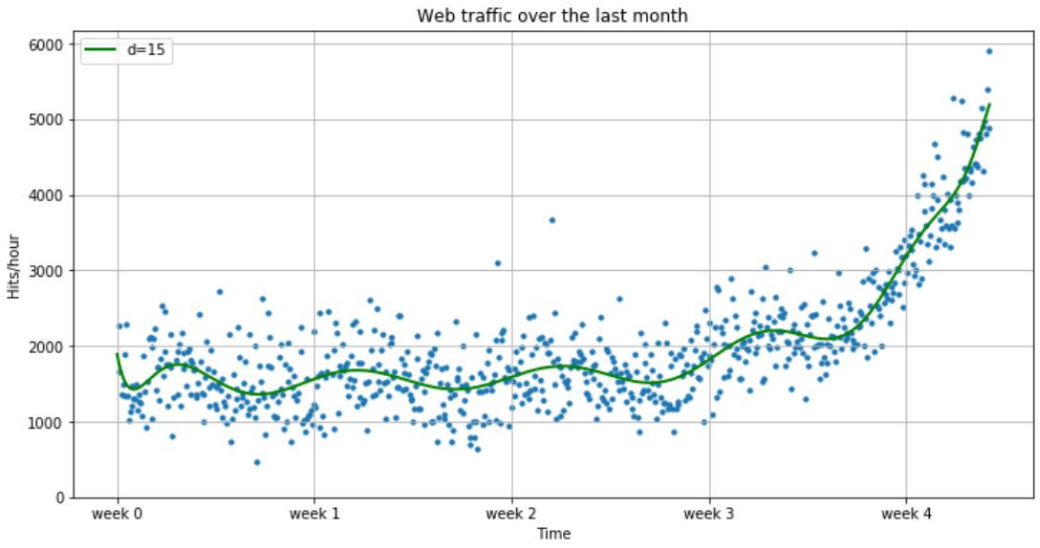
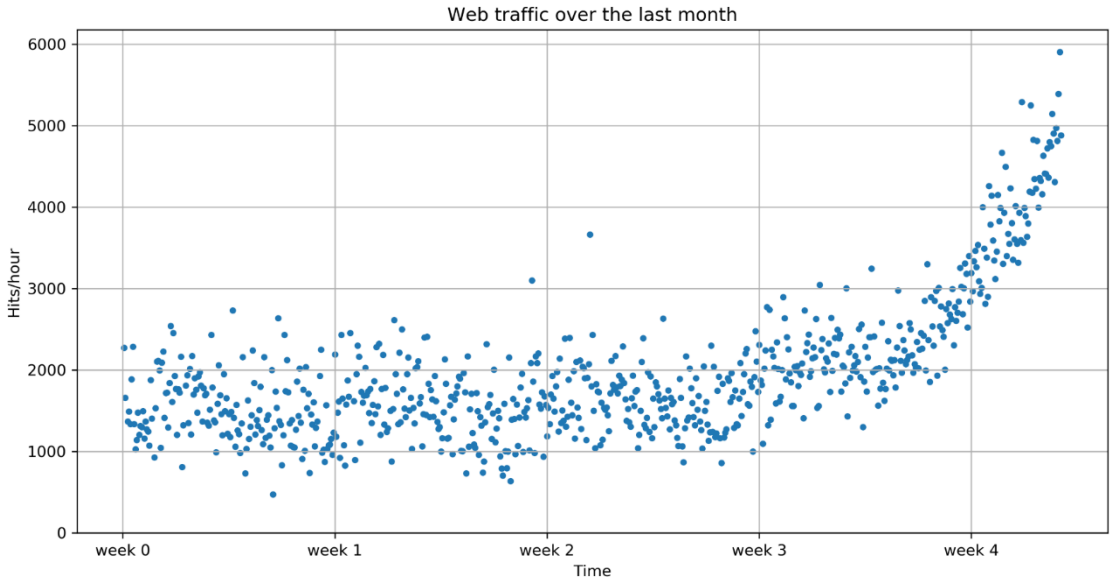
Chapter 1: Getting Started with Python Machine Learning

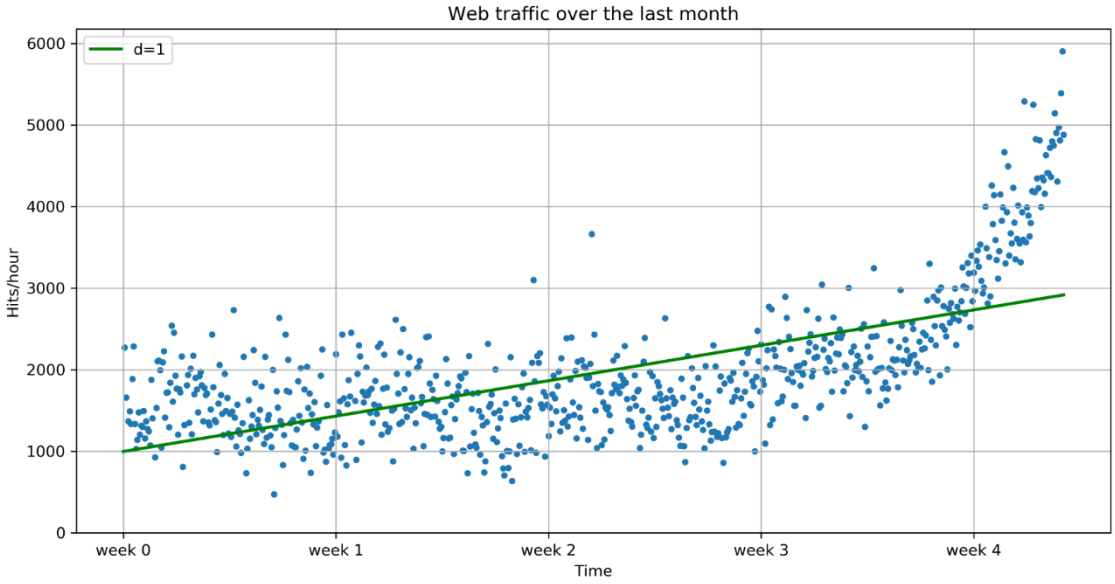
In [55]:
```
from ipywidgets import interactive
import ipywidgets as widgets

def play_with_dim(dim=1):
    f = np.poly1d(np.polyfit(x, y, dim))
    plot_web_traffic(x, y, [f])
    print("Error for d=%i: %f" % (f.order, error(f, x, y)))

interactive_plot = interactive(play_with_dim, dim=(1,100))
output = interactive_plot.children[-1]
output.layout.height = '500px'
interactive_plot
```

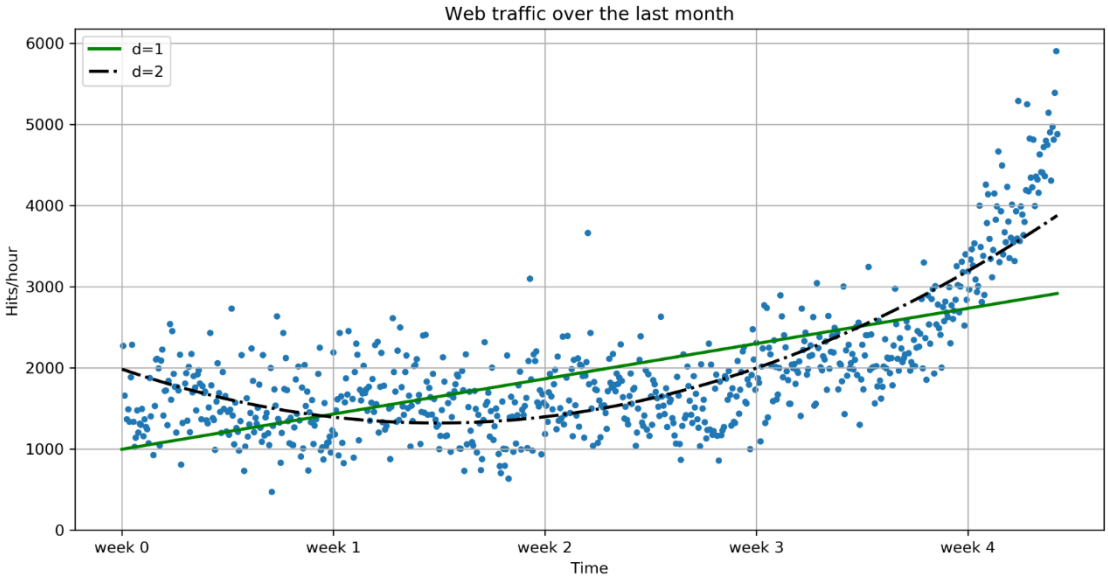
dim 15

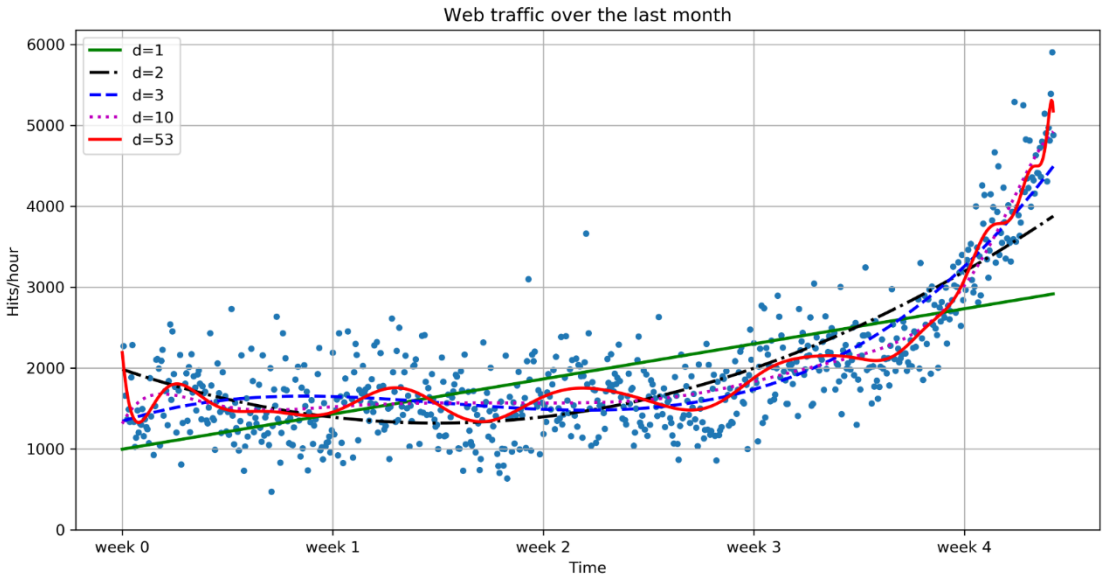
```
Error for d=15: 113617150.429347
```

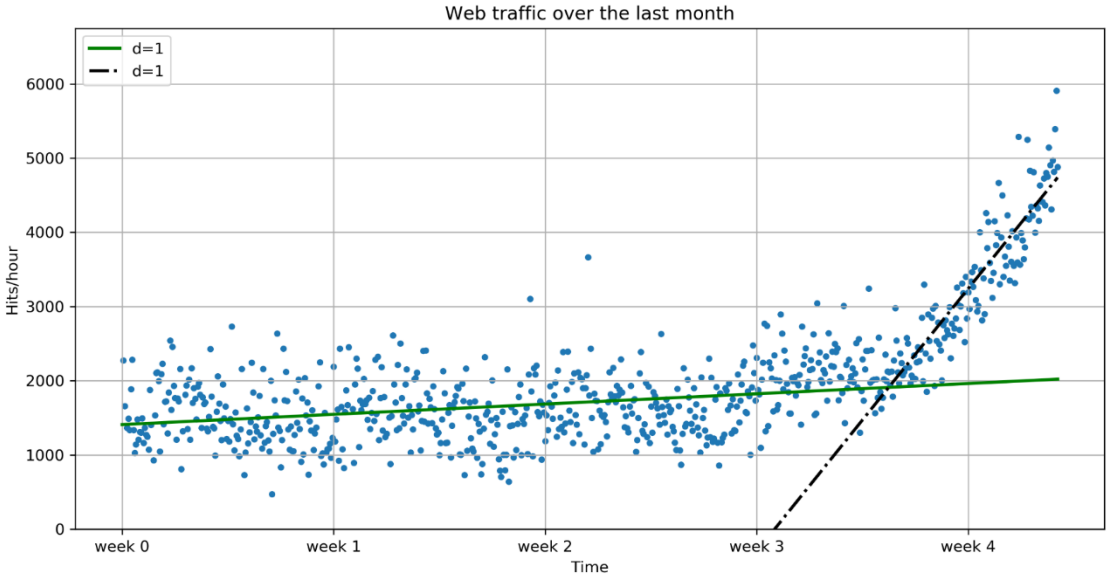


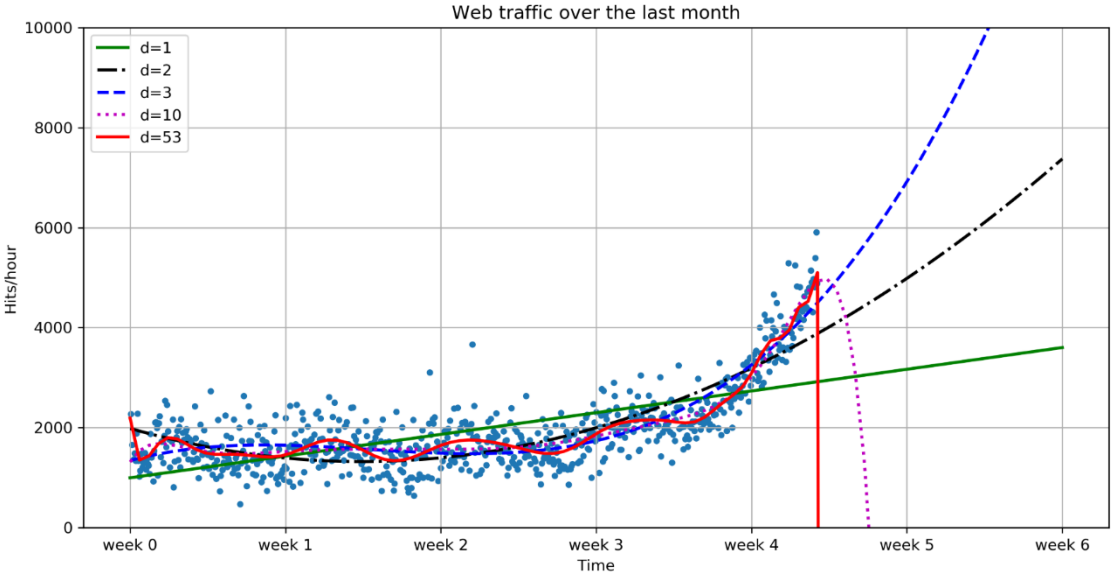

Web traffic over the last month
Hits/hour
Time
week 0
week 1
week 2
week 3
week 4
0
1000
2000
3000
4000
5000
6000

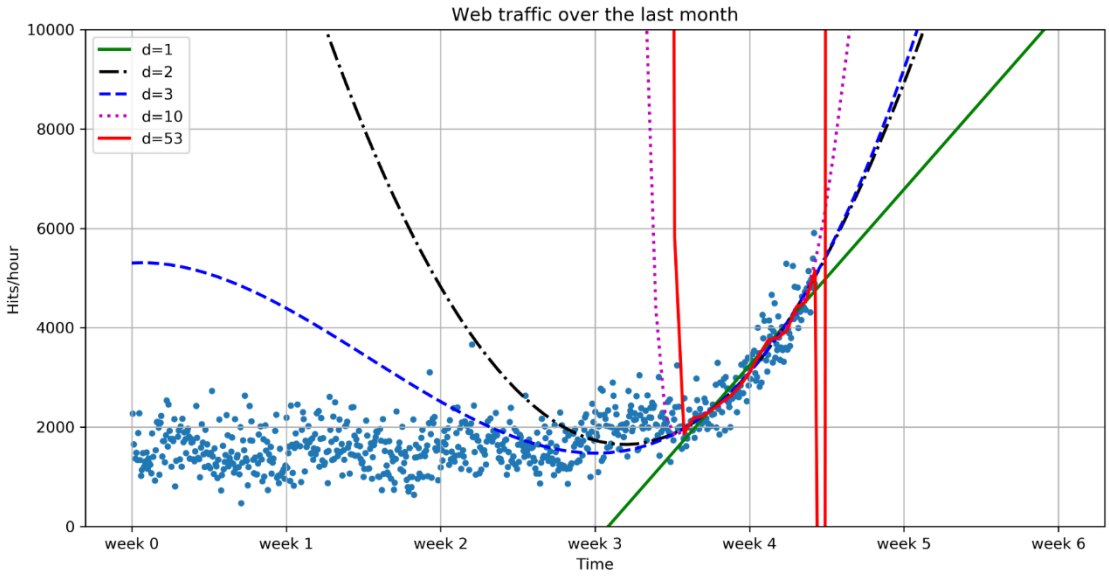

Web traffic over the last month
d=1
Hits/hour
Time
week 0
week 1
week 2
week 3
week 4
0
1000
2000
3000
4000
5000
6000

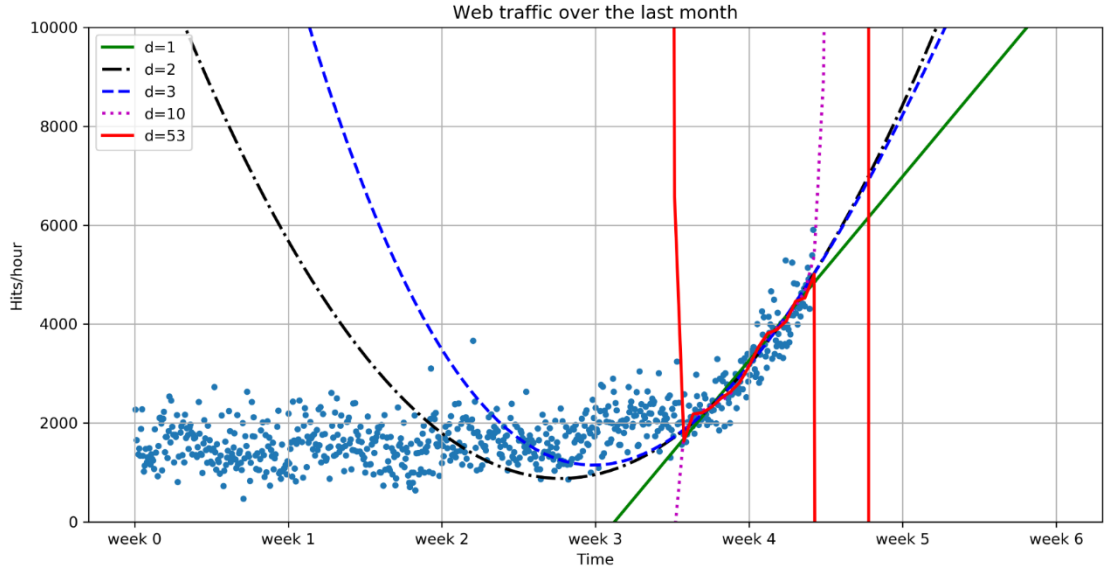

Web traffic over the last month
d=1
d=2
Hits/hour
Time
week 0
week 1
week 2
week 3
week 4
0
1000
2000
3000
4000
5000
6000

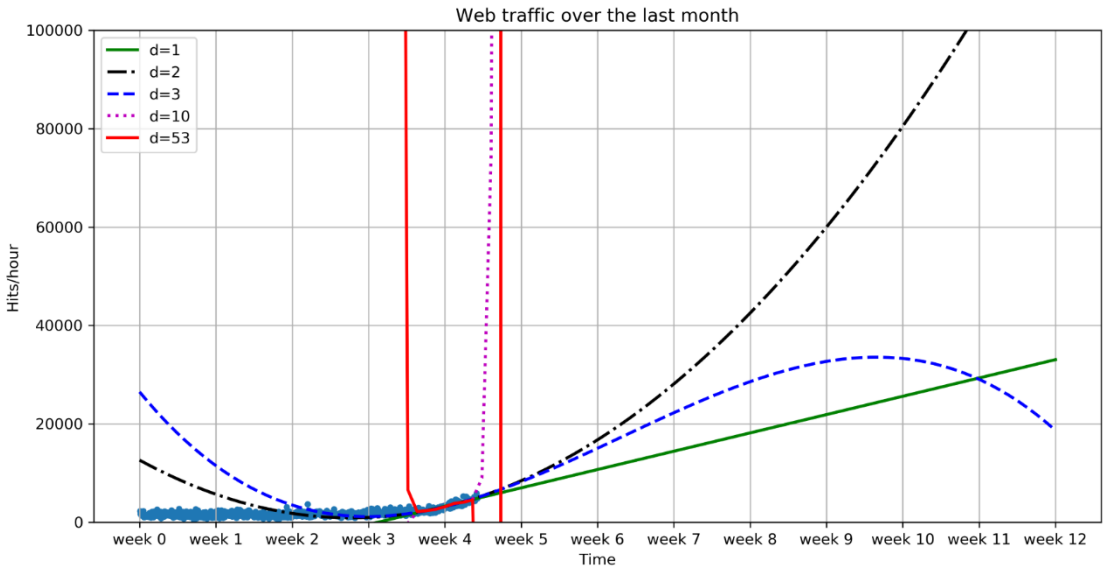

Web traffic over the last month
d=1
d=2
d=3
d=10
d=53
Hits/hour
Time
week 0
week 1
week 2
week 3
week 4
0
1000
2000
3000
4000
5000
6000


Web traffic over the last month
d=1
d=1
Hits/hour
0
1000
2000
3000
4000
5000
6000
week 0
week 1
week 2
week 3
week 4
Time

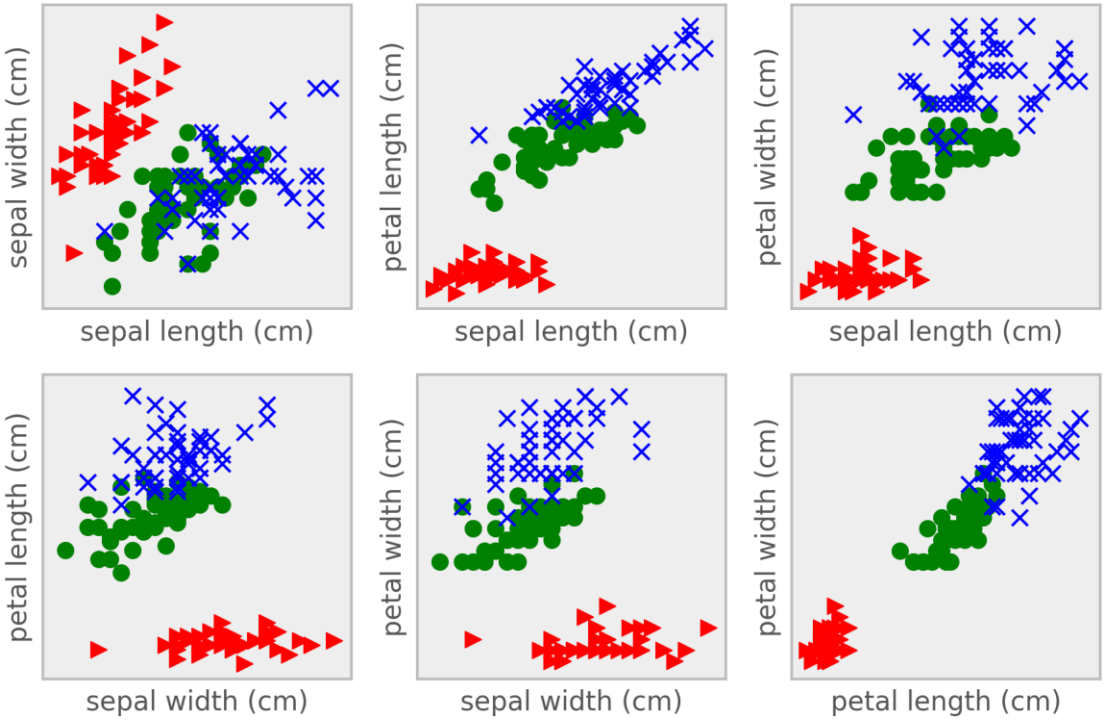

Web traffic over the last month
d=1
d=2
d=3
d=10
d=53
Hits/hour
0
2000
4000
6000
8000
10000
week 0
week 1
week 2
week 3
week 4
week 5
week 6
Time

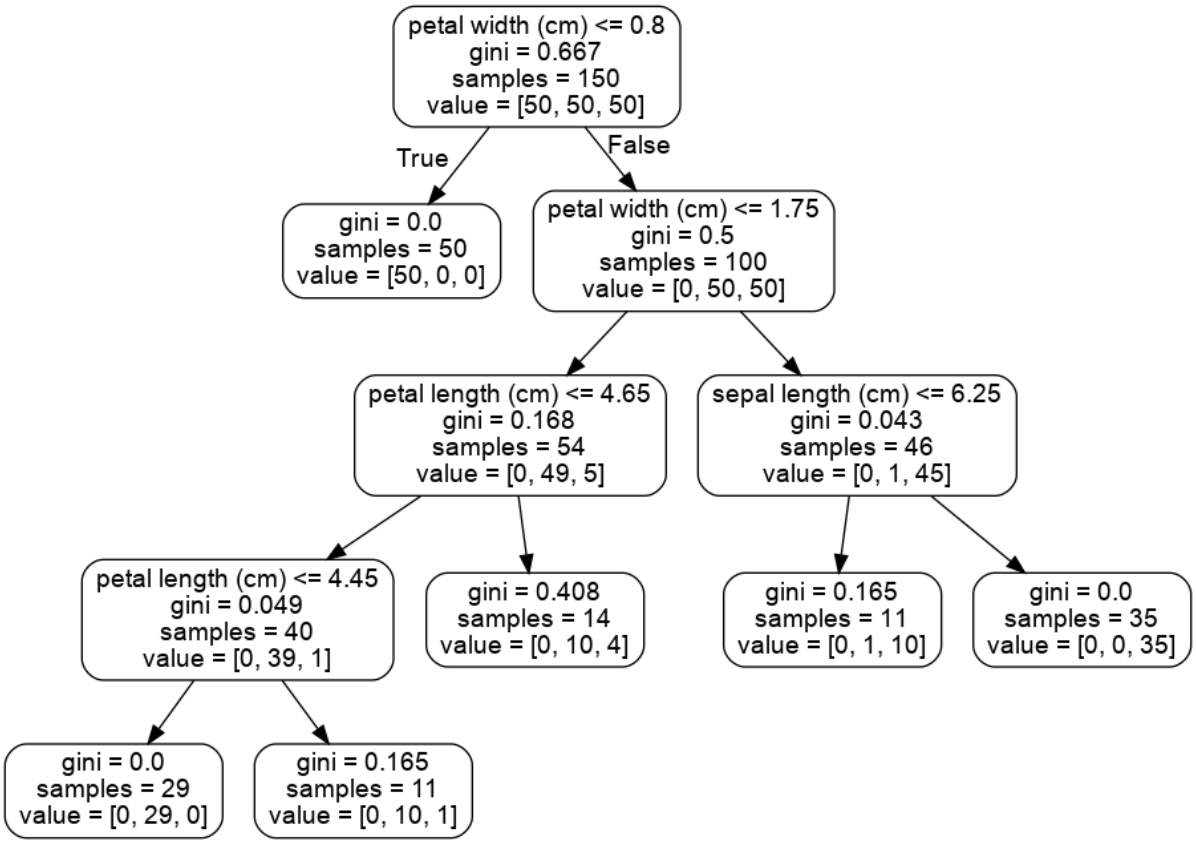

Web traffic over the last month
d=1
d=2
d=3
d=10
d=53
Hits/hour
Time
week 0
week 1
week 2
week 3
week 4
week 5
week 6
0
2000
4000
6000
8000
10000

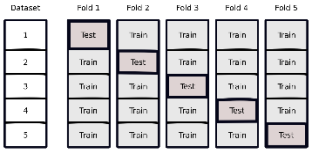

Web traffic over the last month
d=1
d=2
d=3
d=10
d=53
Hits/hour
Time
week 0
week 1
week 2
week 3
week 4
week 5
week 6
0
2000
4000
6000
8000
10000

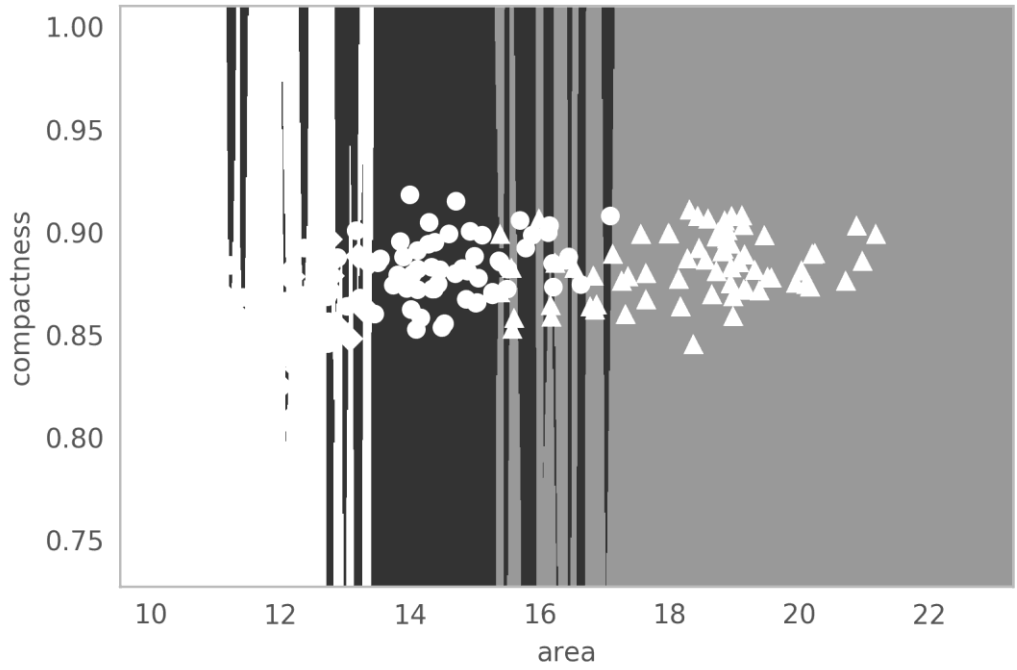

Web traffic over the last month
d=1
d=2
d=3
d=10
d=53
100000
80000
60000
40000
20000
0
Hits/hour
week 0
week 1
week 2
week 3
week 4
week 5
week 6
week 7
week 8
week 9
week 10
week 11
week 12
Time

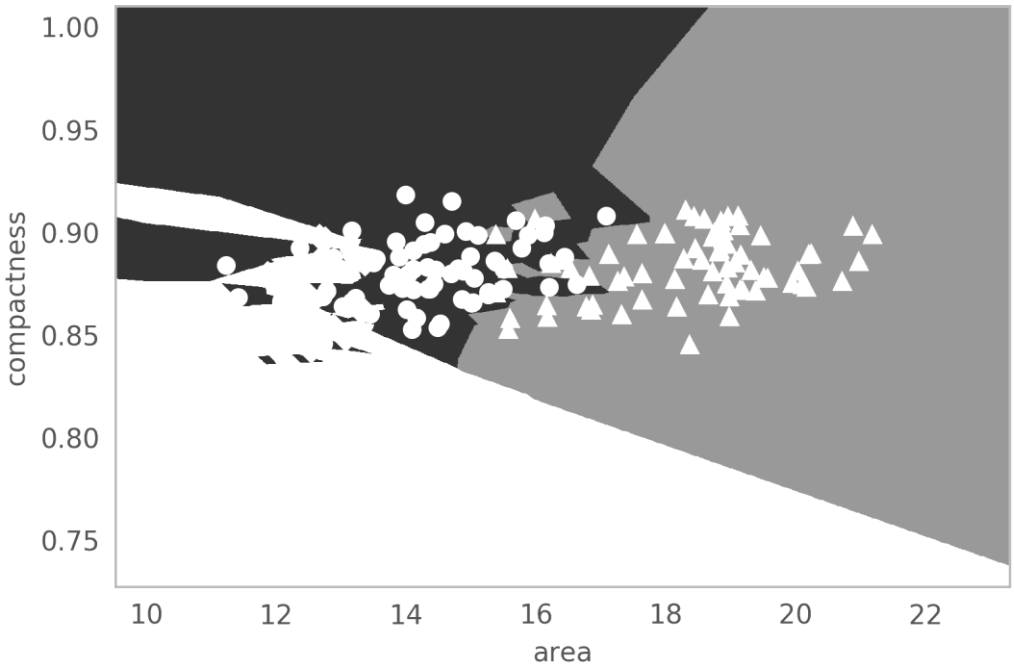
Chapter 2: Classifying with Real-world Examples



