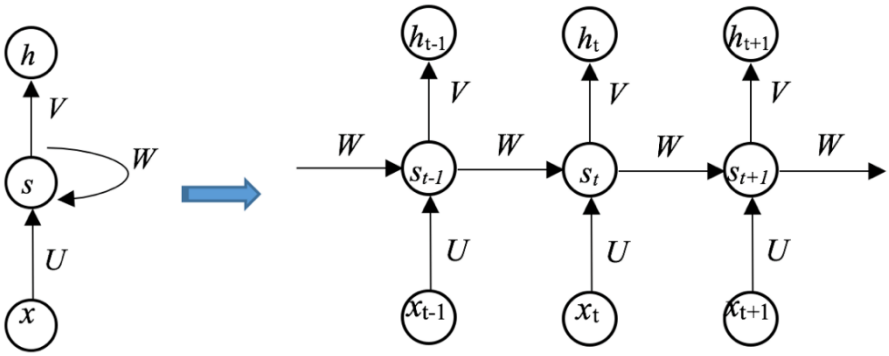
One-to-many RNN



Many-to-many (synced) RNN

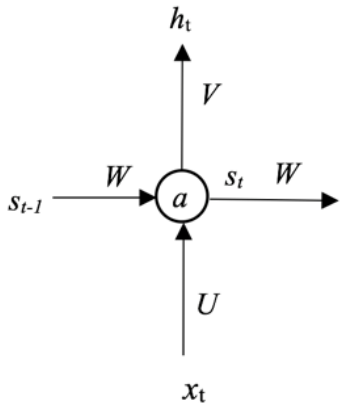


Many-to-many (unsynced) RNN

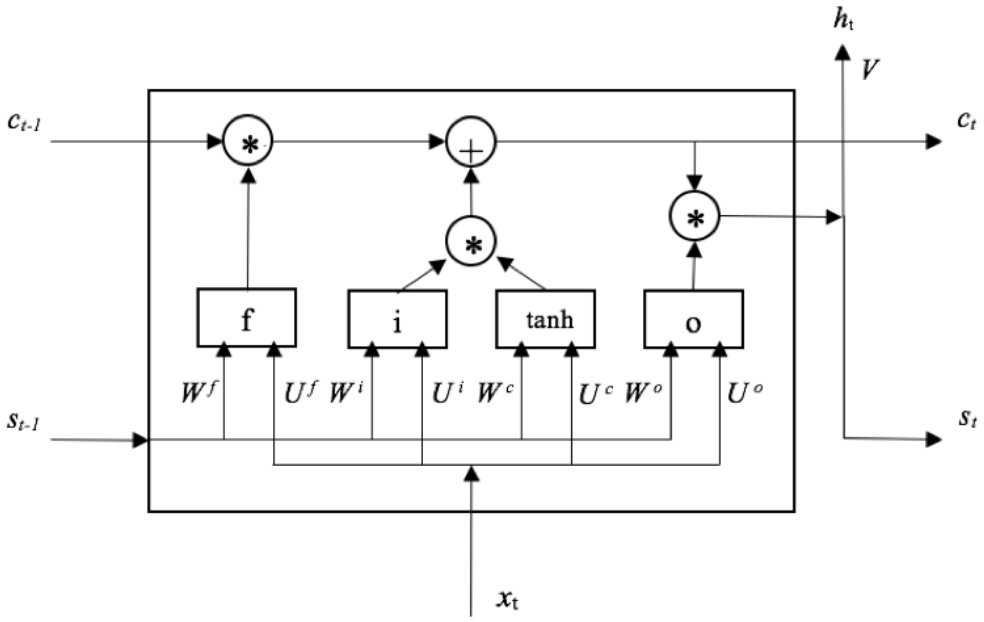


A basic one-layered RNN

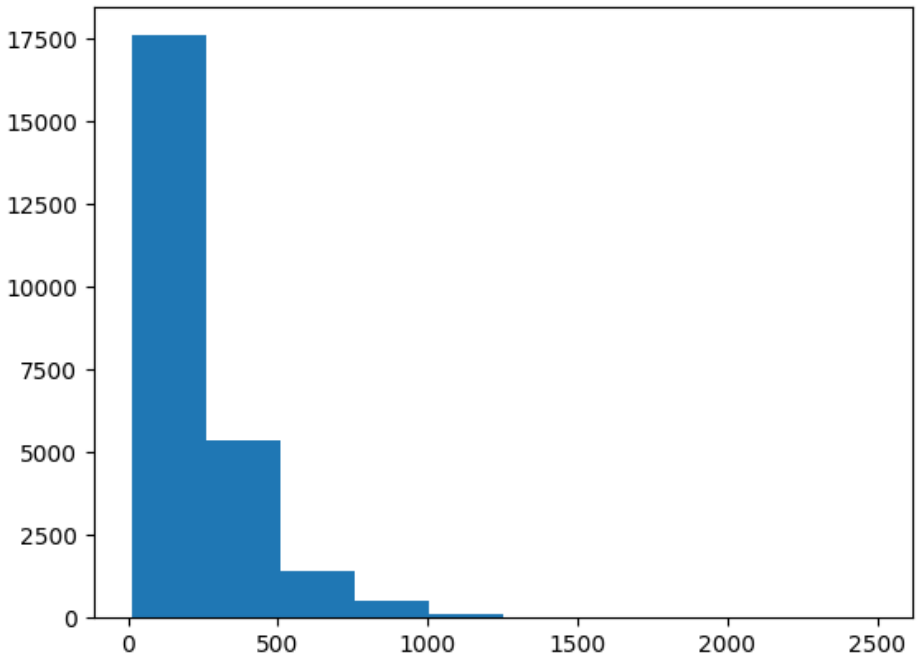
Unfolded version

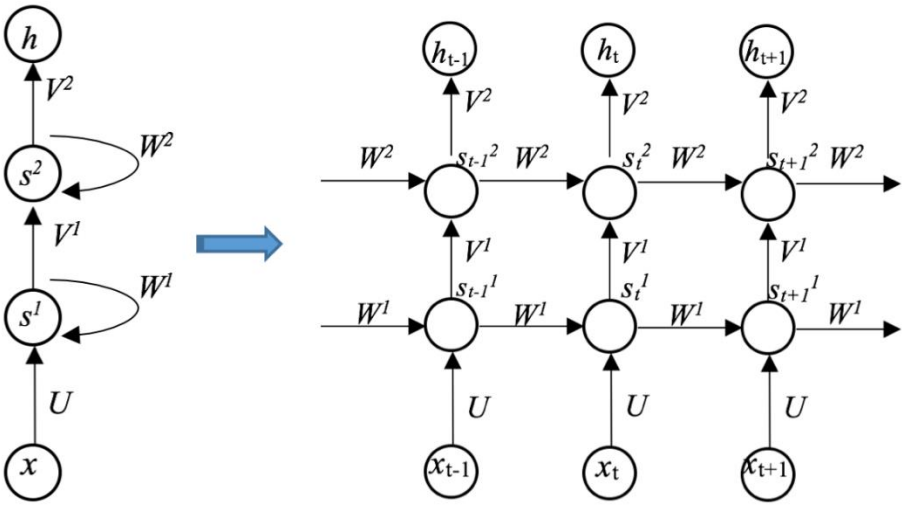


Recurrent cell of a vanilla RNN



