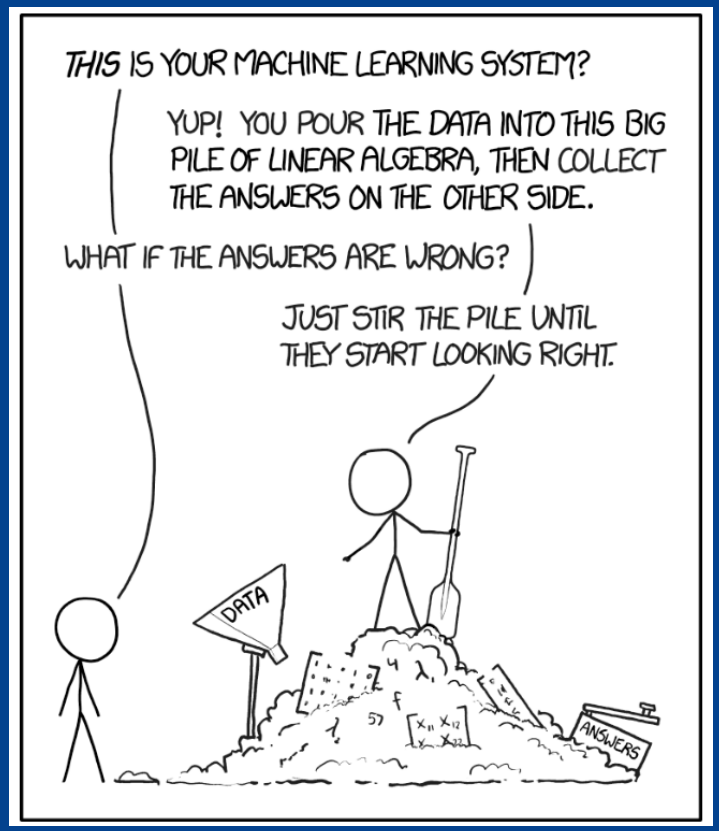
AARHUS UNIVERSITY

Lesson 12: Real-world Examples, Dimensionality Reduction, Unsupervised Learning (and Reinforced Learning)

CARSTEN EIE FRIGAARD

SPRING 2023

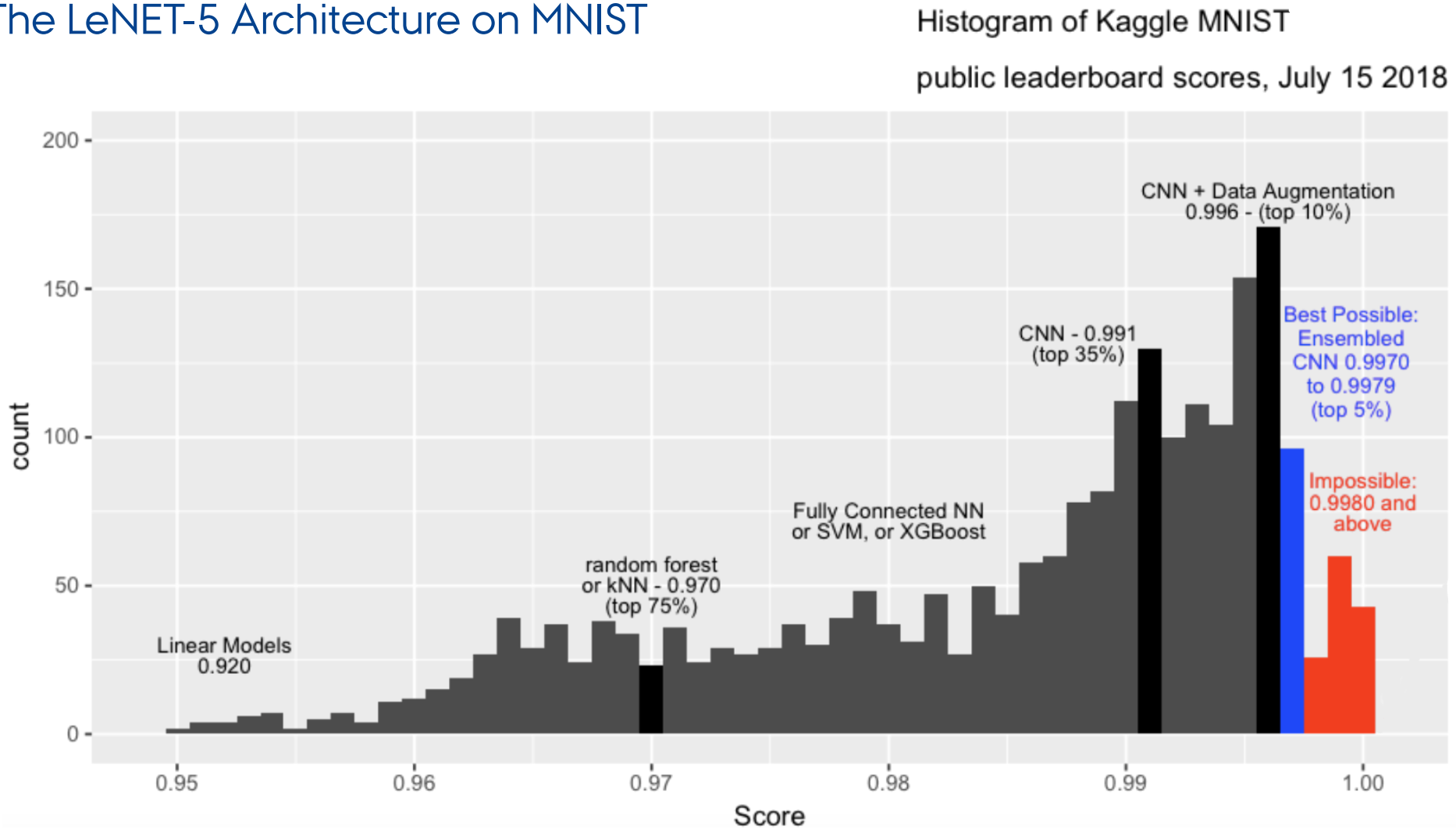


Agenda

- MNIST Search Quest Hi-Score Cake,
- Real-world Examples,
- Dimensionality Reduction,
- Unsupervised Learning,
- (Reinforced Learning.)

Qd MNIST Search Quest II

The LeNET-5 Architecture on MNIST



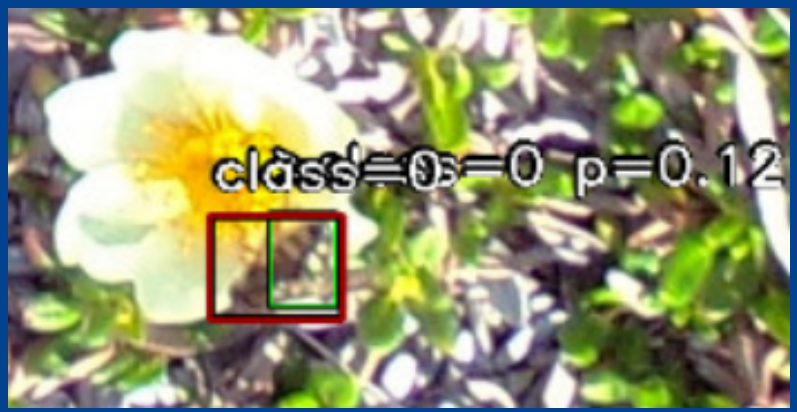
Qd MNIST Search Quest II

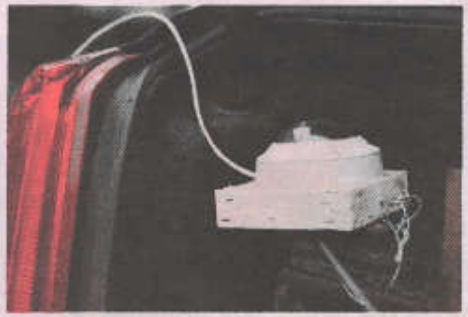
```
E22:  Grp05:  score=0.983, SVC, F22:  Grp05:  score=0.990, SGDClassifier
E21:  Grp28:  score=0.973, KNN, F21:  Grp20:  score=0.979, SVC
```

F23: Grp01, SVC traning time??

```
19/04 09:09> Grp 01:  best:  dat=mnist, score=0.98095, model=SVC(..)
             CTOR: SVC(C=2, degree=1, gamma='auto', kernel='poly')
Grp01:  score=0.98095, model=SVC(..)
Grp08:  score=0.97788, model=SVC(..)
Grp02:  score=0.97673, model=SVC(..)
Grp01:  score=0.97143, model=SVC(..)
Grp29:  score=0.96780, model=RandomForestClassifier(..)
Grp12:  score=0.96682, model=RandomForestClassifier(..)
Grp04:  score=0.96680, model=RandomForestClassifier(..)
Grp15:  score=0.96603, model=RandomForestClassifier(..)
Grp09:  score=0.91192, model=SGDClassifier(..)
Grp07:  score=0.90908, model=SGDClassifier(..)
Grp24:  score=0.90284, model=SGDClassifier(..)
Grp22:  score=0.89976, model=SGDClassifier(..)
Grp11:  score=0.89651, model=SGDClassifier(..)
Grp10:  score=0.88965, model=SGDClassifier(..)
Grp13:  score=0.88465, model=SGDClassifier(..)
Grp05:  score=0.88245, model=SGDClassifier(..)
Grp14:  score=0.87829, model=SGDClassifier(..)
Grp06:  score=0.87765, model=SGDClassifier(..)
Grp20:  score=0.87547, model=SGDClassifier(..)
Grp16:  score=0.86702, model=LinearSVC(..)
```

REAL-WORLD EXAMPLES





»Det er et forsøg på at skabe en ny brugerflade for at skrive. På en måde har jeg jo skrevet en roman med en bil,« fortæller Ross Goodwin om sin AI-forfattede bog *1 the road*

På roadtrip med en insekthjerne

Af MIKKEL BORIS

I 2016 slog computeren AlphaGo den 18-dobbelte verdensmester i brætspillet Go, Lee Sedol. Go er et kompliceret og abstrakt spil, som kræver intuition og kreativitet, men den kunstige intelligens vandt med en række innovative træk overlegent.

Undervejs i spillet troede kommentatorerne

Goodwin til Weekendavisen fra sin lejlighed i Los Angeles.

Inden køreturen havde han brugt måneder på at træne maskinen. Han satte den til at læse et stort korpus af moderne litteratur fra hele verden, så den kunne lære at skrive af de store forfattere.

»Det fungerer ligesom autokorrekturen på din telefon, bare klogere og trænet på en mere litterær kilde. Den skriver bogstav for bogstav, så den har lært sig selv at forudsige det næste

når du dekonstruerer dem. Efter at have læst den i ét stræk og fået turen lidt på afstand har romanen fået en universalitet, så jeg kan projicere mine egne oplevelser ind i teksten,« uddyber Goodwin.

– Du har beskrevet projektet som at lære en insekthjerne at skrive. Hvad betyder det?

»Jeg forsøgte at pointere, at maskinen ikke er på niveau med den menneskelige hjerne. Et artificielt neuralt net er en algoritme, der er lavet, som vi tror, hjerner fungerer. Jeg synes


SmartBus
Powered by holo
Hop ombord
på Danmarks første
førerløse bus.

Hvad siger pressen?

Dato: 26 november 2021..