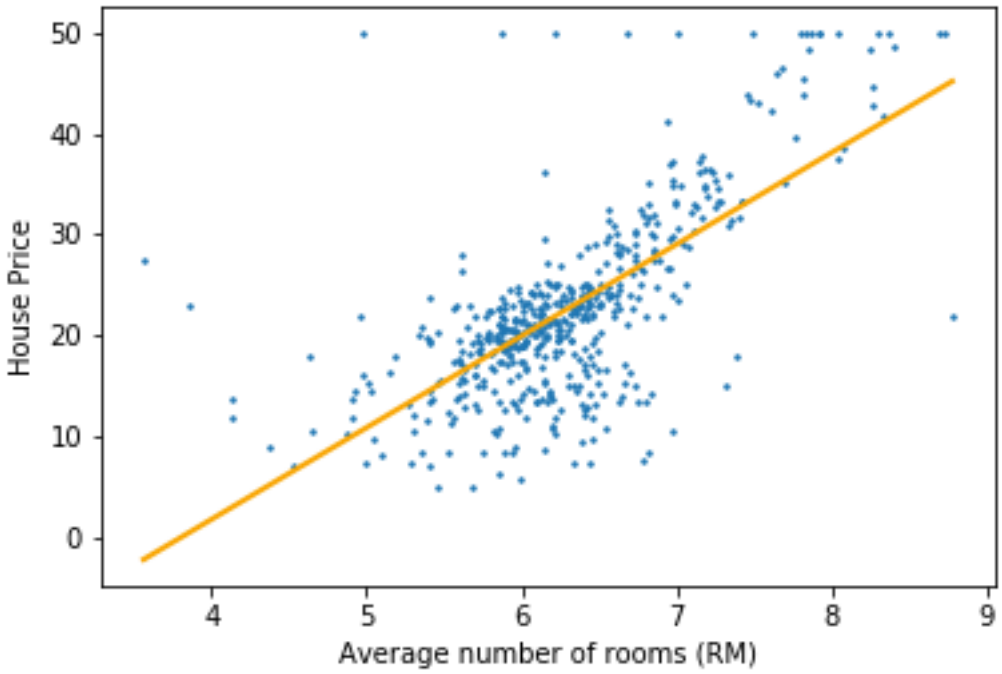

petal width (cm) <= 0.8
gini = 0.667
samples = 150
value = [50, 50, 50]
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
petal width (cm) <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
petal length (cm) <= 4.65
gini = 0.168
samples = 54
value = [0, 49, 5]
sepal length (cm) <= 6.25
gini = 0.043
samples = 46
value = [0, 1, 45]
petal length (cm) <= 4.45
gini = 0.049
samples = 40
value = [0, 39, 1]
gini = 0.408
samples = 14
value = [0, 10, 4]
gini = 0.165
samples = 11
value = [0, 1, 10]
gini = 0.0
samples = 35
value = [0, 0, 35]
gini = 0.0
samples = 29
value = [0, 29, 0]
gini = 0.165
samples = 11
value = [0, 10, 1]

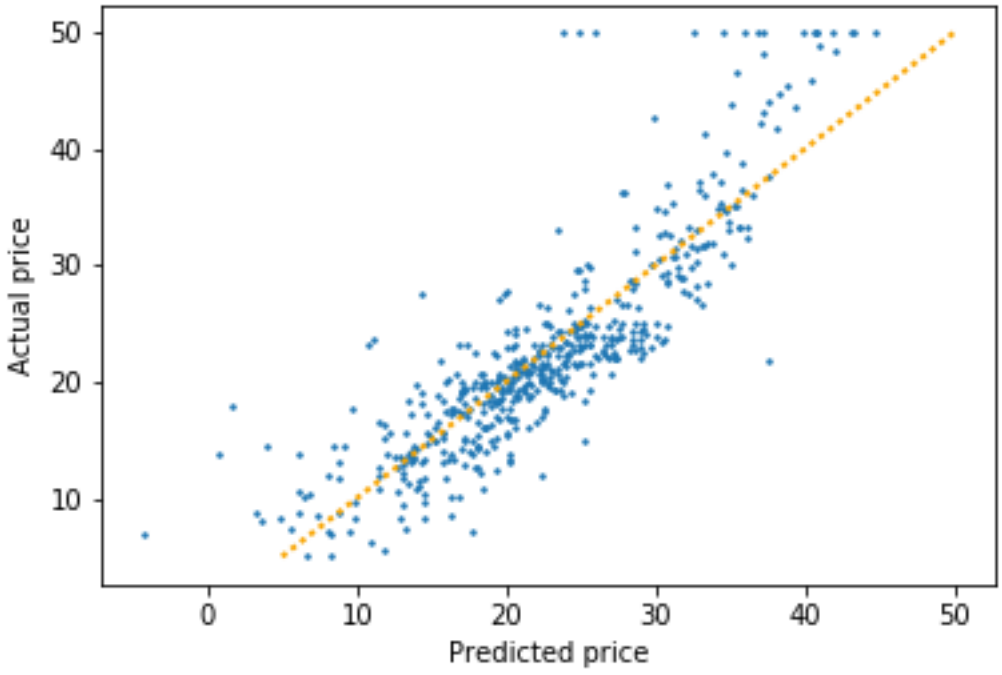

Dataset
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
1
2
3
4
5
Test
Train

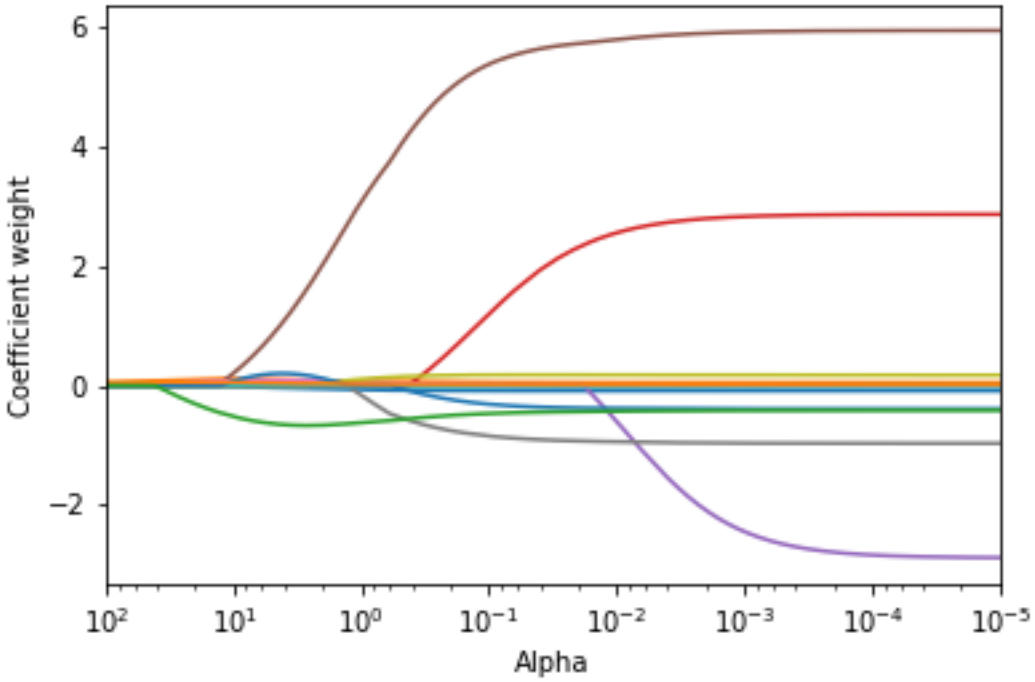

1.00
0.95
0.90
0.85
0.80
0.75
compactness
10
12
14
16
18
20
22
area

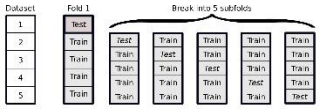

1.00
0.95
0.90
0.85
0.80
0.75
compactness
10
12
14
16
18
20
22
area

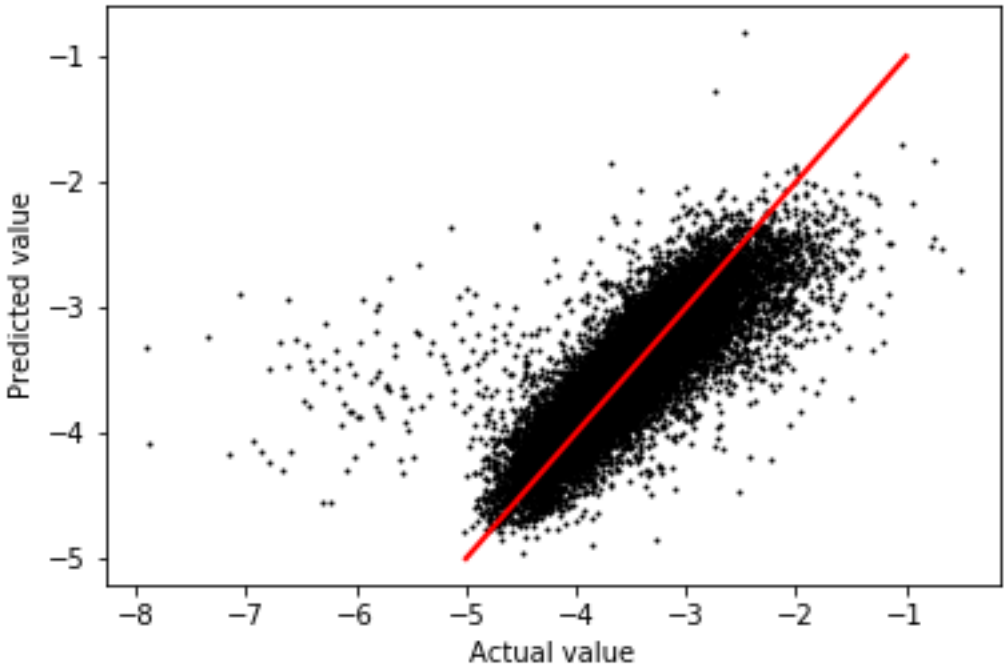
Chapter 3: Regression

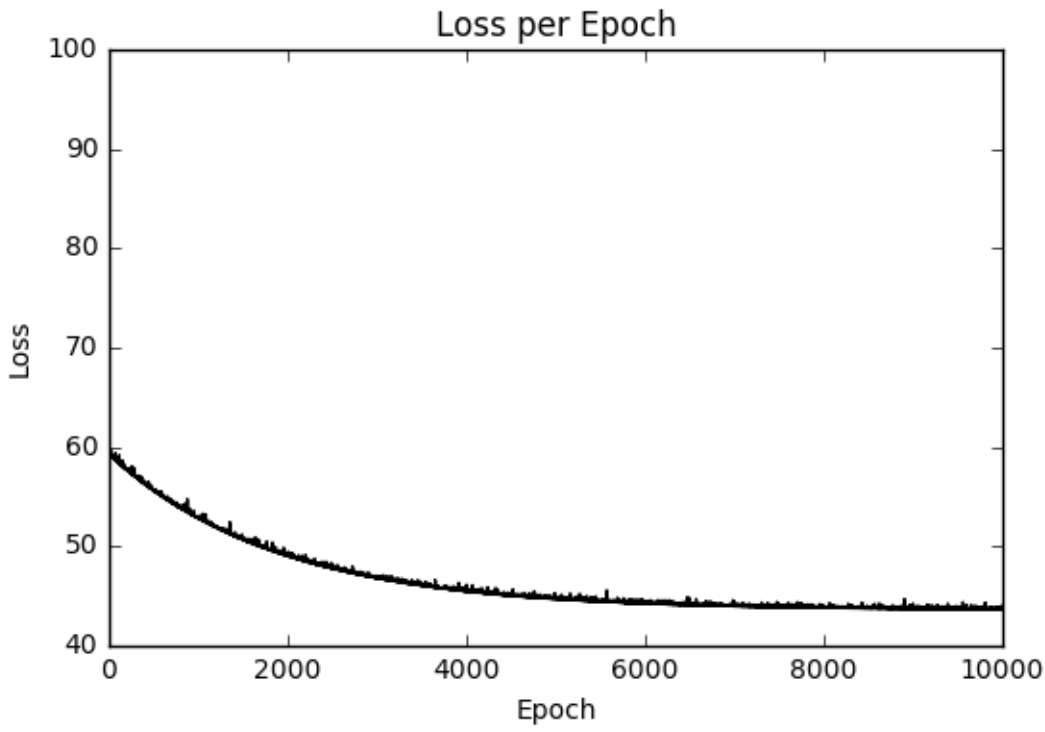


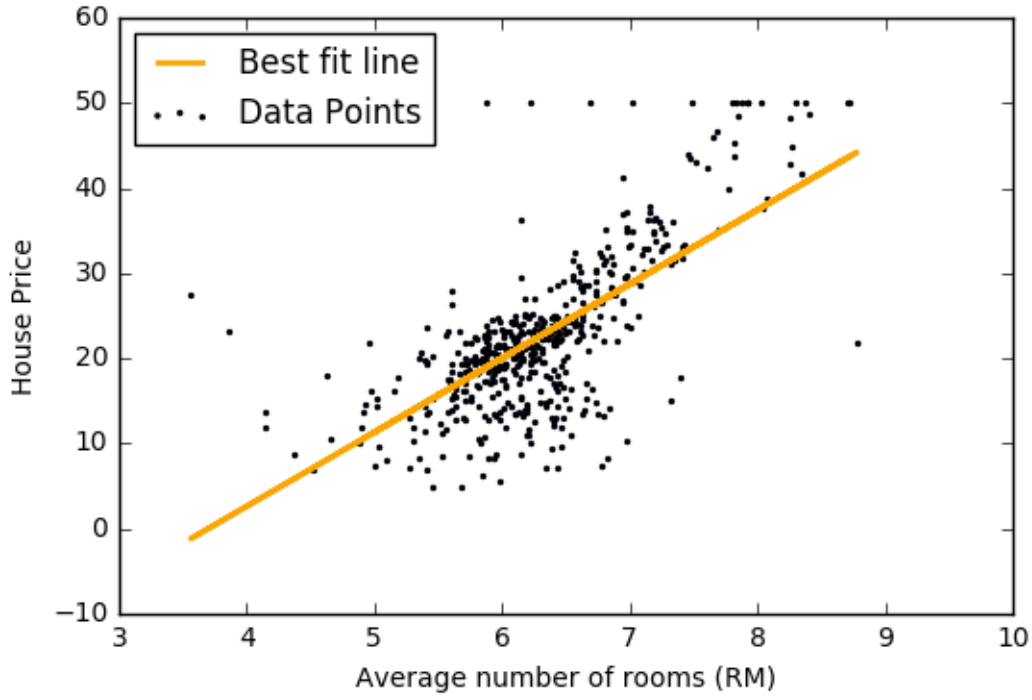



Coefficient weight
6
4
2
0
−2
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
Alpha

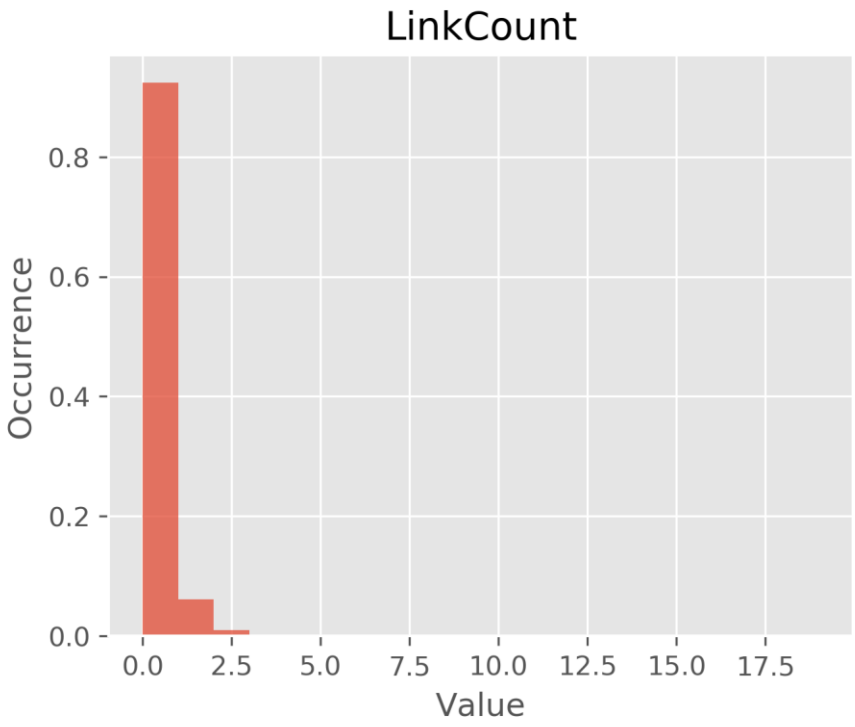

Dataset
1
2
3
4
5
Fold 1
Test
Train
Train
Train
Train
Break into 5 subfolds

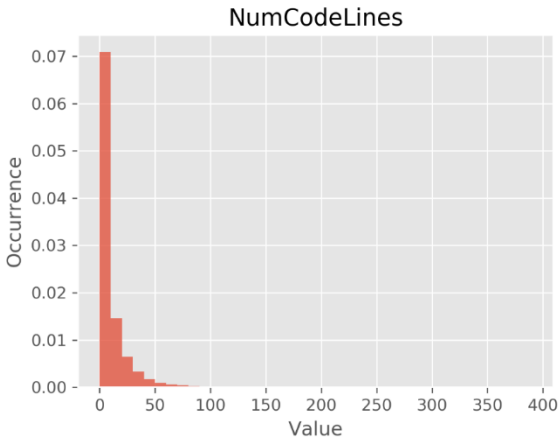

Predicted value
−1
−2
−3
−4
−5
−8
−7
−6
−5
−4
−3
−2
−1
Actual value

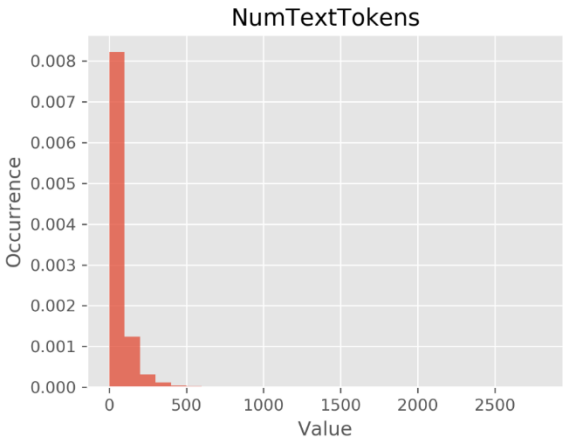

Loss per Epoch
Loss
Epoch
100
90
80
70
60
50
40
0
2000
4000
6000
8000
10000

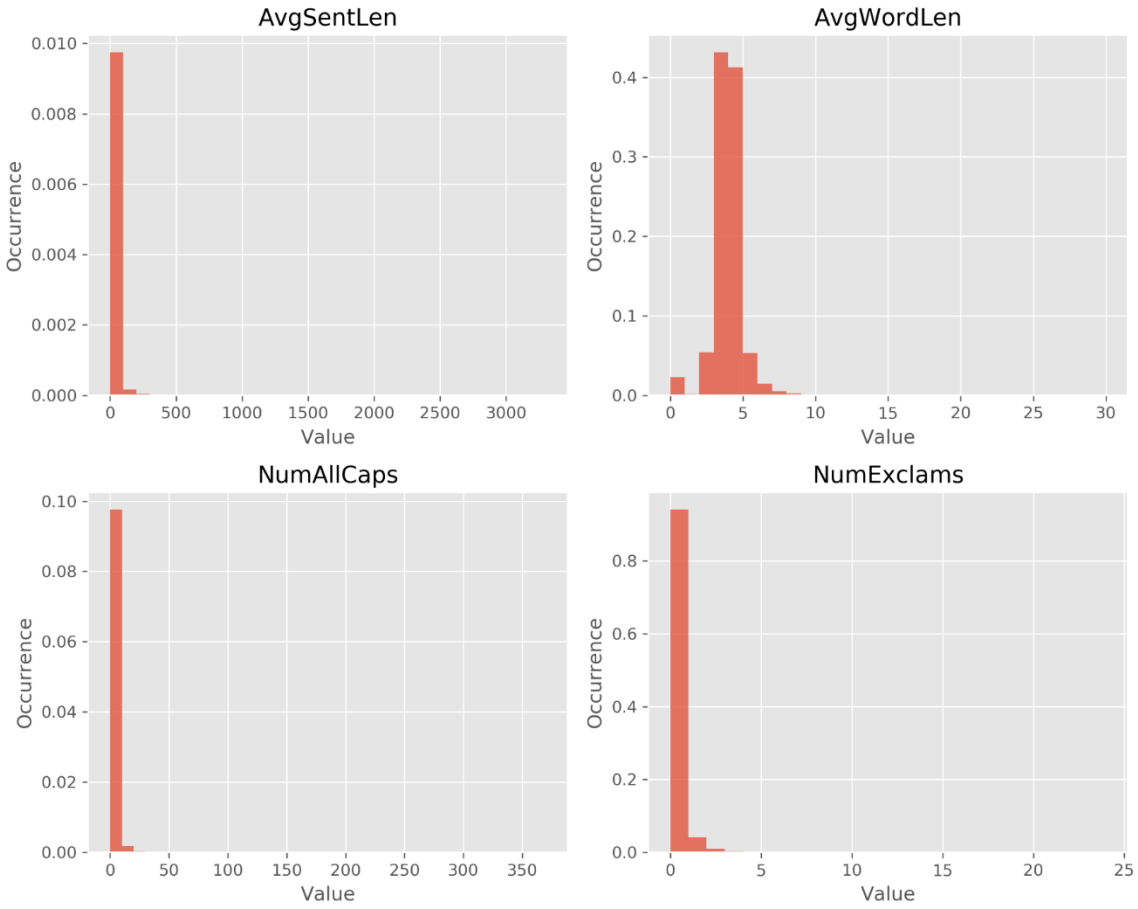

Best fit line
Data Points
House Price
Average number of rooms (RM)
60
50
40
30
20
10
0
−10
3
4
5
6
7
8
9
10

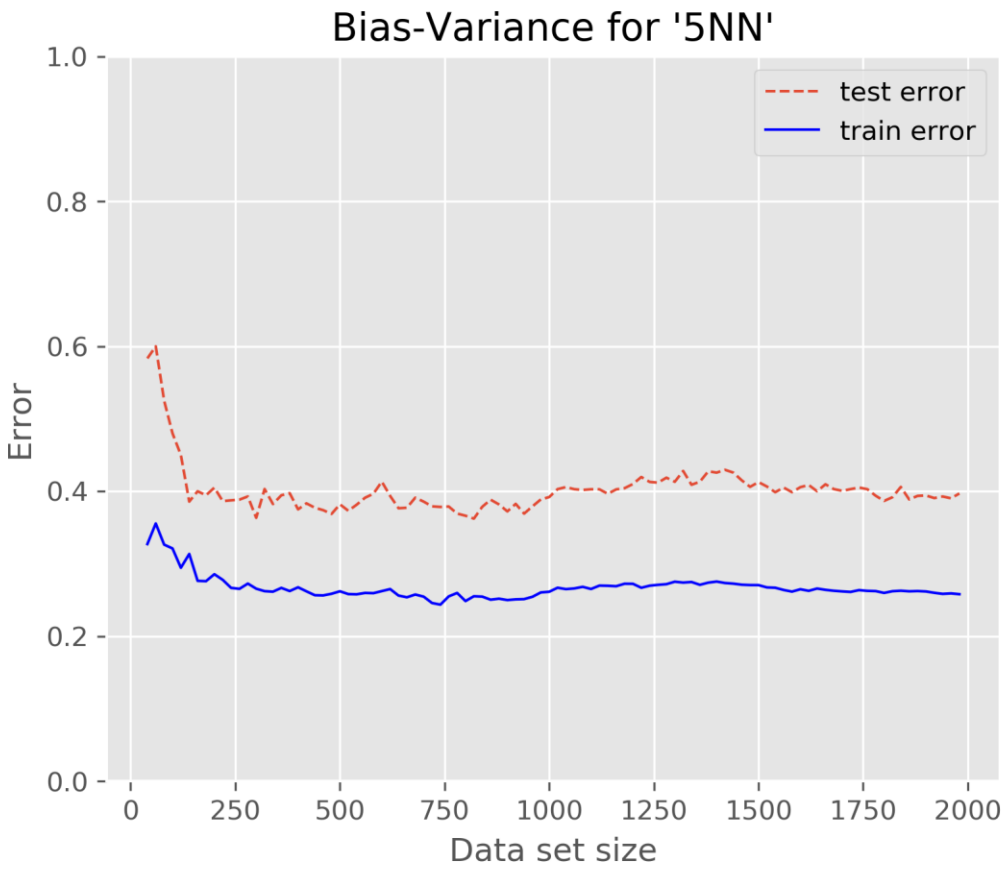
Chapter 4: Classification I - Detecting Poor Answers

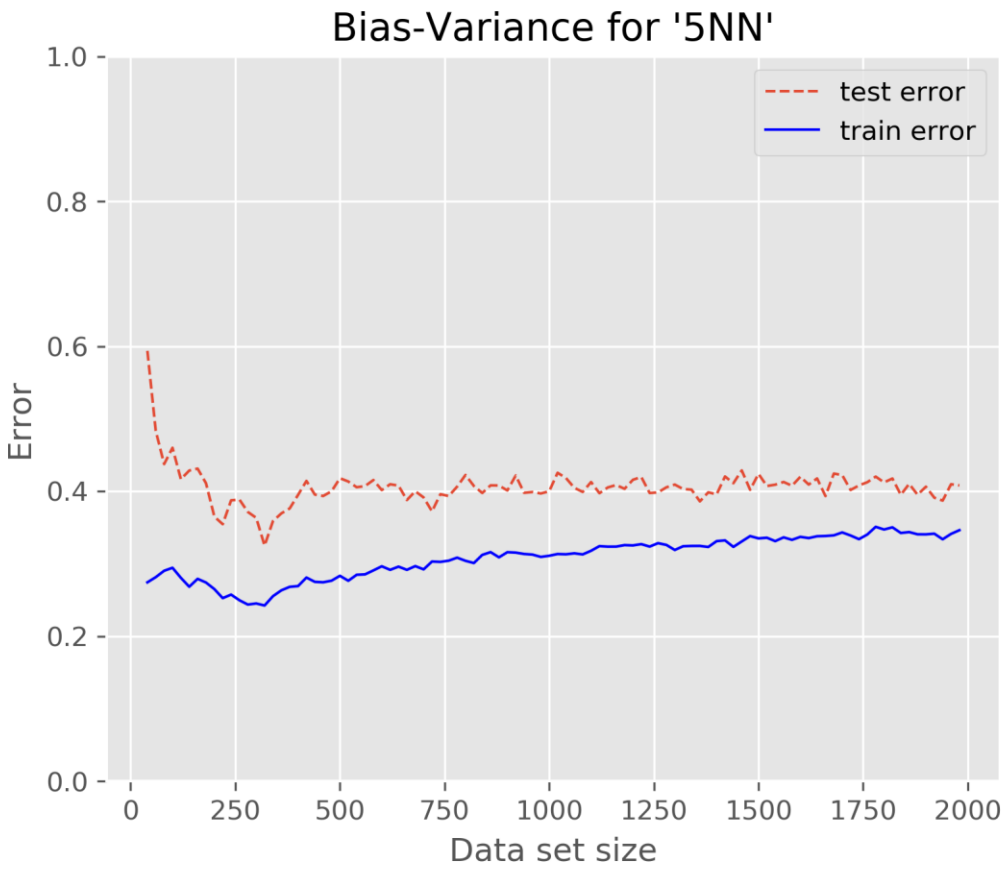


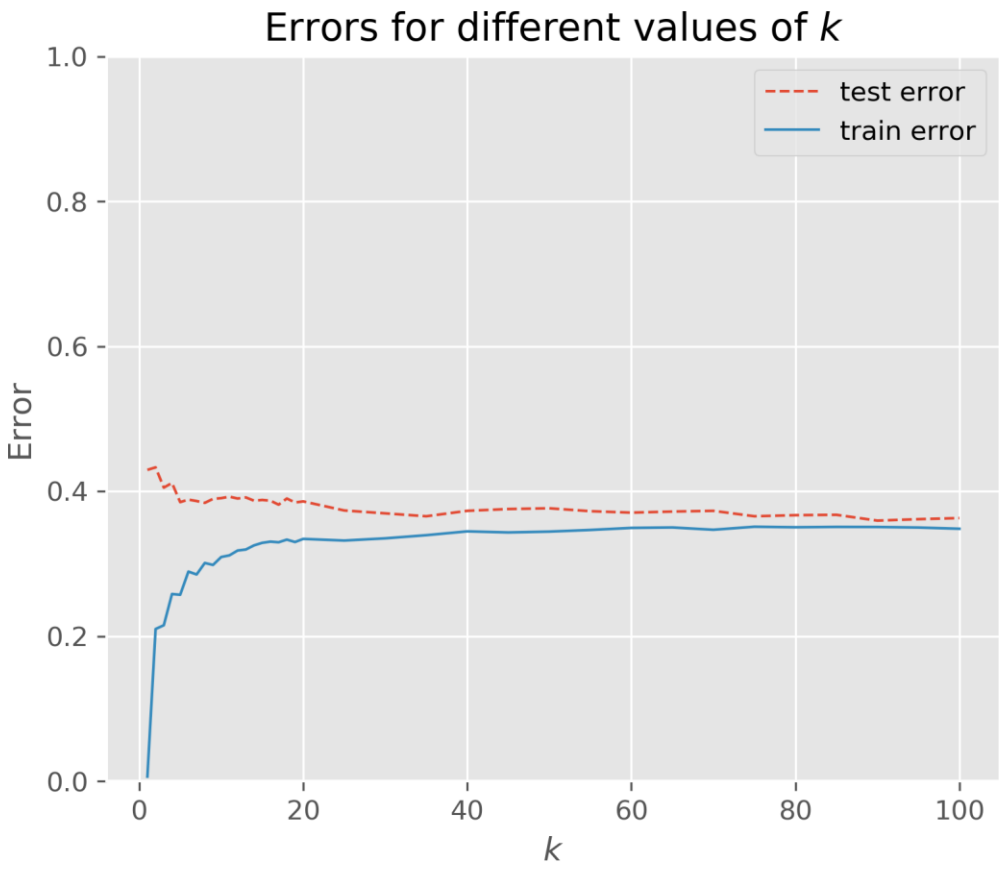


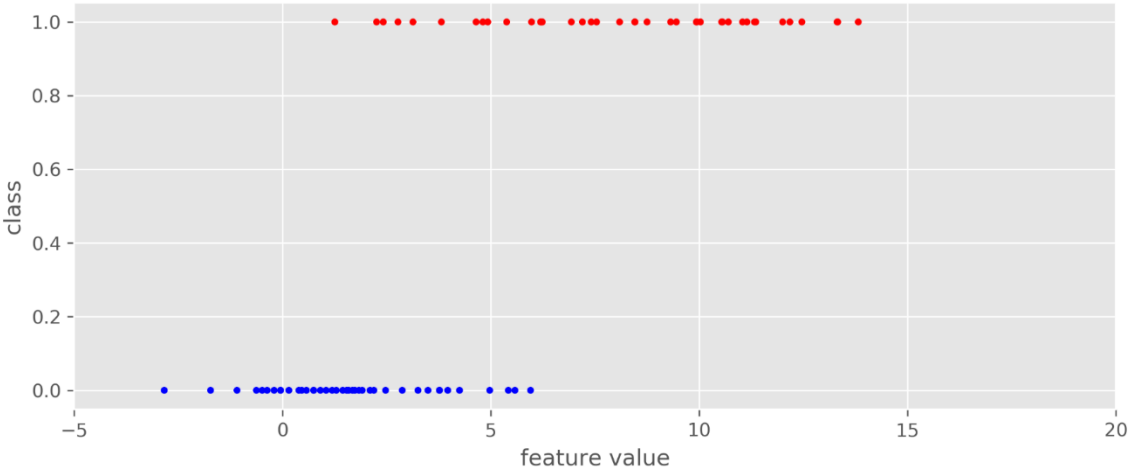


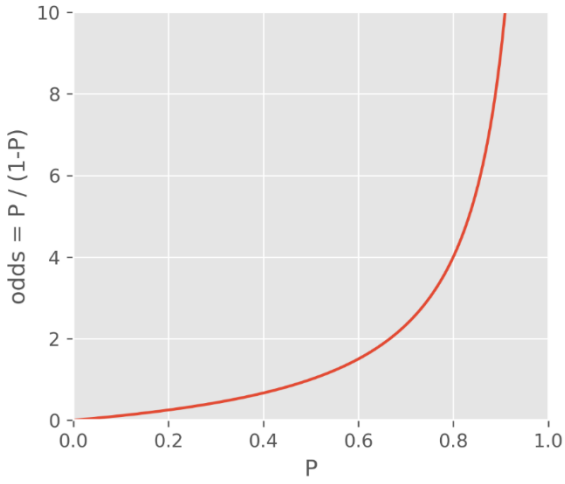

AvgSentLen
Occurrence
Value
0.010
0.008
0.006
0.004
0.002
0.000
0
500
1000
1500
2000
2500
3000
AvgWordLen
Occurrence
Value
0.4
0.3
0.2
0.1
0.0
0
5
10
15
20
25
30
NumAllCaps
Occurrence
Value
0.10
0.08
0.06
0.04
0.02
0.00
0
50
100
150
200
250
300
350
NumExclams
Occurrence
Value
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25

