Mais qu'est-ce que je fous?

comment glander et apprendre les maths en même temps

Jeff Abrahamson

9 novembre 2016

I will lie to you

train vs plane

Le problème

Le code

Avant j'utilisais ratpoison :

```
task=$(ratpoison -c 'windows %s%t' | egrep '^\*' | perl -
echo $(date +%s) $task >> $task_file
```

Le code

Maintenant j'utilise i3 :

```
id=$(xprop -root |  awk '/_NET_ACTIVE_WINDOW\(WINDOW\)/{p
task=$(xprop -id $id | awk '/_NET_WM_NAME/{$1=$2="";print
echo $(date +%s) $task >> $task_file
```

Les données

Et ça donne :

```
1478605245 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605305 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605365 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605425 talk.pdf -- Mais qu'est-ce que je fous? - comment glander et apprendre les maths en même temps
1478605486 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605546 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605606 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605666 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605726 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605786 talk.pdf -- Mais qu'est-ce que je fous? - comment glander et apprendre les maths en même temps
1478605846 emacs@birdsong - talk.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/talk.tex
1478605906 talk.pdf -- Mais qu'est-ce que je fous? - comment glander et apprendre les maths en même temps
1478605966 emacs@birdsong - macros.tex : /home/jeff/src/jma/talks/2016-11__devfest-que-je-fous/macros.tex
```

Et ça donne :

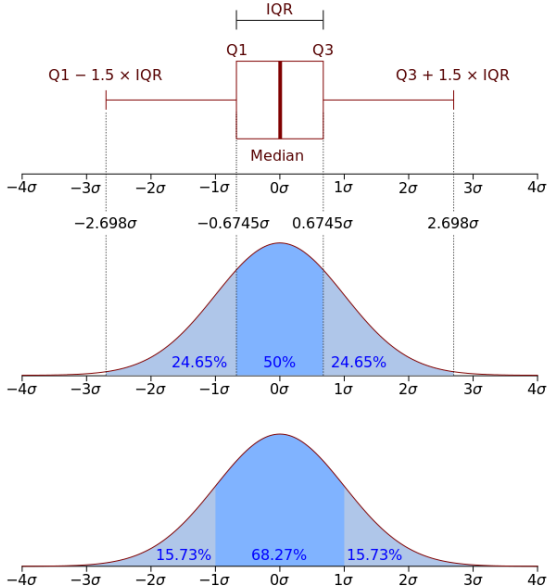
```
1478605245 emacs@birdsong - talk.tex : /home/jeff/
1478605305 emacs@birdsong - talk.tex : /home/jeff/
1478605365 emacs@birdsong - talk.tex : /home/jeff/
1478605425 talk.pdf -- Mais qu'est-ce que je fous
1478605486 emacs@birdsong - talk.tex : /home/jeff/
1478605546 emacs@birdsong - talk.tex : /home/jeff/
1478605606 emacs@birdsong - talk.tex : /home/jeff/
1478605666 emacs@birdsong - talk.tex : /home/jeff/
1478605726 emacs@birdsong - talk.tex : /home/jeff/
```

Quelles questions puis-je poser?