Sidste dag med selvkørende busser i Aalborg

LÆS OP ORDBOG TEKST

AF
Jesper Knox Sørensen

Efter næsten to års drift er et forsøg med selvkørende busser i Aalborg Øst slut, og der er ingen planer om at genoptage det.

Det til trods for, at beboerne i området har været glade for busserne.

- Det er selvfølgelig ærgerligt for beboerne, for det er dem, vi har sat projektet i søen for at hjælpe, siger projektleder Maria Vestergaard fra Aalborg Kommune.

Hun forventer dog, at vi vil se selvkørende busser igen i fremtiden.

Men det nuværende projekt bliver altså ikke forlænget.

- Vores tilladelse udløber. Og de nuværende køretøjer er efterhånden forældet teknologi.

De selvkørende busser rullede ud på Astrupstien i Aalborg Øst i marts 2020, lige inden den første corona-nedlukning.

om migrantkrise i D

54 MIN. SIDEN
Sidste dag med se
Aalborg

I DAG KL. 09:04
To ministre udspør
minkkommissione

I DAG KL. 09:03
Aarhusiansk borgr
formand for KL: Ra
Venstre

I DAG KL. 08:57
EU-formand efter
ant: Alle fly fra den
bør standses

I DAG KL. 08:40
Udvalg drøfter mul
gendanne sms'er

I DAG KL. 08:30
Sydafrika kalder in

BREAKING
NYHEDER SPORT UNDERHOLDNING

LIGE NU:

ALLE AI OVERSKRIFTER HAVDE MODALVERBER I NUTID (*skal give*)

MEN SÅ KOM DENNE.....

BREAKING NEWS

Første ML artikel i datid: *"Sådan gav ML resultater.."*


e Count: 8
e Count: 0
A T
[0 0]
0 0]
[0 0]
side
front
side
front
front

A computer vision system to monitor the infestation level of Varroa destructor in a honeybee colony

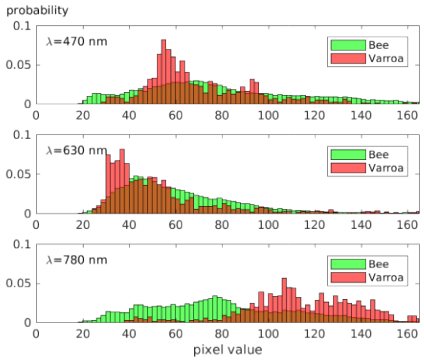


Figure 5: Histograms of bee and varroa pixel intensity values, for the spectral wavelengths 470 nm, 630 nm, and 780 nm respectively, recorded with the JAI camera. The image path is via the mirror-window-mirror, i.e. data were sampled with the setup given in figure 1. The image data for the histogram is the single bee with mite seen in figure 6.

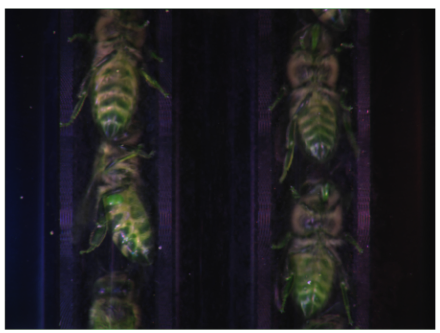


Figure 6: The actual unprocessed camera view of the bees

spectively. The CM analysis was able to rank all wavelengths combinations, using one, two, three or four district wavelengths to give a ranking list of 'best' combination also taking the JAI camera spectrum into account.

The CM value of the actual choose wavelengths combination (470-630-780 nm) gave a rank just below the CM average score. This CM analysis was conducted after picking the actual used wavelengths, so later versions of the VMU might want to investigate a CM combination with a higher rank.

A specially designed diffuser and a number of narrow spectral LE were mounted in the camera focal diffuse illuminat flections.

Figure 6 disp along the passa era, with the gre with the NIR m

2.3.3. Real-time processin

A color and tion of 1296×96 from the camer over two separat sary sustained b ing frames real-t

These data w line post-proces first matching t rally coalescing producing a 24 the later image

Lossless real-t can be applied bandwidth than

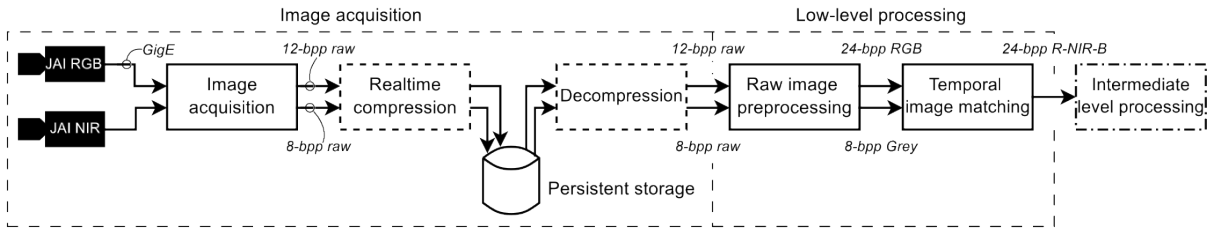


Figure 7: The low-level image processing pipeline. Raw camera images are stored on disk for later retrieval and post-processing. 12- and 8 bits-per-pixel are used as the raw JAI/Bayer packed pixel format for the RGB and IR images respectively. Lossless, real-time compression can be introduced if persistent storage bandwidth is less than the raw-stream image rate of 93 MiB/sec. The 12- and 8-bpp raw images from the network arrives out-of-order with respect to each other, hence the need for the temporal image matching.

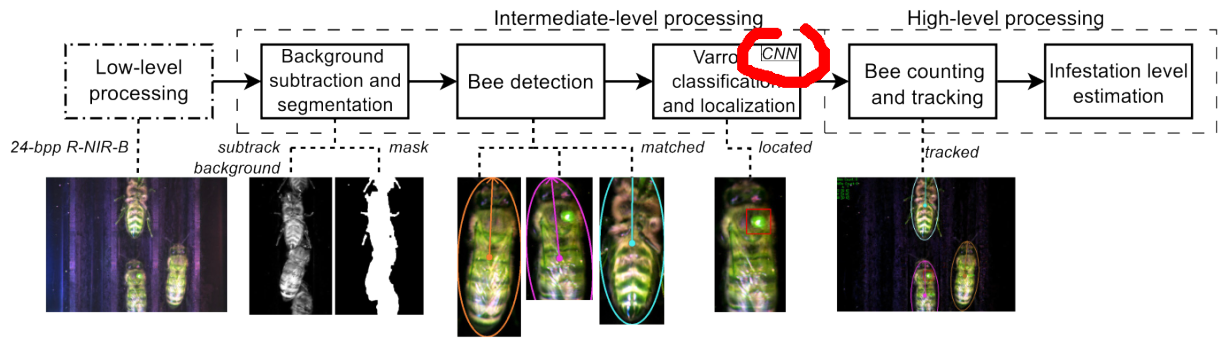
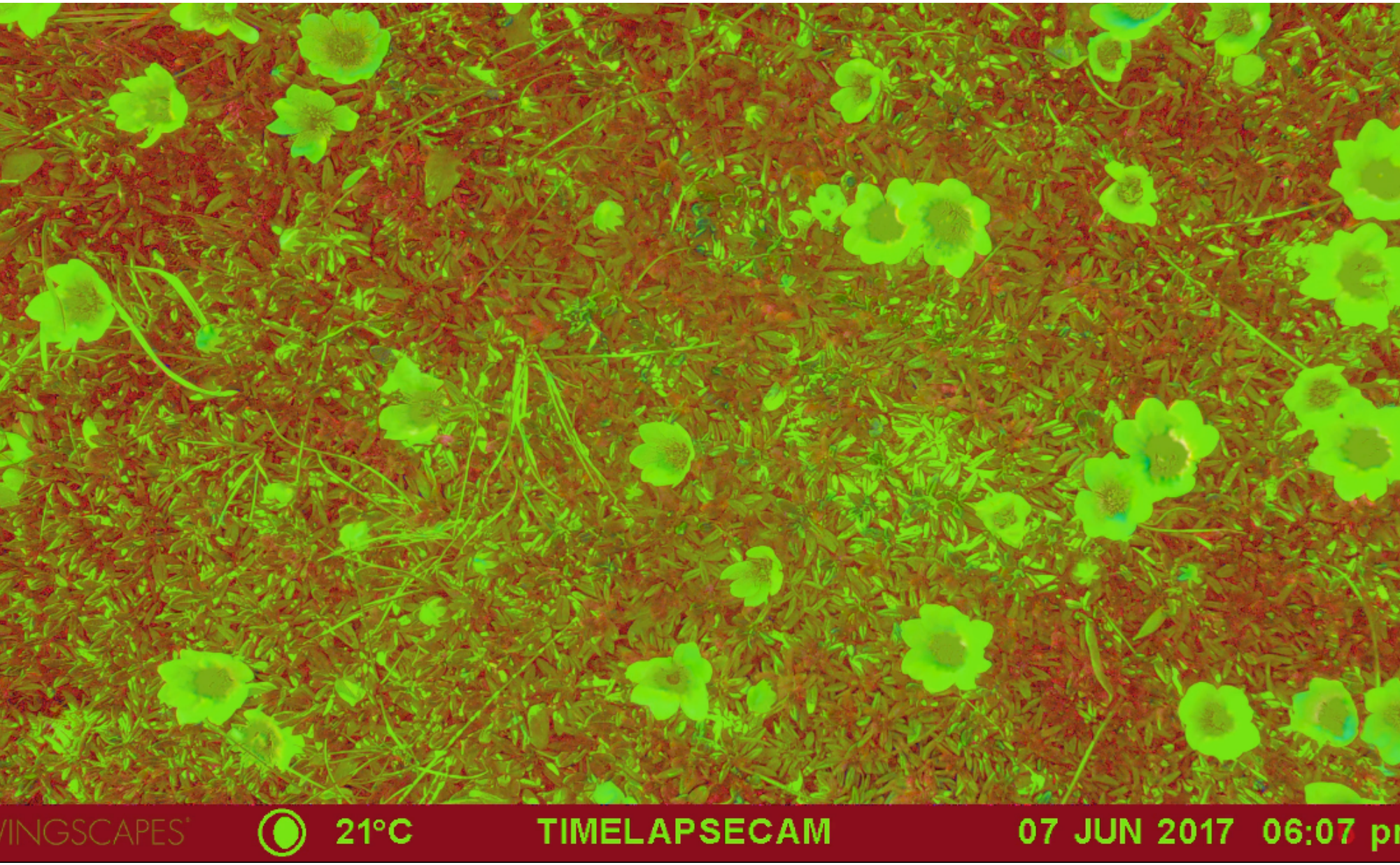
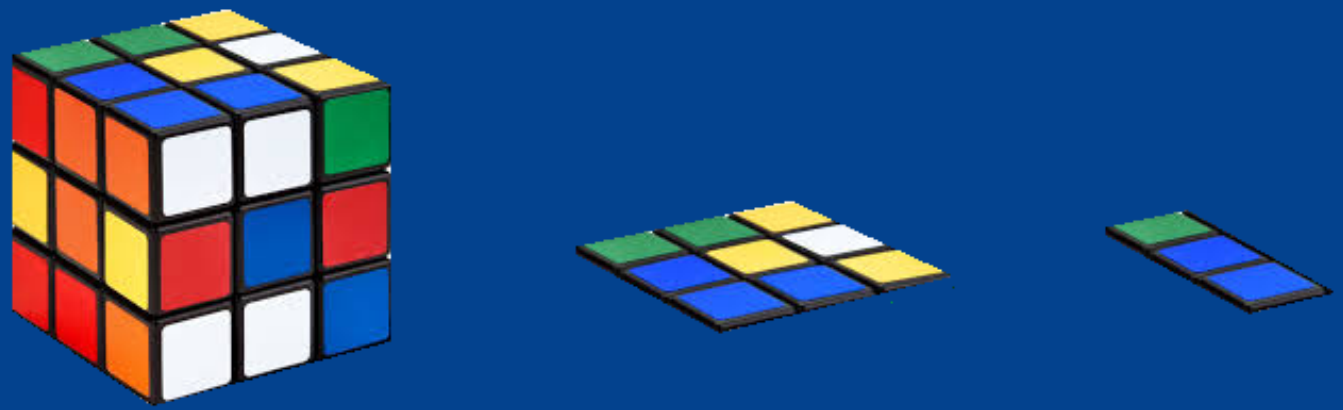


Figure 8: The processing pipeline of the intermediate- to high-level image processing algorithms to analyze and count the number of bees with *Varroa destructor*. A trained convolutional neural network (CNN) was used for the Varroa classification and localization stage.