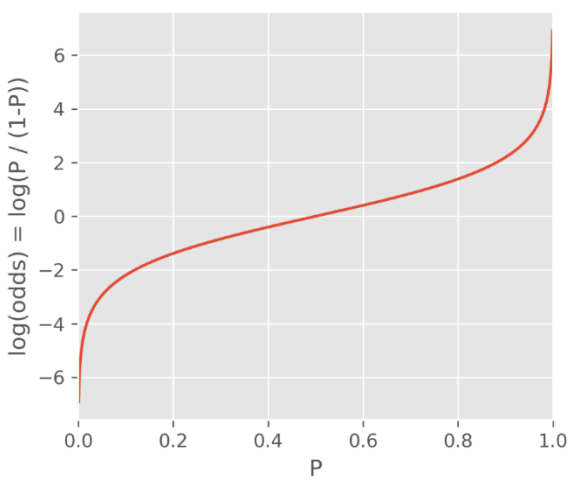

Bias-Variance for '5NN'
test error
train error
Error
Data set size
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000

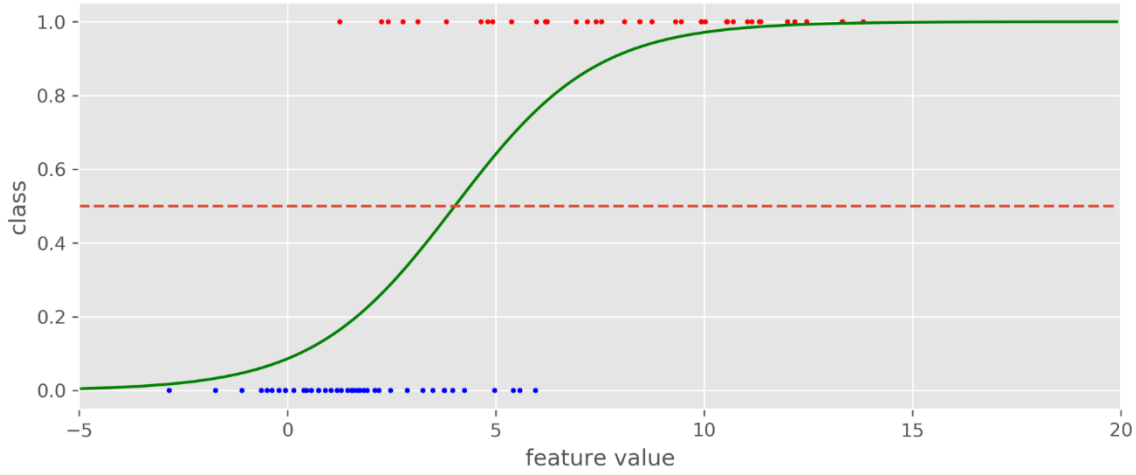

Bias-Variance for '5NN'
test error
train error
Error
Data set size
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000

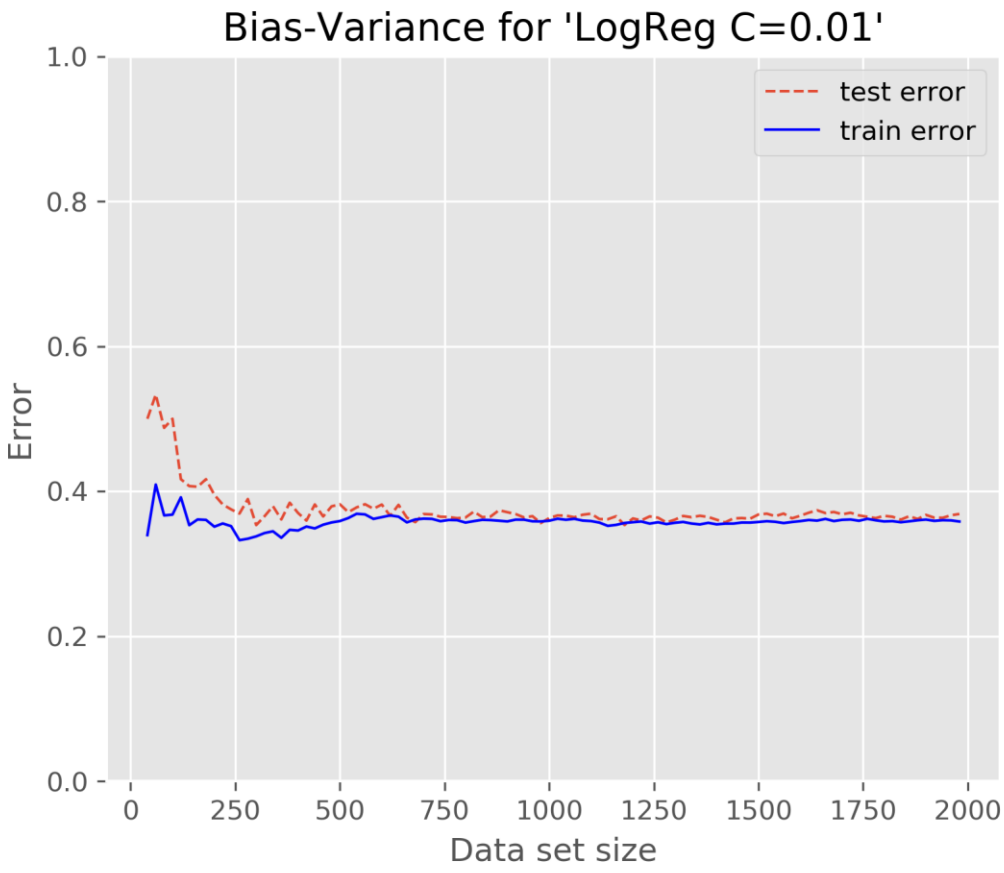

Errors for different values of k
test error
train error
Error
k
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100

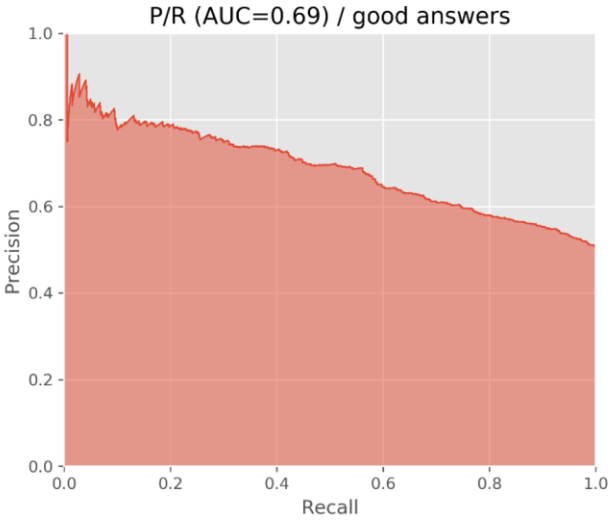

class
feature value
1.0
0.8
0.6
0.4
0.2
0.0
−5
0
5
10
15
20

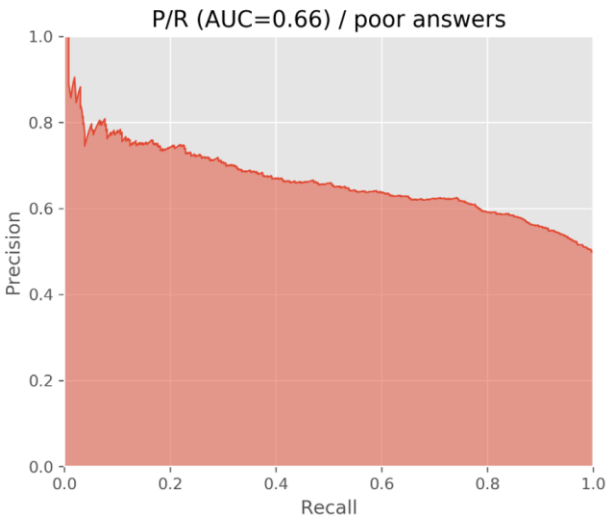

odds = P / (1-P)
P
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10

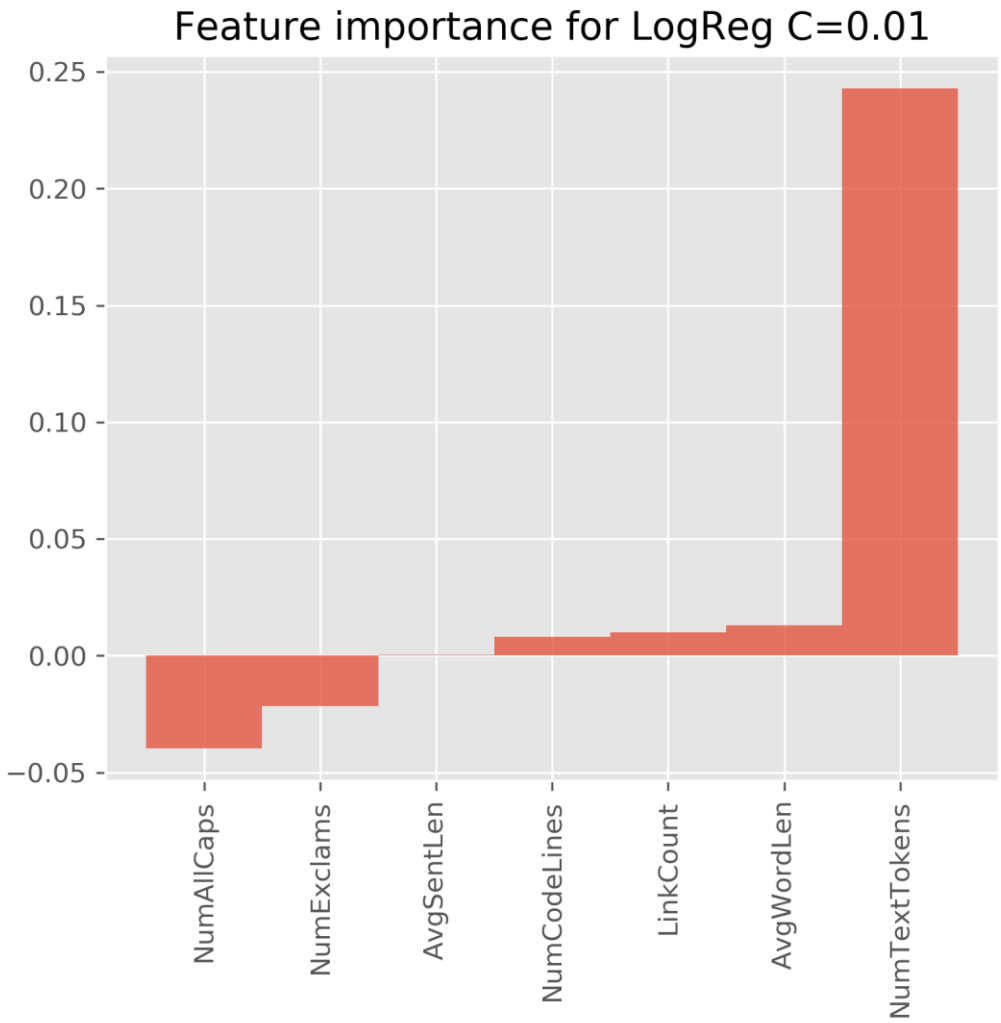

log(odds) = log(P / (1-P))
P
0.0
0.2
0.4
0.6
0.8
1.0
−6
−4
−2
0
2
4
6

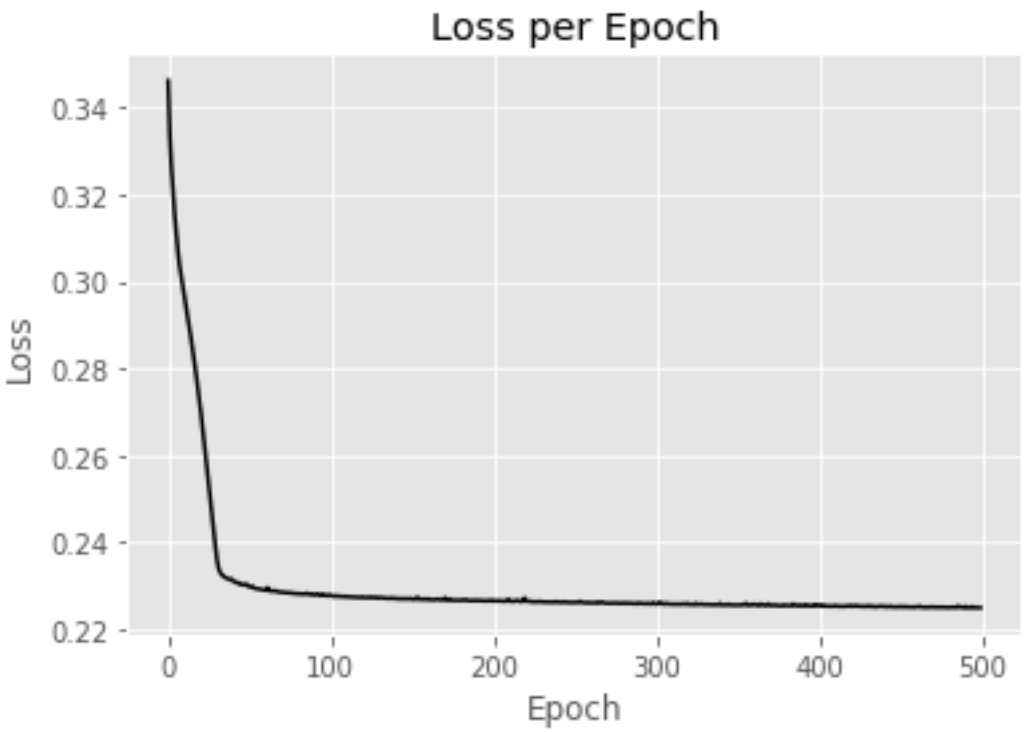

class
feature value
0.0
0.2
0.4
0.6
0.8
1.0
−5
0
5
10
15
20

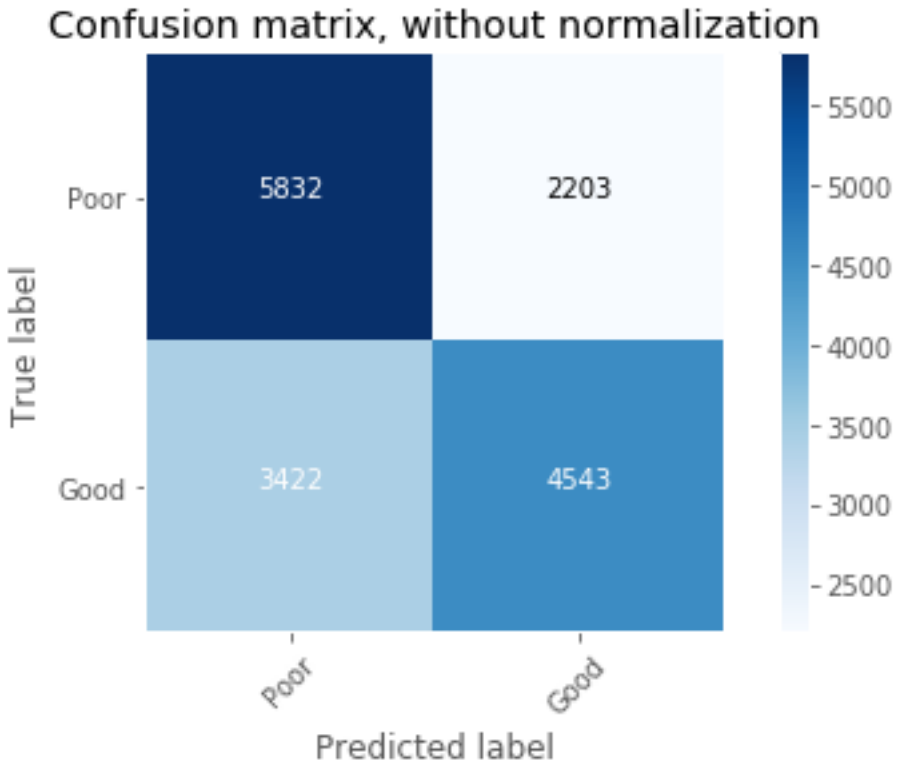

Bias-Variance for 'LogReg C=0.01'
test error
train error
Error
Data set size
1.0
0.8
0.6
0.4
0.2
0.0
0
250
500
750
1000
1250
1500
1750
2000

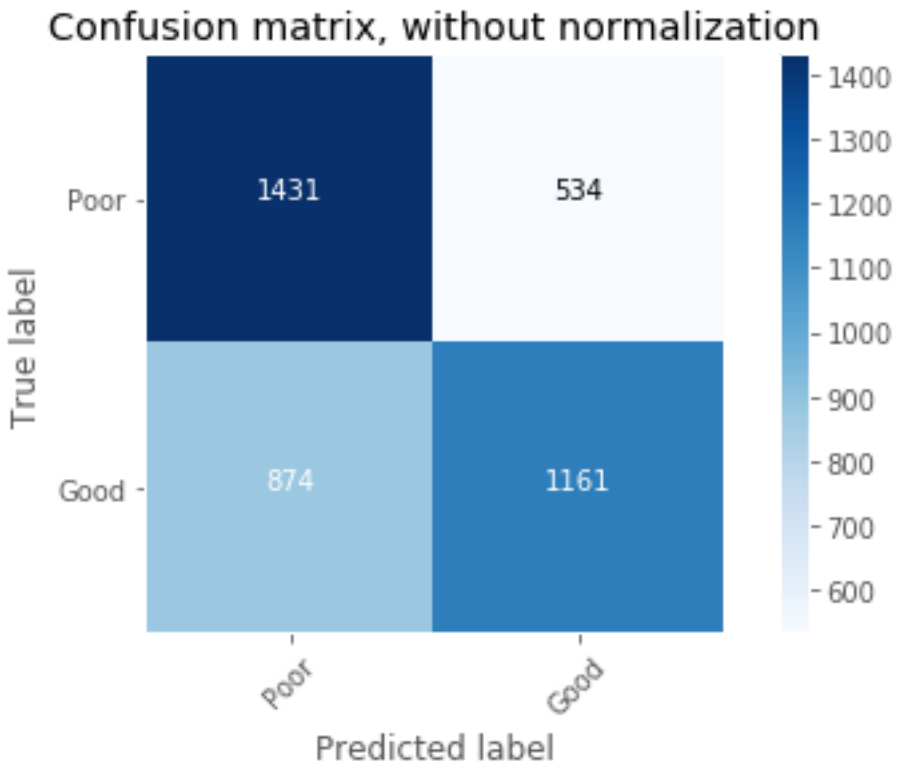

P/R (AUC=0.69) / good answers
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


P/R (AUC=0.66) / poor answers
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

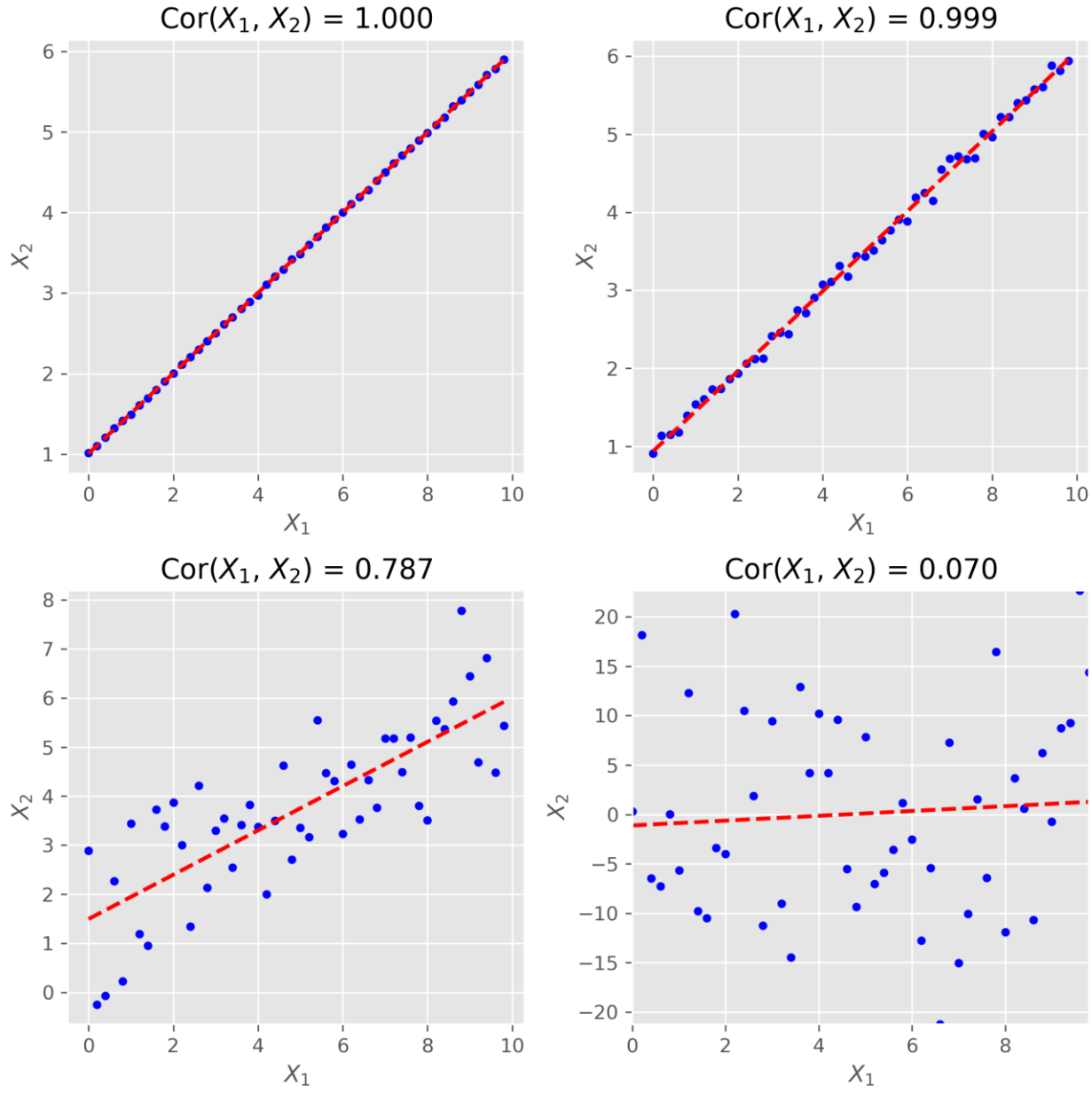

Feature importance for LogReg C=0.01
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
NumAllCaps
NumExclams
AvgSentLen
NumCodeLines
LinkCount
AvgWordLen
NumTextTokens

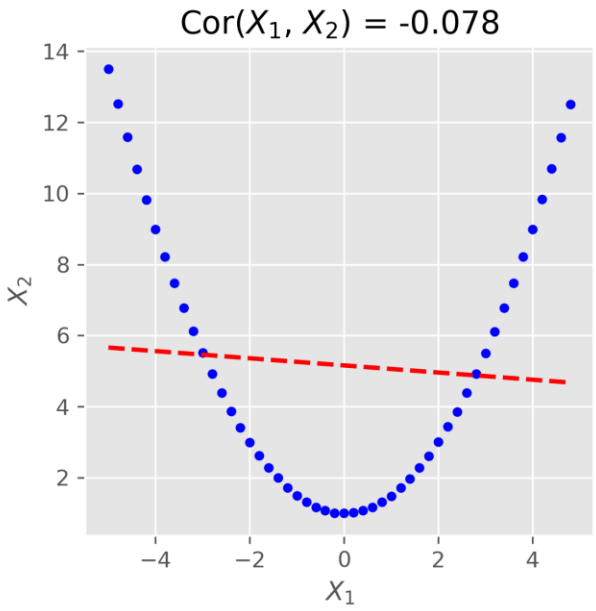

Loss per Epoch
Loss
0.34
0.32
0.30
0.28
0.26
0.24
0.22
0
100
200
300
400
500
Epoch

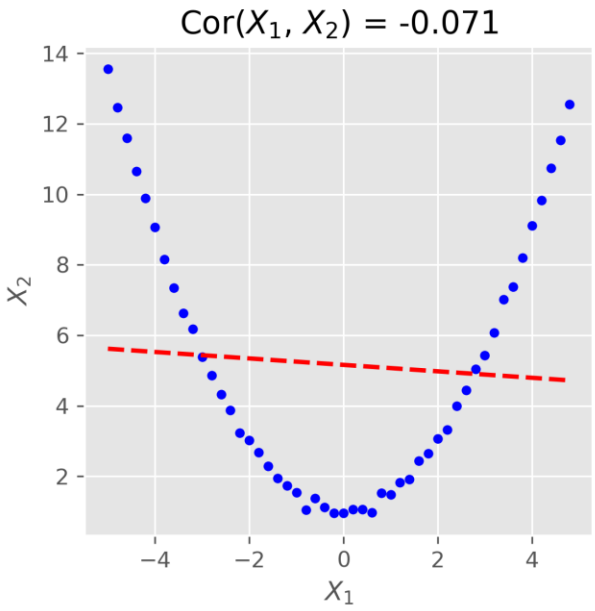

Confusion matrix, without normalization
True label
Poor
Good
5832
2203
3422
4543
5500
5000
4500
4000
3500
3000
2500
Poor
Good
Predicted label

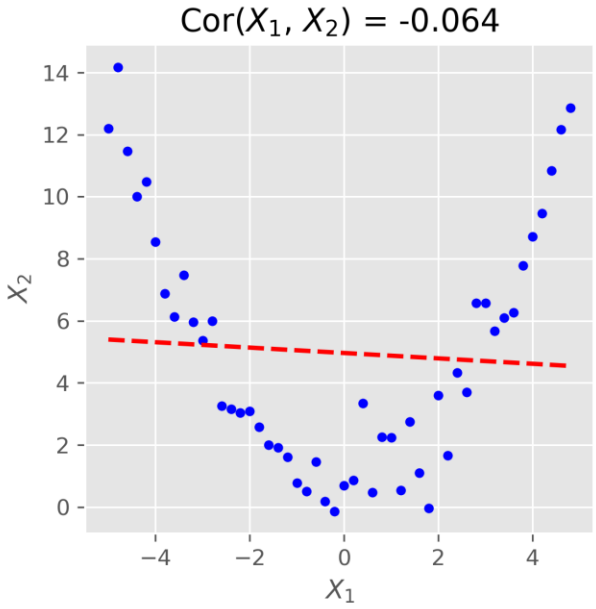

Confusion matrix, without normalization
1431
534
874
1161
Poor
Good
True label
Poor
Good
Predicted label
1400
1300
1200
1100
1000
900
800
700
600