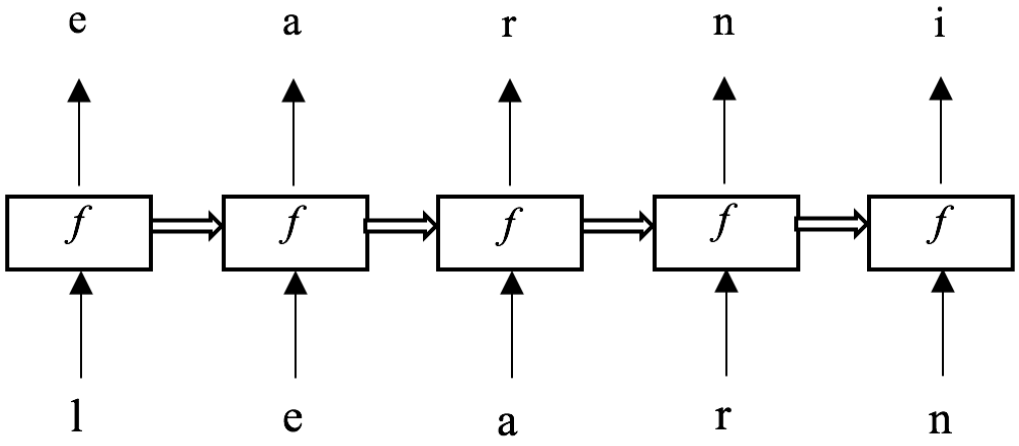
Recurrent cell of an LSTM RNN


17500
15000
12500
10000
7500
5000
2500
0
0
500
1000
1500
2000
2500



Two-layered RNN

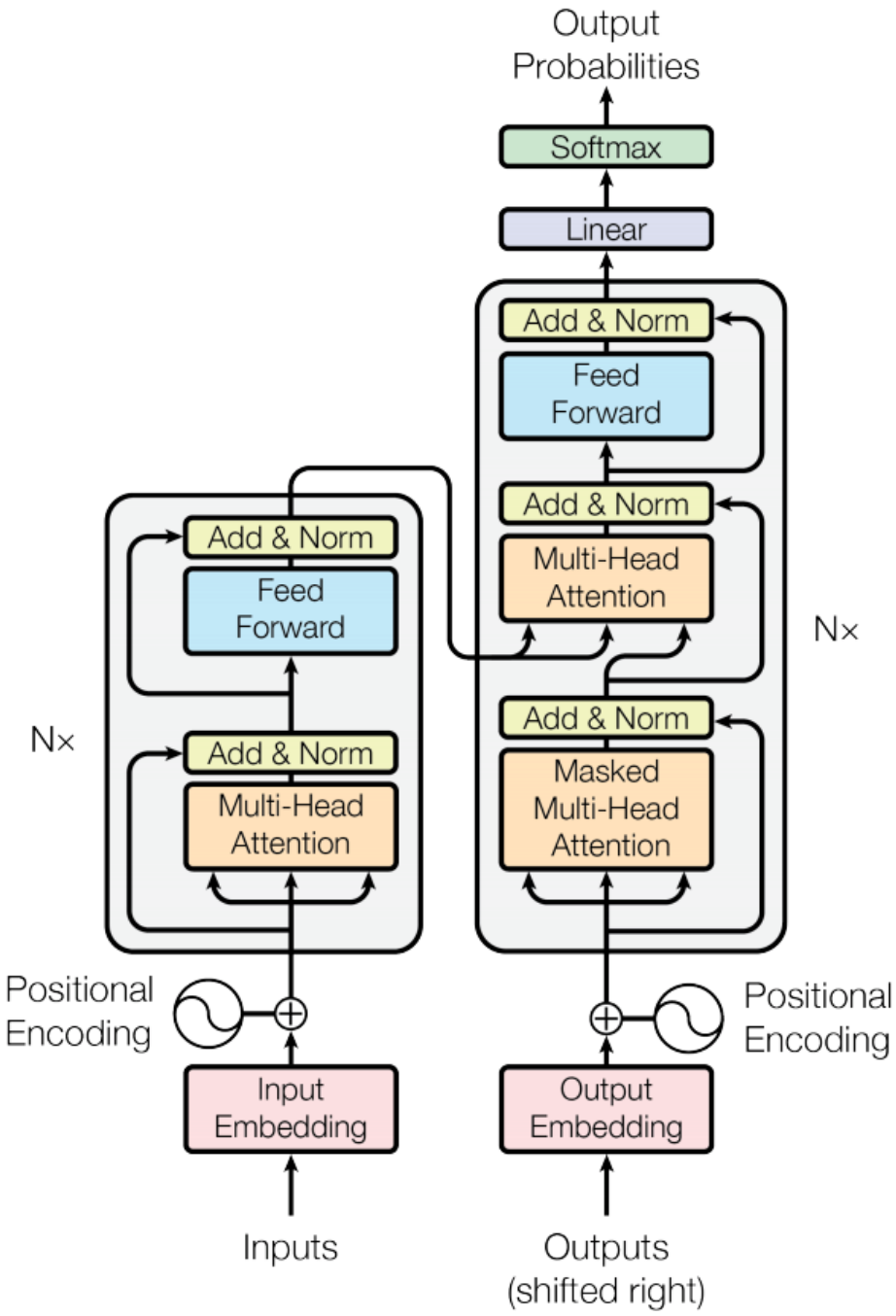
Unfolded version



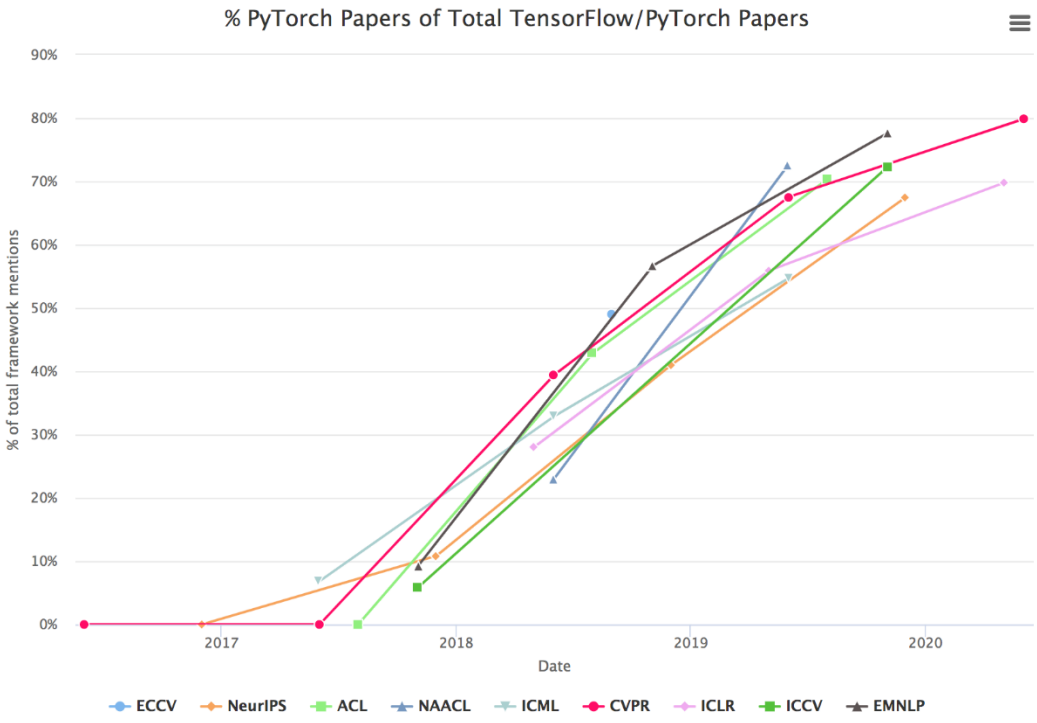
Input sequence: learn

Output sequence: earni

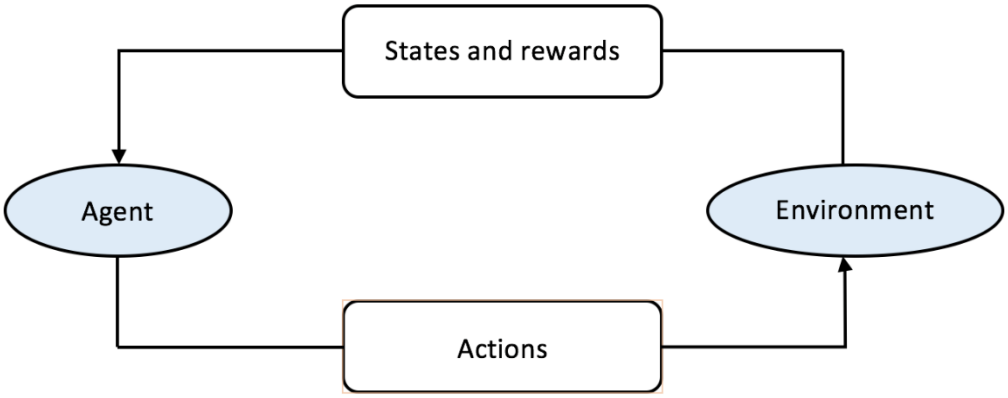
Input	Output