Machine Learning

Data Science

Statistics

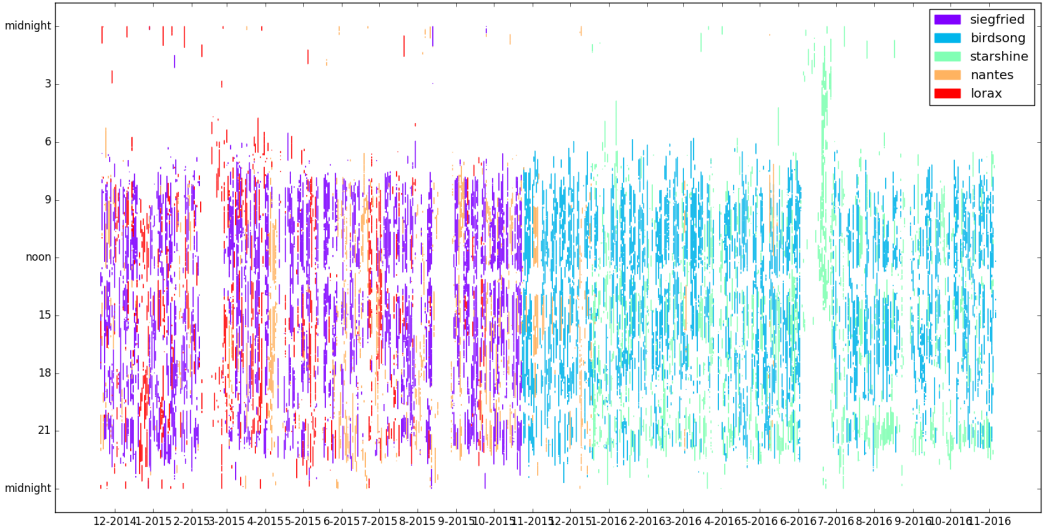


Vector Space

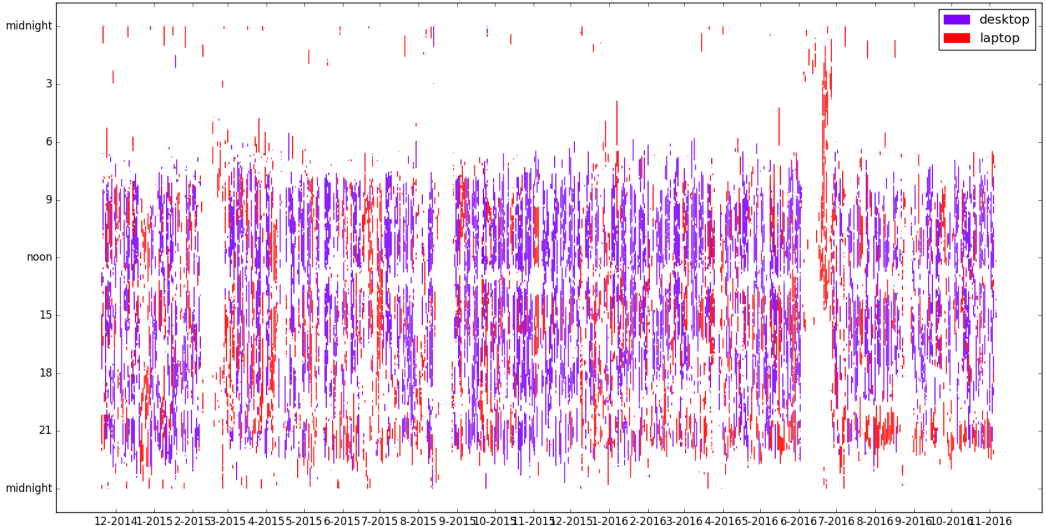
Features

Feature engineering

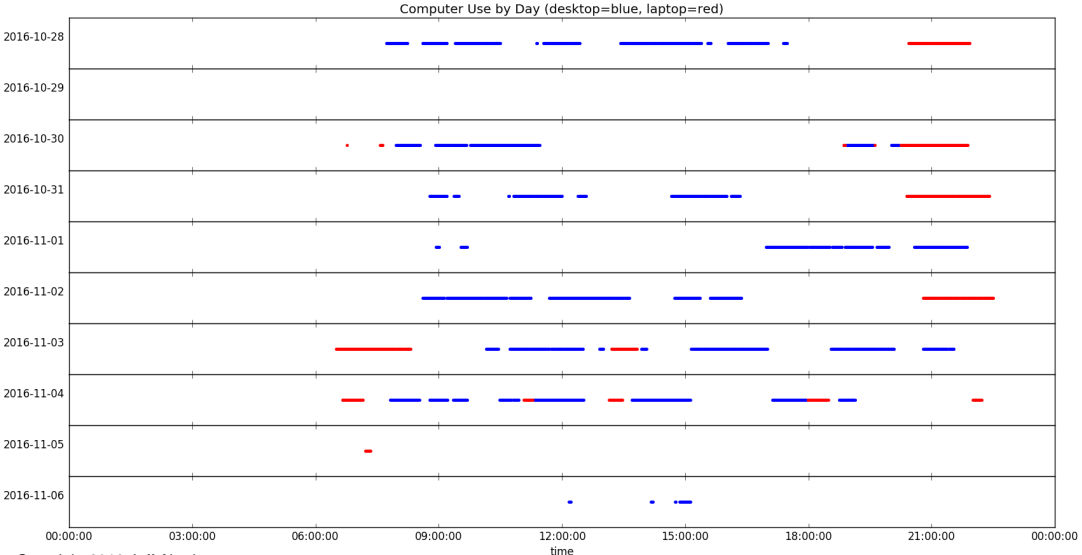
Activity by host



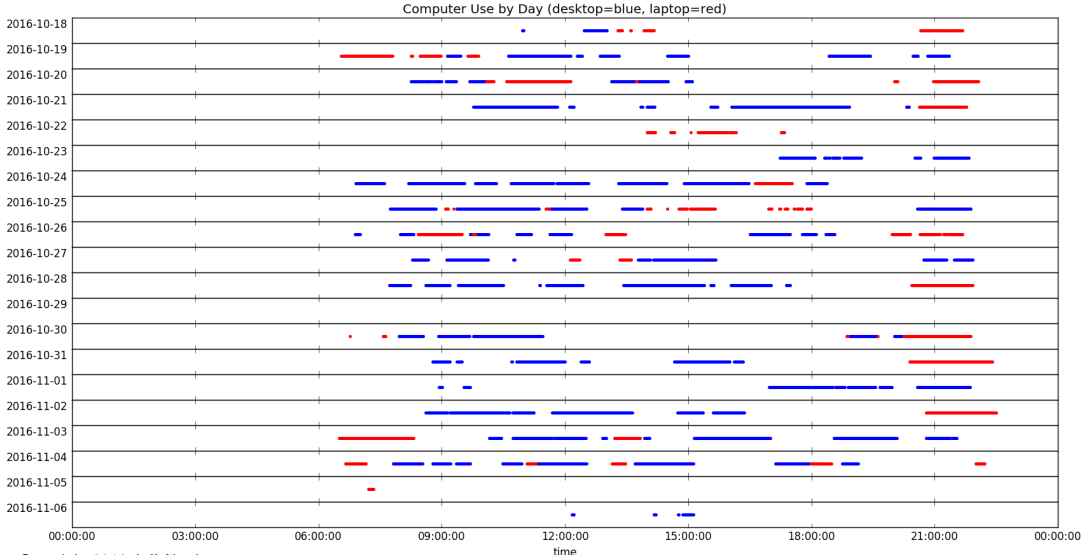
Activity by host class



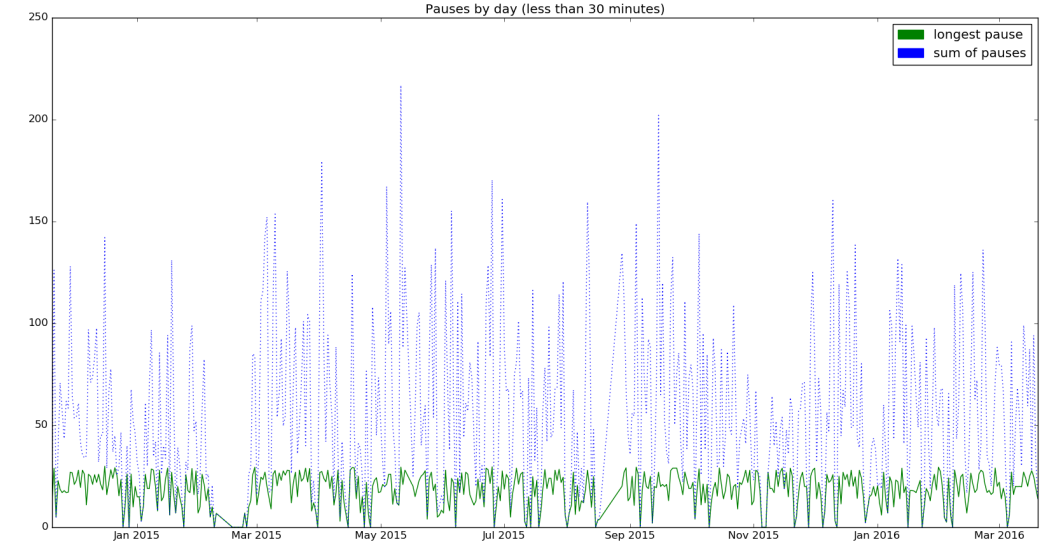
Recent activity



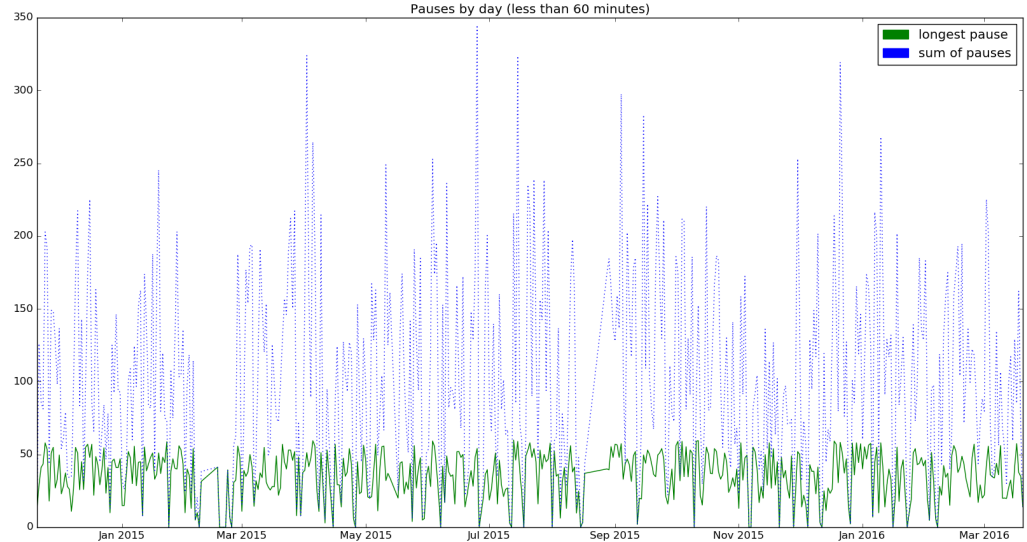
Recent activity



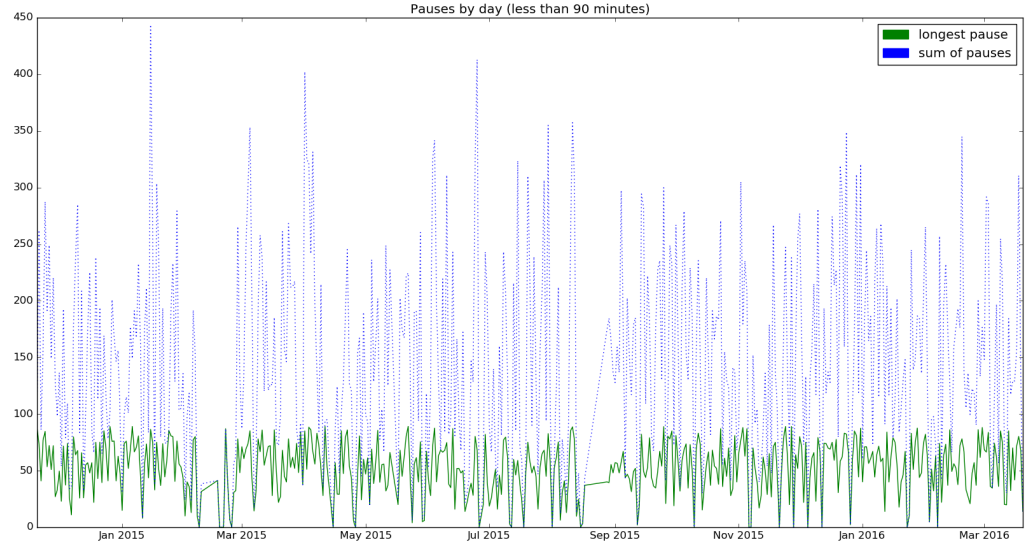
Pauses



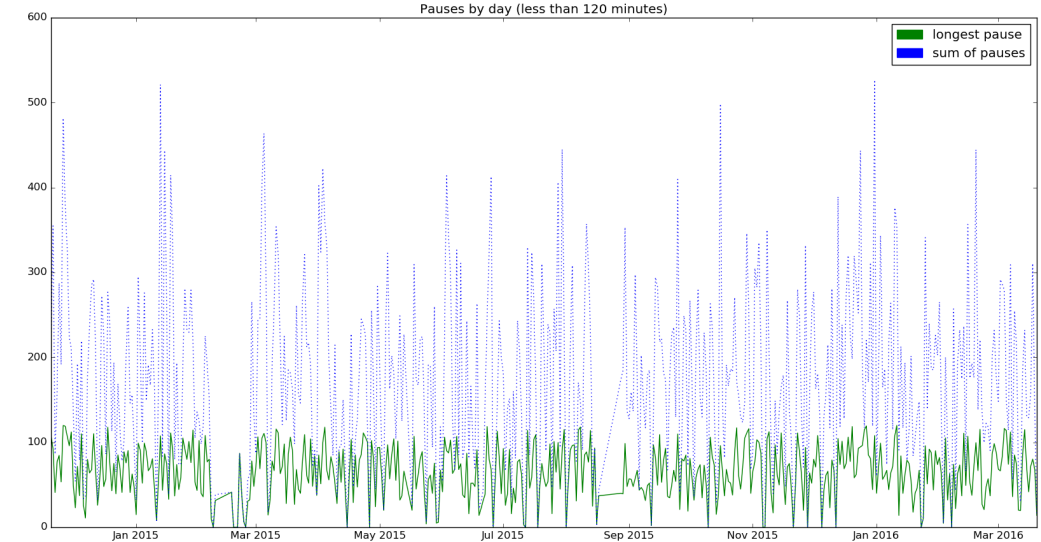
Pauses



Pauses



Pauses



sac de mots

Bag of Words

```
Le chat est orange.
Le chien court vite.
```

Bag of Words

```
6   1    7    2
Le chat est orange.
Le chien court vite.
6    3     4    5
```

Bag of Words

```
6    1     7     2
Le chat est orange.
Le chien court vite.
6     3      4     5
```

Bag of Words

```
[[6, 1, 7, 2],

 [6, 3, 4, 5]]
```

Bag of Words

```
[[6, 1, 7, 2]
 [1, 1, 0, 0, 0, 1, 1]
 [0, 0, 1, 1, 1, 1, 0]
 [6, 3, 4, 5]]
```

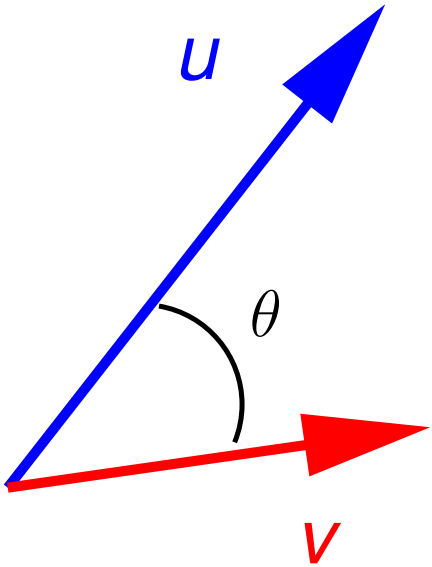
Bag of Words

```
[1, 1, 0, 0, 0, 1, 1]
[0, 0, 1, 1, 1, 1, 0]
```

Bag of Words