Insects...where?

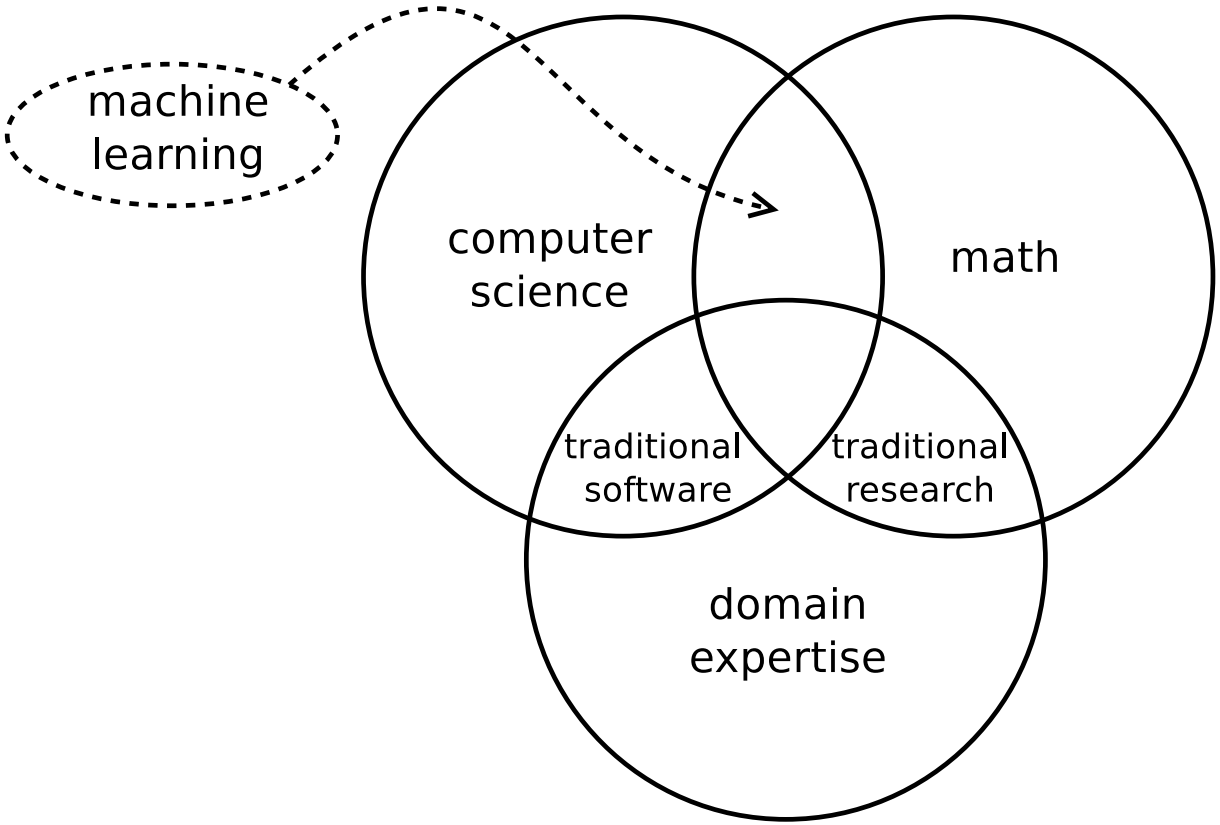


DIMENSIONALITY REDUCTION



Machine learning baggrund

Machine learning: ekspertise

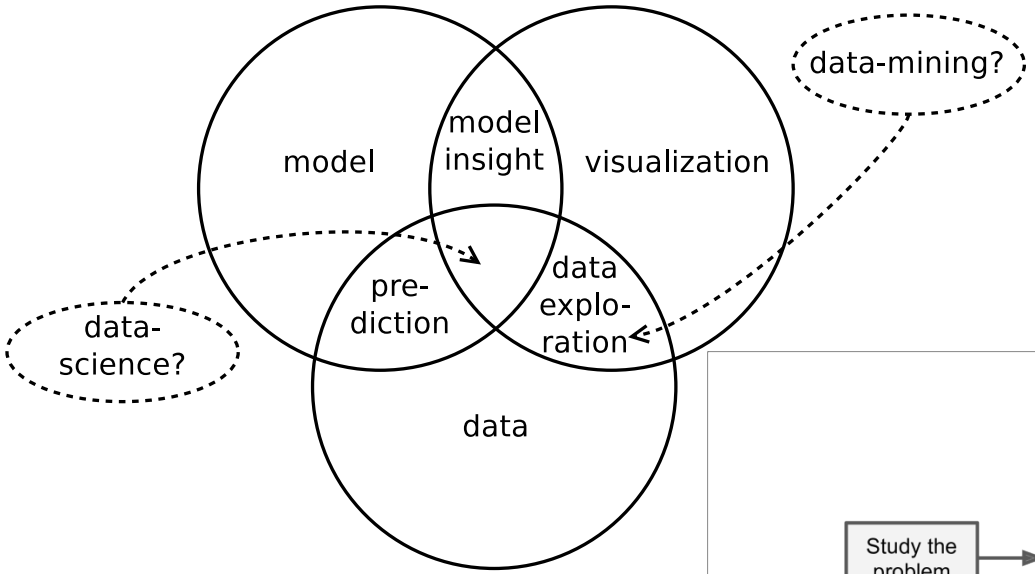


- ML ekspert er computer science og matematik ekspert.
- ML ekspert er IKKE (nødvendigvis) domæne ekspert!

NOTE:
[https://imarticus.org/what-are-the-skills-you-need-to-become-a-machine-learning-engineer/]

Machine learning baggrund

Machine learning: data science ekspert



- fra white-box domæne ekspert til black-box ML data scientist,
- stadig polytekniker:
  - math- og computer science,
  - pattern-recognition,
  - neurocompution,
  - datamining,
  - visualization,
  - etc..

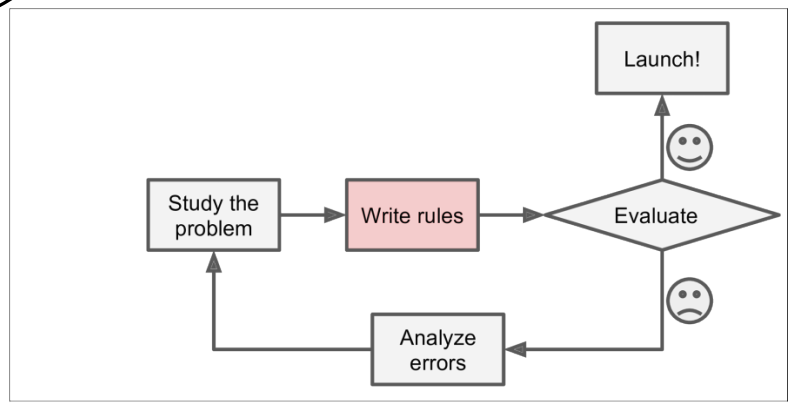


Figure 1-1. The traditional approach

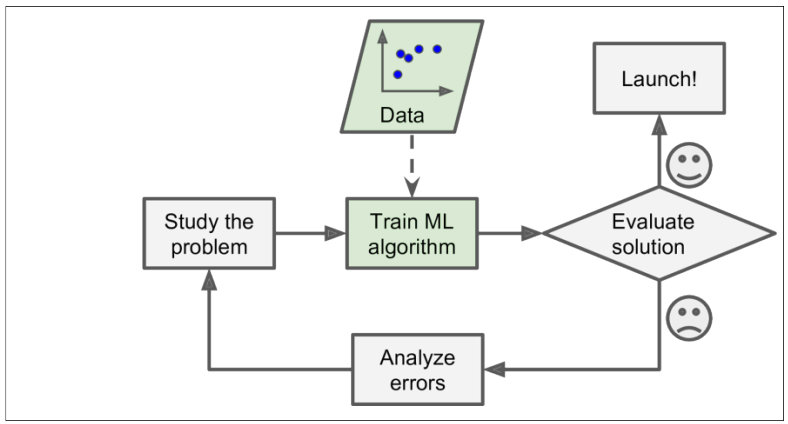


Figure 1-2. Machine Learning approach

The Course of Dimensionality

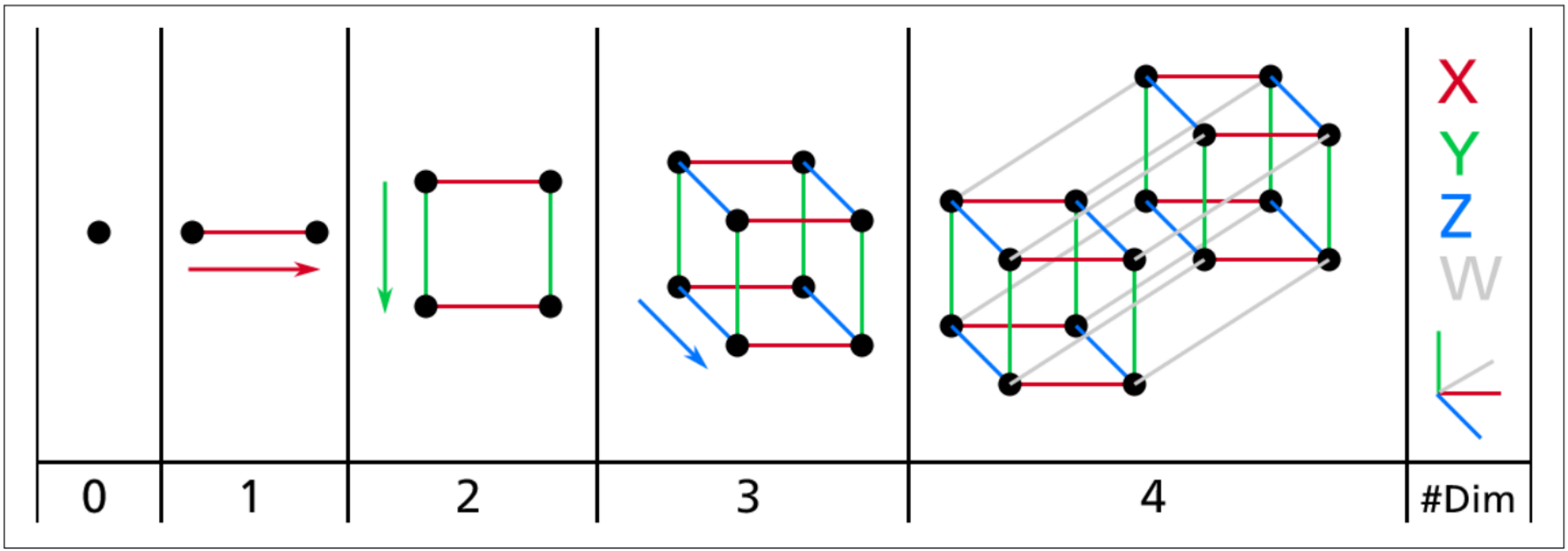


Figure 8-1. Point, segment, square, cube, and tesseract (0D to 4D hypercubes)[2]