machi	achin
ne□le	e□lea
arnin	rning
g□by□	□by□e
examp	xampl


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Chapter 14: Making Decisions in Complex Environments with Reinforcement Learning



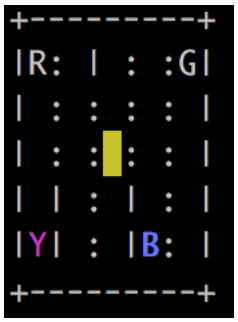
PyTorch Build	Stable (1.5.1)	Preview (Nightly)		
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
CUDA	9.2	10.1	10.2	None
Run this Command:	`conda install pytorch torchvision -c pytorch`			

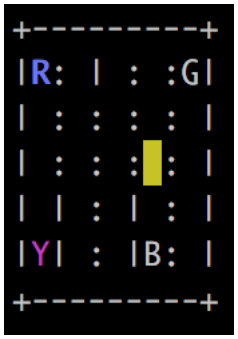

States and rewards
Agent
Environment
Actions

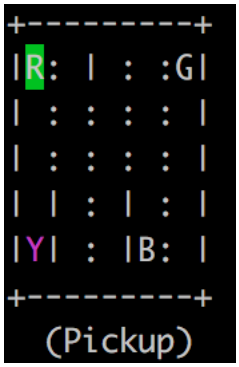

SFFF
FHFH
FFFH
HFFG

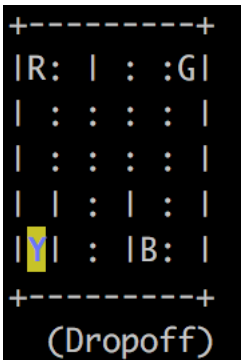

(Right)
SFFF
FHFH
FFFH
HFFG

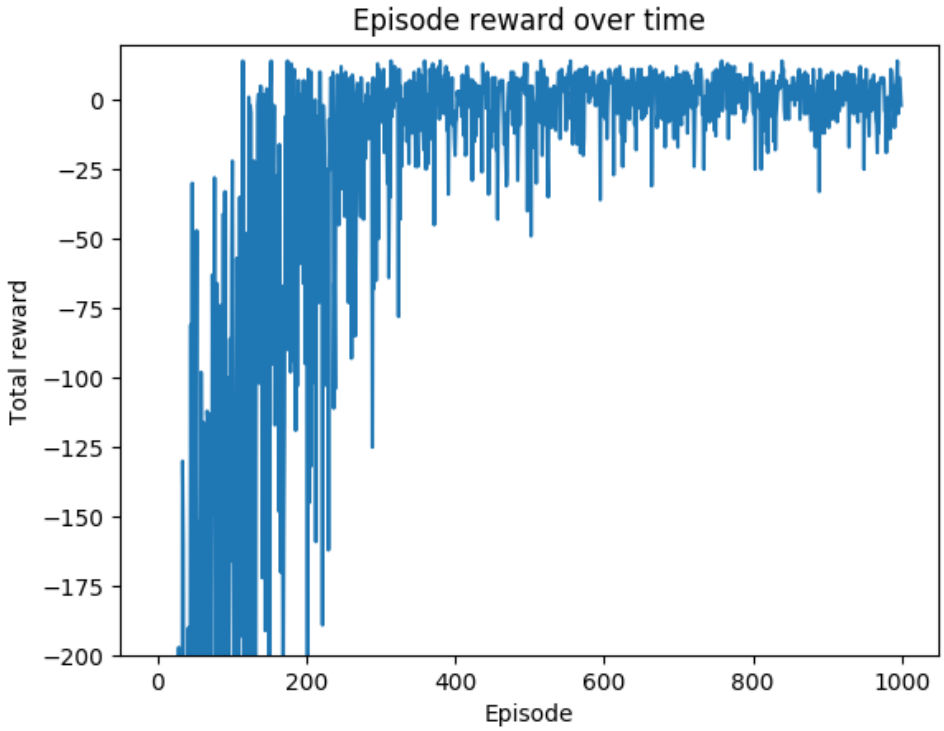
Scenario	Preference	Reason
A large number of actions	Policy iteration	Policy iteration can converge faster
A small number of actions	Value iteration	Less computation in value iteration
A fair policy exists (obtained either by intuition or domain knowledge)	Policy iteration	Policy iteration from a fair policy can converge faster
Others	No preference	Policy iteration and value iteration are comparable

