Preprocessing and Data Transformation

by Dane Brown

Appropriate data as input for ML

Preprocessing for Machine Learning

- **Feature extraction:** e.g. Use Hough circles to only keep oranges and eliminate bananas
- **Data formatting:** Format the extracted data according to Scikit’s standards (later)
- **Data cleaning:** Fix invalid or missing data entries
- **Data sampling:** Start small. Be smart, be selective

Preprocessing for Machine Learning

- Effectively separating good features from noise still might not be good enough for classification even if it is:
  - correctly formatted
  - complete (no missing data)
  - errorless

Check it out -> CV_ML

Data Transformation

- **Scaling:** simple example km to m. Different kinds:

- **Standardized Scaling**: Features put into a common range of values. Data should have **zero mean** and **one unit variance**
  - General solution: (x - μ) / σ
- **Normalized Scaling**: Scaling individual samples to have unit norm (length of a vector). L1 or L2
- **Range Scaling**: Get features that lie between a given minimum and maximum value (default 0 and 1)

- **Binarization Scaling**: The exact feature values of the data is unimportant. Shows whether the feature exists or whether positive or negative data

Data Transformation

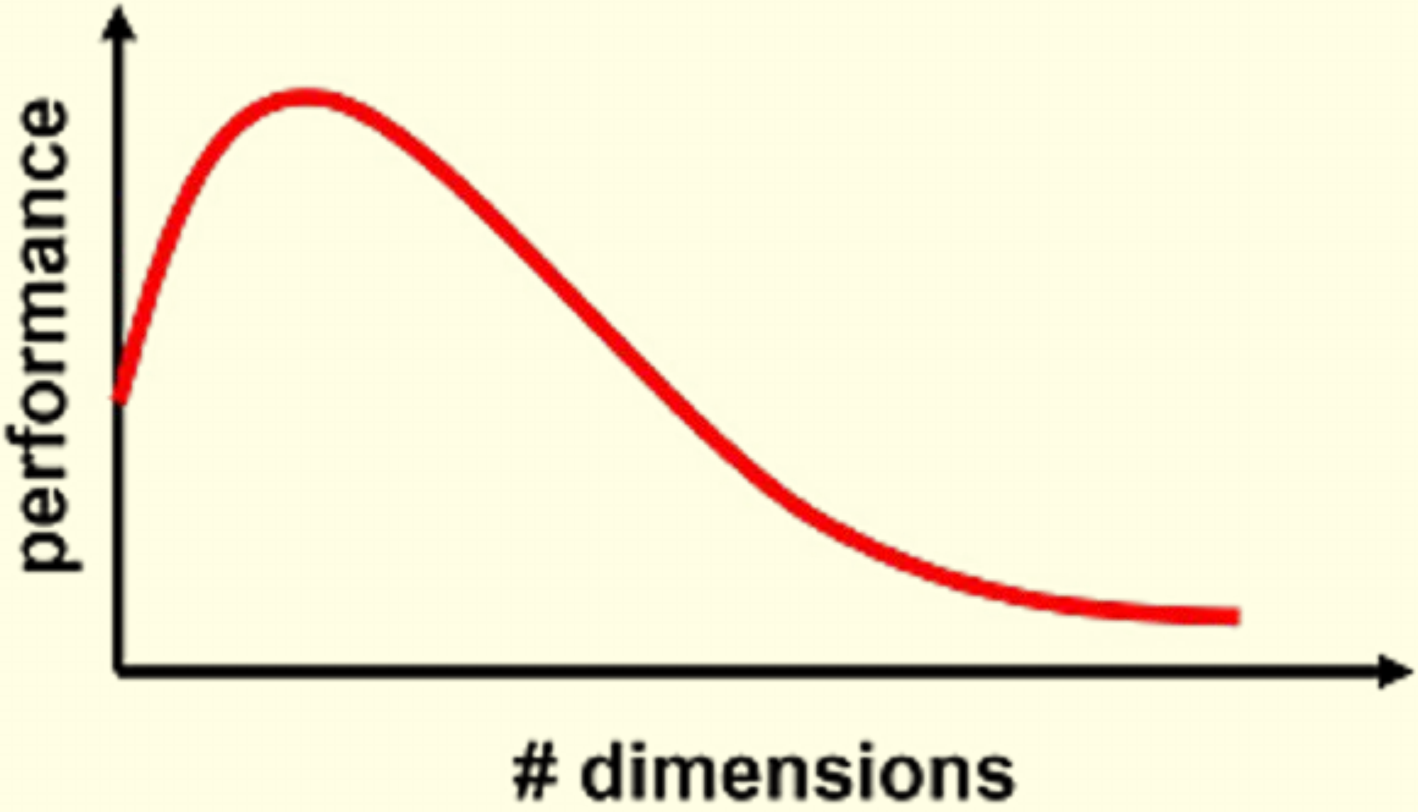
The curse of dimensionality: Number of data points needed to fill the available space grows exponentially with the number of dimensions. But you cannot use 1D or 2D to represent all values (in most cases). Solutions:

- **Aggregation**: Some features can be aggregated into a single feature that would be more meaningful or not change the meaning.