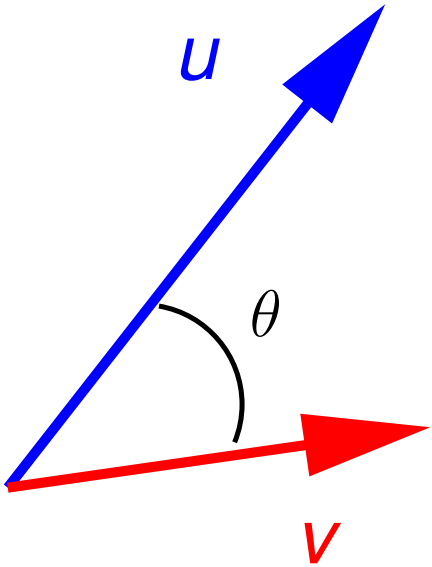
```
Le chat est orange.
 [1, 1, 0, 0, 0, 1, 1]
 [0, 0, 1, 1, 1, 1, 0]
Le chien court vite.
```

Cosine Similarity



$$\cos\theta = \frac{u \cdot v}{\| u \| \| v \|}$$

Cosine Similarity



$$\cos\theta = u \cdot v$$

(if u and v have norm 1)

Cosine Similarity

```
Le chat est orange.
 [1, 1, 0, 0, 0, 1, 1]
 [0, 0, 1, 1, 1, 1, 0]
Le chien court vite.
```

Cosine Similarity

$u = [1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 1]$

$v = [0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0]$

$u \cdot v = 0 + 0 + 0 + 0 + 0 + 1 + 0 = 1$

$$\cos\theta = \frac{u \cdot v}{\| u \| \| v \|} = \frac{1}{\sqrt{4} \cdot \sqrt{4}} = \frac{1}{4}$$

Hamming Distance

$$u = [1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 1]$$
$$v = [0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0]$$

$$H(u, v) = 0 + 0 + 0 + 0 + 0 + 1 + 0$$

Jaccard Index

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Bigrams

more context

Bigrams

Le chat est orange.
Le chien court vite.

Bigrams

Le chat est orange.
Le chien court vite.