Manifold Learning

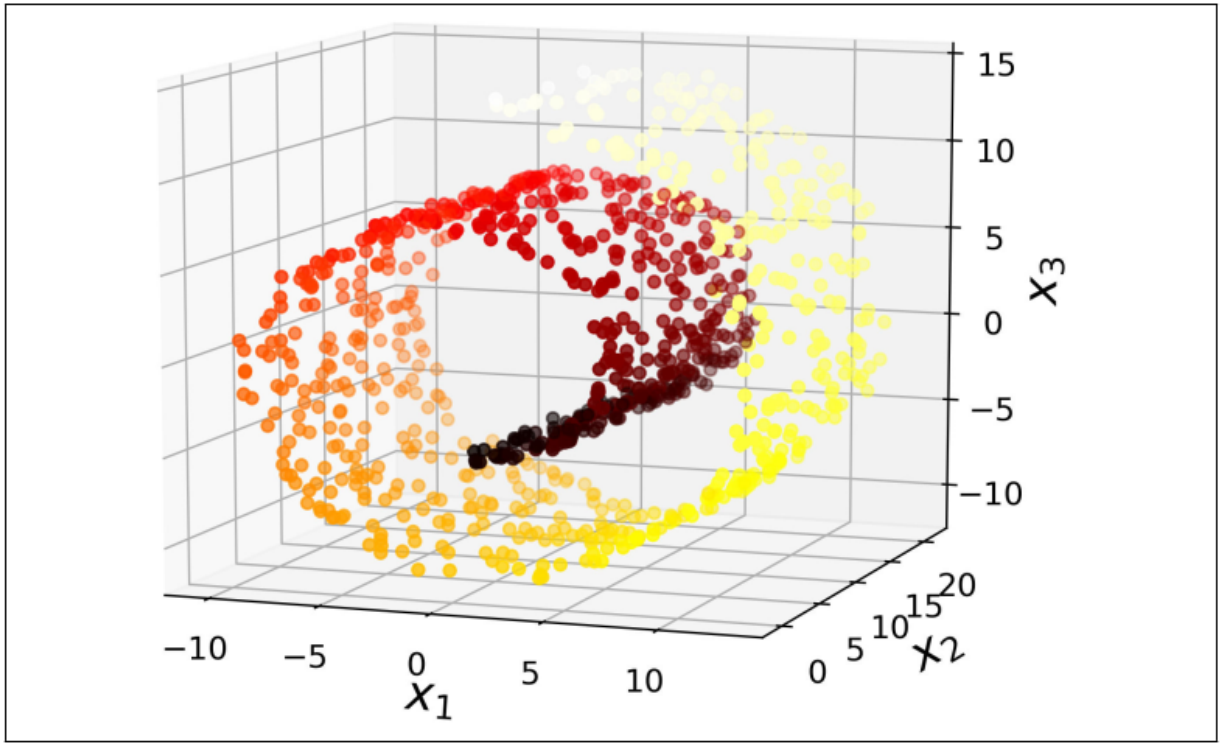


Figure 8-4. Swiss roll dataset

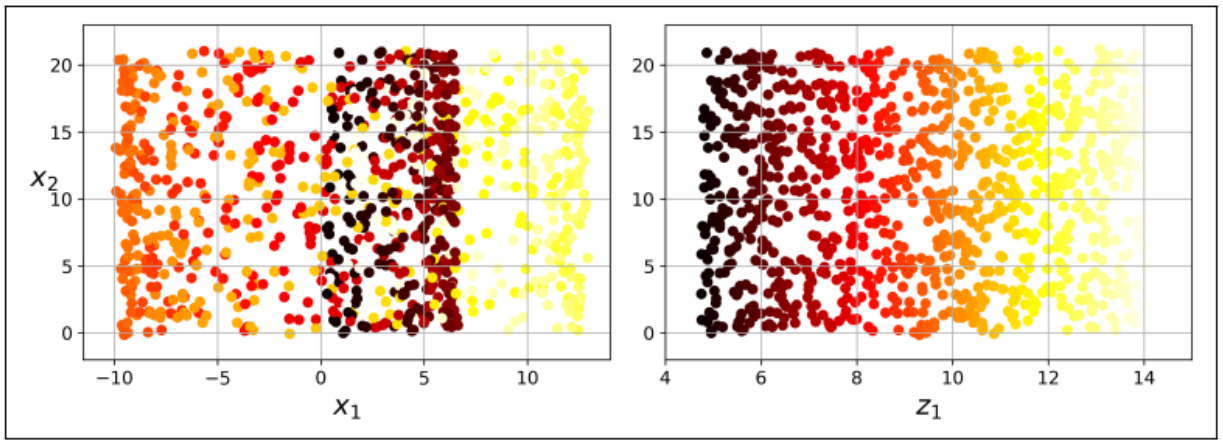


Figure 8-5. Squashing by projecting onto a plane (left) versus unrolling the Swiss roll (right)

Manifold Learning

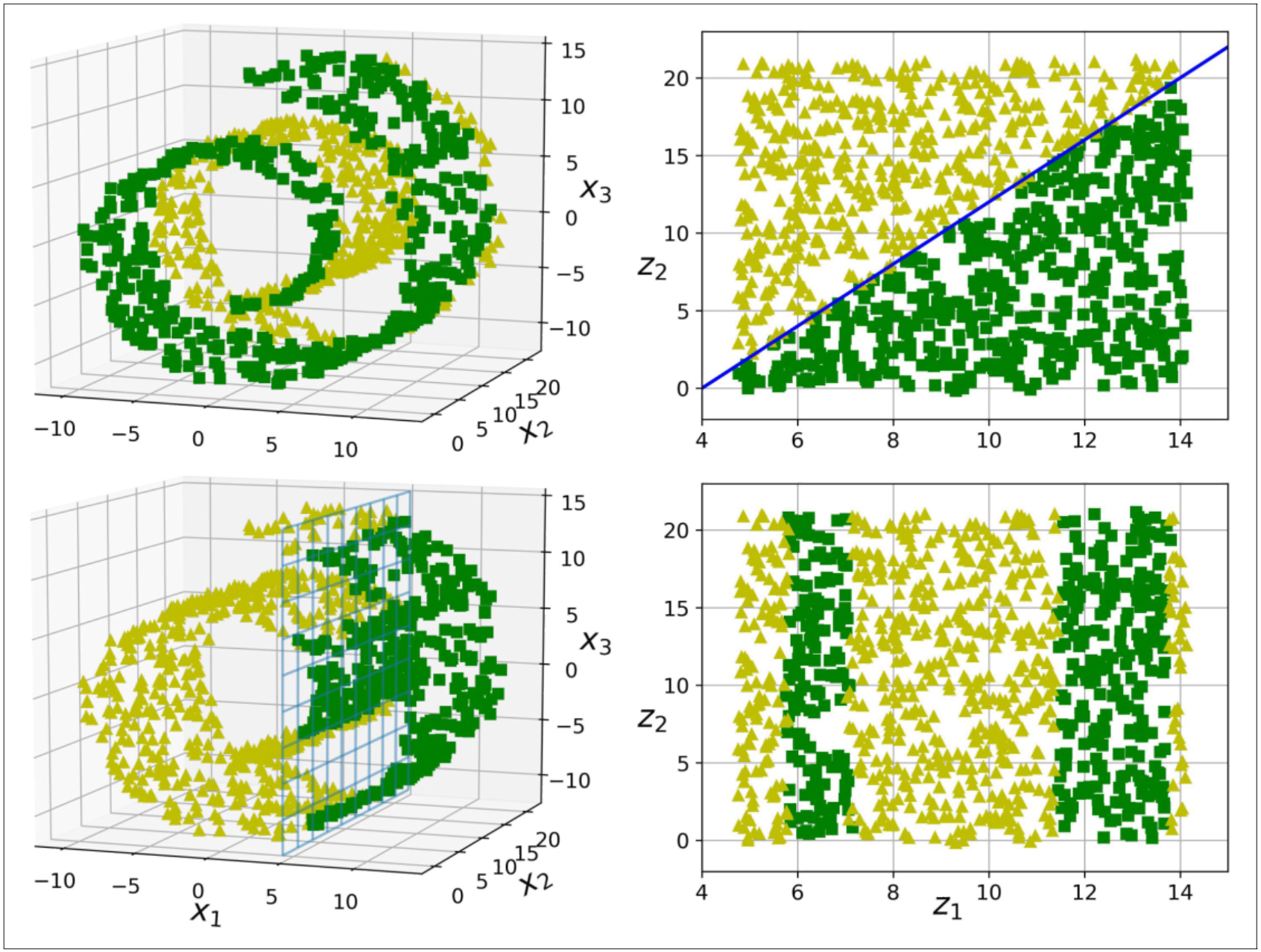
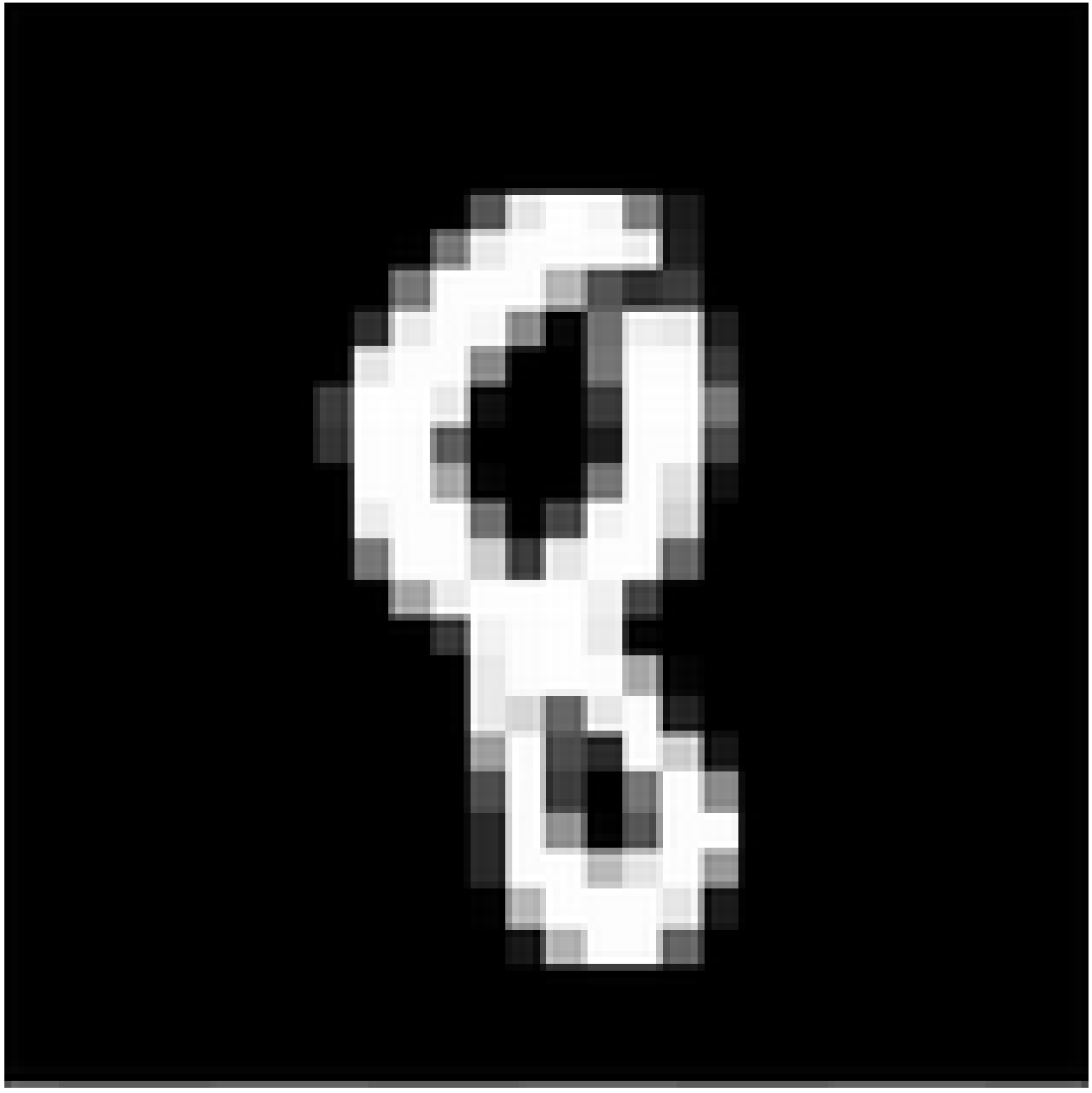