- **Decomposition**: Reduces no. of features to process. Compressing data into a smaller number of highly informative data components.

Data Transformation

- More is not always better!



Data Transformation

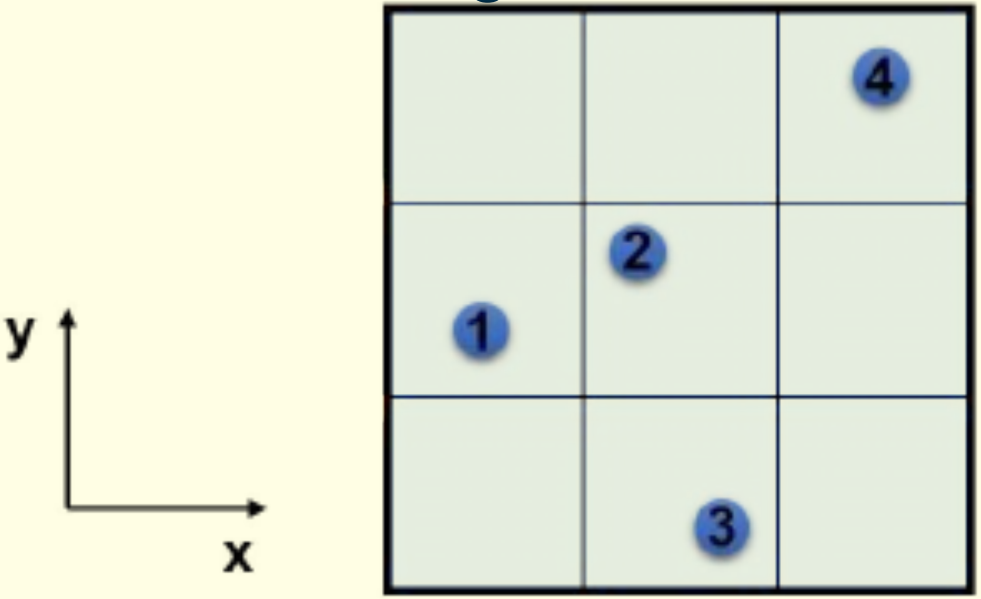
- Consider two features (2D space):
  - GDP as x-axis
  - no. of citizens as y-axis

- Say the first country has a small GDP and an average no. of citizens.
- Draw a point that represents this country

- Also add a 2nd, 3rd, and 4th country.
- 4th country has a high GDP and a large no of citizens

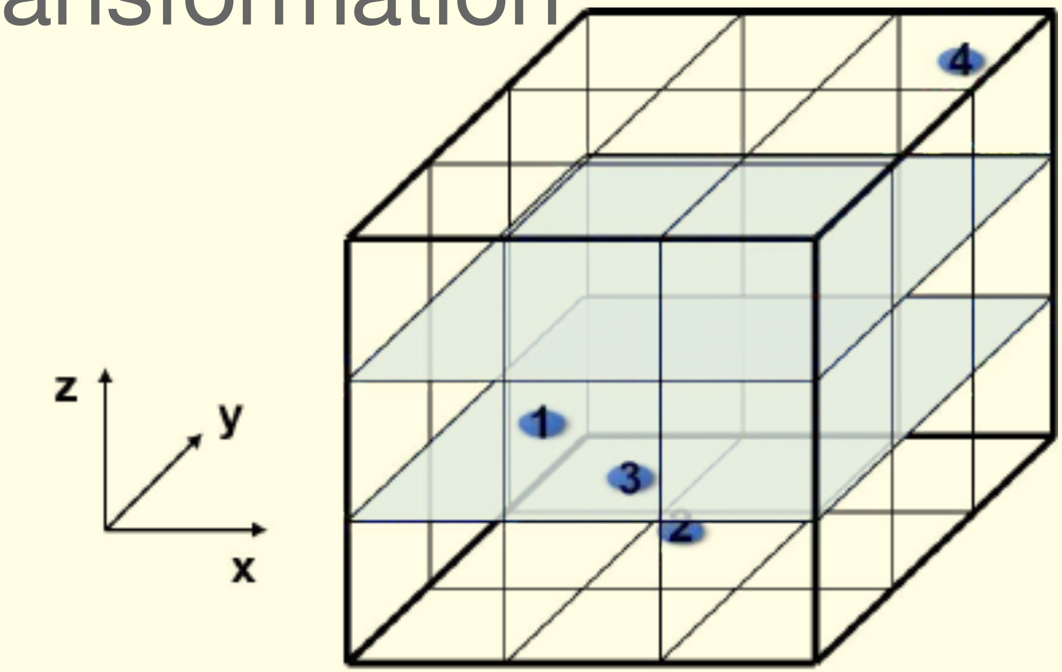
Data Transformation

If you cannot model data point 4 for some reason, try representing the model using an extra dimension