{ le, chat, est, orange, chien, court, vite,
  le chat, chat est, est orange,
  le chien, chien court, court vite }

Bag of Words

Exemple :

Il est nuit. La cabane est pauvre, mais bien close.
Le logis est plein d'ombre et l'on sent quelque chose
Qui rayonne à travers ce crépuscule obscur.
Des filets de pêcheur sont accrochés au mur.
Au fond, dans l'encoignure où quelque humble vaisselle
Aux planches d'un bahut vaguement étincelle,
On distingue un grand lit aux longs rideaux tombants.
Tout près, un matelas s'étend sur de vieux bancs,

Bag of Words

Exemple (plus simple) :

Il est nuit. La cabane est pauvre, mais bien close.
Le logis est plein d'ombre et l'on sent quelque chose

Bag of Words

Exemple (plus simple) :

Il est nuit. La cabane est pauvre, mais bien close.
Le logis est plein d'ombre et l'on sent quelque chose

```
vectorizer = CountVectorizer(analyzer='word')
ft = vectorizer.fit_transform(pauvres_gens)
ft.todense()

[1, 1, 0, 1, 2, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0]
[0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1]
```

Bag of Words

```
vectorizer = CountVectorizer(analyzer='word')
ft = vectorizer.fit_transform(les_pauvres_gens)
cosine_distance = 1 - cosine_similarity(ft)

for i in range(cosine_distance.shape[0]):
    for j in range(cosine_distance.shape[1]):
        if i < j:
            if cosine_distance[i, j] < .43:
                print('{i:3}, {j:3}: d={d:.2}\n   {t1}\n   {t2}'.
                    i=i, j=j, d=cosine_distance[i, j],
                    t1=les_pauvres_gens[i], t2=les_pauvres_gens[j
```

Bag of Words

Pluie ou bourrasque, il faut qu'il sorte, il faut qu'il aille,
Il n'avait pas assez de peine ; il faut que j'aille

Pluie ou bourrasque, il faut qu'il sorte, il faut qu'il aille,
Quand il verra qu'il faut nourrir avec les nôtres

l s'en va dans l'abîme et s'en va dans la nuit.
Or, la nuit, dans l'ondée et la brume, en décembre,

Bag of Words

```
bigram_vectorizer = CountVectorizer(ngram_range=(1,2))
bigram_ft = bigram_vectorizer.fit_transform(les_pauvres_gens)
bigram_analyze = bigram_vectorizer.build_analyzer()
bigram_analyze(phrase_1)
bigram_cosine_distance = 1 - cosine_similarity(bigram_ft)

for i in range(bigram_cosine_distance.shape[0]):
    for j in range(bigram_cosine_distance.shape[1]):
        if i < j:
            if bigram_cosine_distance[i, j] < .43:
                print('{i:3}, {j:3}: d={d:.2}\n   {t1}\n   {t2}'.
                    i=i, j=j, d=bigram_cosine_distance[i, j],
                    t1=les_pauvres_gens[i], t2=les_pauvres_gens[j
```

Bag of Words

```
'Comme il faut calculer la marée et le vent !'
```

```
['comme', 'il', 'faut', 'calculer', 'la',
'marée', 'et', 'le', 'vent', 'comme il', 'il
faut', 'faut calculer', 'calculer la', 'la
marée', 'marée et', 'et le', 'le vent']
```

Bag of Words

Pluie ou bourrasque, il faut qu'il sorte, il faut qu'il aille,
Quand il verra qu'il faut nourrir avec les nôtres

Comme il faut calculer la marée et le vent !
Et l'onde et la marée et le vent en colère.

Qu'est-ce donc que Jeannie a fait chez cette morte ?
Qu'est-ce donc que Jeannie emporte en s'en allant ?

TF - IDF

$$TF_{td} = \frac{f_{td}}{\max_k f_{kd}} \qquad IDF_t = \log_2 \left(\frac{N}{n_t} \right)$$

$$TF\text{-}IDF_{td} = TF_{td} \cdot IDF_t$$

with

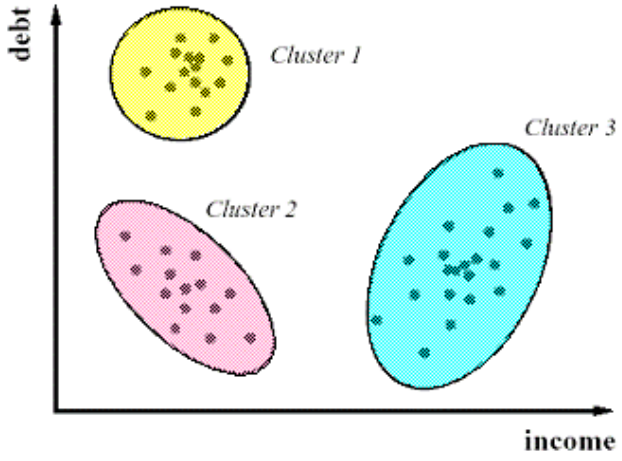
f_{td} = frequency of word (term) t in document d

N = number of documents

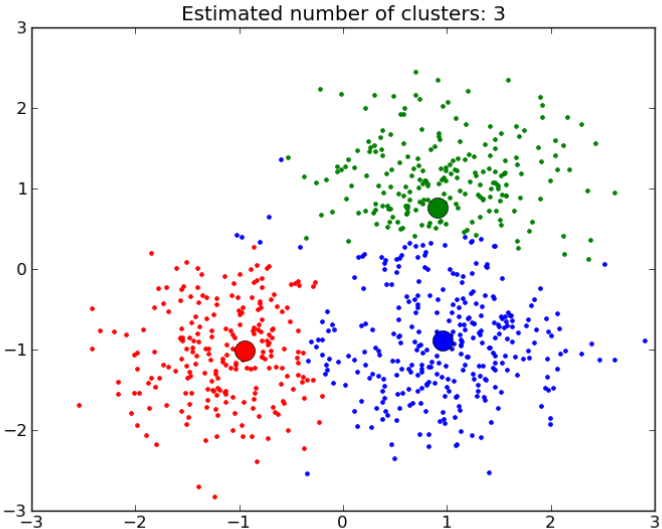
n_t = number of documents containing term t