Figure 8-6. The decision boundary may not always be simpler with lower dimensions

Manifold Learning

Random generated 28x28 image? MNIST is on a manifold..



Principle Component Analysis (PCA)

Via Singular Value Decomposition (SVD), $X = U\Sigma V^\top$

Projecting Down to *d* Dimensions

Once you have identified all the principal components, you can reduce the dimensionality of the dataset down to *d* dimensions by projecting it onto the hyperplane defined by the first *d* principal components. Selecting this hyperplane ensures that the projection will preserve as much variance as possible. For example, in Figure 8-2 the 3D dataset is projected down to the 2D plane defined by the first two principal components, preserving a large part of the dataset's variance. As a result, the 2D projection looks very much like the original 3D dataset.

To project the training set onto the hyperplane, you can simply compute the matrix multiplication of the training set matrix $\mathbf{X}$ by the matrix $\mathbf{W}_d$, defined as the matrix containing the first *d* principal components (i.e., the matrix composed of the first *d* columns of $\mathbf{V}$), as shown in Equation 8-2.

Equation 8-2. Projecting the training set down to d dimensions

$$\mathbf{X}_{d\text{-proj}} = \mathbf{X}\mathbf{W}_d$$

The following Python code projects the training set onto the plane defined by the first two principal components:

```
W2 = Vt.T[:, :2]
X2D = X_centered.dot(W2)
```

PCA

Projecting Down to *d* Dimensions

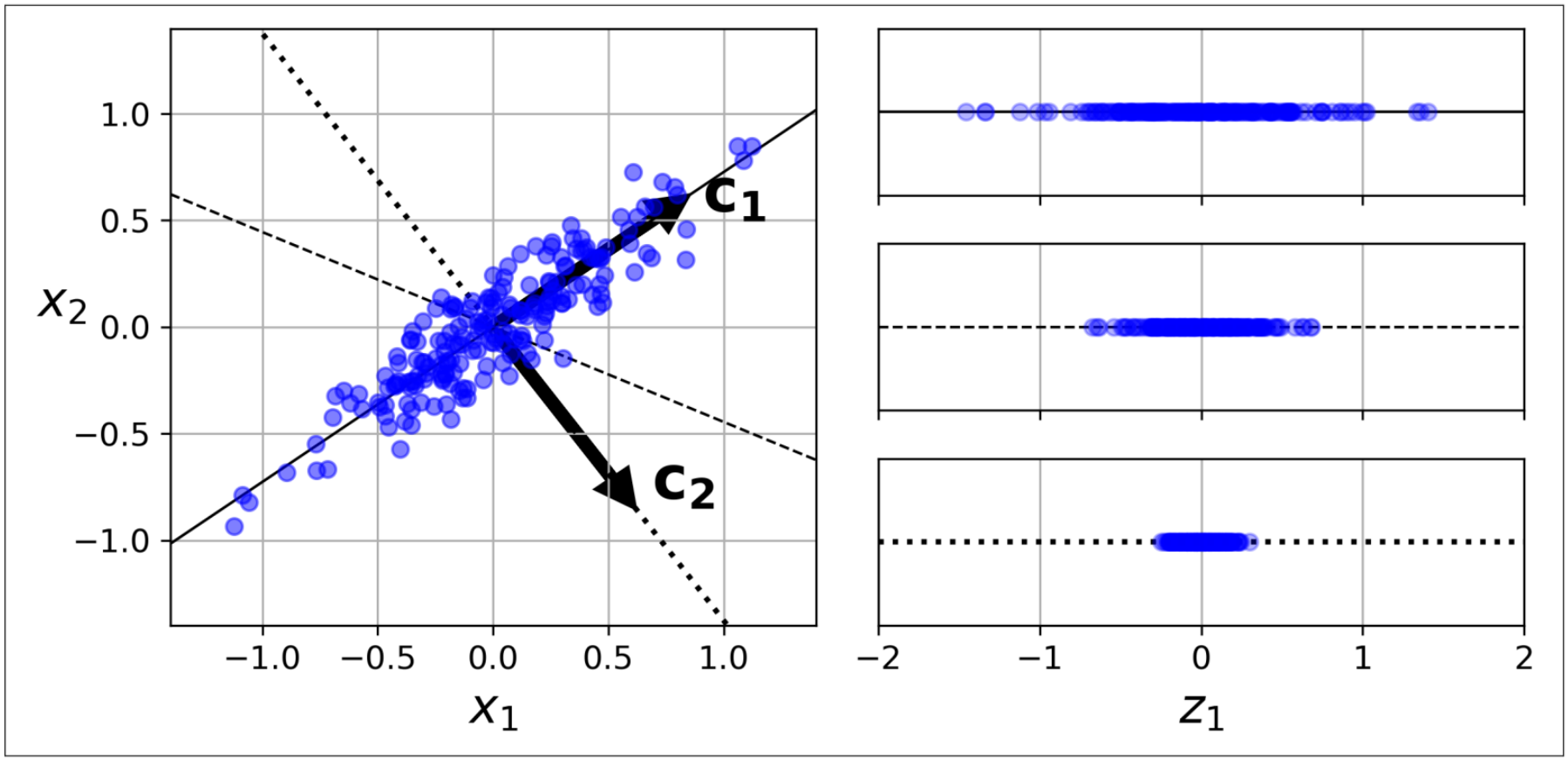


Figure 8-7. Selecting the subspace onto which to project

PCA for Image Compression

```
pca = PCA(n_components = 154)
X_reduced = pca.fit_transform(X_train)
X_recovered = pca.inverse_transform(X_reduced)
```

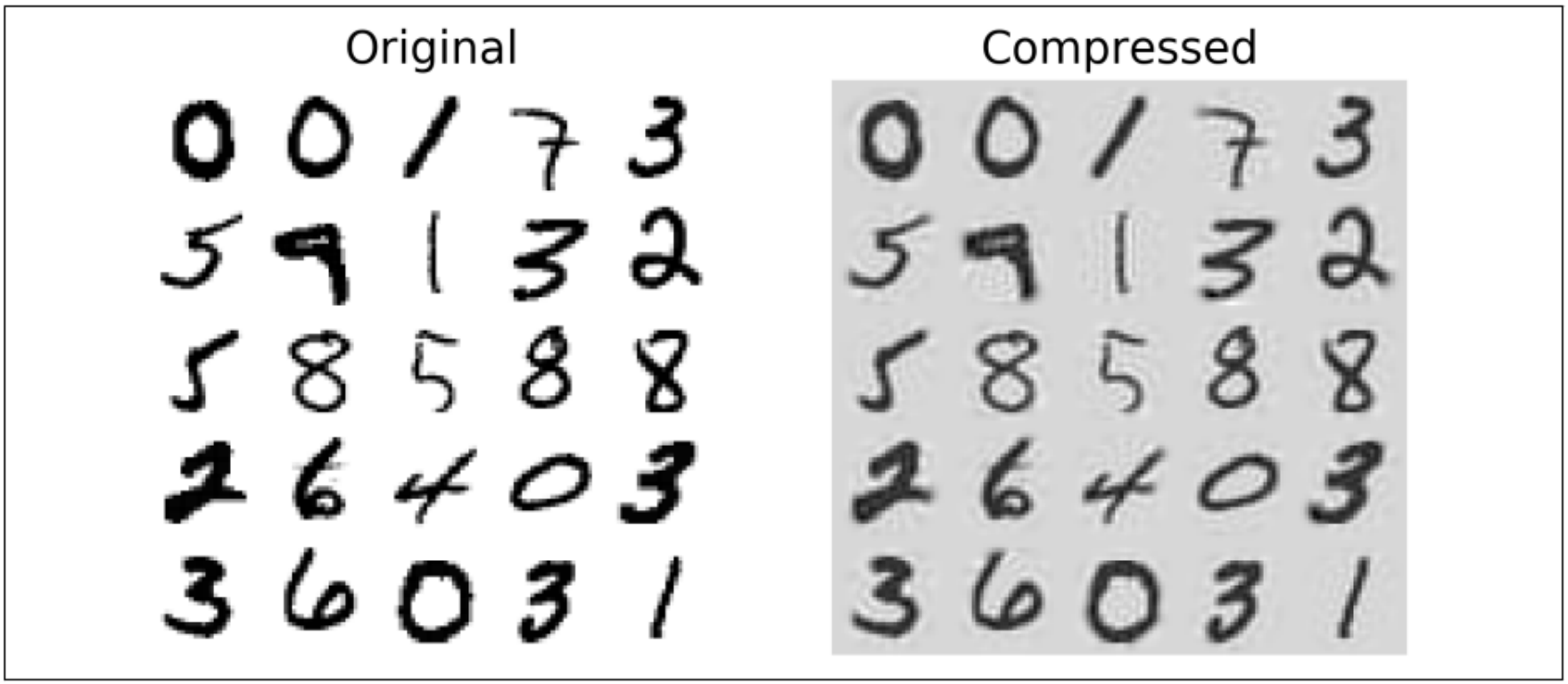


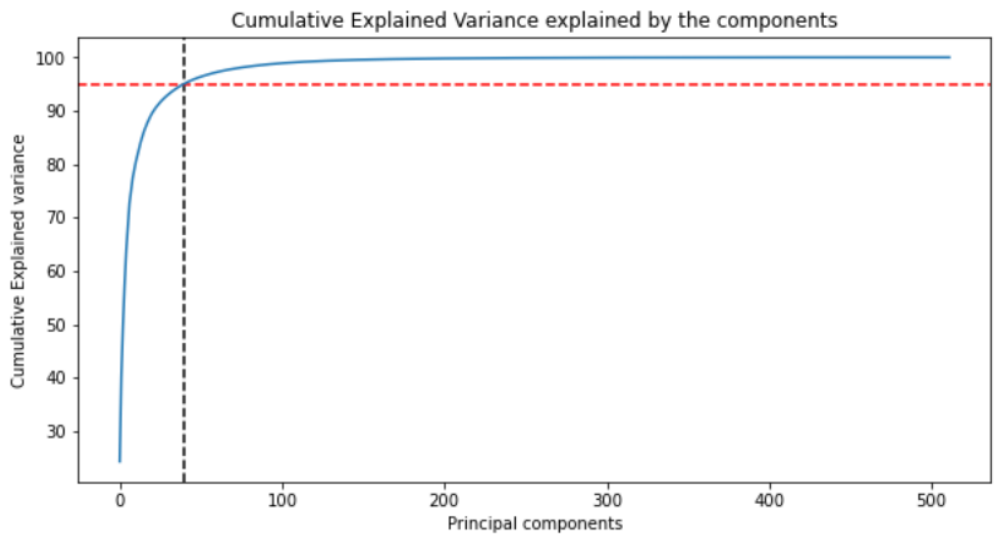
Figure 8-9. MNIST compression preserving 95% of the variance