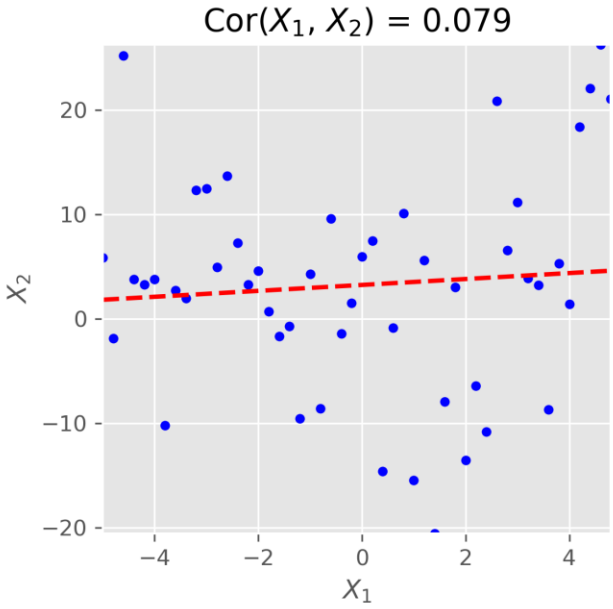
Chapter 5: Dimensionality Reduction

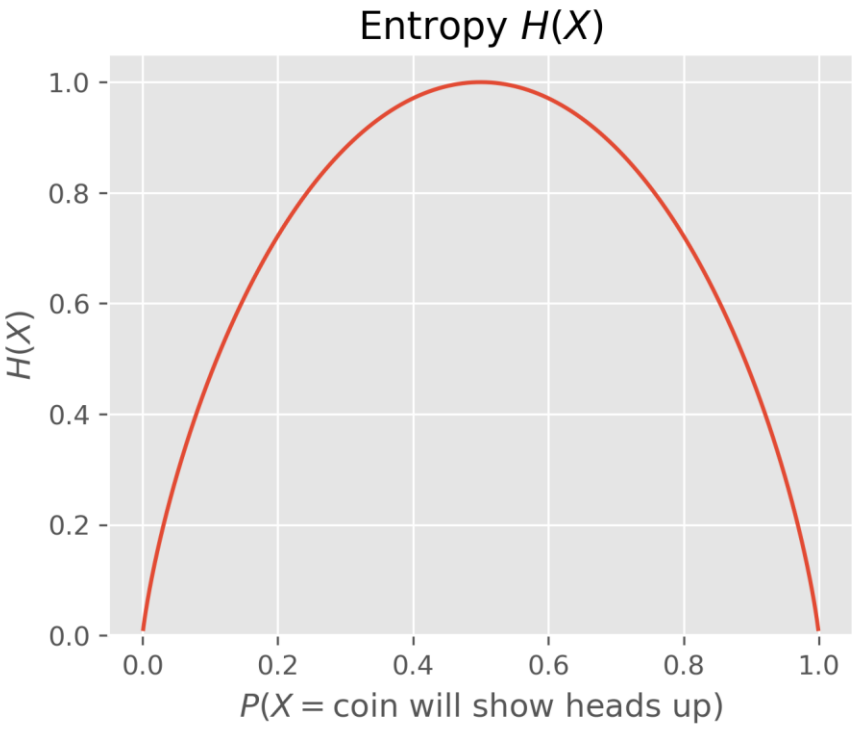


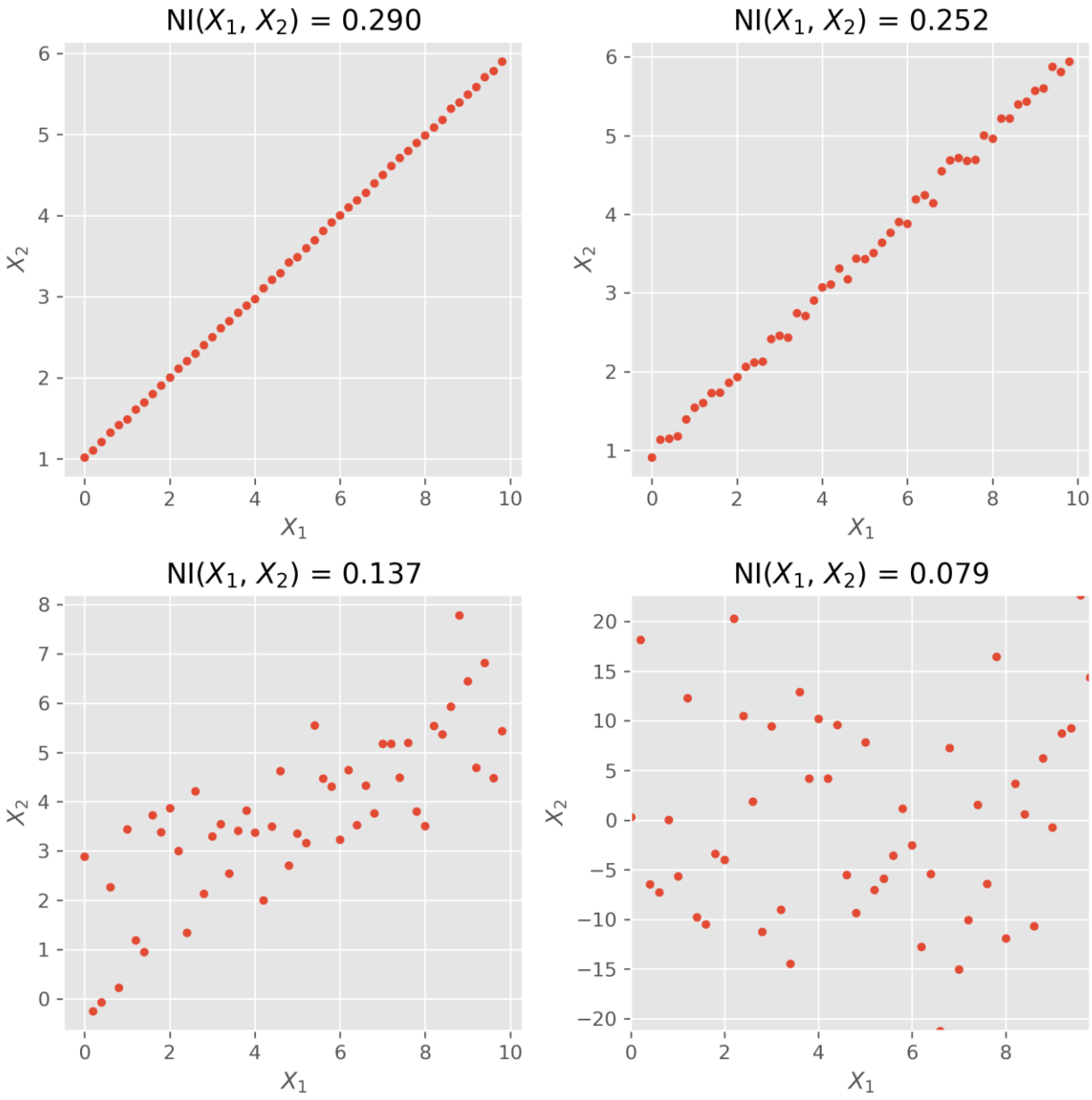

Cor(X_1, X_2) = -0.078
X_2
X_1

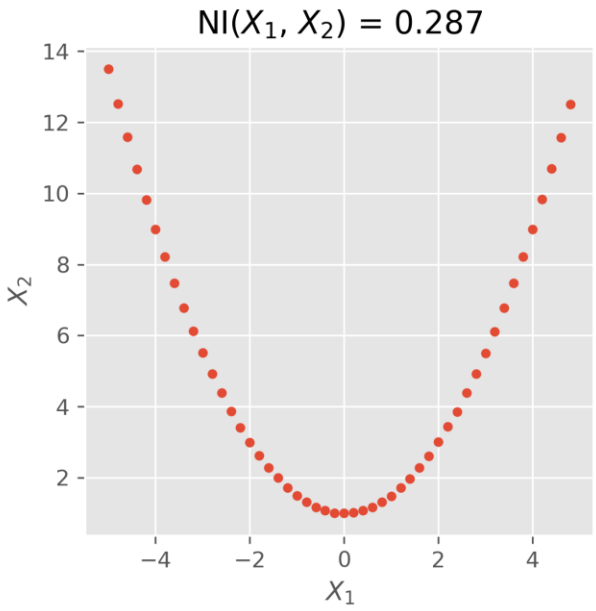

Cor(X_1, X_2) = -0.071
X_2
X_1

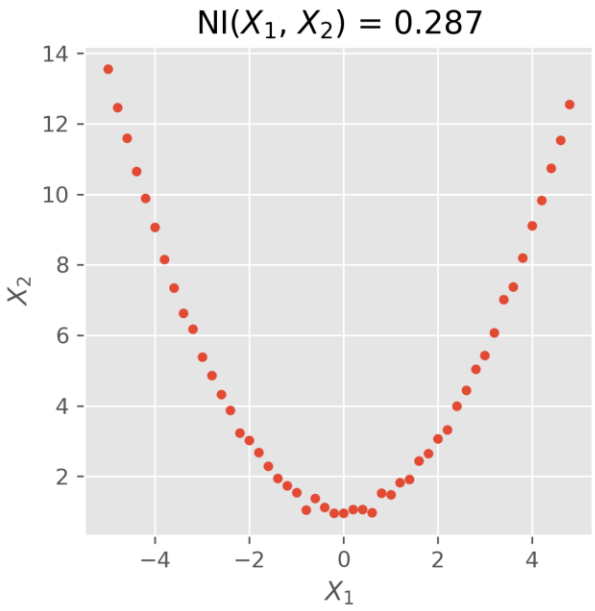

Cor(X_1, X_2) = -0.064
X_2
X_1

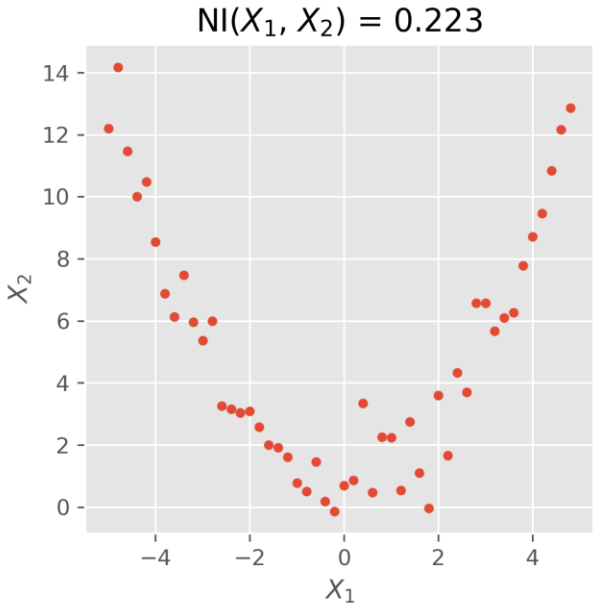

Cor(X_1, X_2) = 0.079
X_2
X_1

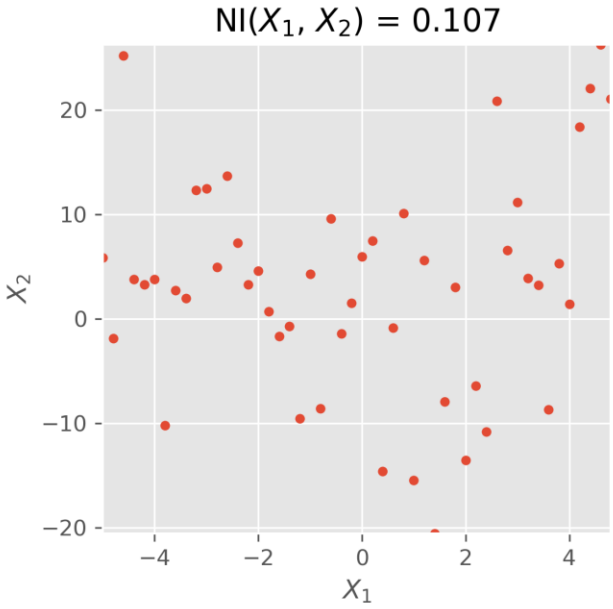

Entropy $H(X)$
1.0
0.8
0.6
0.4
0.2
0.0
$H(X)$
0.0
0.2
0.4
0.6
0.8
1.0
$P(X = \text{coin will show heads up})$

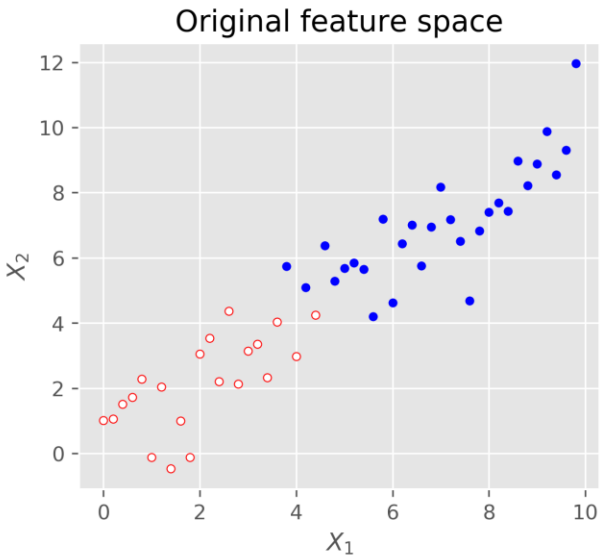

NI(X_1, X_2) = 0.290
NI(X_1, X_2) = 0.252
NI(X_1, X_2) = 0.137
NI(X_1, X_2) = 0.079
X_1
X_2

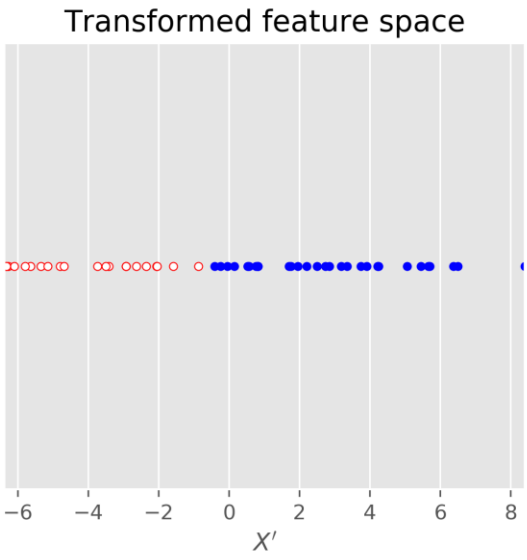

NI(X_1, X_2) = 0.287
X_2
X_1

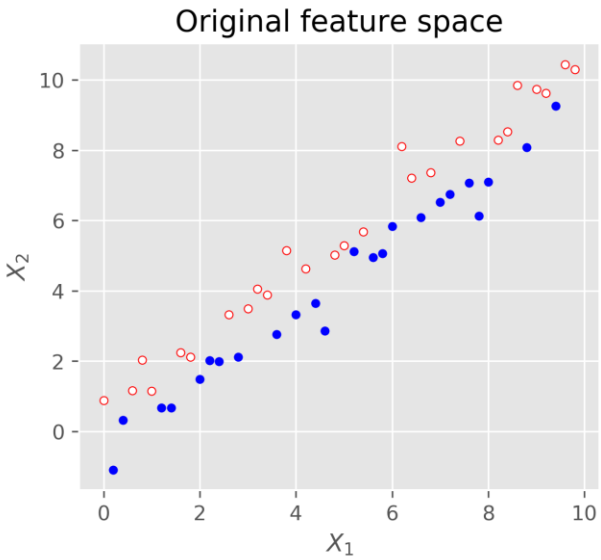

NI(X_1, X_2) = 0.287
X_2
X_1

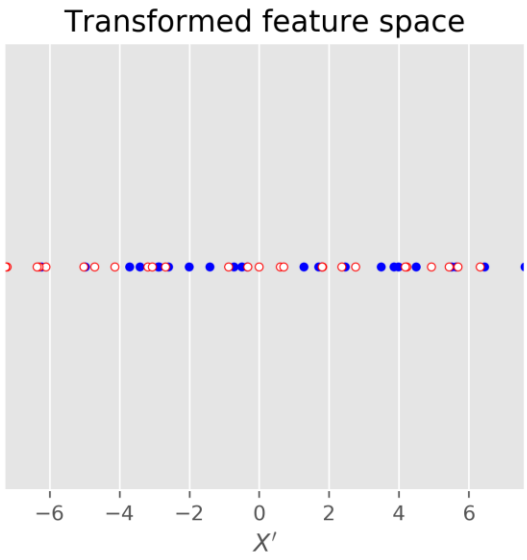

NI(X_1, X_2) = 0.223
X_2
X_1

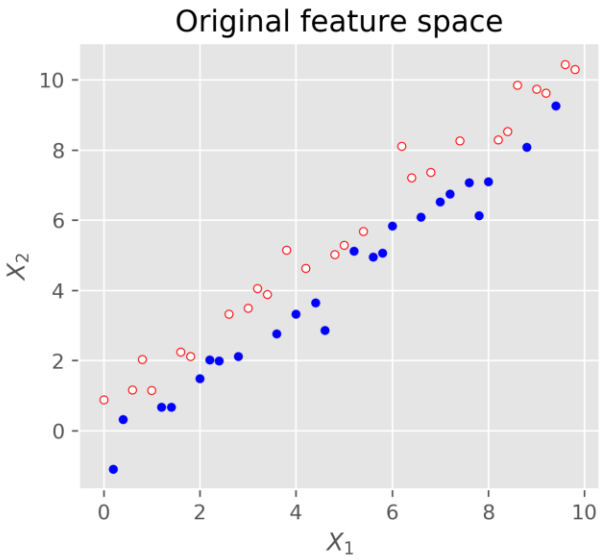

NI(X_1, X_2) = 0.107
X_2
X_1

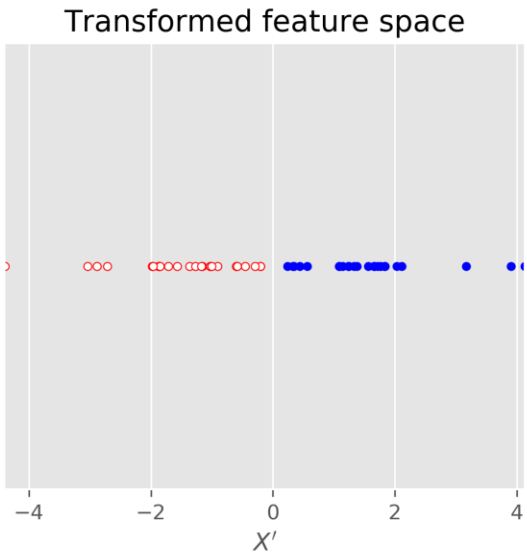

Original feature space
X_2
X_1

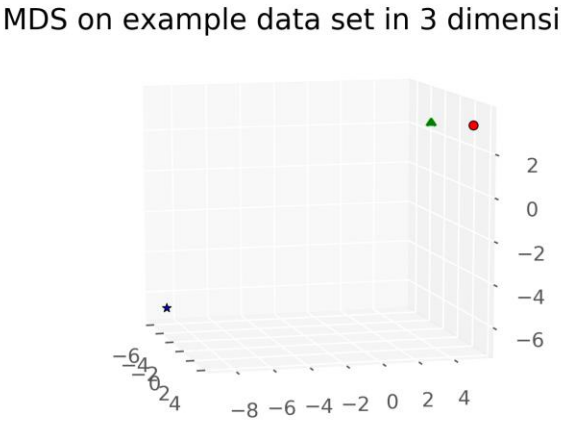

Transformed feature space
X'

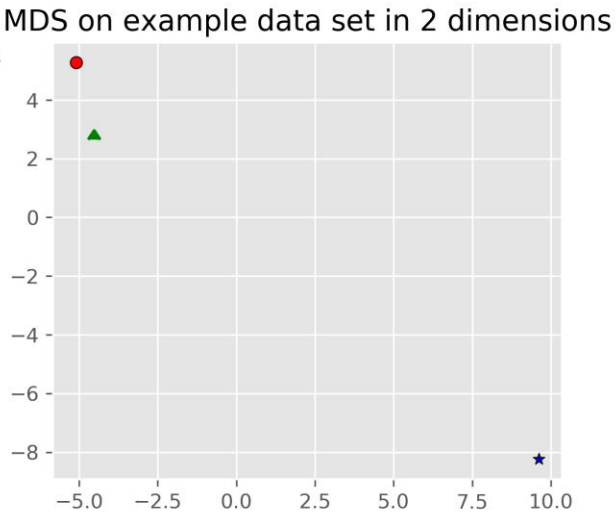

Original feature space
X_2
X_1


Transformed feature space
X'

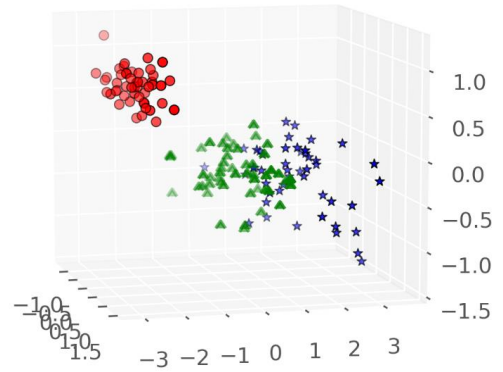

Original feature space
X_1
X_2


Transformed feature space
X'

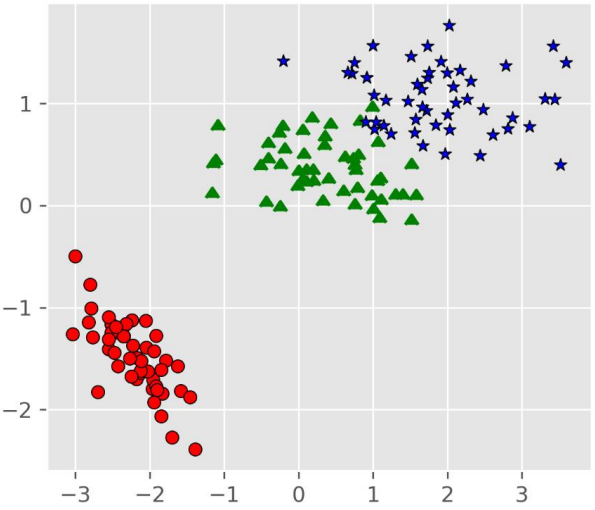

MDS on example data set in 3 dimensions


MDS on example data set in 2 dimensions

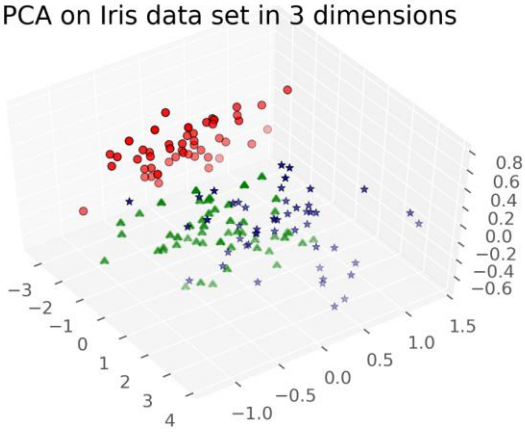
MDS on Iris data set in 3 dimensions



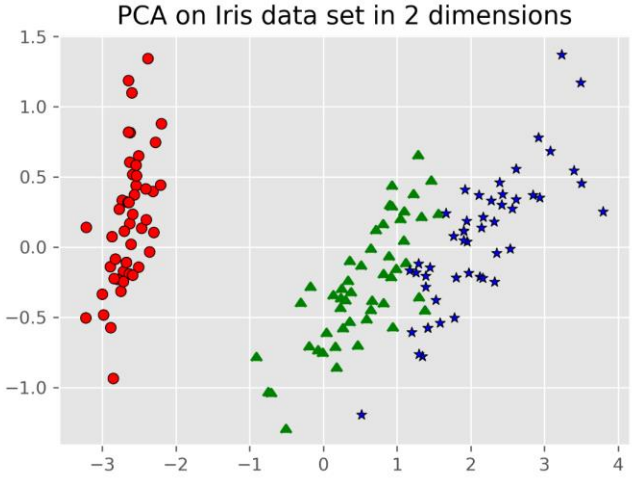
MDS on Iris data set in 2 dimensions

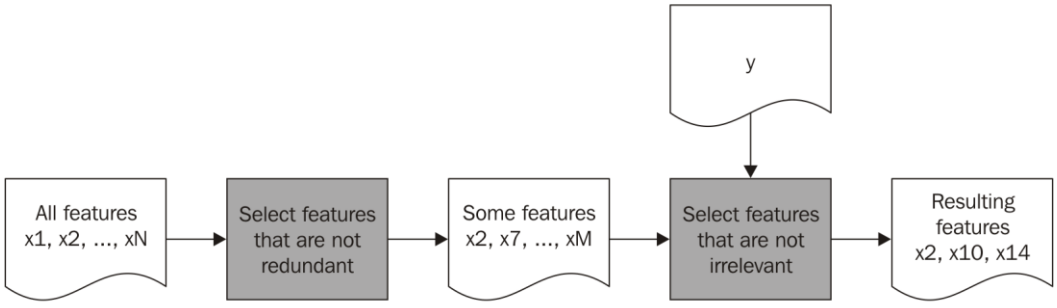


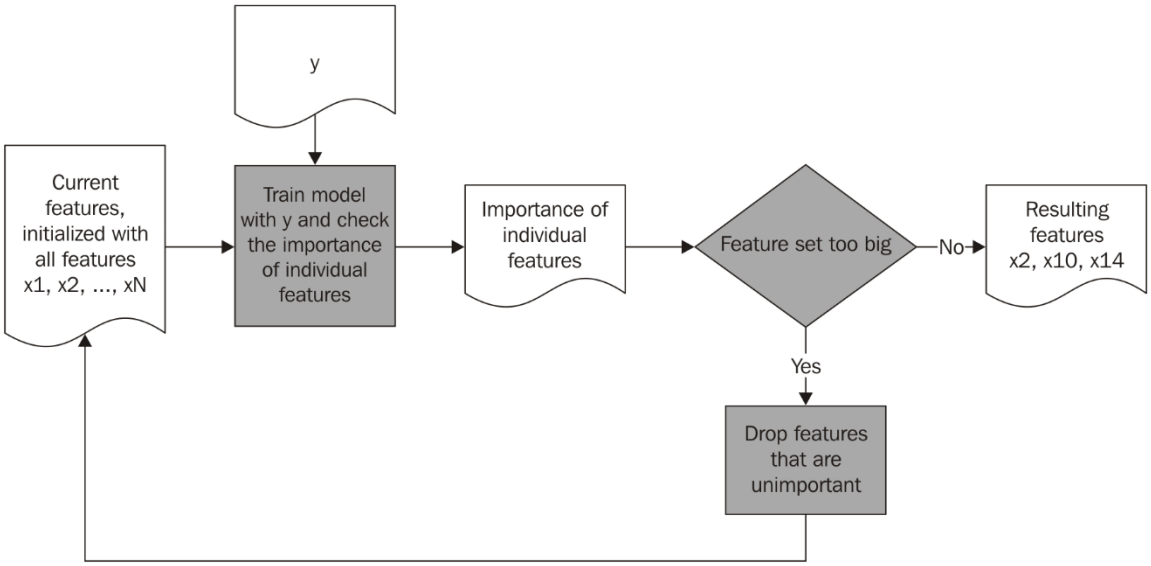
PCA on Iris data set in 3 dimensions

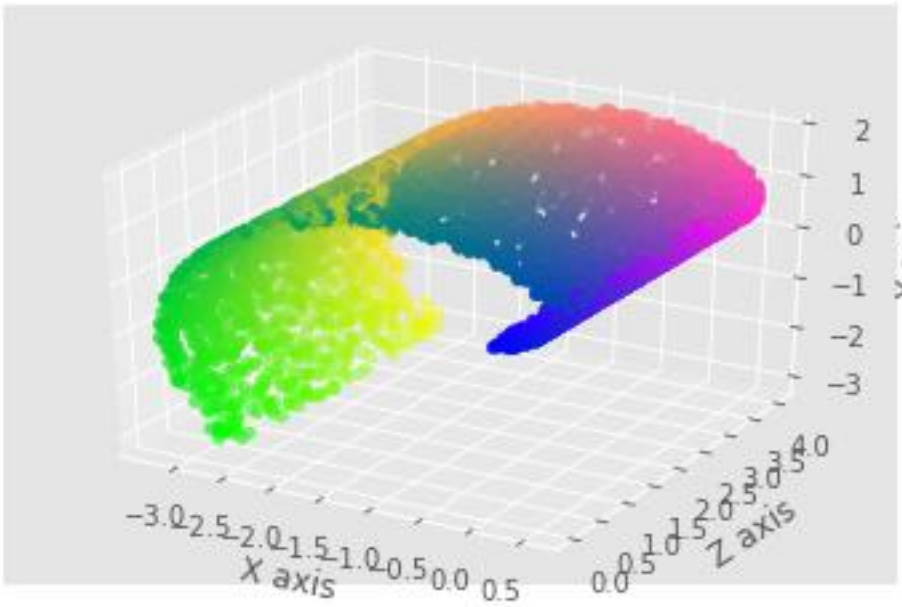


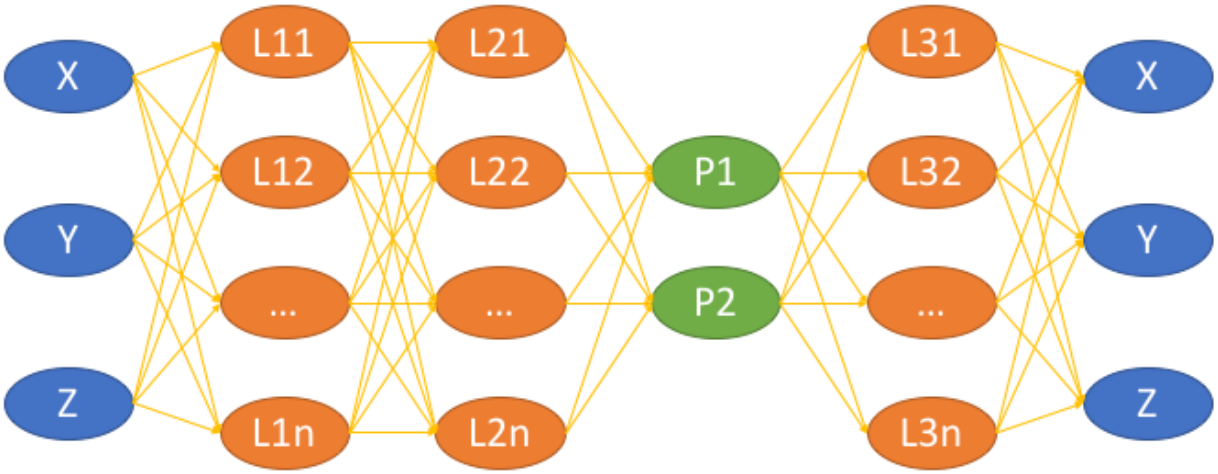
PCA on Iris data set in 2 dimensions

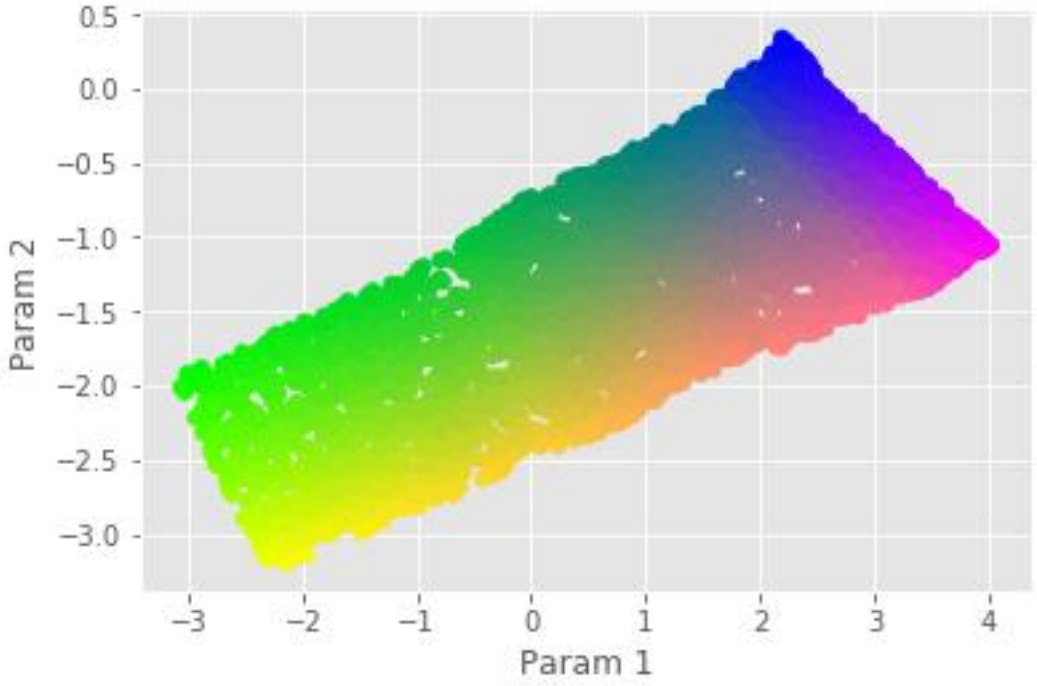


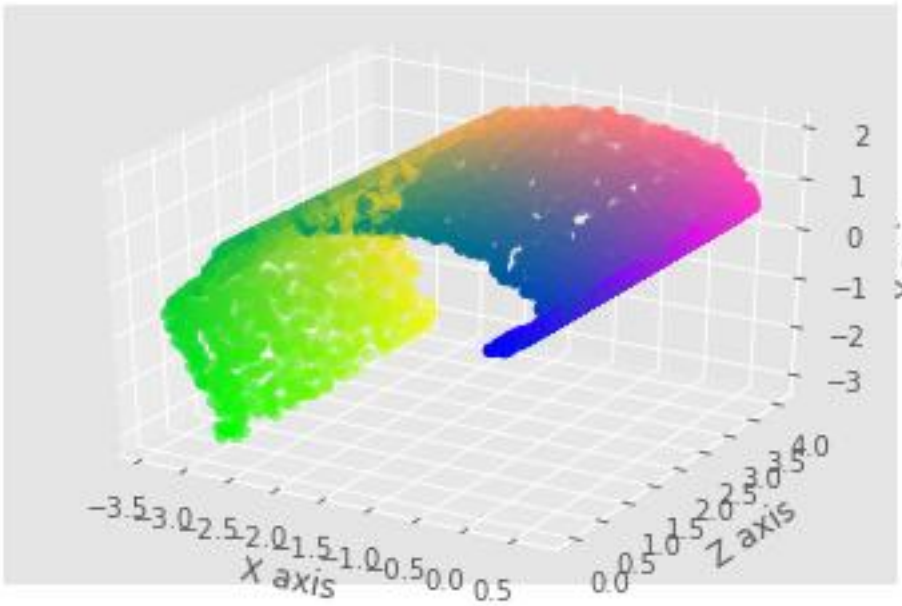



y
Current features, initialized with all features x1, x2, ..., xN
Train model with y and check the importance of individual features
Importance of individual features
Feature set too big
No
Resulting features x2, x10, x14
Yes
Drop features that are unimportant