k-means

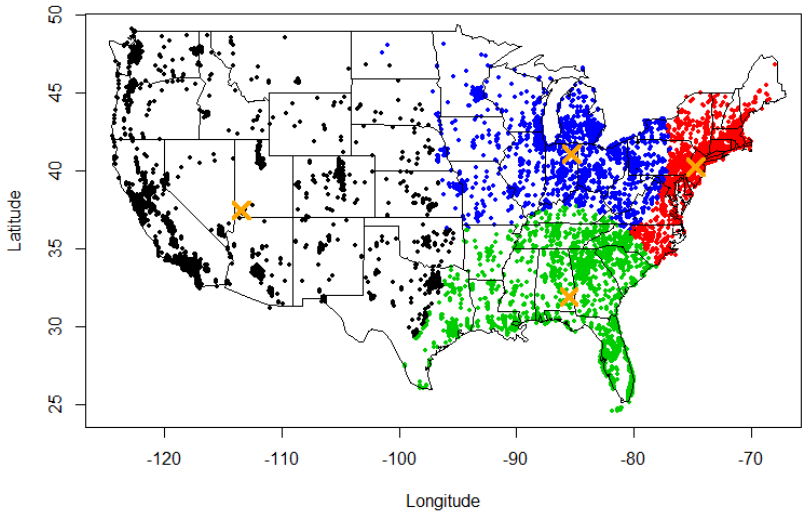
k-means



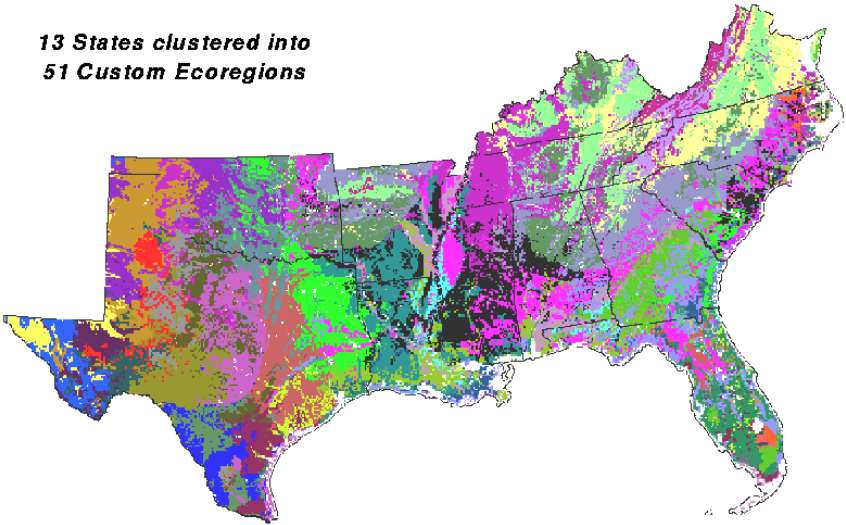
k-means



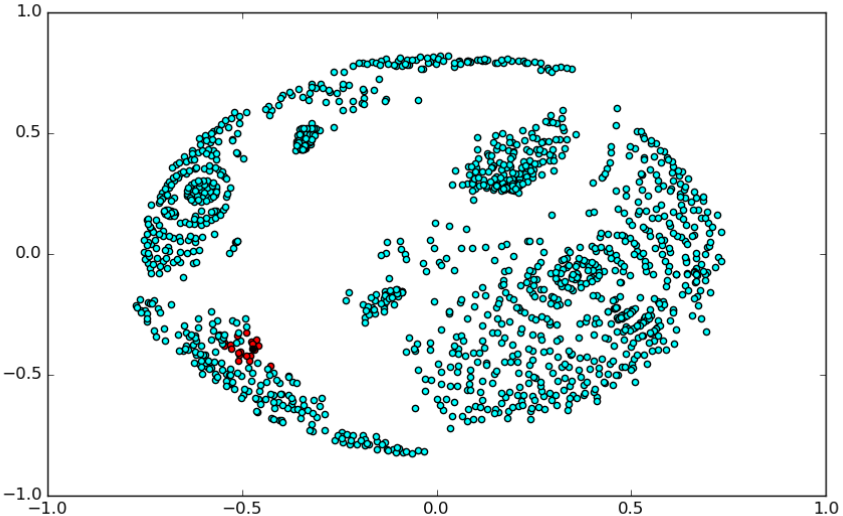
k-means



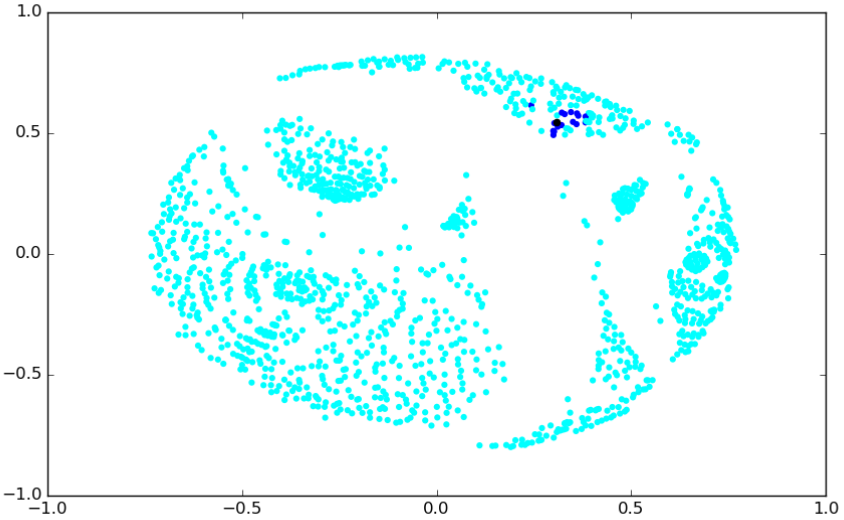
k-means



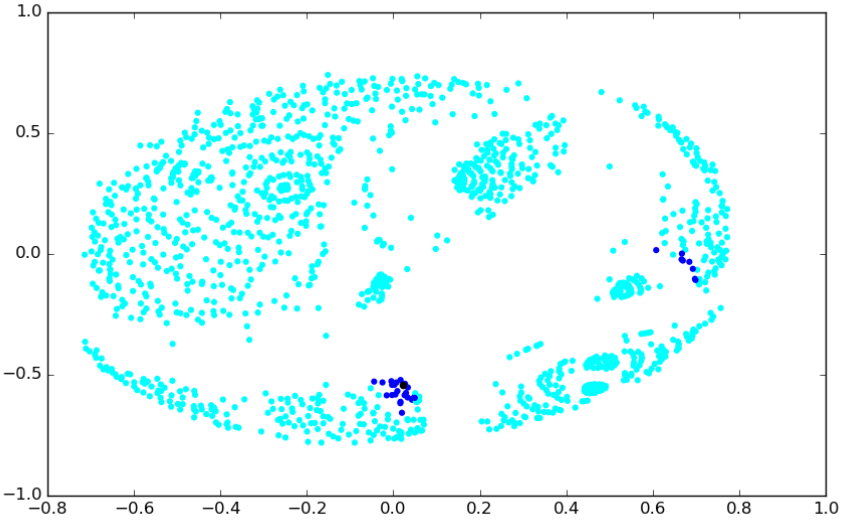
k Nearest Neighbours



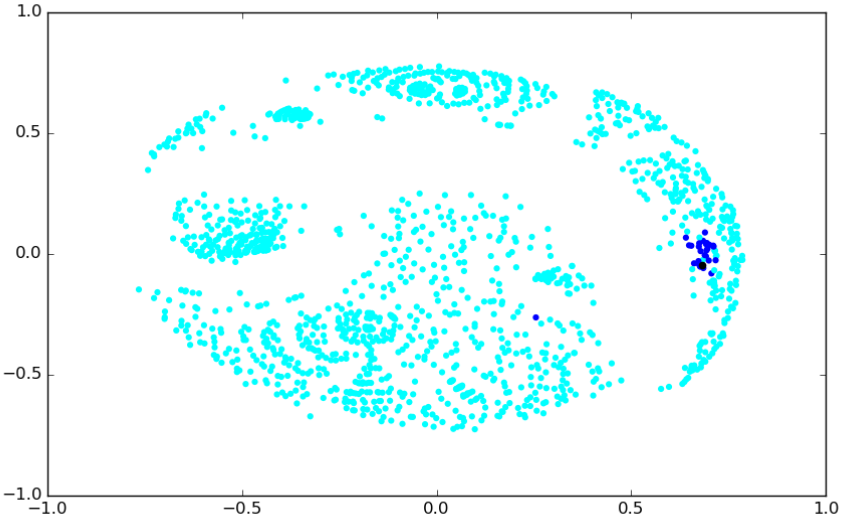
k Nearest Neighbours



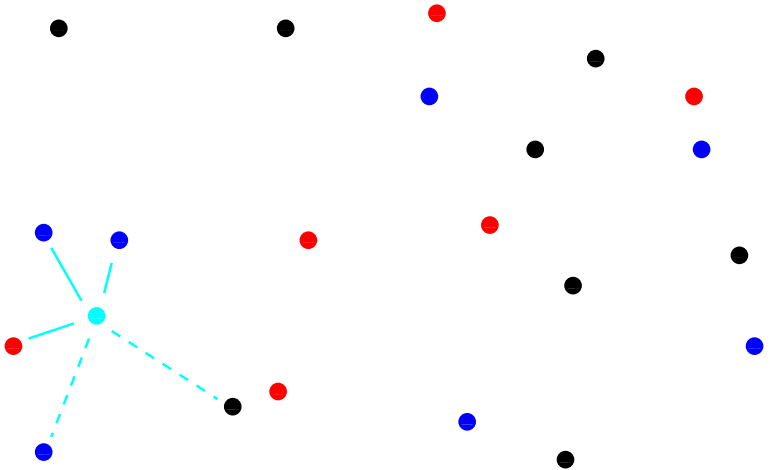
k Nearest Neighbours