PCA for Image Compression

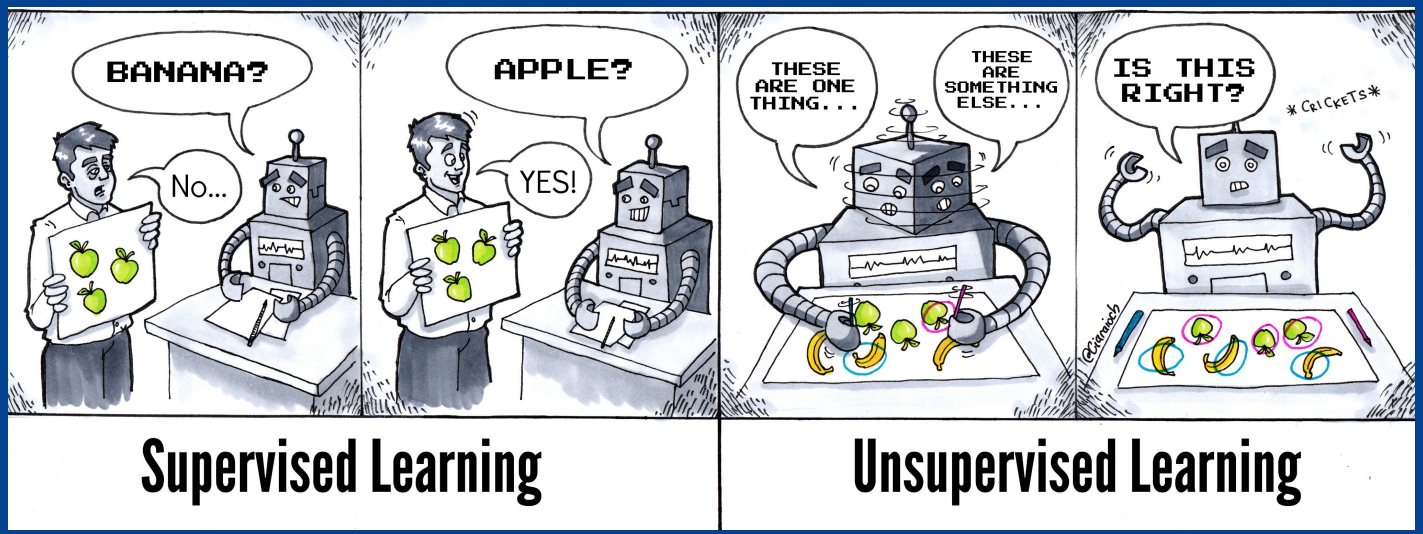
Scikit-learn demo..

```
from sklearn.decomposition import PCA, IncrementalPCA

pca = PCA()
pca.fit(image_bw)

# Getting the cumulative variance
var_cumu = np.cumsum(pca.explained_variance_ratio_)*100

# How many PCs explain 95% of the variance?
k = np.argmax(var_cumu>95)
print("Number of components explaining 95% variance: "+ str(k))
#print("\n")

plt.figure(figsize=[10,5])
plt.title('Cumulative Explained Variance explained by the components')
plt.ylabel('Cumulative Explained variance')
plt.xlabel('Principal components')
plt.axvline(x=k, color="k", linestyle="--")
plt.axhline(y=95, color="r", linestyle="--")
ax = plt.plot(var_cumu)

print(f"  => found k={k}, OK")
```

```
Number of components explaining 95% variance: 40
  => found k=40, OK
```

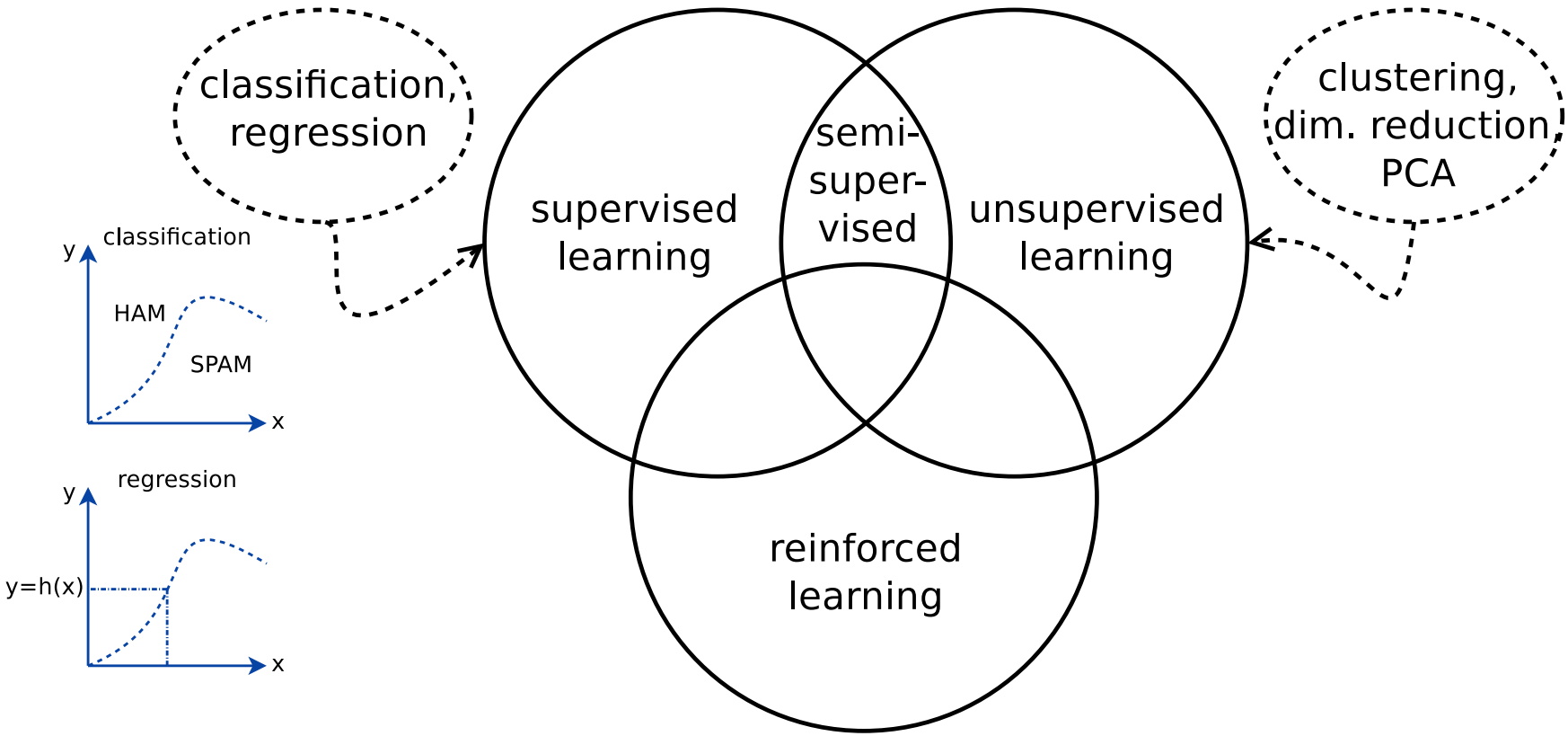


UNSUPERVISED LEARNING



Machine learning taksonomi

Machine learning læringstyper



K-Means

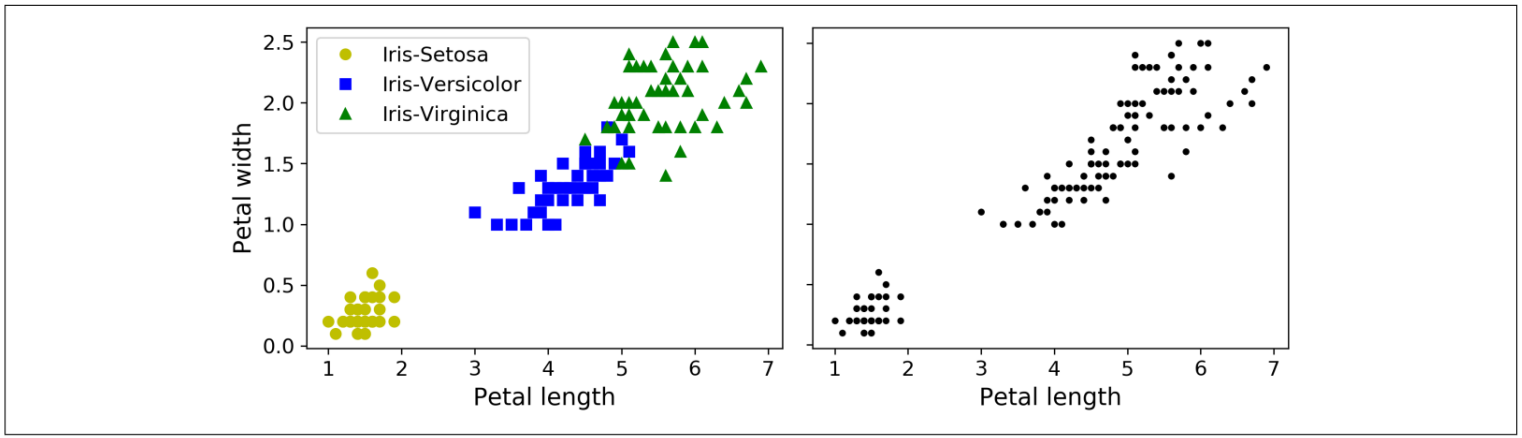


Figure 9-1. Classification (left) versus clustering (right)