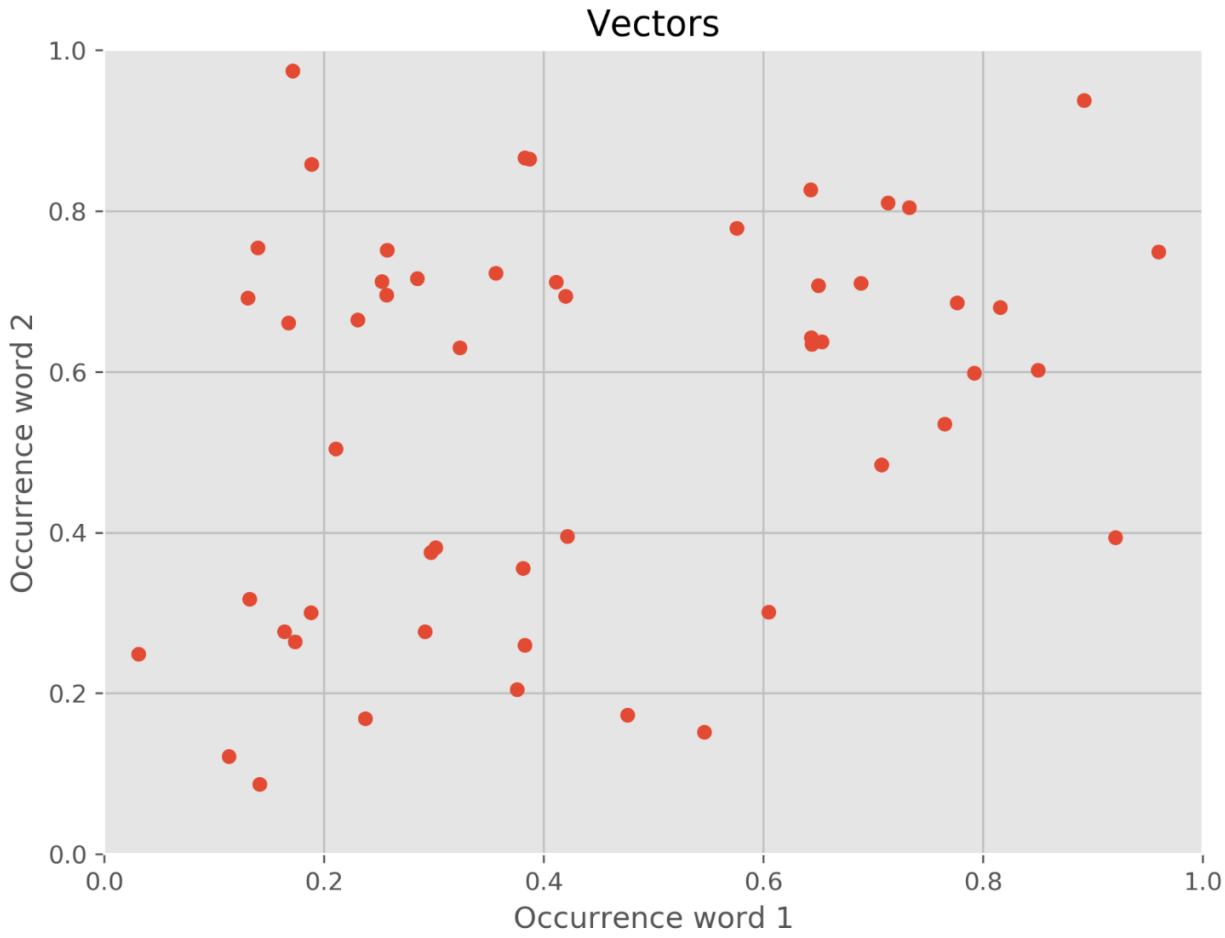

X axis
Z axis
Y axis

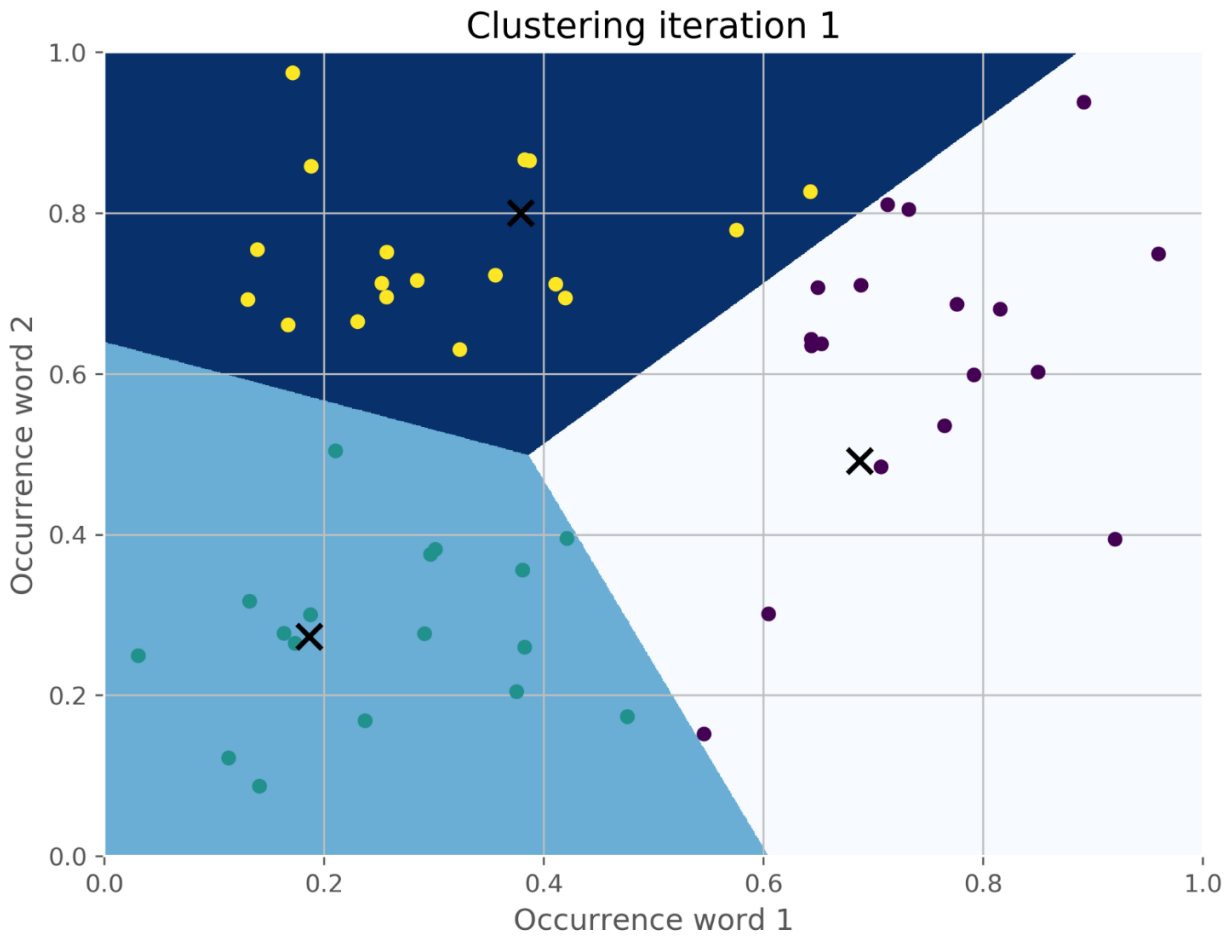

X
Y
Z
L11
L12
...
L1n
L21
L22
...
L2n
P1
P2
L31
L32
...
L3n
X
Y
Z

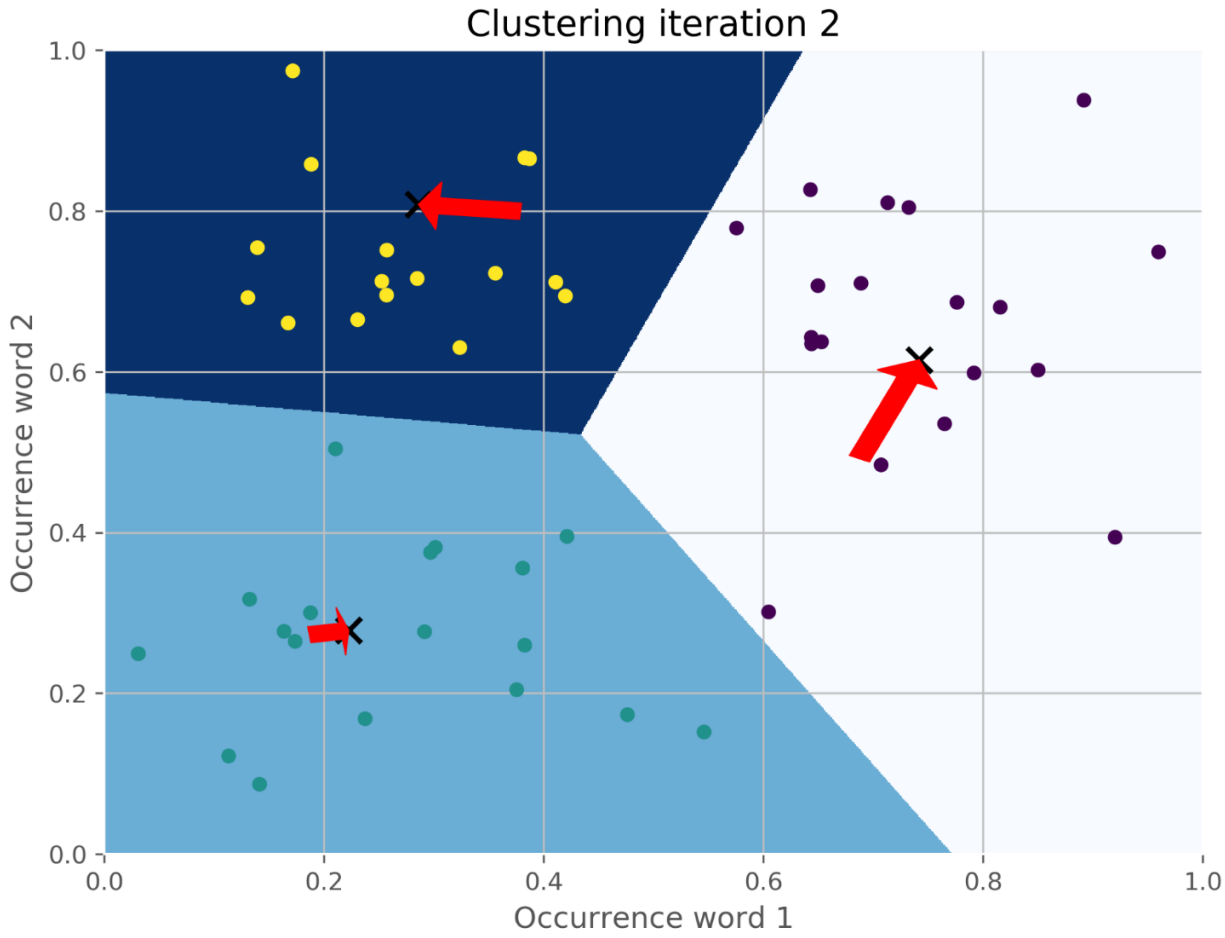

0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3.0
Param 2
−3
−2
−1
0
1
2
3
4
Param 1

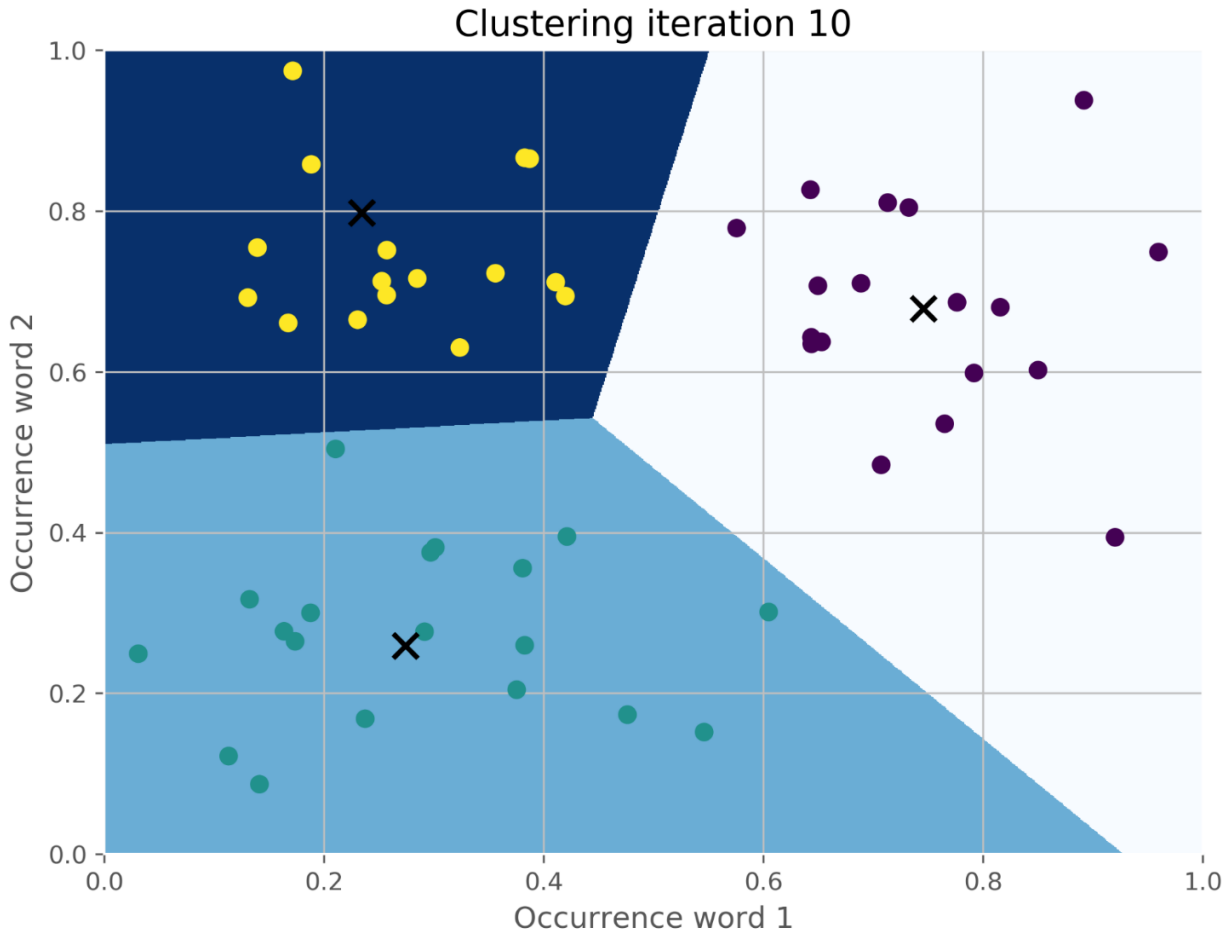

2
1
0
−1
−2
−3
Y axis
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
X axis
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Z axis

Chapter 6: Clustering - Finding Related Posts

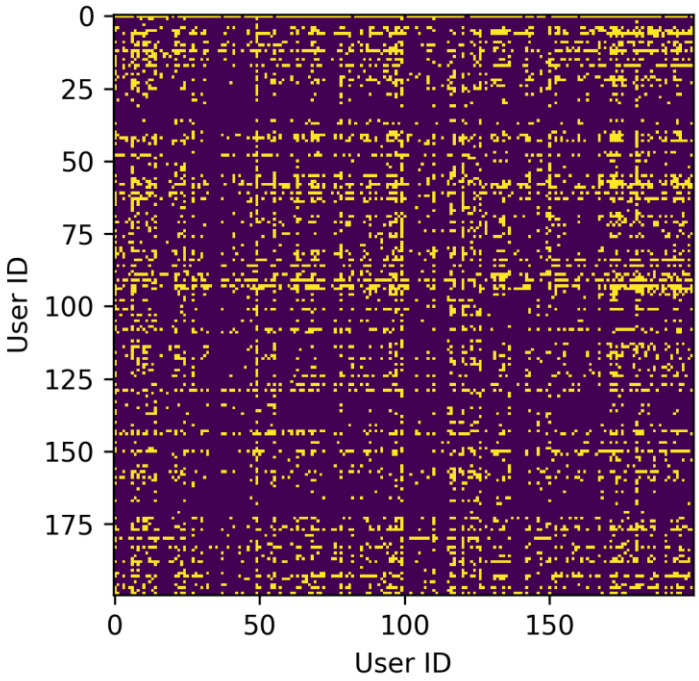


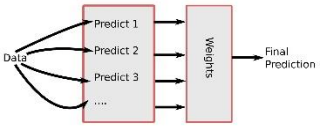

Clustering iteration 1
Occurrence word 2
Occurrence word 1
0.0
0.2
0.4
0.6
0.8
1.0


Clustering iteration 2
Occurrence word 2
Occurrence word 1
0.0
0.2
0.4
0.6
0.8
1.0

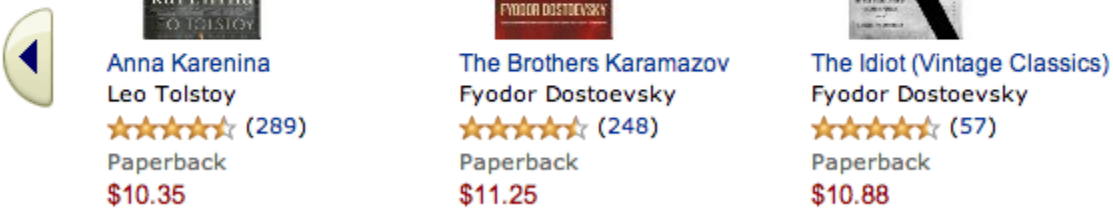

Clustering iteration 10
Occurrence word 2
Occurrence word 1
0.0
0.2
0.4
0.6
0.8
1.0

Chapter 7: Recommendations

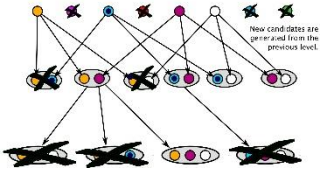




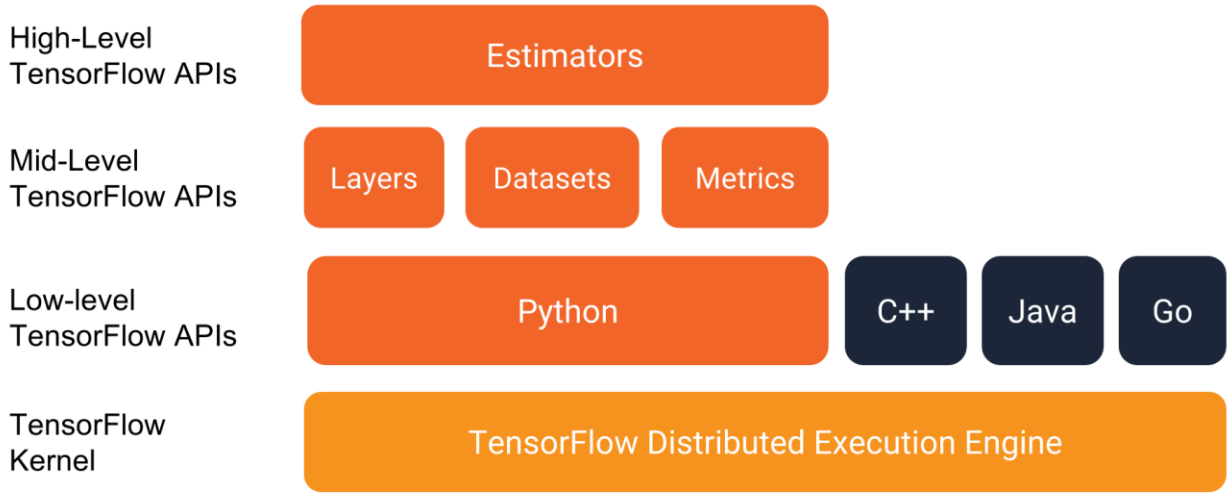
Customers Who Bought This Item Also Bought

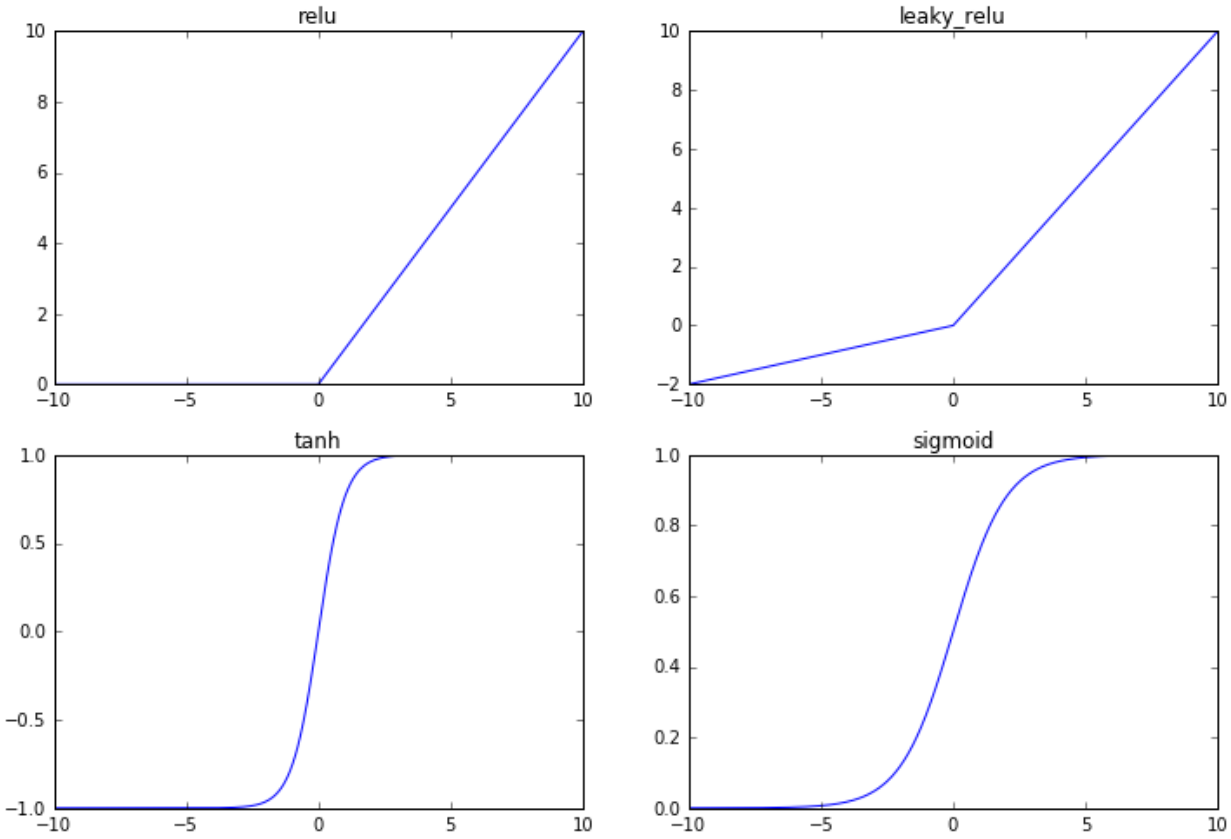


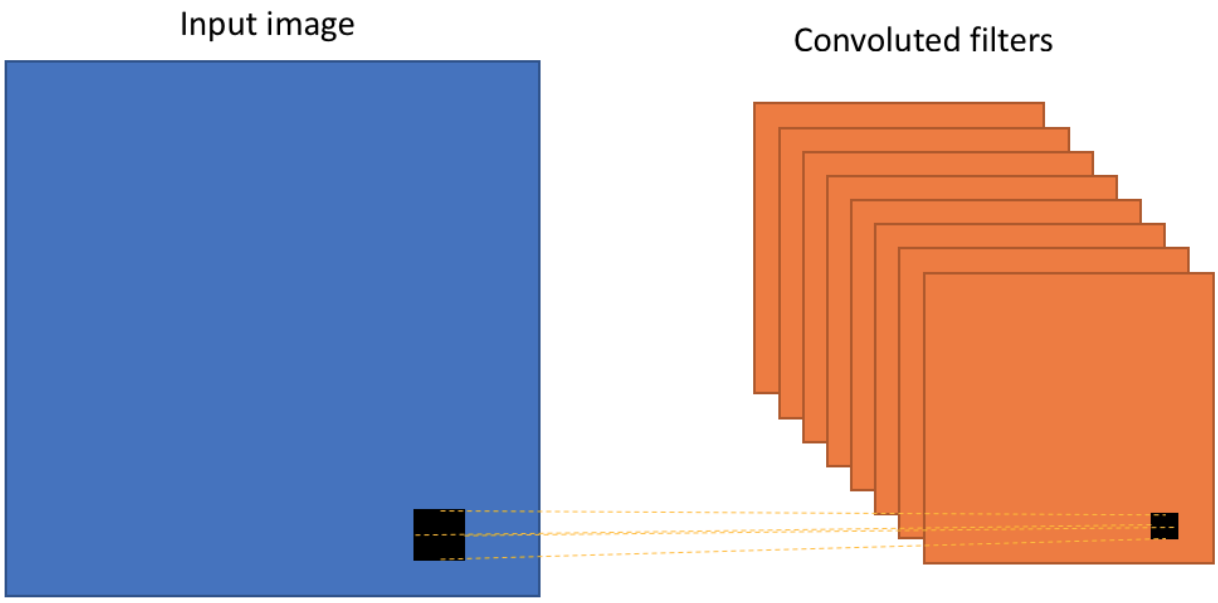
At each level, groups that do not have *minsupport* are filtered out.

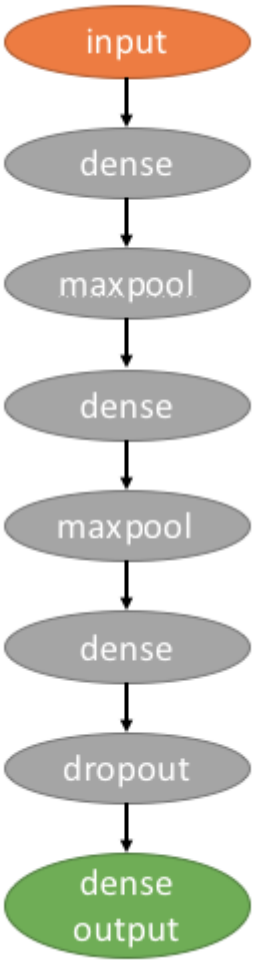


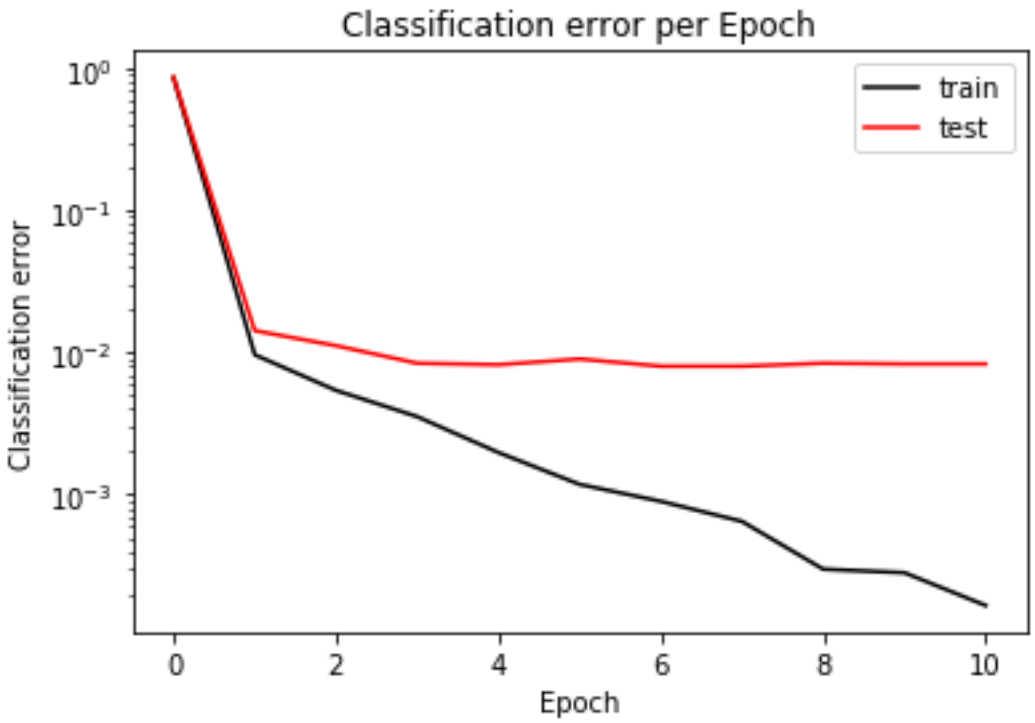
Chapter 8: Artificial Neural Networks and Deep Learning