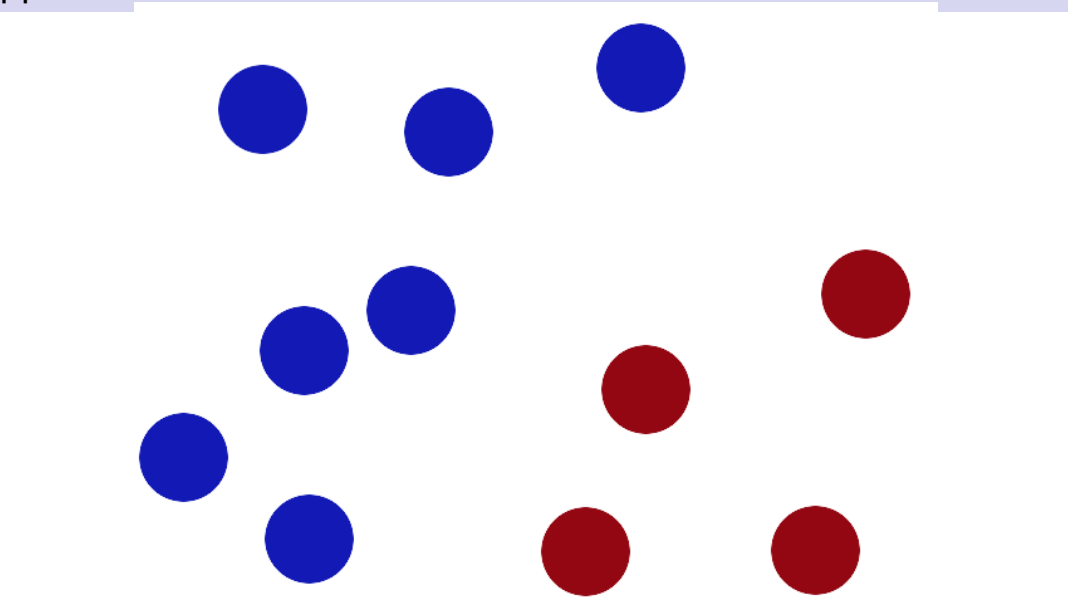
k Nearest Neighbours



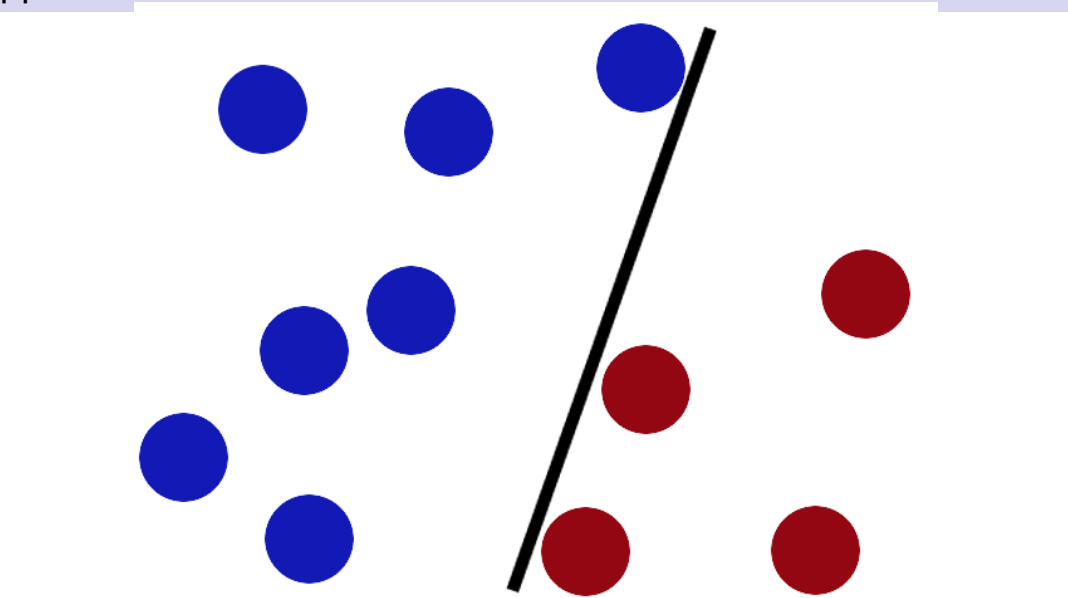
k Nearest Neighbours



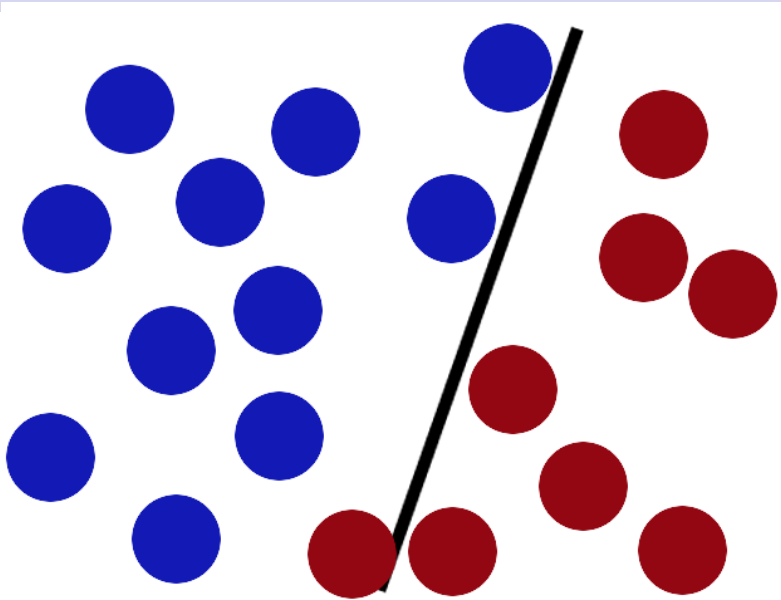
Support Vector Machine



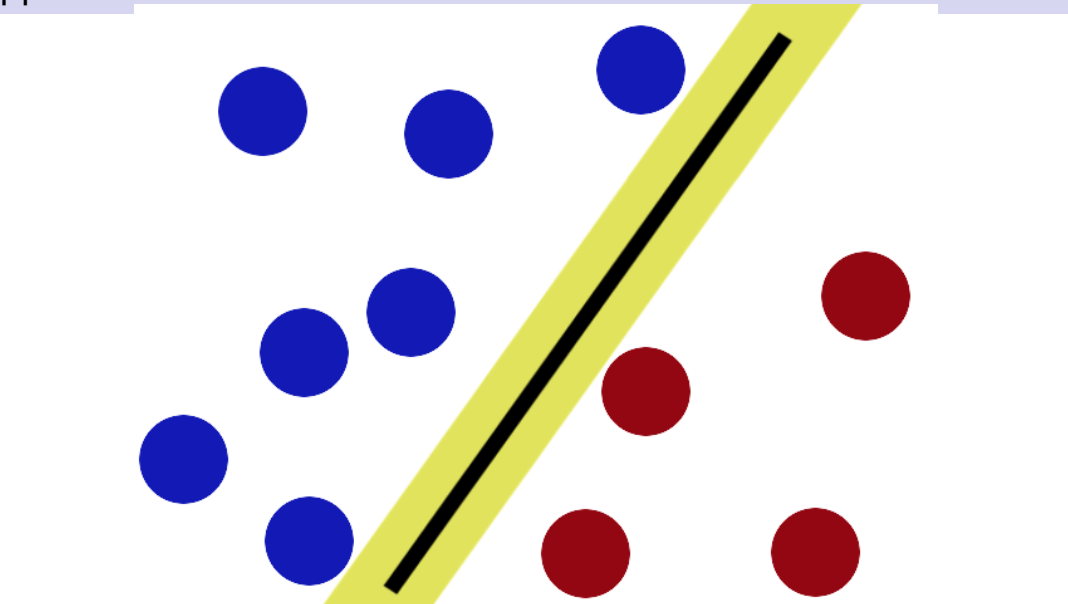
Support Vector Machine



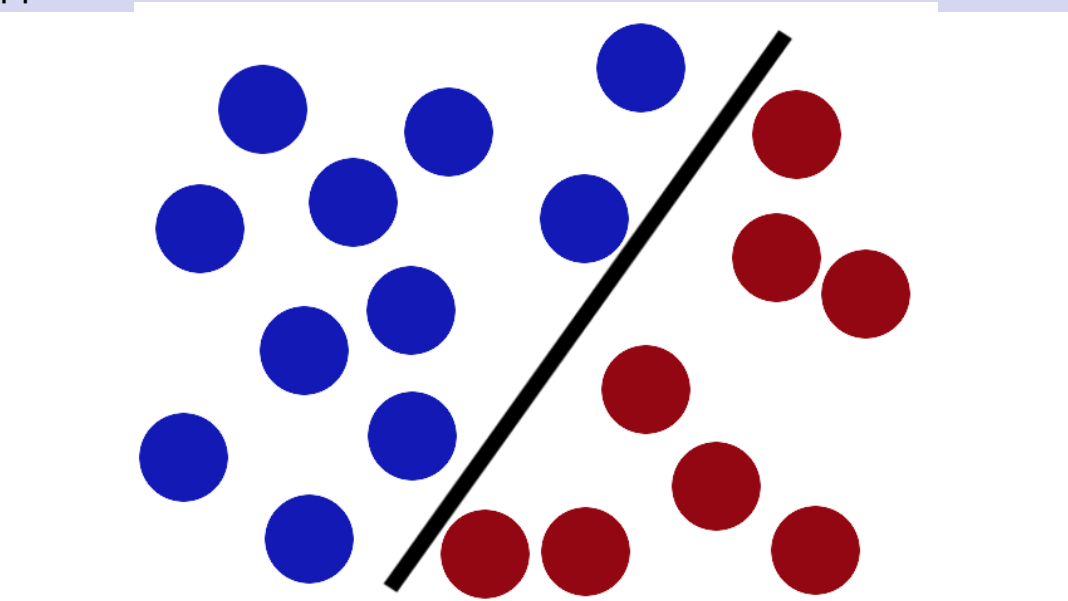
Support Vector Machine



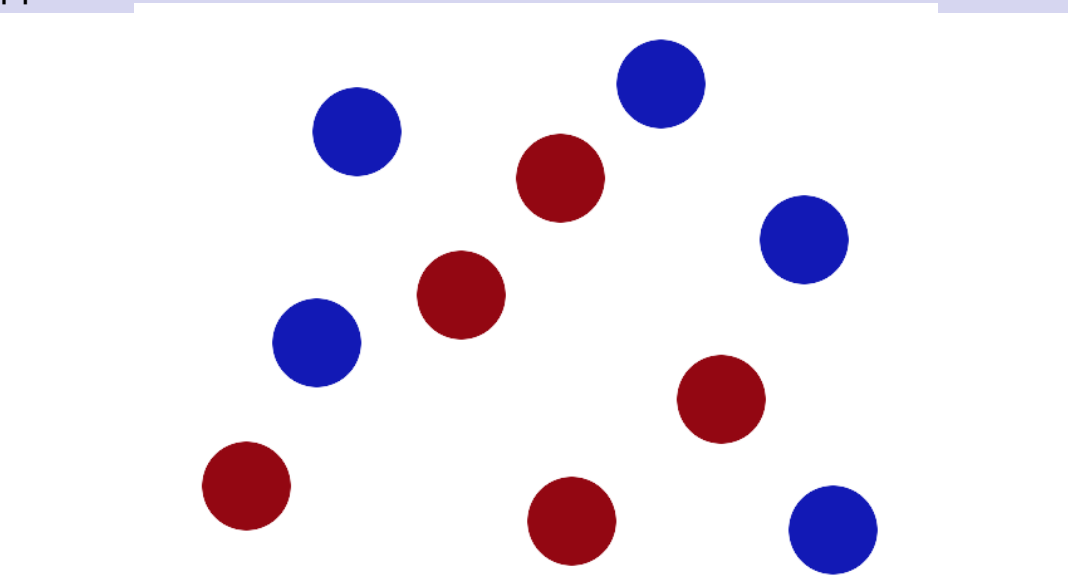
Support Vector Machine