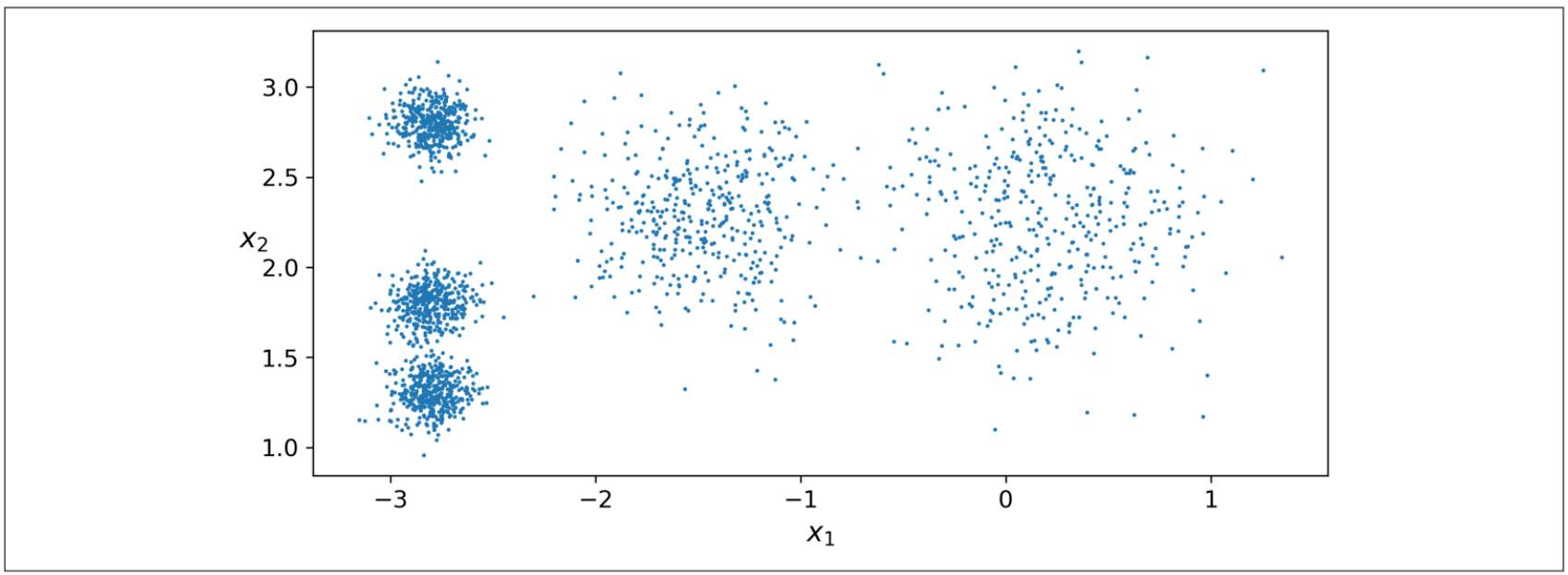


Figure 9-2. An unlabeled dataset composed of five blobs of instances

K-Means

Scikit-learn demo..

jupyter demo_k-means Last Checkpoint: 5 timer siden (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

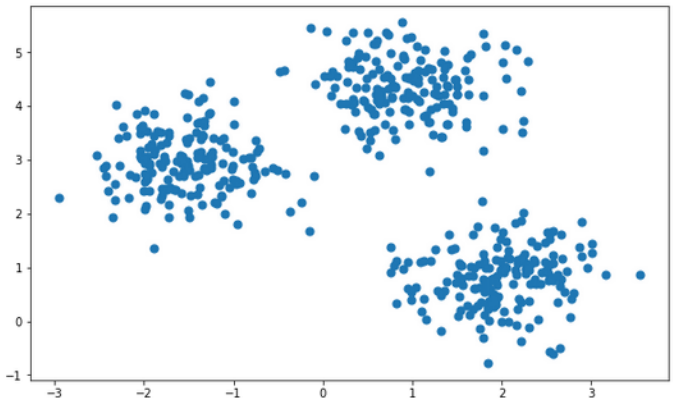
Trusted | Python 3 (ipykernel)

Run Markdown

K-Means Demo

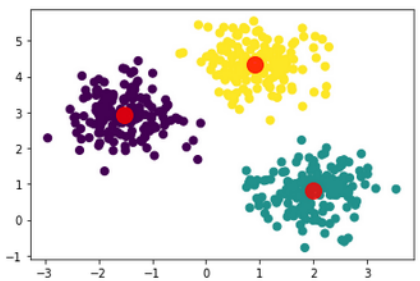
Source: http://deciphertoknow.com/mnist-k-means-clustering/

In [7]:
```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs

features, label = make_blobs(n_samples=500, centers=3, cluster_std=0.55, random_state=0)

plt.figure(figsize=(10, 6))
plt.scatter(features[:, 0], features[:, 1], s=50);
```



In [12]:
```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3)
kmeans.fit(features)
kmeansLabels = kmeans.predict(features)
plt.scatter(features[:, 0], features[:, 1], c=kmeansLabels, s=50, cmap='viridis')

clusterCenters = kmeans.cluster_centers_
plt.scatter(clusterCenters[:, 0], clusterCenters[:, 1], c='red', s=200, alpha=0.8)

print("OK")
```

```
OK
```



K-Means

Algorithm..

- Choose K random points as cluster centers or cluster means.
- For all the N data points: Assign each data point $\mathbf{x}_i$ to one of the K clusters—i.e that cluster whose center is closest to the data point
- For K clusters repeat

$$C(i) = \arg\min_{\mathbf{m}_k} ||\mathbf{x}_i - \mathbf{m}_k||^2, \quad i = 1, \cdots, N$$

- Update the cluster center by taking the average of points within cluster.
- Repeat above two steps until converge or clusters mean doesn't change.

[http://deciphertoknow.com/mnist-k-means-clustering/]

K-Means

```
from sklearn.cluster import KMeans
k = 5
kmeans = KMeans(n_clusters=k)
y_pred = kmeans.fit_predict(X)
```

```
>>> y_pred
array([4, 0, 1, ..., 2, 1, 0], dtype=int32)
>>> y_pred is kmeans.labels_
True
```

We can also take a look at the 5 centroids that the algorithm found:

```
>>> kmeans.cluster_centers_
array([[-2.80389616,  1.80117999],
       [ 0.20876306,  2.25551336],
       [-2.79290307,  2.79641063],
       [-1.46679593,  2.28585348],
       [-2.80037642,  1.30082566]])
```

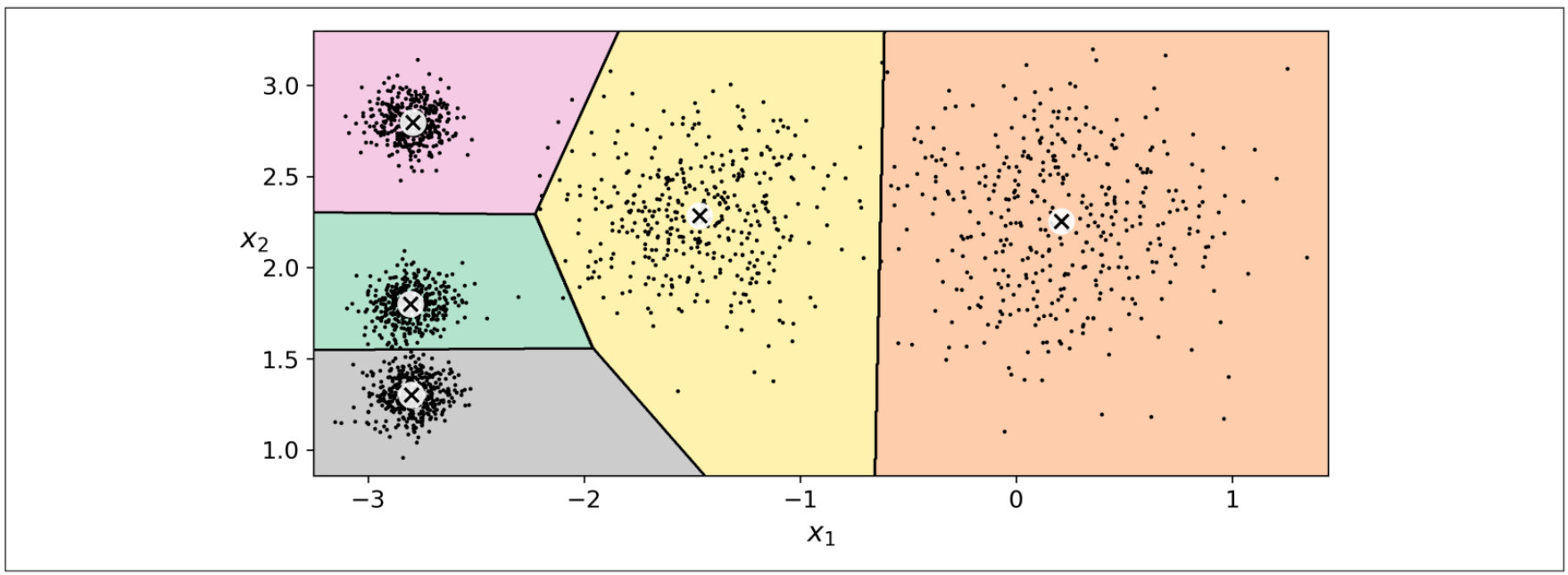


Figure 9-3. K-Means decision boundaries (Voronoi tessellation)