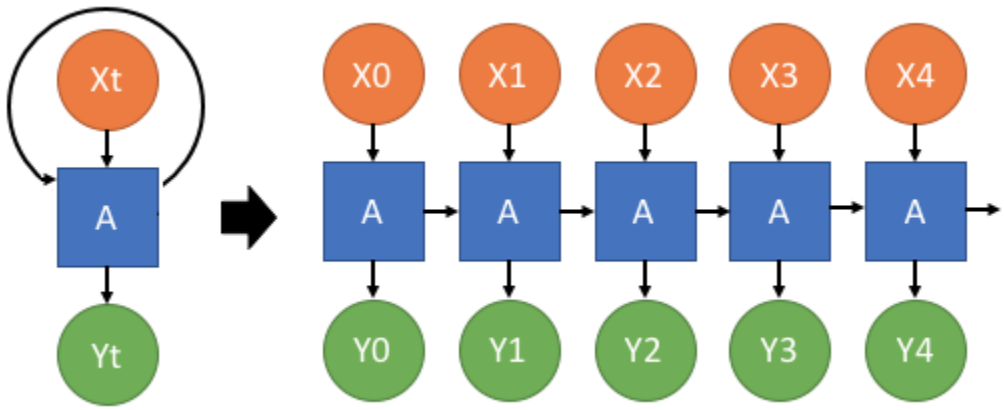


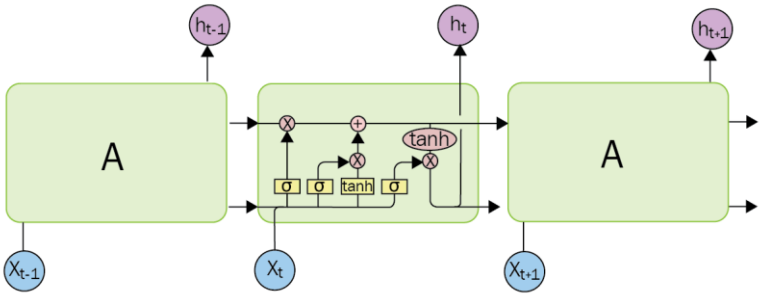



Input image
Convoluted filters

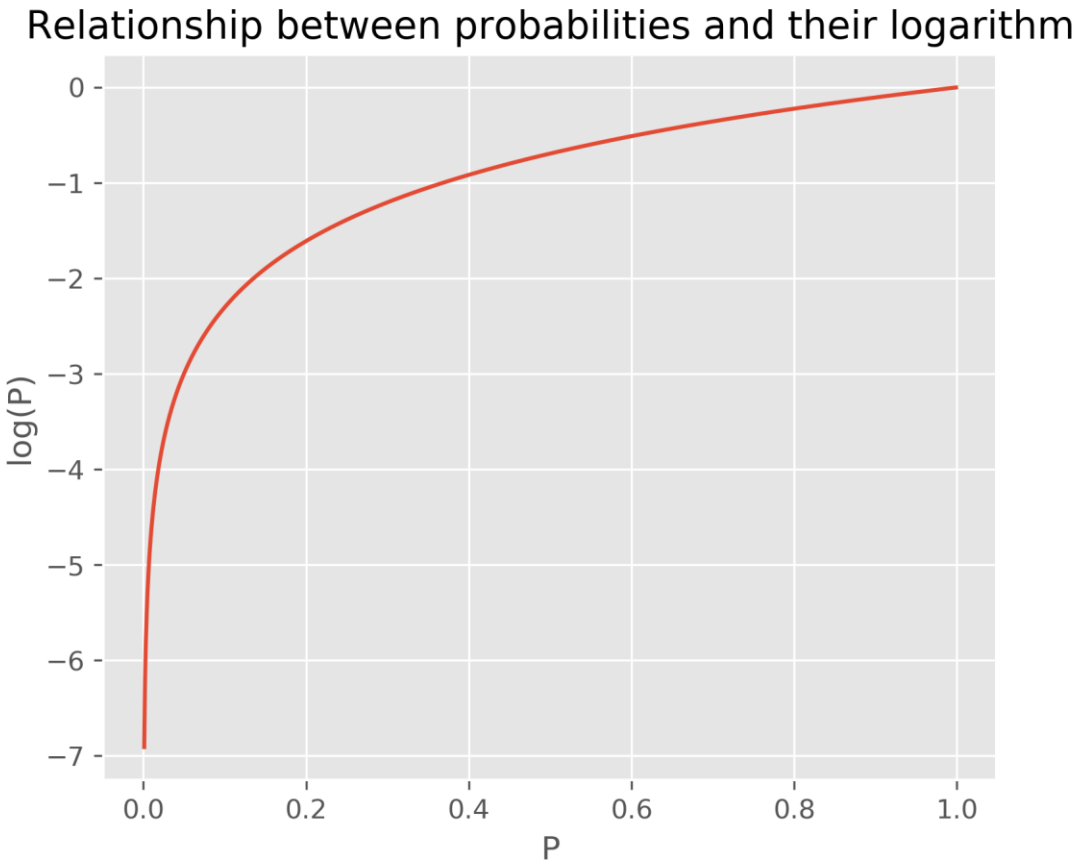

input
dense
maxpool
dense
maxpool
dense
dropout
dense
output

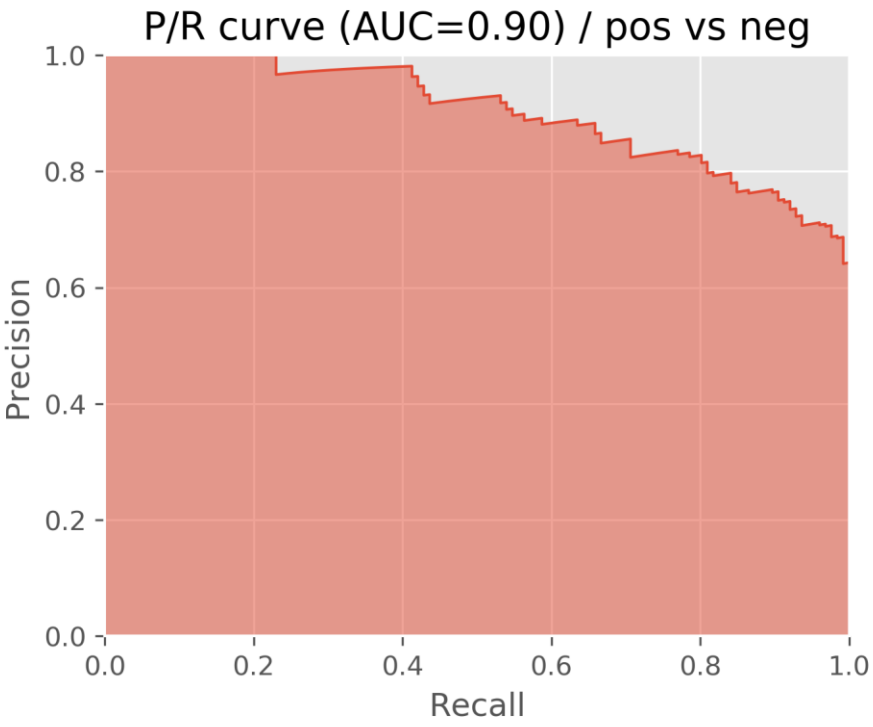

Classification error per Epoch
train
test
Classification error
Epoch
10^{0}
10^{-1}
10^{-2}
10^{-3}
0
2
4
6
8
10

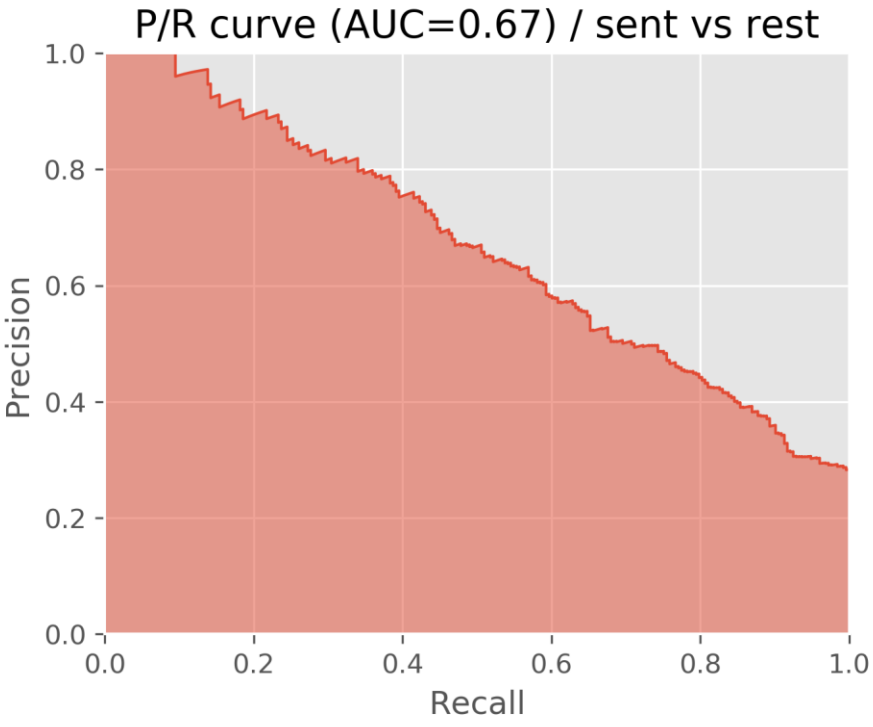

Xt
A
Yt
X0
X1
X2
X3
X4
A
A
A
A
A
Y0
Y1
Y2
Y3
Y4

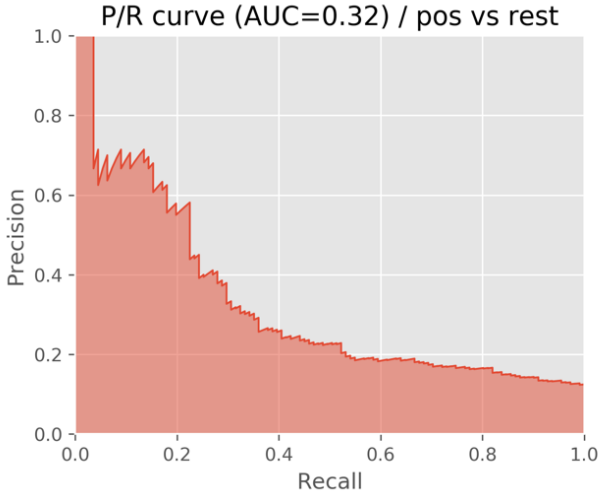

h_{t-1}
h_t
h_{t+1}
A
A
tanh
σ
σ
tanh
σ
X_{t-1}
X_t
X_{t+1}

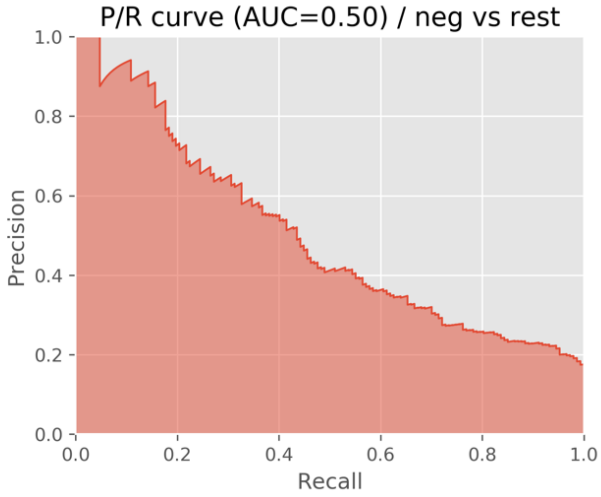
Chapter 9: Classification II - Sentiment Analysis

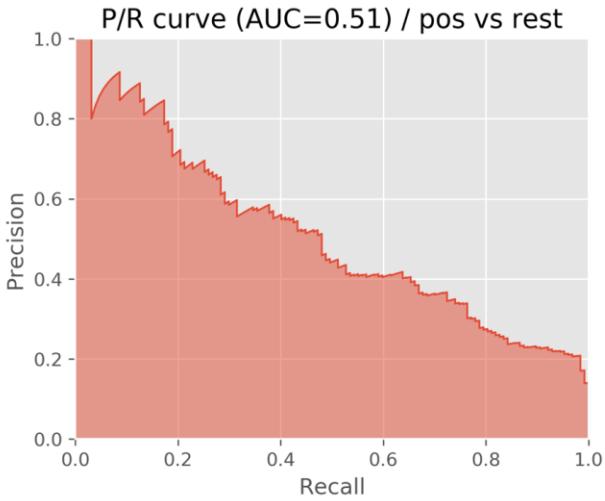


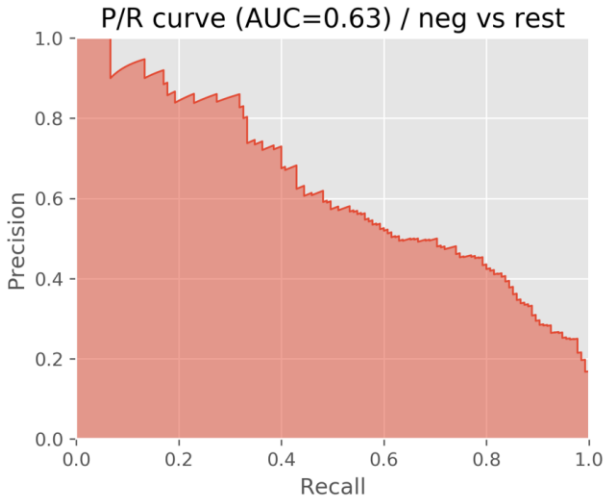

P/R curve (AUC=0.90) / pos vs neg
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


P/R curve (AUC=0.67) / sent vs rest
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

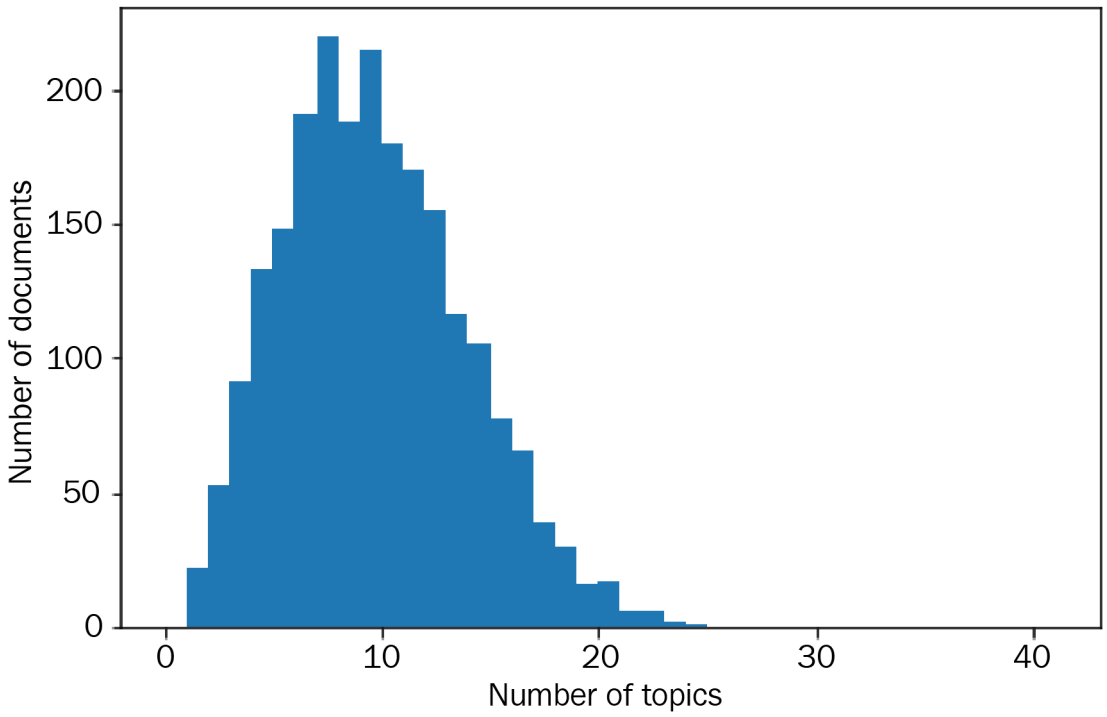

P/R curve (AUC=0.32) / pos vs rest
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

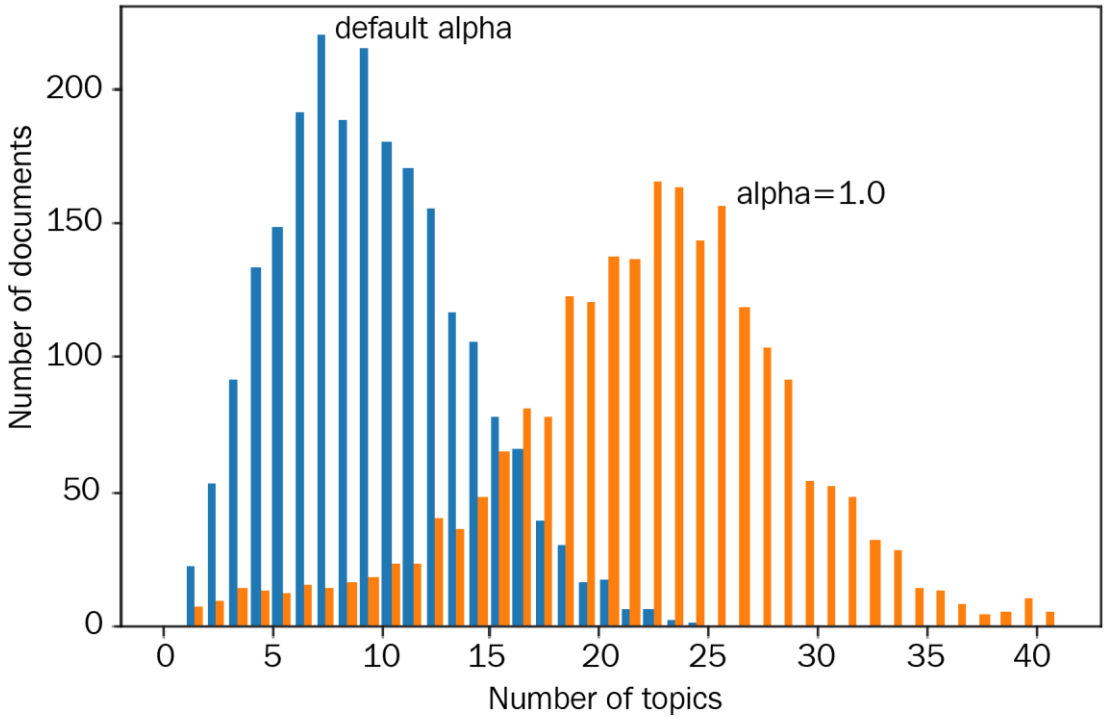

P/R curve (AUC=0.50) / neg vs rest
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

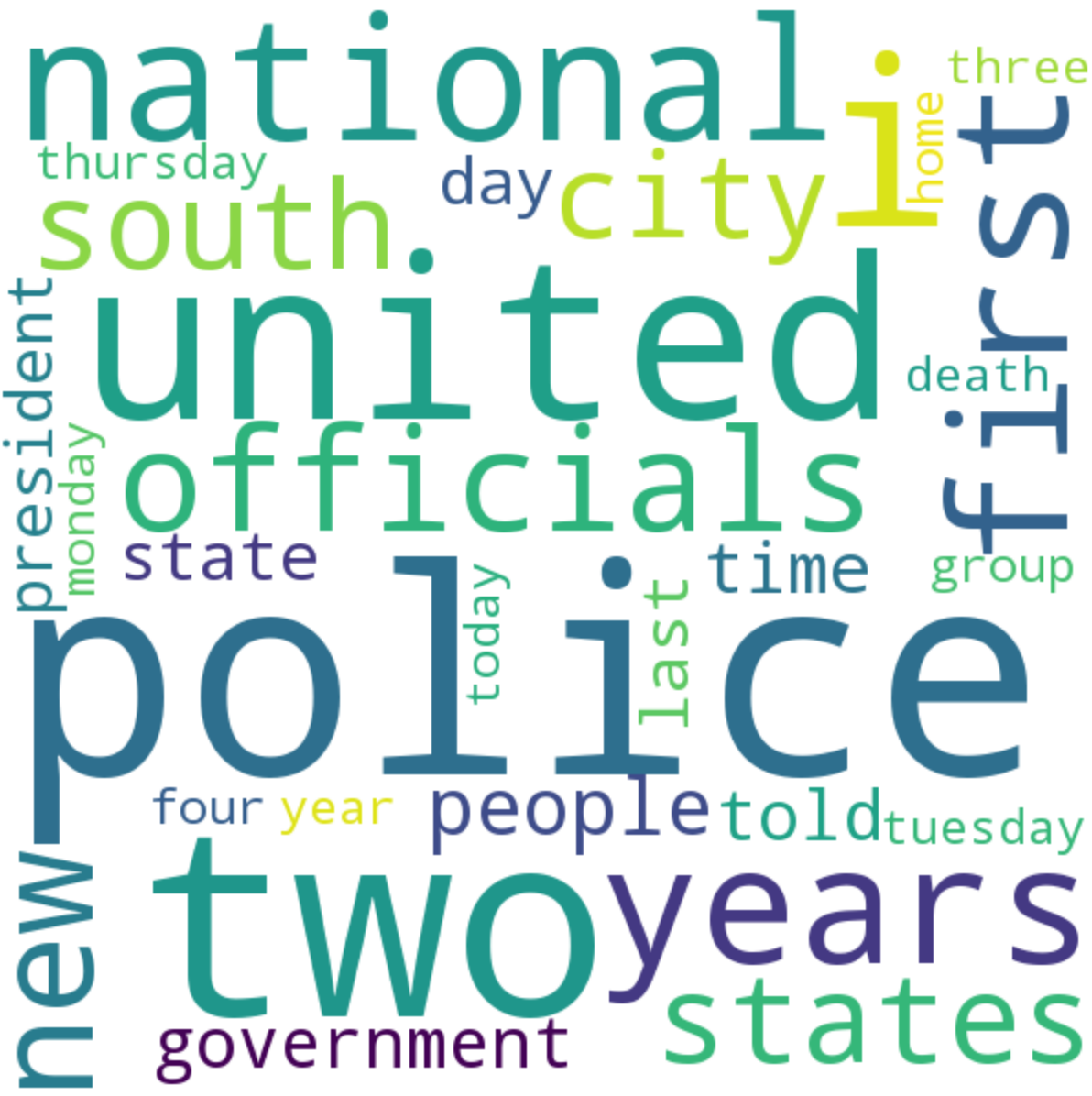

P/R curve (AUC=0.51) / pos vs rest
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

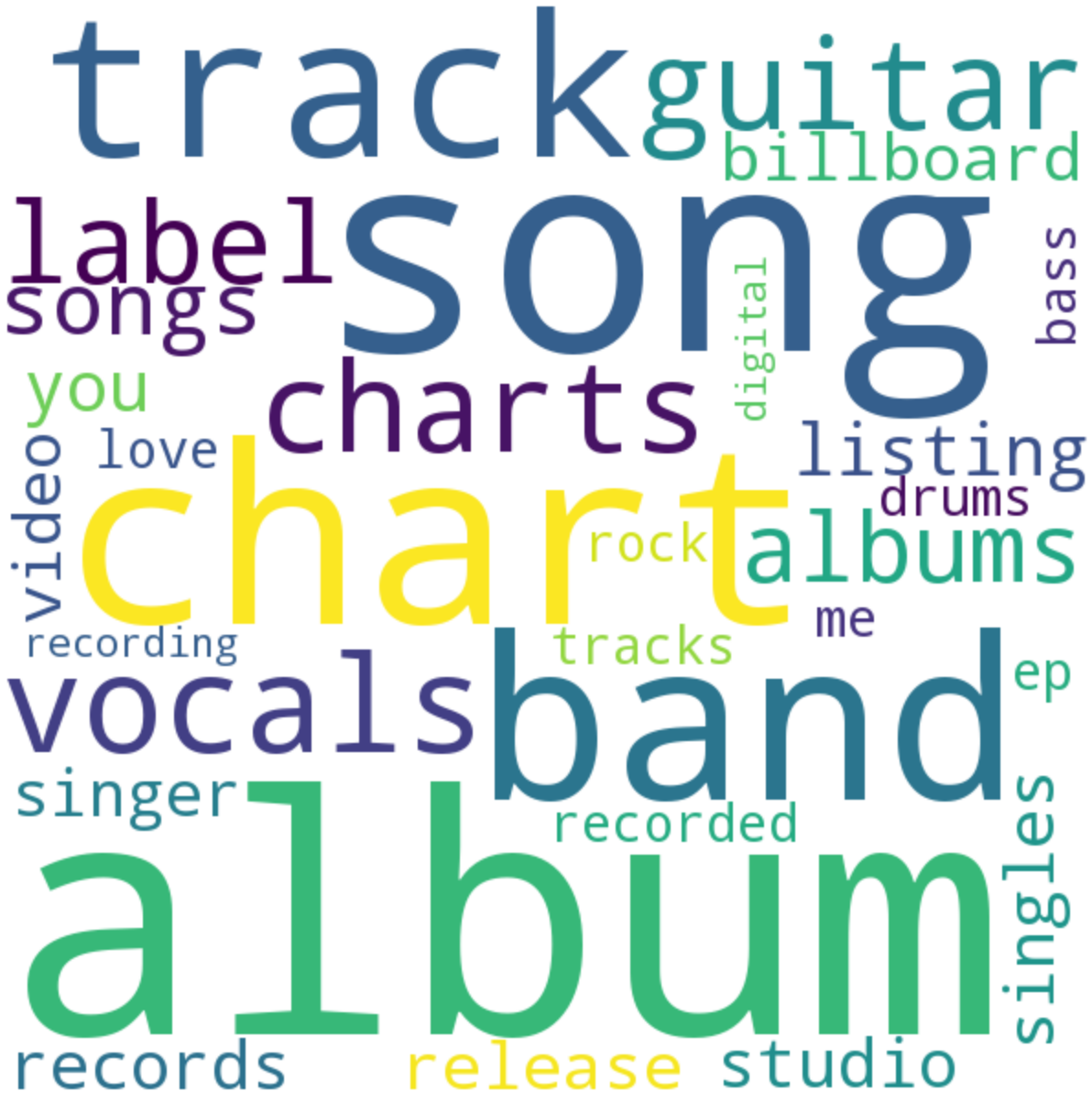

P/R curve (AUC=0.63) / neg vs rest
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

Chapter 10: Topic Modeling



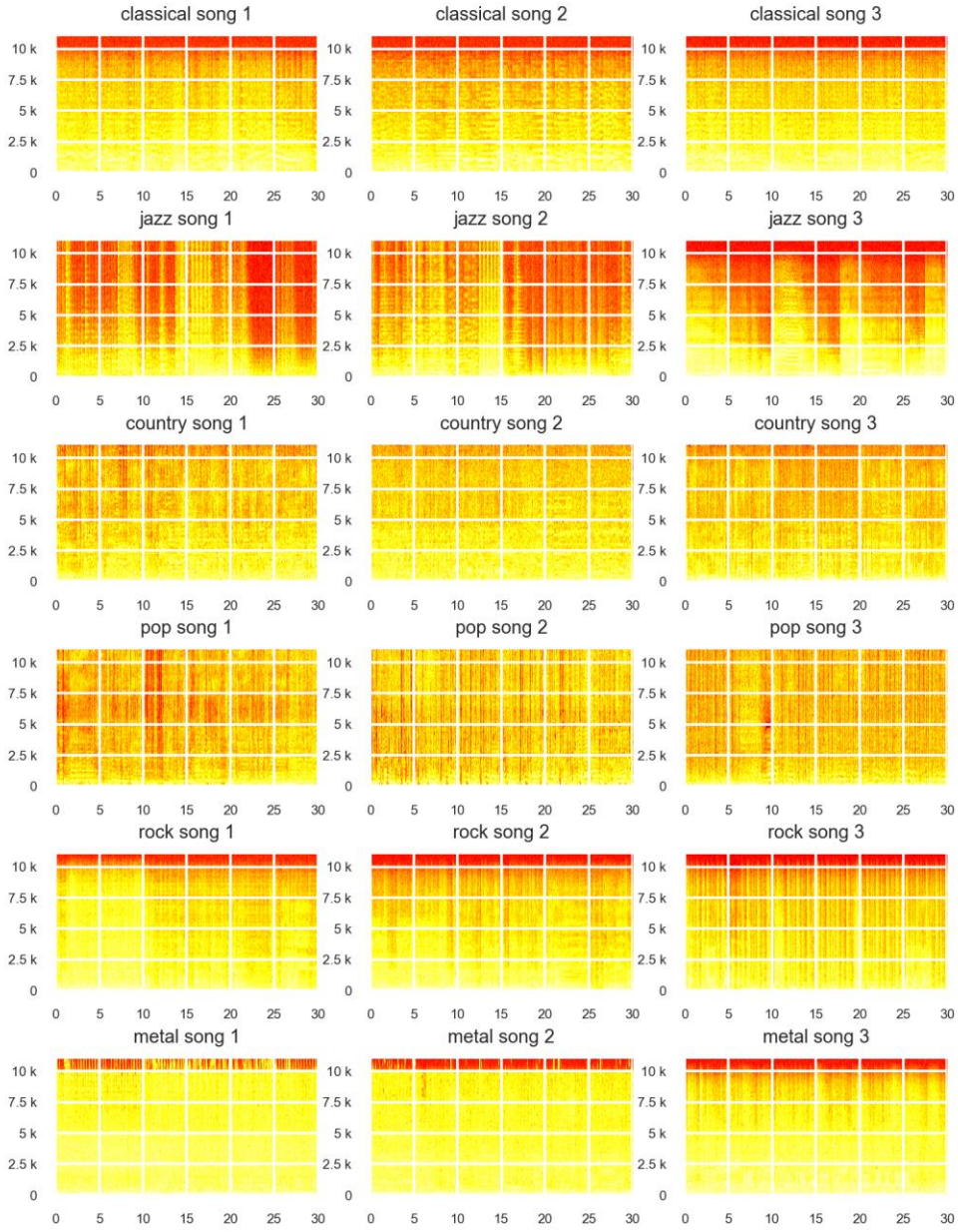

default alpha
alpha=1.0
Number of documents
Number of topics
0
50
100
150
200
0
5
10
15
20
25
30
35
40

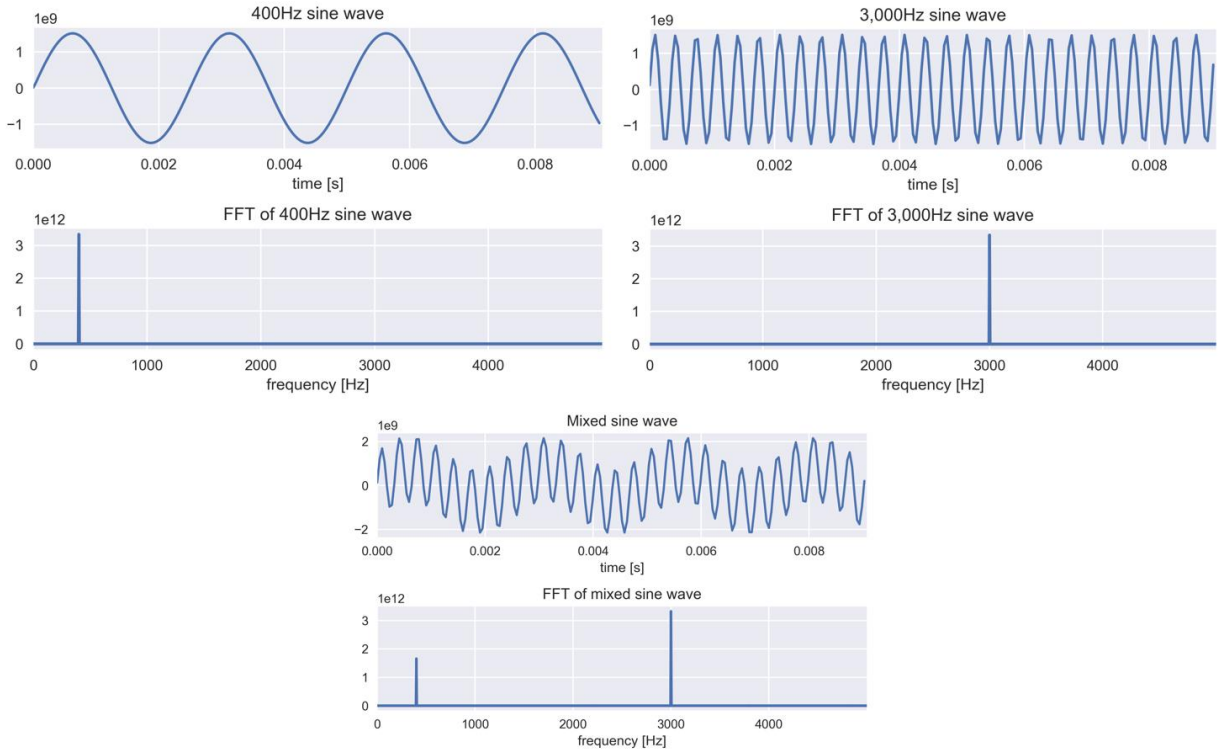

national
i
three
thursday
day
city
home
south.
first
united
president
death
officials
monday
state
time
group
police
today
last
four
year
people
told
tuesday
new
two
years
government
states

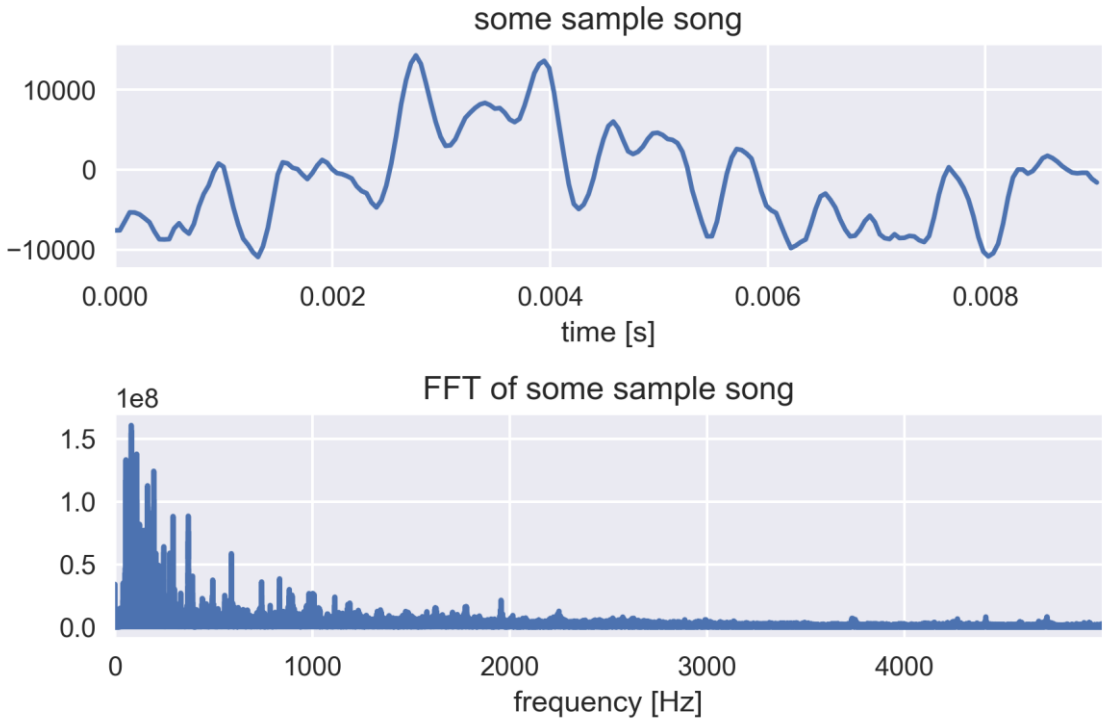

track
guitar
billboard
label
song
songs
bass
digital
you
charts
video
love
listing
drums
chart
albums
rock
me
recording
tracks
ep
vocals
band
singer
recorded
singles
album
records
release
studio