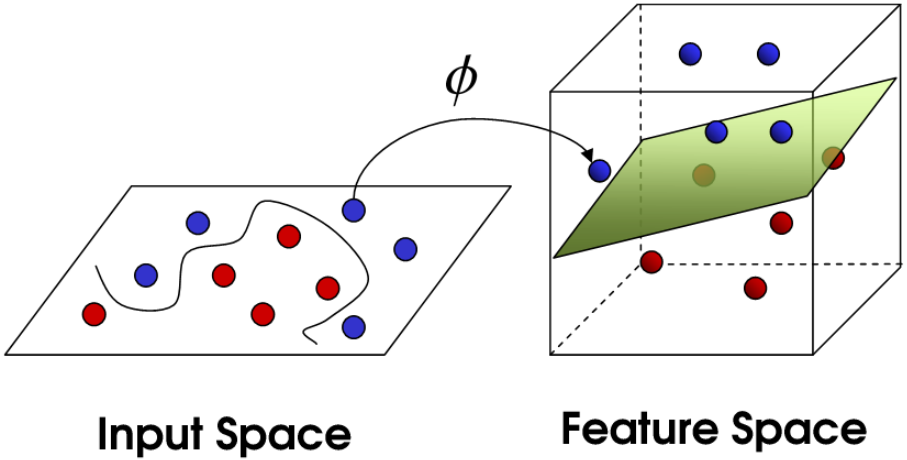
Support Vector Machine



Support Vector Machine

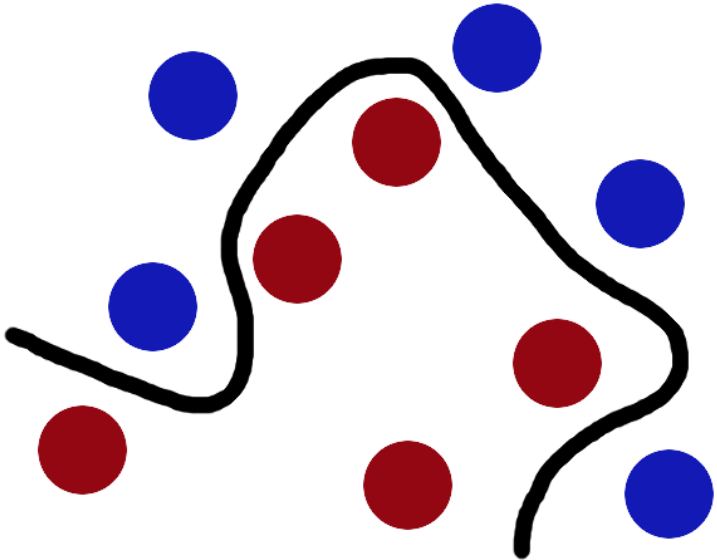


Support Vector Machine

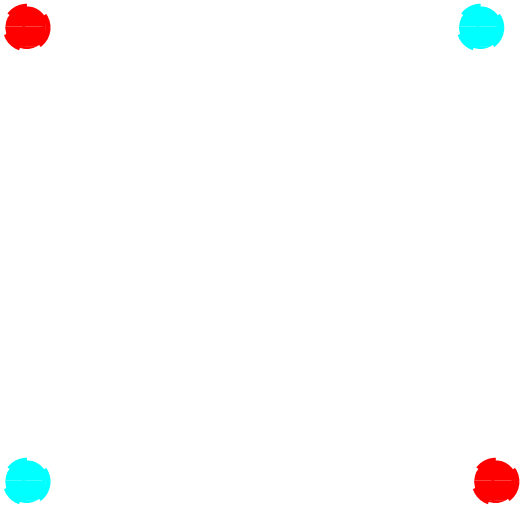


kernel trick

Support Vector Machine



Support Vector Machine



Support Vector Machine

Goals

point $\mapsto$ label

Goals

labels so far $\mapsto$ next labels

Predicting the Future

Formulation:

$$(x_1, \ldots, x_{100}) \rightsquigarrow (x_{101}, \ldots, x_{110})$$

A (too) quick example

Linear neuron

$$y = b + \sum_i x_i w_i$$

Linear neuron

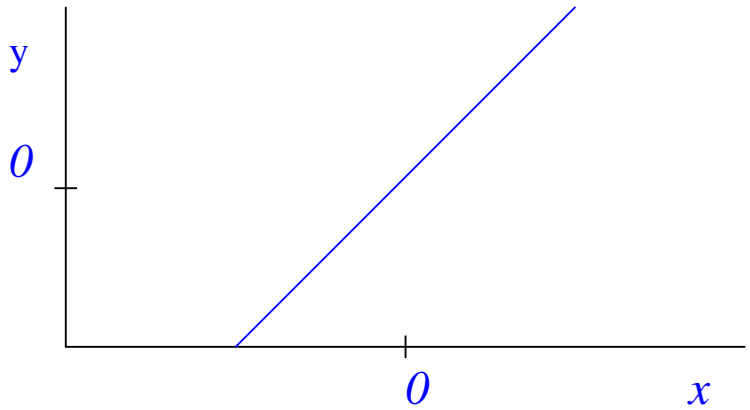
$$y = b + \sum_i x_i w_i$$

where

$$y = \text{output}$$
$$b = \text{bias}$$
$$x_i = i^{\text{th}} \text{ input}$$
$$w_i = \text{weight on } i^{\text{th}} \text{ input}$$

Linear neuron

$$y = b + \sum_i x_i w_i$$



Example: handwriting recognition of digits

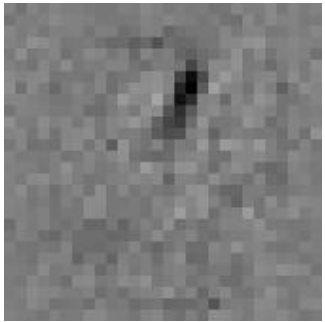
- Input neurons: pixels
- Output neurons: classes (digits)
- Connect them all! *(bipartite)*

Example: handwriting recognition of digits

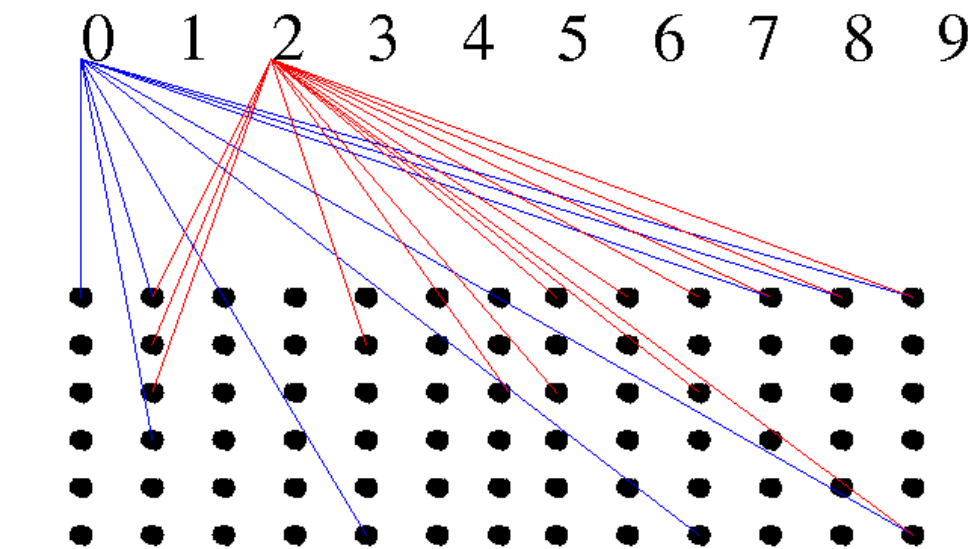
- Input neurons: pixels
- Output neurons: classes (digits)
- Connect them all! *(bipartite)*
- Initialize input weights to random

Example: handwriting recognition of digits

- Input neurons: pixels
- Output neurons: classes (digits)
- Connect them all! *(bipartite)*
- Initialize input weights to random



Example: handwriting recognition of digits



Example: handwriting recognition of digits

To train this ANN:

- Increment weights from active pixels going to correct class
- Decrement weights from active pixels going to predicted class

Example: handwriting recognition of digits

To train this ANN:

- Increment weights from active pixels going to correct class
- Decrement weights from active pixels going to predicted class

When it's right, nothing happens. This is good.

Remember about lying?

Do this, but a bit more complicated

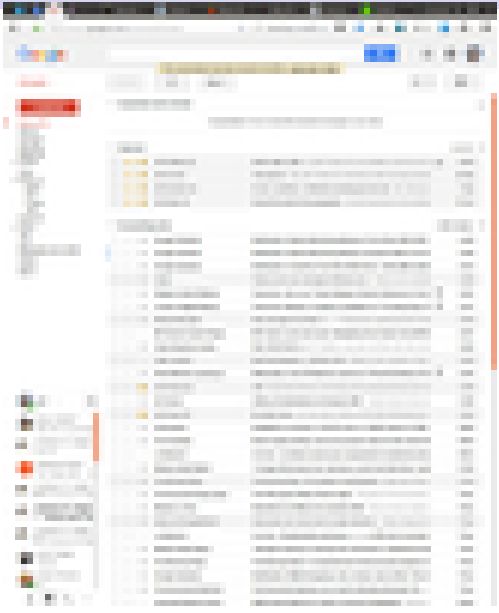
Does it work?

Yes, but it's a work in progress. . .

There’s more!

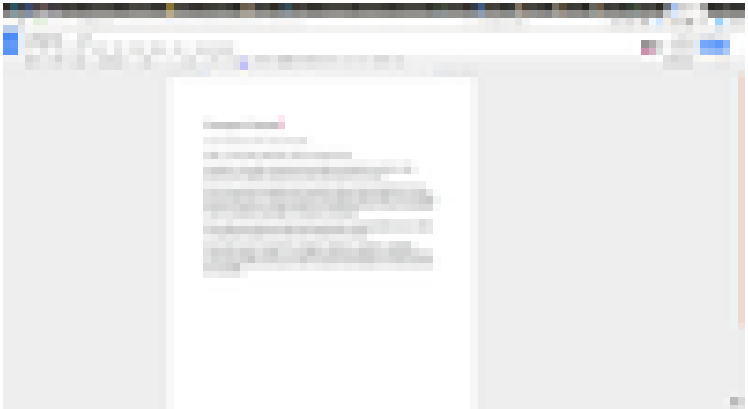
I could capture the active window

There's more!



There’s more!

There’s more!



There's more!

Resources

http://www.meetup.com/Nantes-Machine-Learning-Meetup/

Resources

http://www.ml-week.com/

Questions?