K-Means

Algorithm

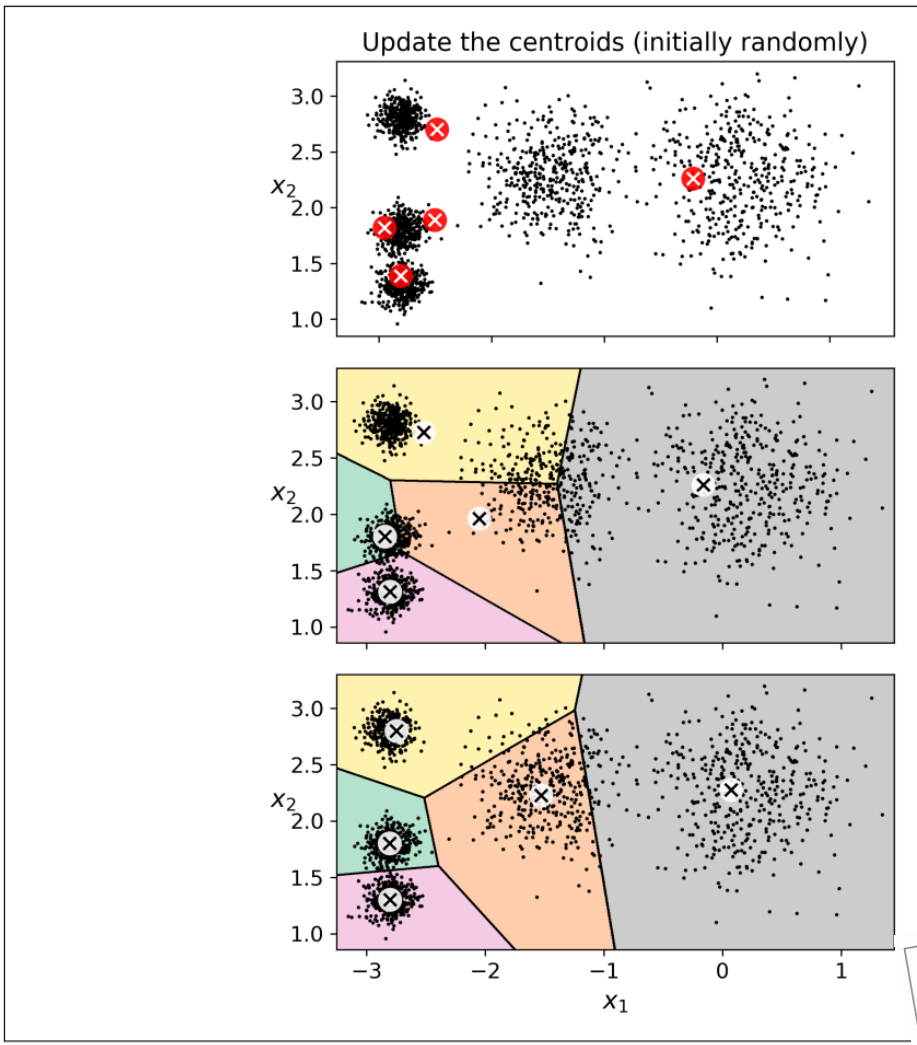


Figure 9-4. The K-Means algorithm

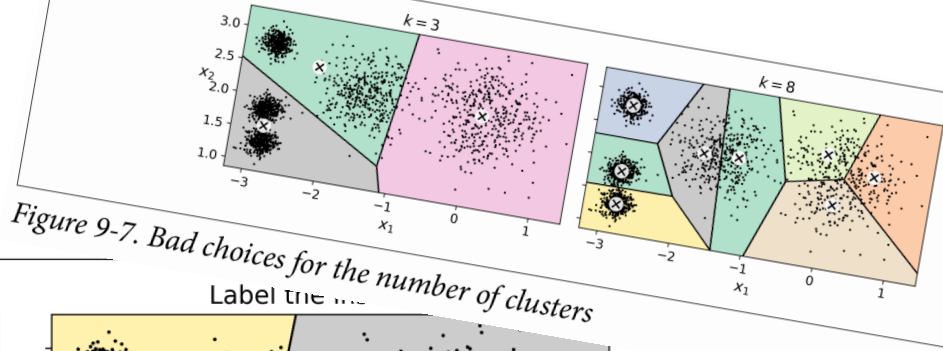


Figure 9-7. Bad choices for the number of clusters

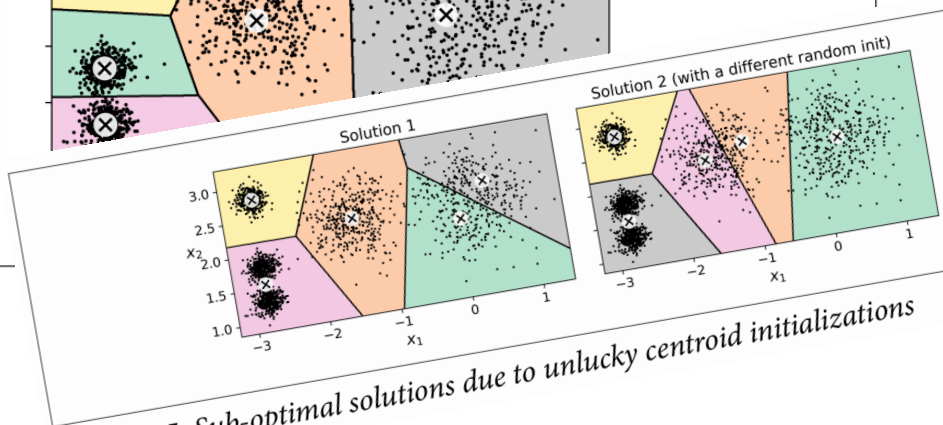


Figure 9-5. Sub-optimal solutions due to unlucky centroid initializations

K-Means on MNIST

Using Clustering for Semi-Supervised Learning

```
k = 50
kmeans = KMeans(n_clusters=k)
X_digits_dist = kmeans.fit_transform(X_train)
representative_digit_idx = np.argmin(X_digits_dist, axis=0)
X_representative_digits = X_train[representative_digit_idx]
```

Figure 9-13 shows these 50 representative images:

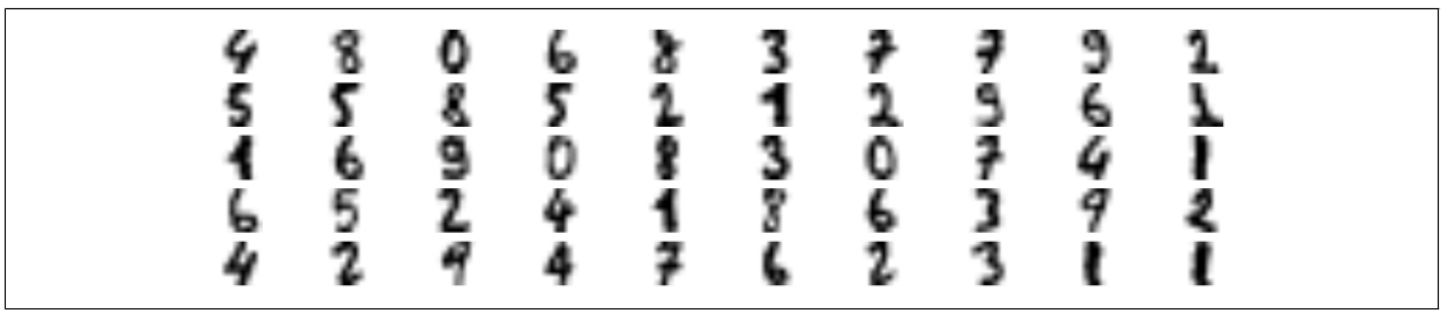


Figure 9-13. Fifty representative digit images (one per cluster)

Now let's look at each image and manually label it:

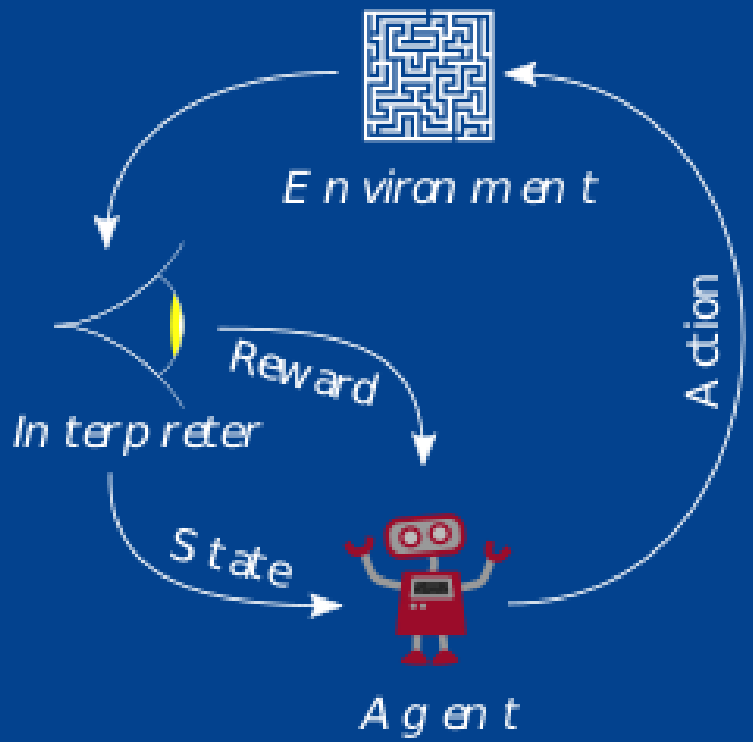
```
y_representative_digits = np.array([4, 8, 0, 6, 8, 3, ..., 7, 6, 2, 3, 1, 1])
```

Now we have a dataset with just 50 labeled instances, but instead of being completely random instances, each of them is a representative image of its cluster. Let's see if the performance is any better:

```
>>> log_reg = LogisticRegression()
>>> log_reg.fit(X_representative_digits, y_representative_digits)
>>> log_reg.score(X_test, y_test)
0.9244444444444444
```

Wow! We jumped from 82.7% accuracy to 92.4%, although we are still only training

REINFORCED LEARNING



Reinforced Learning

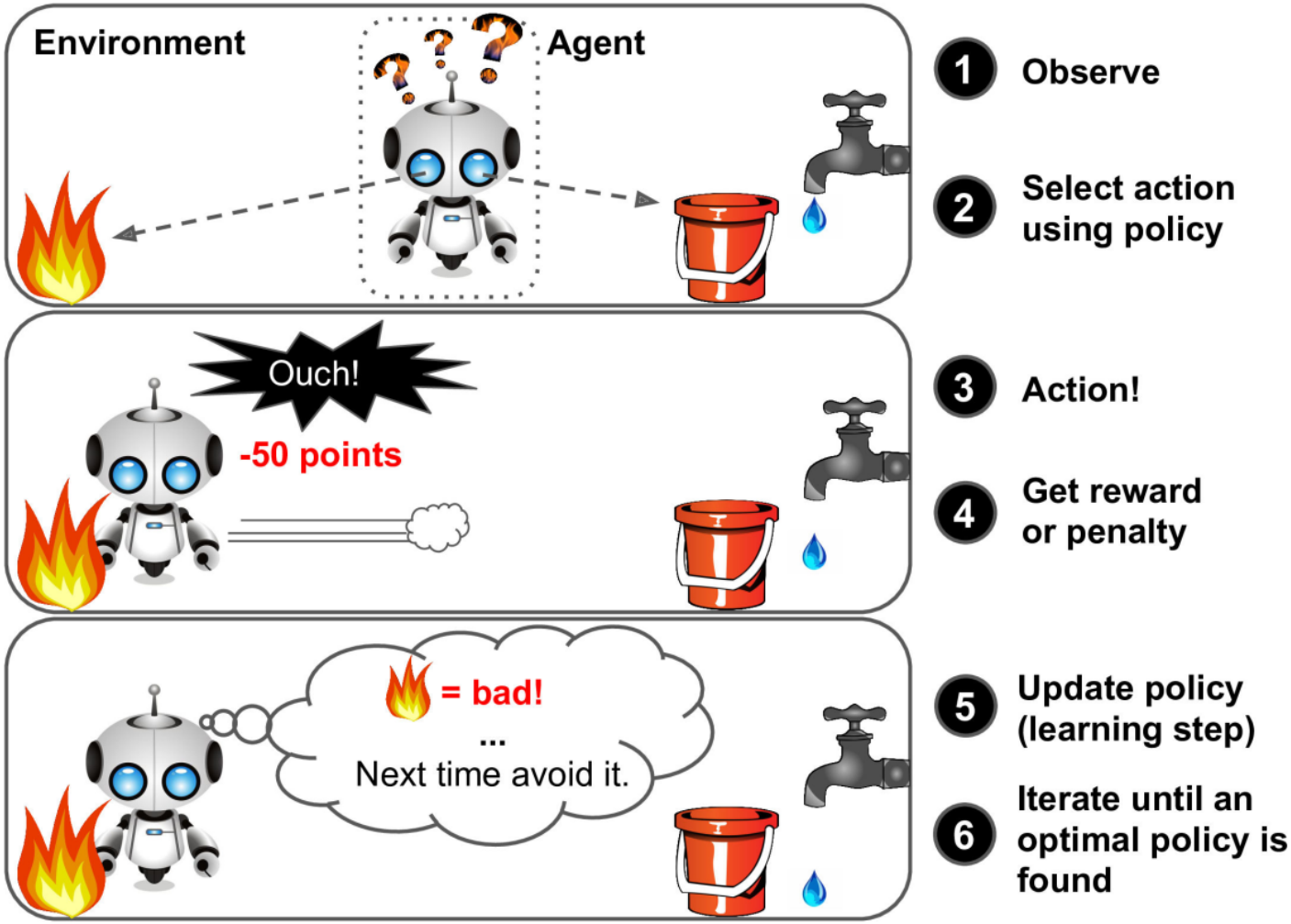


Figure 1-12. Reinforcement Learning

Reinforced Learning