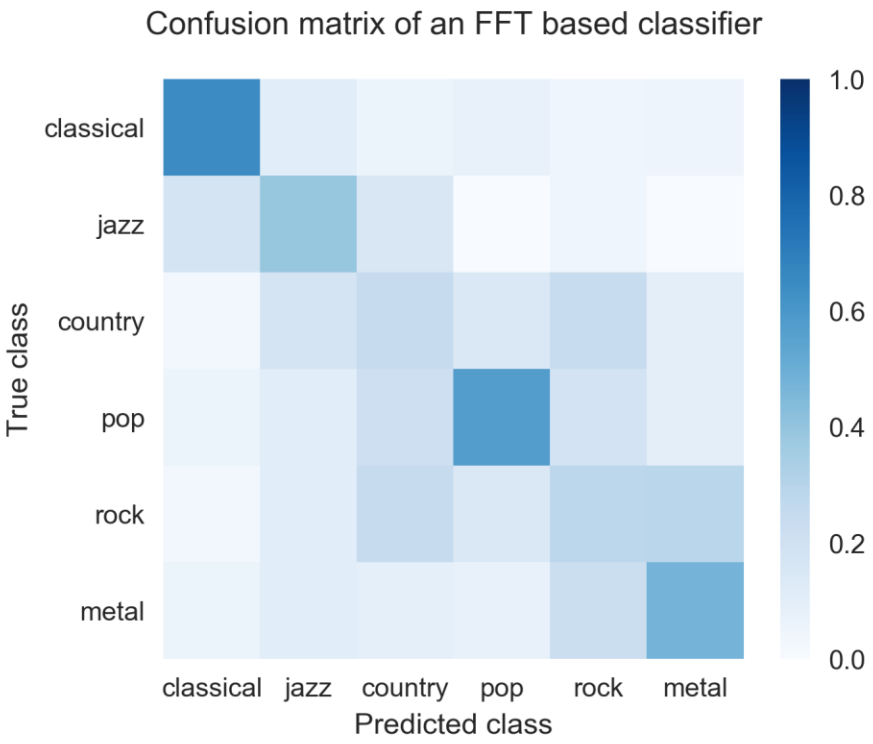

petersen
foals
liberian
ammonites
leone
pinkish
eventing
liberia
pura
nahiya
subcostal
sire
bukit
taman
sierra
yearling
silurian
freetown
wikimedia
jasmin
jalan
sired
tentacles
silvery
aloe
grambling
posteriorly
henchman
mirna
therapeutics

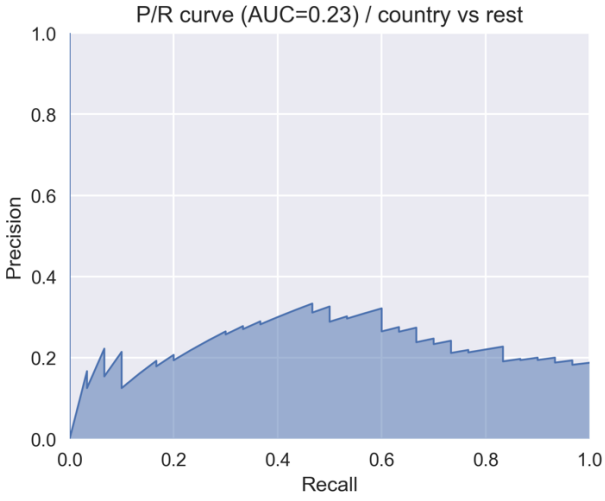
Chapter 11: Classification III - Music Genre Classification

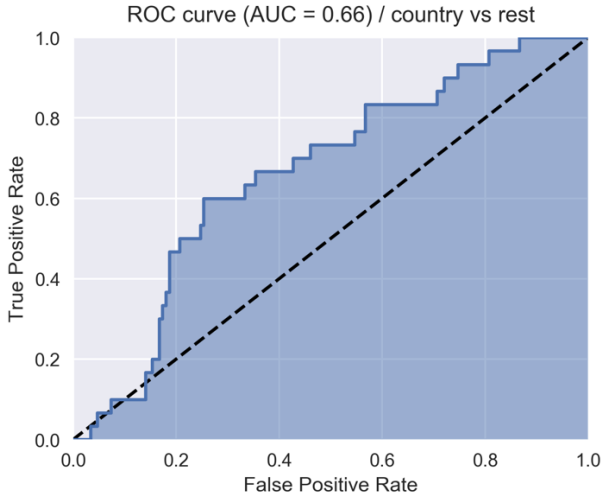


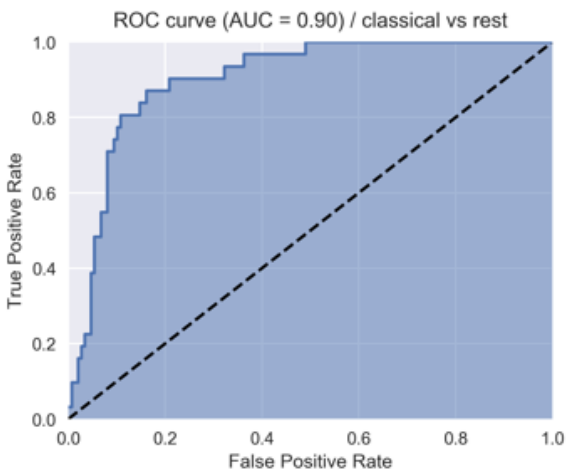

400Hz sine wave
1e9
time [s]
3,000Hz sine wave
1e9
time [s]
FFT of 400Hz sine wave
1e12
frequency [Hz]
FFT of 3,000Hz sine wave
1e12
frequency [Hz]
Mixed sine wave
1e9
time [s]
FFT of mixed sine wave
1e12
frequency [Hz]

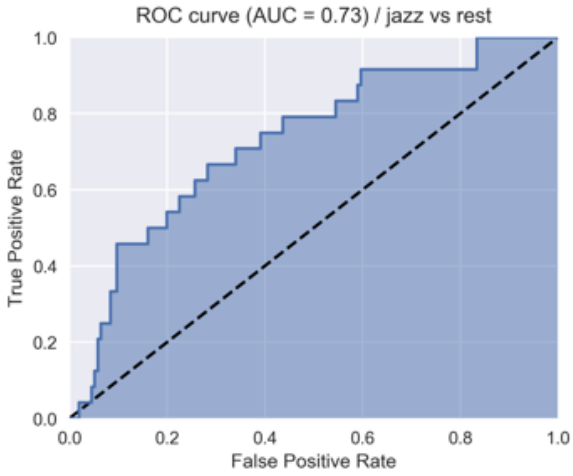

some sample song
time [s]
FFT of some sample song
1e8
frequency [Hz]

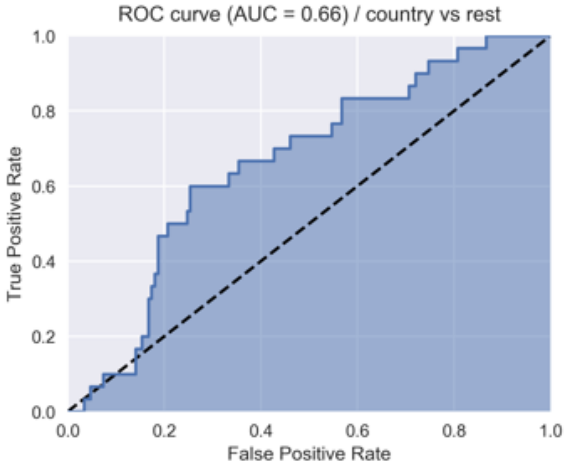

Confusion matrix of an FFT based classifier
classical
jazz
country
pop
rock
metal
True class
classical
jazz
country
pop
rock
metal
Predicted class
1.0
0.8
0.6
0.4
0.2
0.0

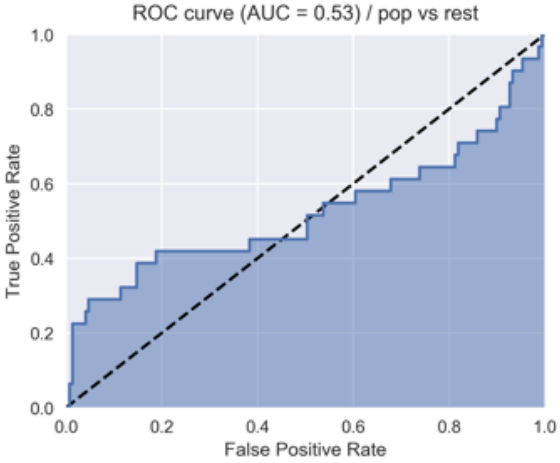

P/R curve (AUC=0.23) / country vs rest
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

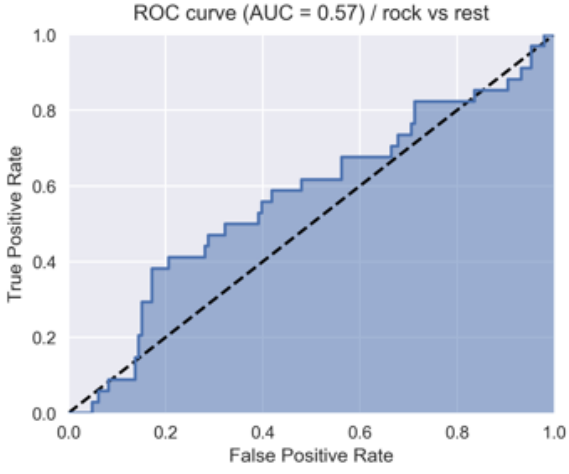

ROC curve (AUC = 0.66) / country vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

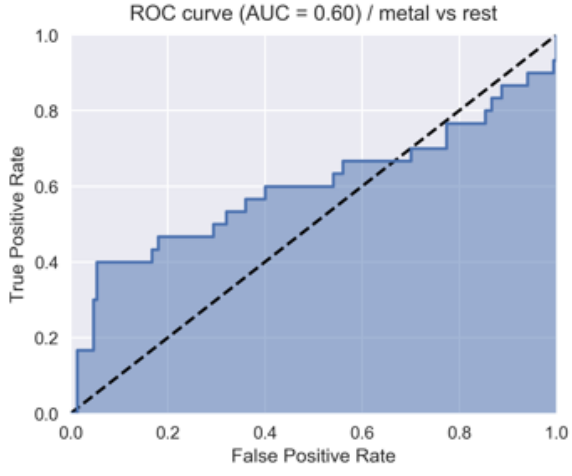

ROC curve (AUC = 0.90) / classical vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

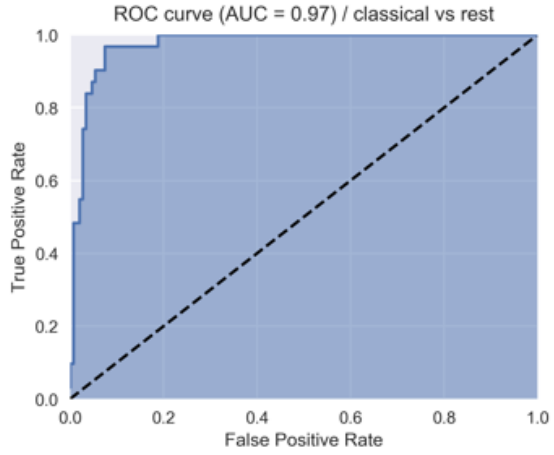

ROC curve (AUC = 0.73) / jazz vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

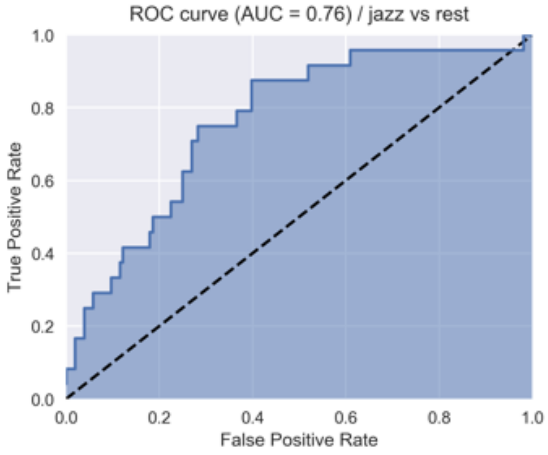

ROC curve (AUC = 0.66) / country vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

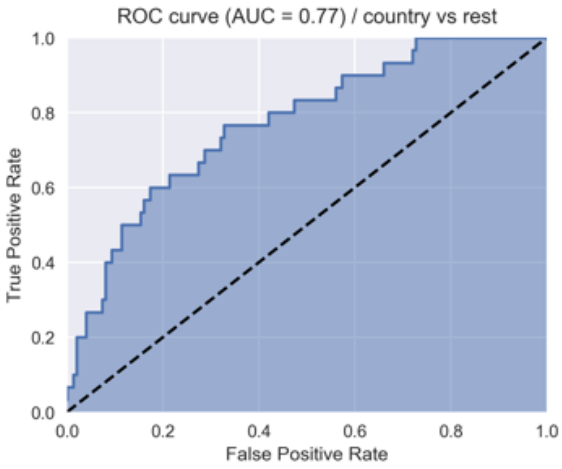

ROC curve (AUC = 0.53) / pop vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

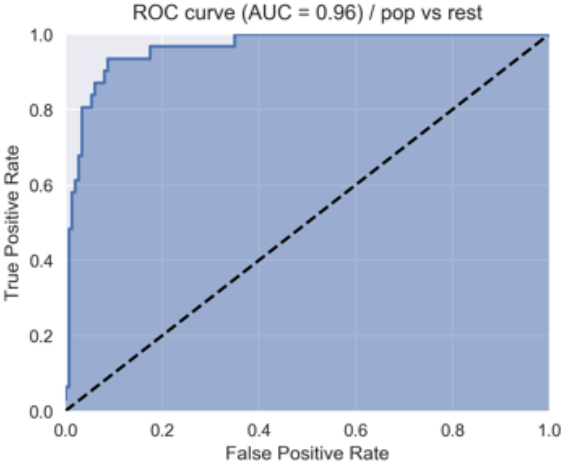

ROC curve (AUC = 0.57) / rock vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

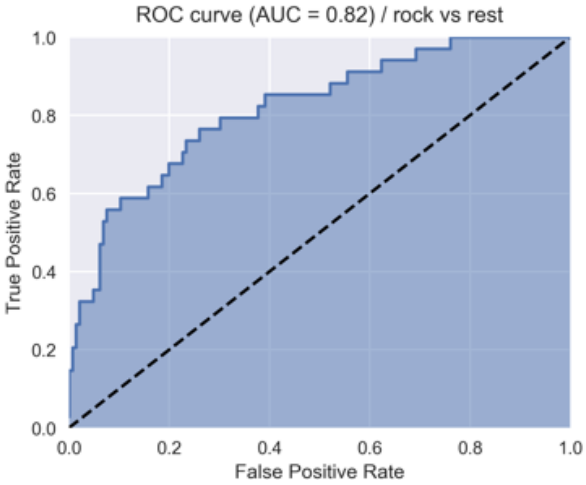

ROC curve (AUC = 0.60) / metal vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

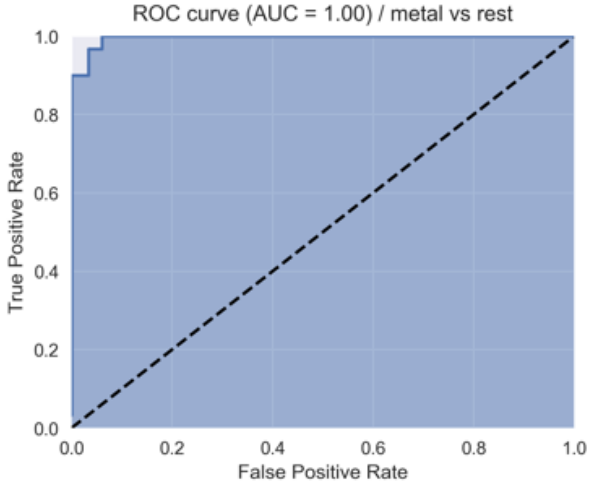

ROC curve (AUC = 0.97) / classical vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

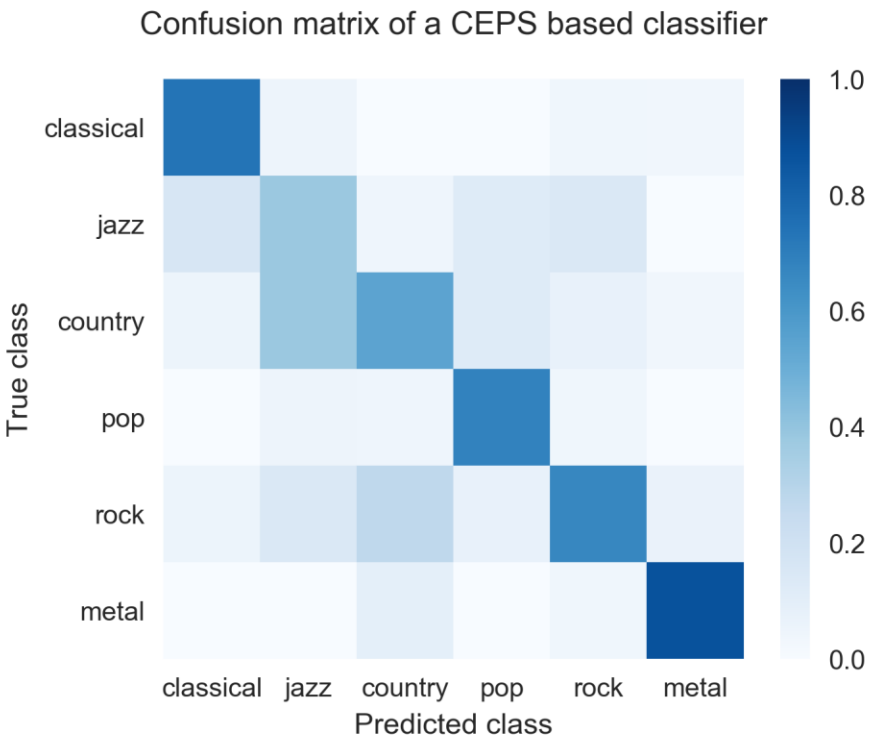

ROC curve (AUC = 0.76) / jazz vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

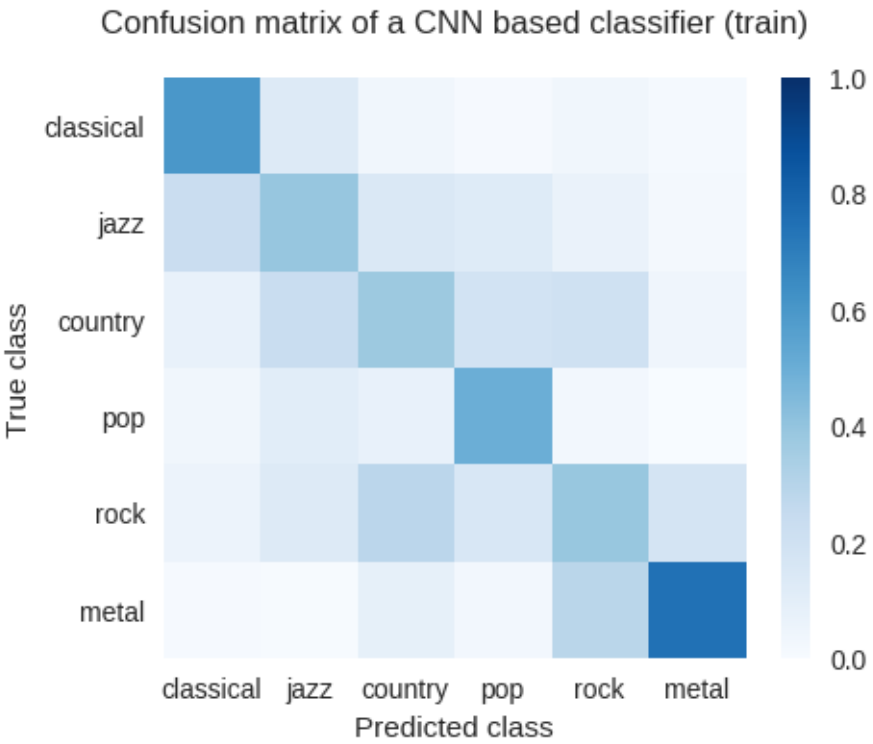

ROC curve (AUC = 0.77) / country vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

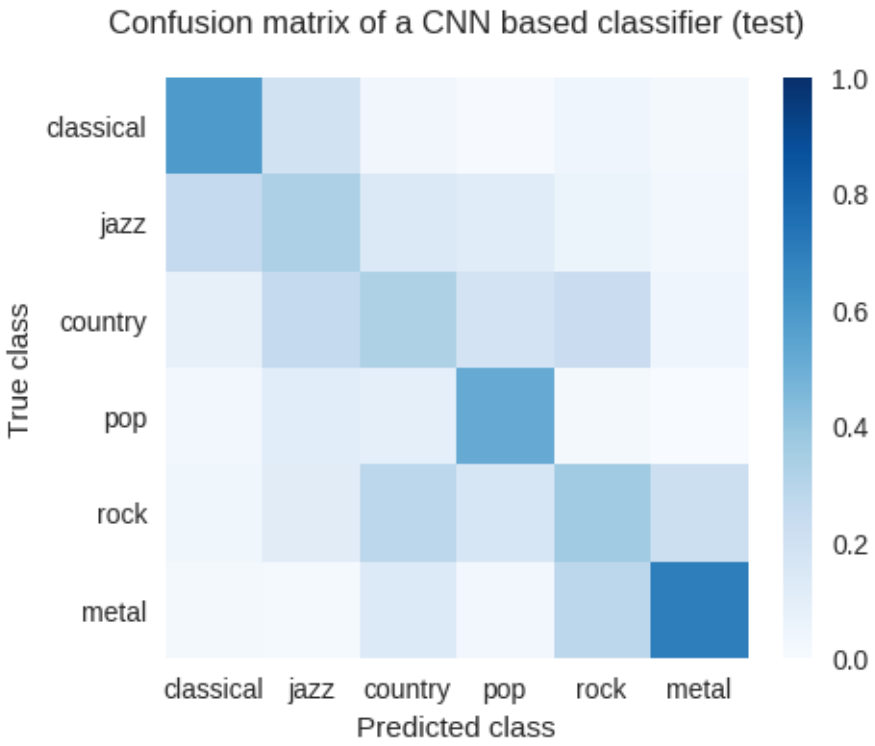

ROC curve (AUC = 0.96) / pop vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC curve (AUC = 0.82) / rock vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

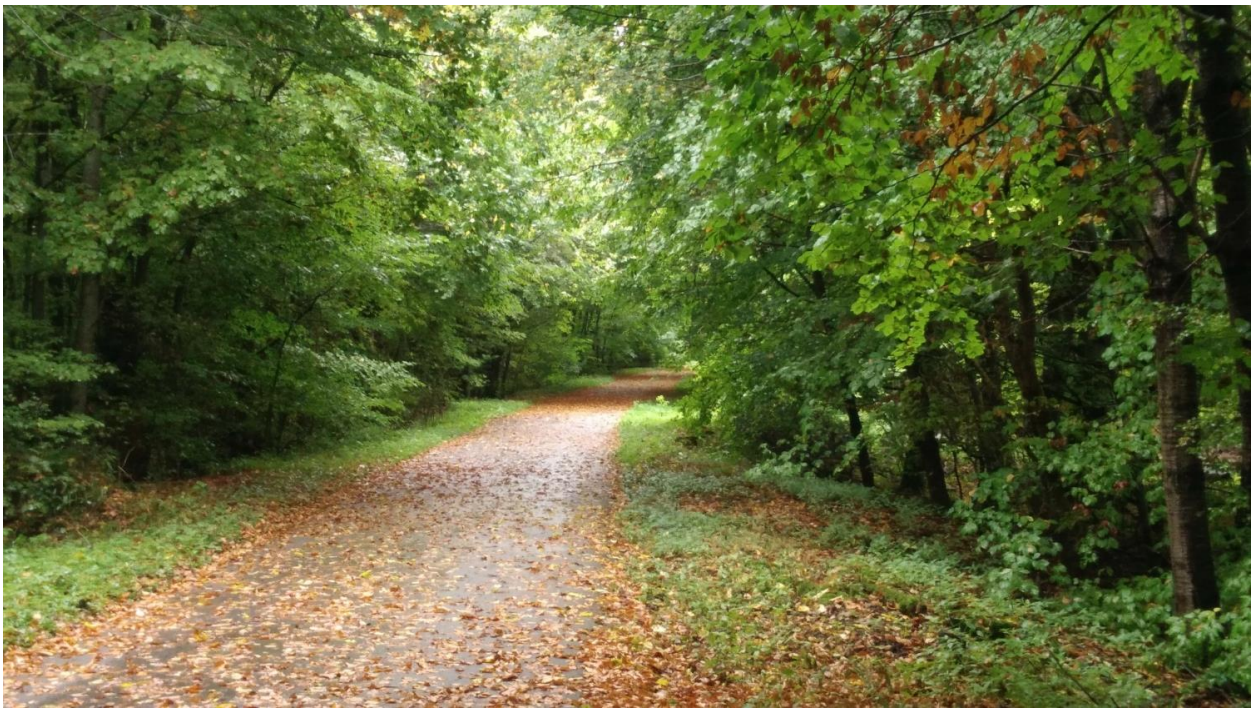

ROC curve (AUC = 1.00) / metal vs rest
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

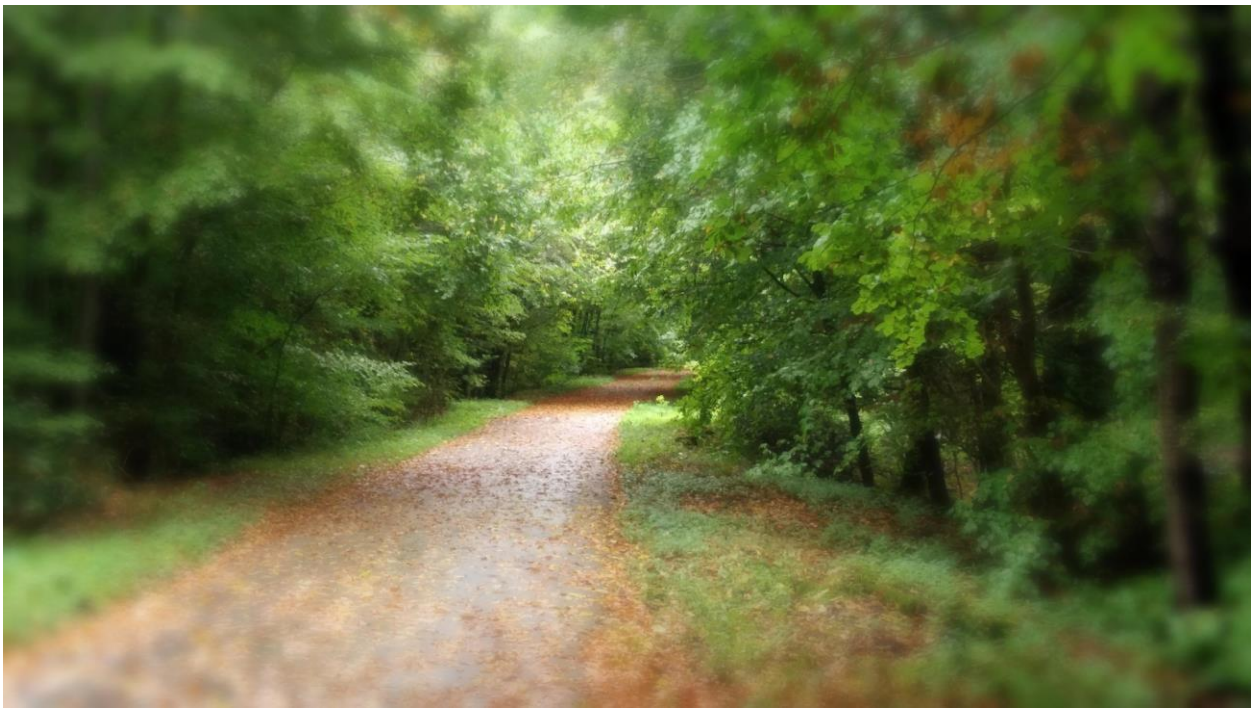

Confusion matrix of a CEPS based classifier
classical
jazz
country
pop
rock
metal
True class
classical
jazz
country
pop
rock
metal
Predicted class
1.0
0.8
0.6
0.4
0.2
0.0

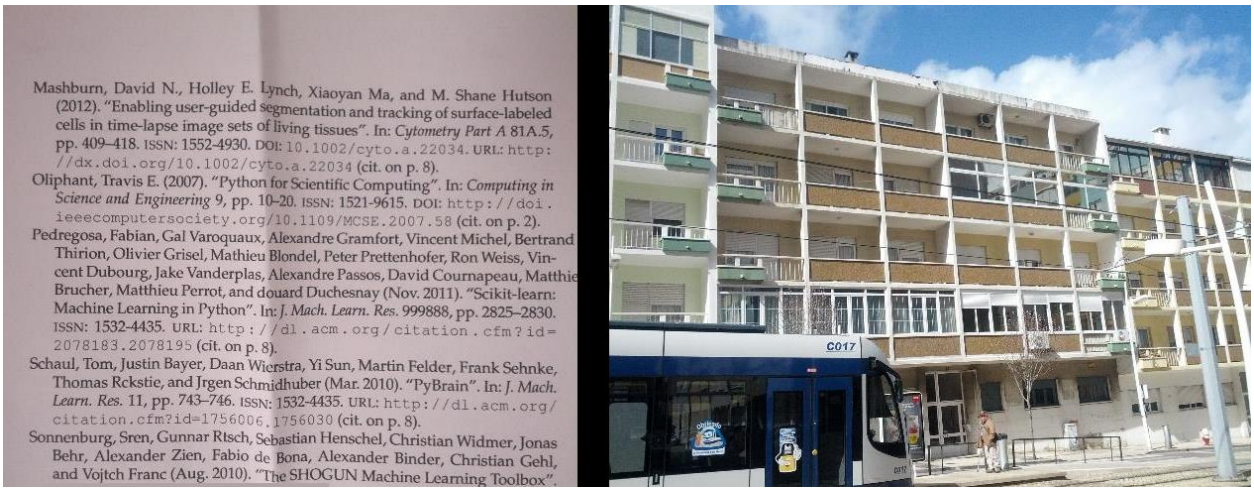

Confusion matrix of a CNN based classifier (train)
classical
jazz
country
pop
rock
metal
True class
classical
jazz
country
pop
rock
metal
Predicted class
1.0
0.8
0.6
0.4
0.2
0.0


Confusion matrix of a CNN based classifier (test)
True class
classical
jazz
country
pop
rock
metal
Predicted class
classical
jazz
country
pop
rock
metal
1.0
0.8
0.6
0.4
0.2
0.0

Chapter 12: Computer Vision







Mashburn, David N., Holley E. Lynch, Xiaoyan Ma, and M. Shane Hutson (2012). "Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues". In: *Cytometry Part A* 81A.5, pp. 409–418. ISSN: 1552-4930. DOI: 10.1002/cyto.a.22034. URL: http://dx.doi.org/10.1002/cyto.a.22034 (cit. on p. 8).