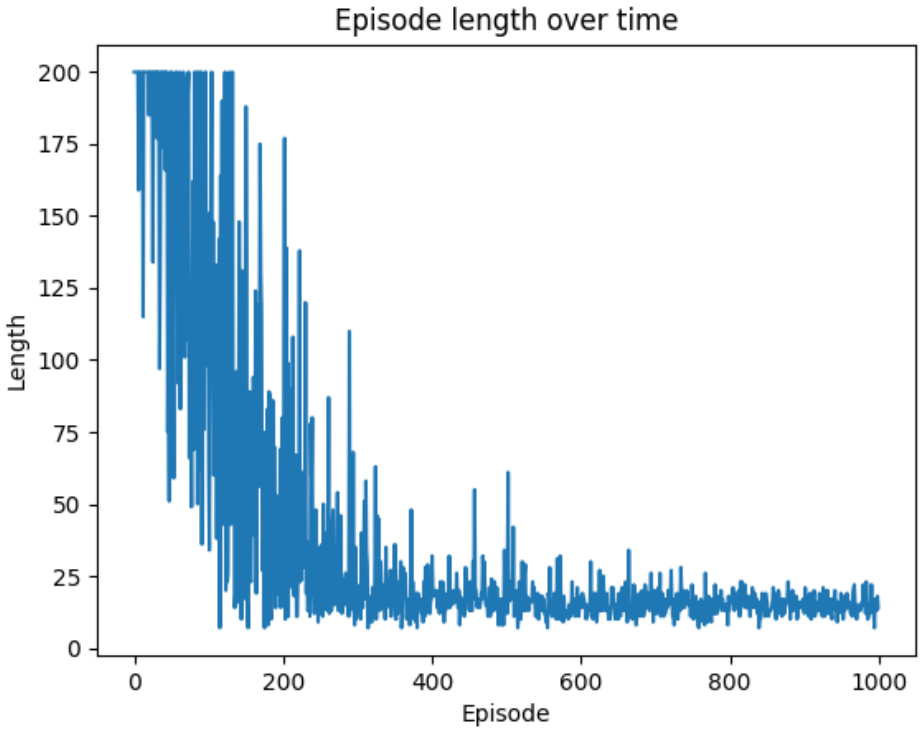





+---------+
|R: | : :G|
| : : : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
(Pickup)


+---------+
|R: | : :G|
| : : : : |
| : : : : |
| | : | : |
|Y| : |B: |
+---------+
(Dropoff)


Episode reward over time
Total reward
0
−25
−50
−75
−100
−125
−150
−175
−200
0
200
400
600
800
1000
Episode


Episode length over time
Length
Episode
0
25
50
75
100
125
150
175
200
0
200
400
600
800
1000