Oliphant, Travis E. (2007). "Python for Scientific Computing". In: *Computing in Science and Engineering* 9, pp. 10–20. ISSN: 1521-9615. DOI: http://doi.ieeecomputersociety.org/10.1109/MCSE.2007.58 (cit. on p. 2).

Pedregosa, Fabian, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and douard Duchesnay (Nov. 2011). "Scikit-learn: Machine Learning in Python". In: *J. Mach. Learn. Res.* 999888, pp. 2825–2830. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2078183.2078195 (cit. on p. 8).

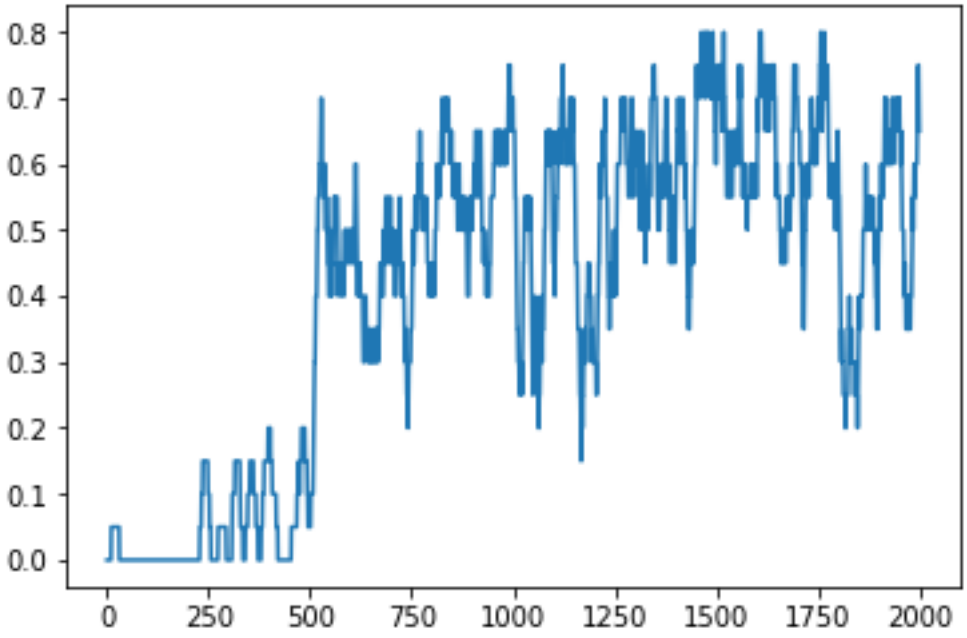
Schaul, Tom, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, Thomas Rckstie, and Jrgen Schmidhuber (Mar. 2010). "PyBrain". In: *J. Mach. Learn. Res.* 11, pp. 743–746. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1756006.1756030 (cit. on p. 8).

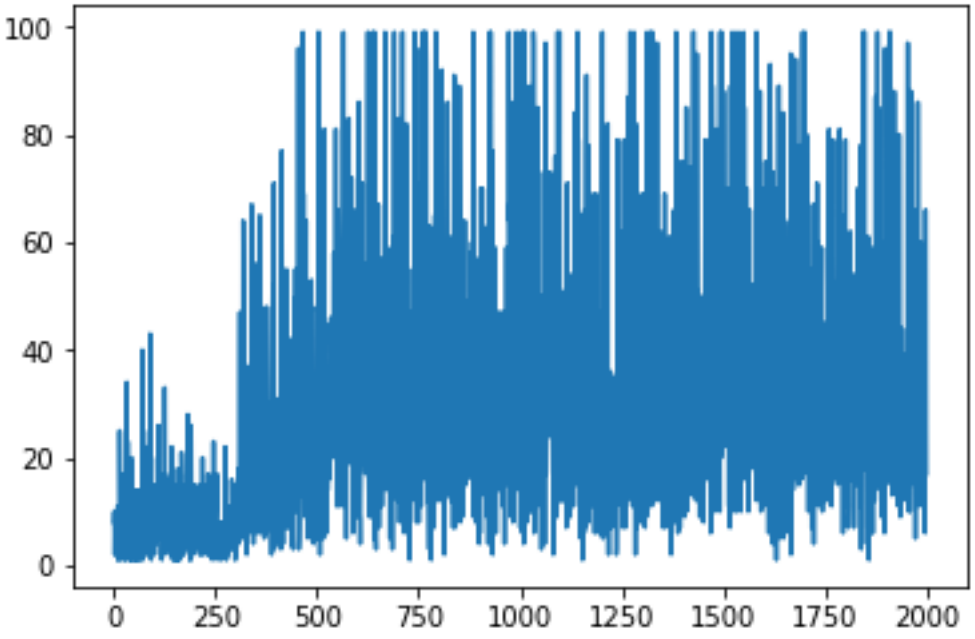
Sonnenburg, Sren, Gunnar Rtsch, Sebastian Henschel, Christian Widmer, Jonas Behr, Alexander Zien, Fabio de Bona, Alexander Binder, Christian Gehl, and Vojtch Franc (Aug. 2010). "The SHOGUN Machine Learning Toolbox".


Fahrräder
abstellen
verboten!


AP
98

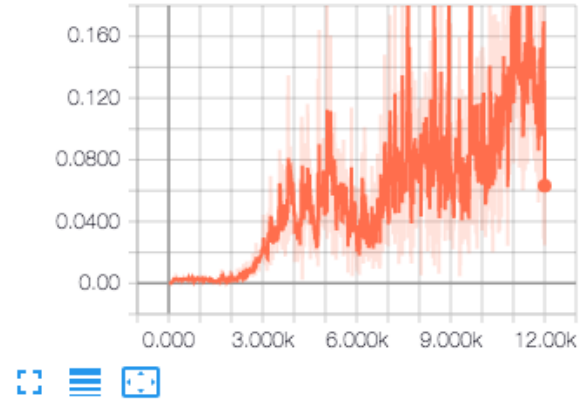
Chapter 13: Reinforcement Learning



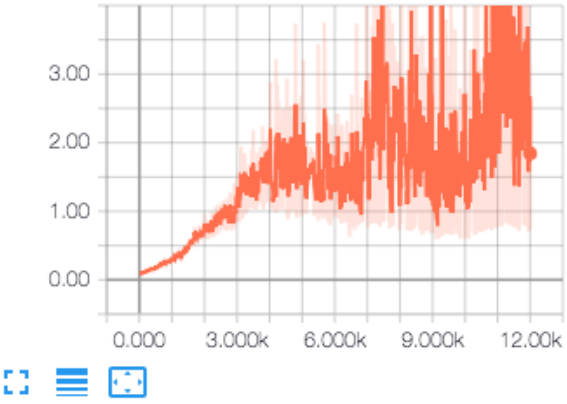


Breakout-v4

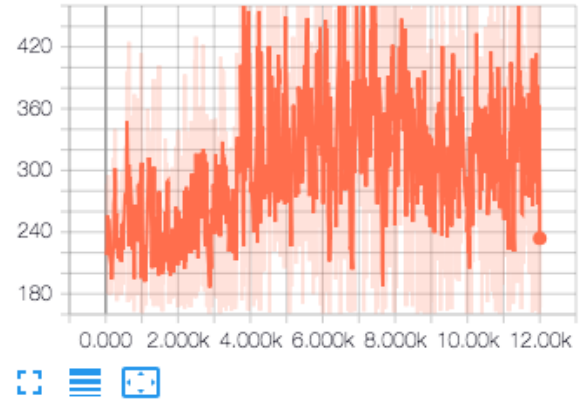
Breakout-v4/Average_Loss/Episode



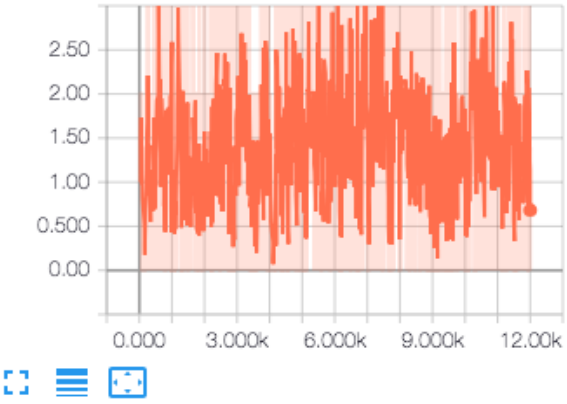
Breakout-v4/Average_Max_Q/Episode

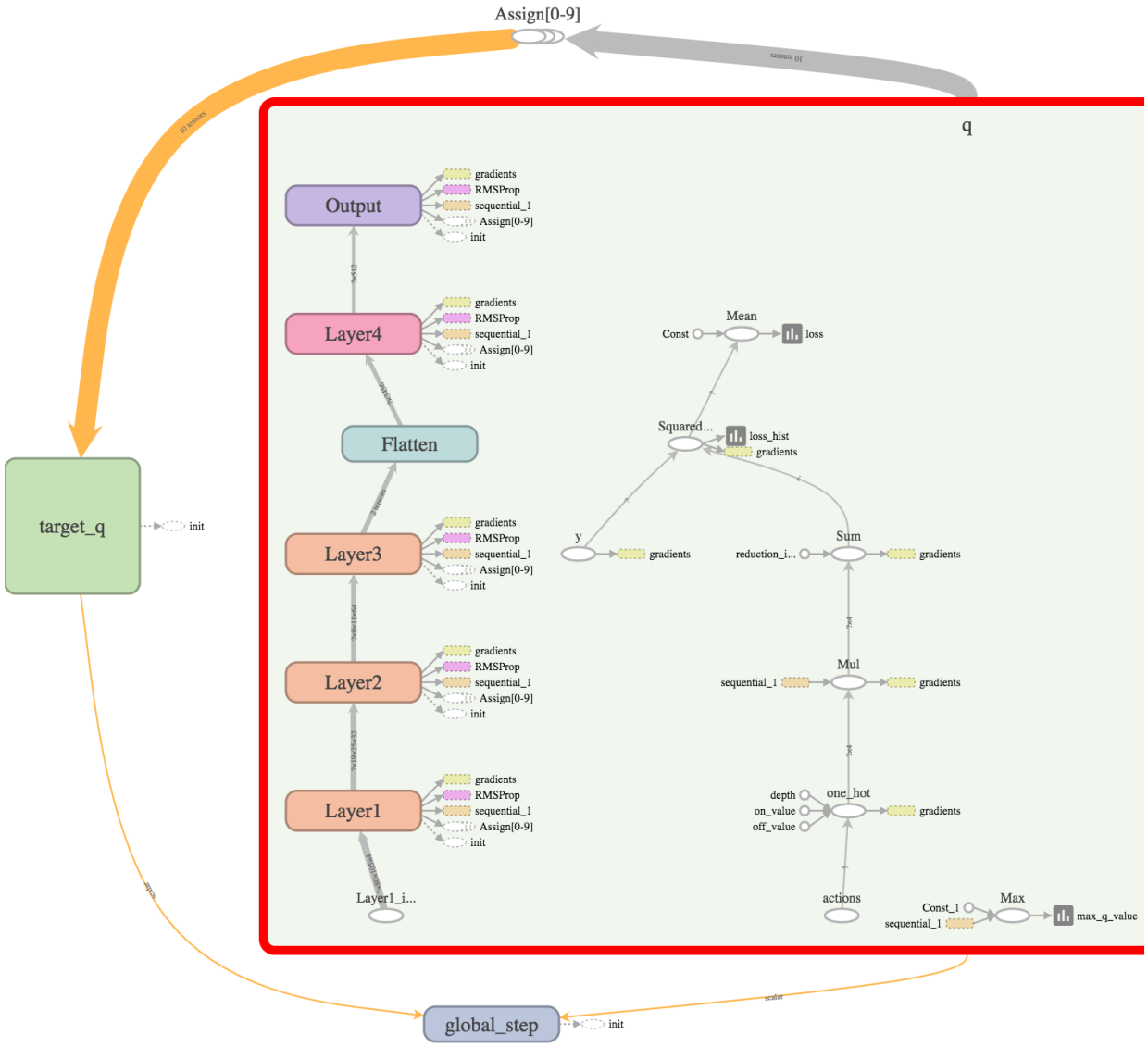


Breakout-v4/Duration/Episode



Breakout-v4/Total_Reward/Episode

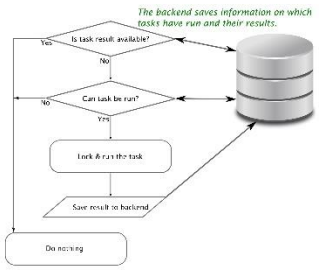



Assign[0-9]
10 tensors
10 tensors
q
Output
gradients
RMSProp
sequential_1
Assign[0-9]
init
Layer4
Flatten
Layer3
Layer2
Layer1
Layer1_i...
target_q
init
Mean
Const
loss
Squared...
loss_hist
gradients
y
Sum
reduction_i...
Mul
sequential_1
one_hot
depth
on_value
off_value
actions
Const_1
Max
max_q_value
global_step
scalar

Chapter 14: Bigger Data

```
  Waiting     Ready    Finished     Running  Task name
-------------------------------------------------------------------------
        1         0           0           0  jugfile.print_final_result
        1         0           0           0  jugfile.add
        2         2           0           0  jugfile.double
.........................................................................
        4         2           0           0  Total
```

```
 Waiting     Ready    Finished     Running  Task name
-------------------------------------------------------------------------
       1         0           0           0  jugfile.print_final_result
       2         0           0           2  jugfile.double
       1         0           0           0  jugfile.add
.........................................................................
       4         2           0           0  Total
```



AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage).

> Recently visited services

v All services

Compute
- EC2
- Lightsail
- Elastic Container Service
- Lambda
- Batch
- Elastic Beanstalk

Storage
- S3
- EFS
- Glacier
- Storage Gateway

Database
- RDS
- DynamoDB
- ElastiCache
- Amazon Redshift

Migration
- AWS Migration Hub
- Application Discovery Service
- Database Migration Service

Management Tools
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- CloudTrail
- Config
- OpsWorks
- Service Catalog
- Systems Manager
- Trusted Advisor
- Managed Services

Media Services
- Elastic Transcoder
- Kinesis Video Streams
- MediaConvert
- MediaLive
- MediaPackage
- MediaStore
- MediaTailor

Machine Learning
- Amazon SageMaker
- Amazon Comprehend
- AWS DeepLens

Mobile Services
- Mobile Hub
- AWS AppSync
- Device Farm
- Mobile Analytics

AR & VR
- Amazon Sumerian

Application Integration
- Step Functions
- Amazon MQ
- Simple Notification Service
- Simple Queue Service
- SWF

Customer Engagement
- Amazon Connect
- Pinpoint
- Simple Email Service

Business Productivity
- Alexa for Business
- Amazon Chime

Add user

1 2 3 4

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* aws_ml

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type* ☑ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☐ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

*** Required** Cancel Next: Permissions

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions. Learn more

Group name EC2_FULL

Create policy Refresh

Filter: Policy type AmazonEC2 Showing 17 results

Policy name	Type	Attachments	Description
AmazonEC2ContainerRegistryReadOnly	AWS managed	0	Provides read-only access to Amazon EC2 Container Registry repositories.
AmazonEC2ContainerServiceAutoscaleRole	AWS managed	0	Policy to enable Task Autoscaling for Amazon EC2 Container Service
AmazonEC2ContainerServiceEventsRole	AWS managed	0	Policy to enable CloudWatch Events for EC2 Container Service
AmazonEC2ContainerServiceforEC2Role	AWS managed	0	Default policy for the Amazon EC2 Role for Amazon EC2 Container Service.
AmazonEC2ContainerServiceFullAccess	AWS managed	0	Provides administrative access to Amazon ECS resources.
AmazonEC2ContainerServiceRole	AWS managed	0	Default policy for Amazon ECS service role.
☑ AmazonEC2FullAccess	AWS managed	1	Provides full access to Amazon EC2 via the AWS Management Console.
AmazonEC2ReadOnlyAccess	AWS managed	0	Provides read only access to Amazon EC2 via the AWS Management Console.
AmazonEC2ReportsAccess	AWS managed	0	Provides full access to all Amazon EC2 reports via the AWS Management Console.
AmazonEC2RoleforAWSCodeDeploy	AWS managed	0	Provides EC2 access to S3 bucket to download revision. This role is needed by the CodeDeploy agent on EC2 instances.

Cancel Create group

Download .csv

User	Access key ID	Secret access key
aws_ml	AKIAJM3ATLOTSE45BIKA	********* Show

Close

Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

- 0 Running Instances
- 0 Dedicated Hosts
- 0 Volumes
- 1 Key Pairs
- 0 Placement Groups
- 0 Elastic IPs
- 0 Snapshots
- 0 Load Balancers
- 3 Security Groups

Learn more about the latest in AWS Compute from AWS re:Invent 2017 by viewing the EC2 Videos.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Service Status:

US East (N. Virginia):
This service is operating normally

Availability Zone Status:

us-east-1a:
Availability zone is operating normally

us-east-1b:
Availability zone is operating normally

Scheduled Events

US East (N. Virginia):

No events

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

1 to 36 of 36 AMIs

My AMIs
AWS Marketplace
Community AMIs
Free tier only

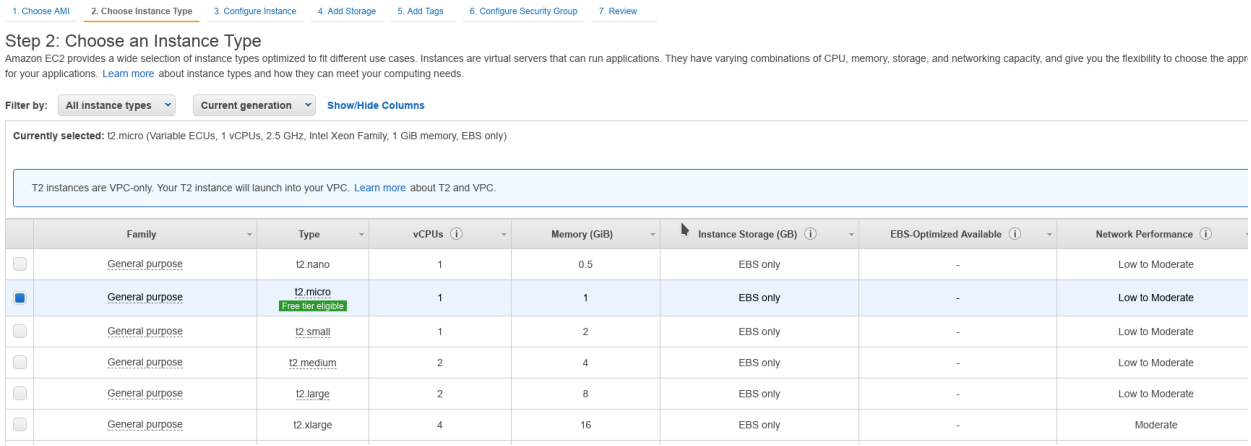
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-14c5486b
Amazon Linux
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit

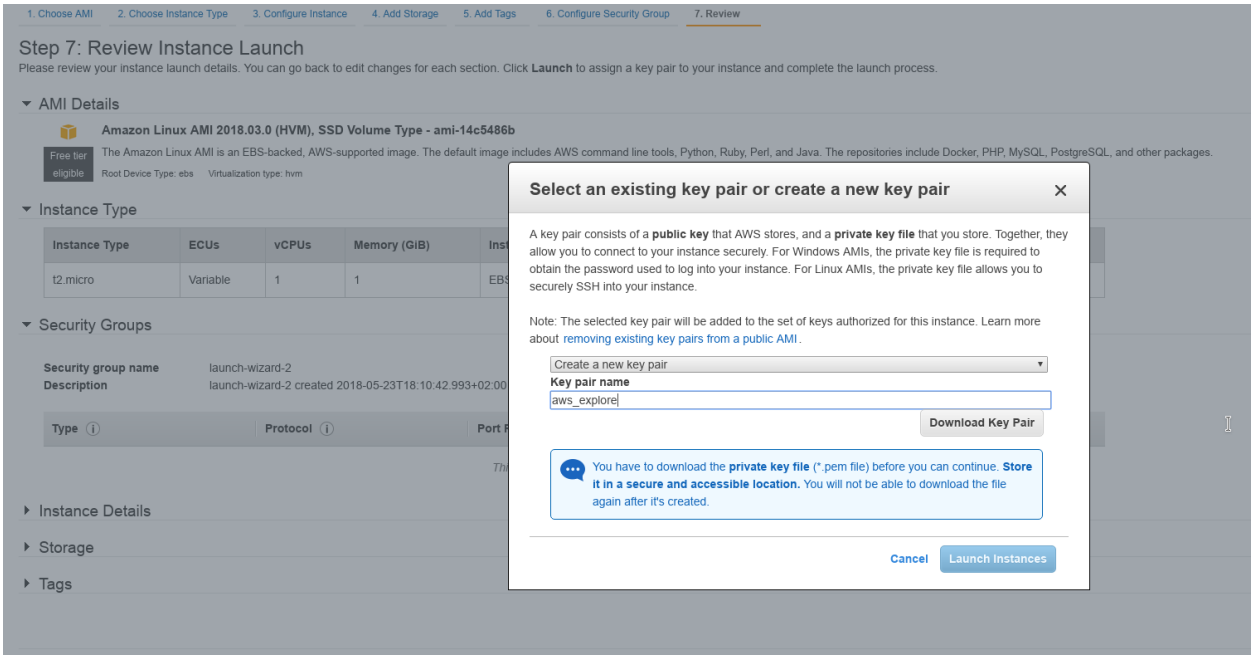
Amazon Linux 2 LTS Candidate 2 AMI (HVM), SSD Volume Type - ami-afd15ed0
Amazon Linux
Free tier eligible
Amazon Linux 2 LTS Candidate 2 provides an updated version of the Linux Kernel (4.14) tuned for EC2, systemd support, a newer compiler (gcc 7.3), an updated C runtime (glibc 2.26), newer tooling (binutils 2.29.1), and the latest software packages through the extras mechanisms.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit

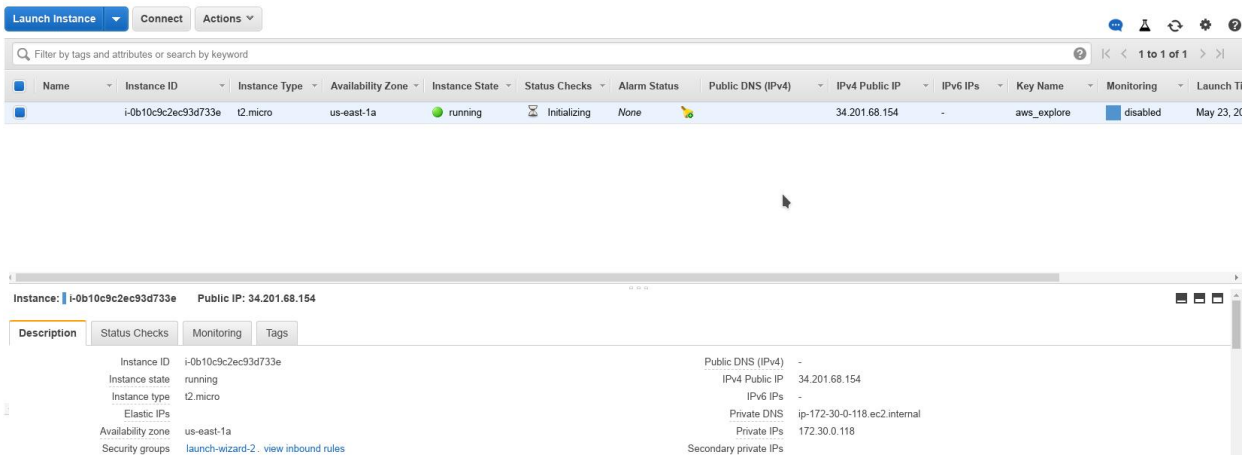
SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type - ami-62bda218
SUSE Linux
Free tier eligible
SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit

Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type - ami-6871a115
Red Hat
Free tier eligible
Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit

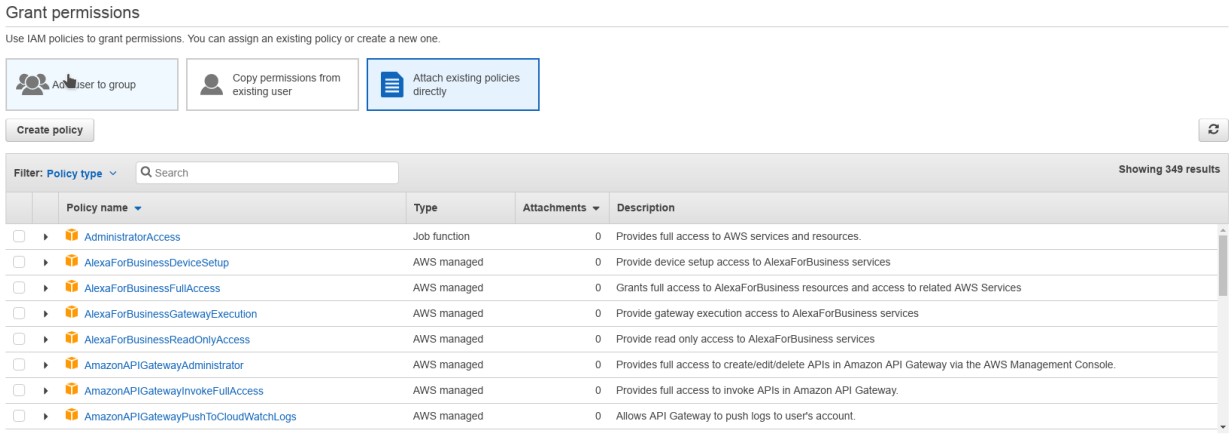
Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-5c66ea23
Free tier eligible
Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select
64-bit

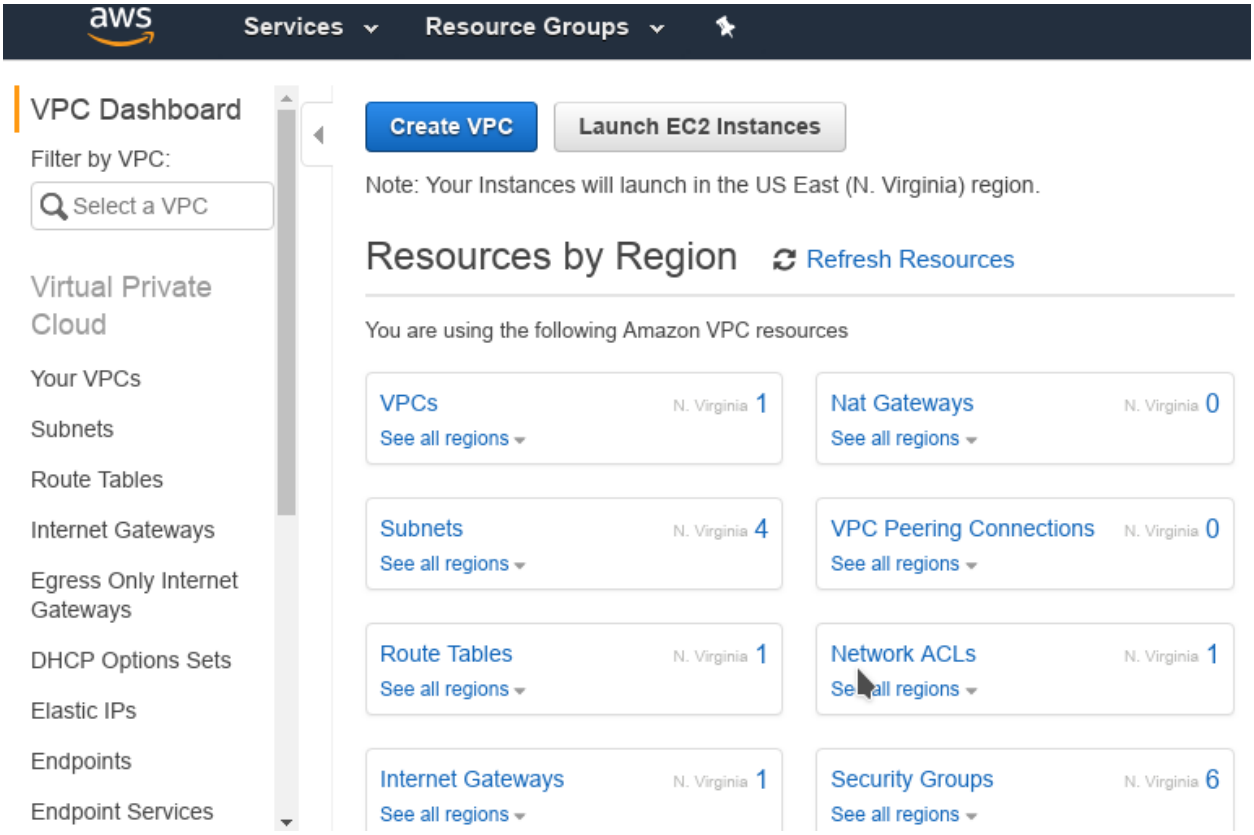

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 2: Choose an Instance Type
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appr
for your applications. Learn more about instance types and how they can meet your computing needs.
Filter by:
All instance types
Current generation
Show/Hide Columns
Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)
T2 instances are VPC-only. Your T2 instance will launch into your VPC. Learn more about T2 and VPC.
Family
Type
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
General purpose
t2.nano
1
0.5
EBS only
-
Low to Moderate
General purpose
t2.micro
Free tier eligible
1
1
EBS only
-
Low to Moderate
General purpose
t2.small
1
2
EBS only
-
Low to Moderate
General purpose
t2.medium
2
4
EBS only
-
Low to Moderate
General purpose
t2.large
2
8
EBS only
-
Low to Moderate
General purpose
t2.xlarge
4
16
EBS only
-
Moderate


1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.
AMI Details
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-14c5486b
Free tier eligible
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Instance Type
ECUs
vCPUs
Memory (GiB)
t2.micro
Variable
1
1
Security Groups
Security group name
launch-wizard-2
Description
launch-wizard-2 created 2018-05-23T18:10:42.993+02:00
Type
Protocol
Instance Details
Storage
Tags
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Create a new key pair
Key pair name
aws_explore
Download Key Pair
You have to download the private key file (*.pem file) before you can continue. Store it in a secure and accessible location. You will not be able to download the file again after it's created.
Cancel
Launch Instances


Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
IPv6 IPs
Key Name
Monitoring
Launch Ti
i-0b10c9c2ec93d733e
t2.micro
us-east-1a
running
Initializing
None
34.201.68.154
-
aws_explore
disabled
May 23, 20
Instance: i-0b10c9c2ec93d733e
Public IP: 34.201.68.154
Description
Status Checks
Monitoring
Tags
Instance ID
i-0b10c9c2ec93d733e
Instance state
running
Instance type
t2.micro
Elastic IPs
Availability zone
us-east-1a
Security groups
launch-wizard-2. view inbound rules
Public DNS (IPv4)
-
IPv4 Public IP
34.201.68.154
IPv6 IPs
-
Private DNS
ip-172-30-0-118.ec2.internal
Private IPs
172.30.0.118
Secondary private IPs

```
$ ssh -i aws_explore.pem ec2-user@34.201.68.154

Last login: Wed May 23 16:14:17 2018 from 94.252.95.61

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
6 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-30-0-118 ~]$
```




aws
Services
Resource Groups
VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
Create VPC
Launch EC2 Instances
Note: Your Instances will launch in the US East (N. Virginia) region.
Resources by Region
Refresh Resources
You are using the following Amazon VPC resources
VPCs N. Virginia 1
See all regions
Nat Gateways N. Virginia 0
See all regions
Subnets N. Virginia 4
See all regions
VPC Peering Connections N. Virginia 0
See all regions
Route Tables N. Virginia 1
See all regions
Network ACLs N. Virginia 1
See all regions
Internet Gateways N. Virginia 1
See all regions
Security Groups N. Virginia 6
See all regions