2-D projection

Data Transformation

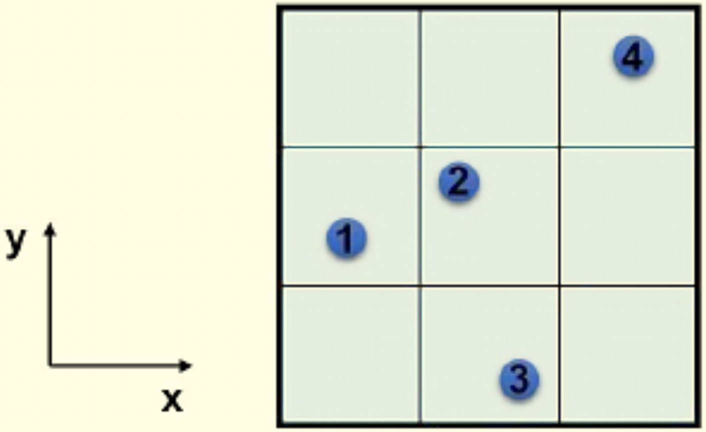


3-D projection

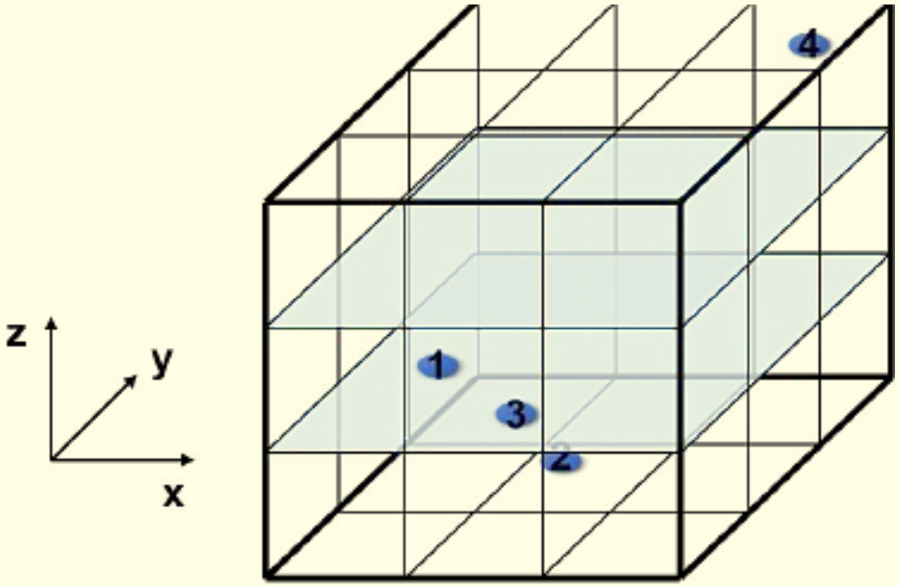
- If a classifier is given data points that do not span the entire feature space, it will not know what to do once a new data point is presented that lies far away from all the previously encountered data points.

Data Transformation

Check it out -> CV_ML



2-D projection



3-D projection

- **Lets explore the following:**
  - Principal Component Analysis (PCA)
  - Independent Component Analysis (ICA)
  - Nonnegative Matrix Factorization (NMF)

PCA is Versatile

- While PCA is not a machine learning algorithm, it is unsupervised and has many uses
  - Dimensionality reduction
  - Visualization
  - Noise filtering,
  - Feature extraction etc.

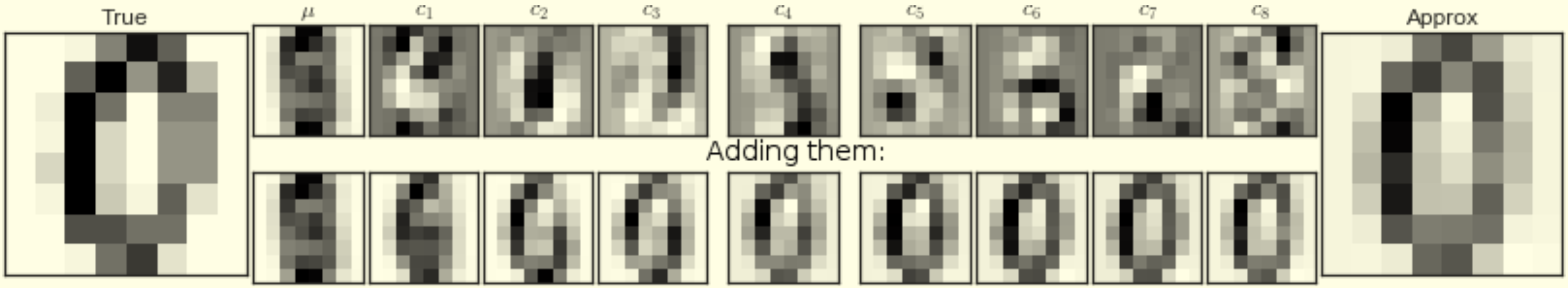
PCA for Noise Filtering (and Dim. Red.)

- Instead of training a classifier on high-dimensional data
  - train the classifier on the PCA processed data; lower-dimensional representation
  - reconstruct image using the lower-dimensional data
  - automatically filters out random noise in the inputs

Check it out -> CV_ML

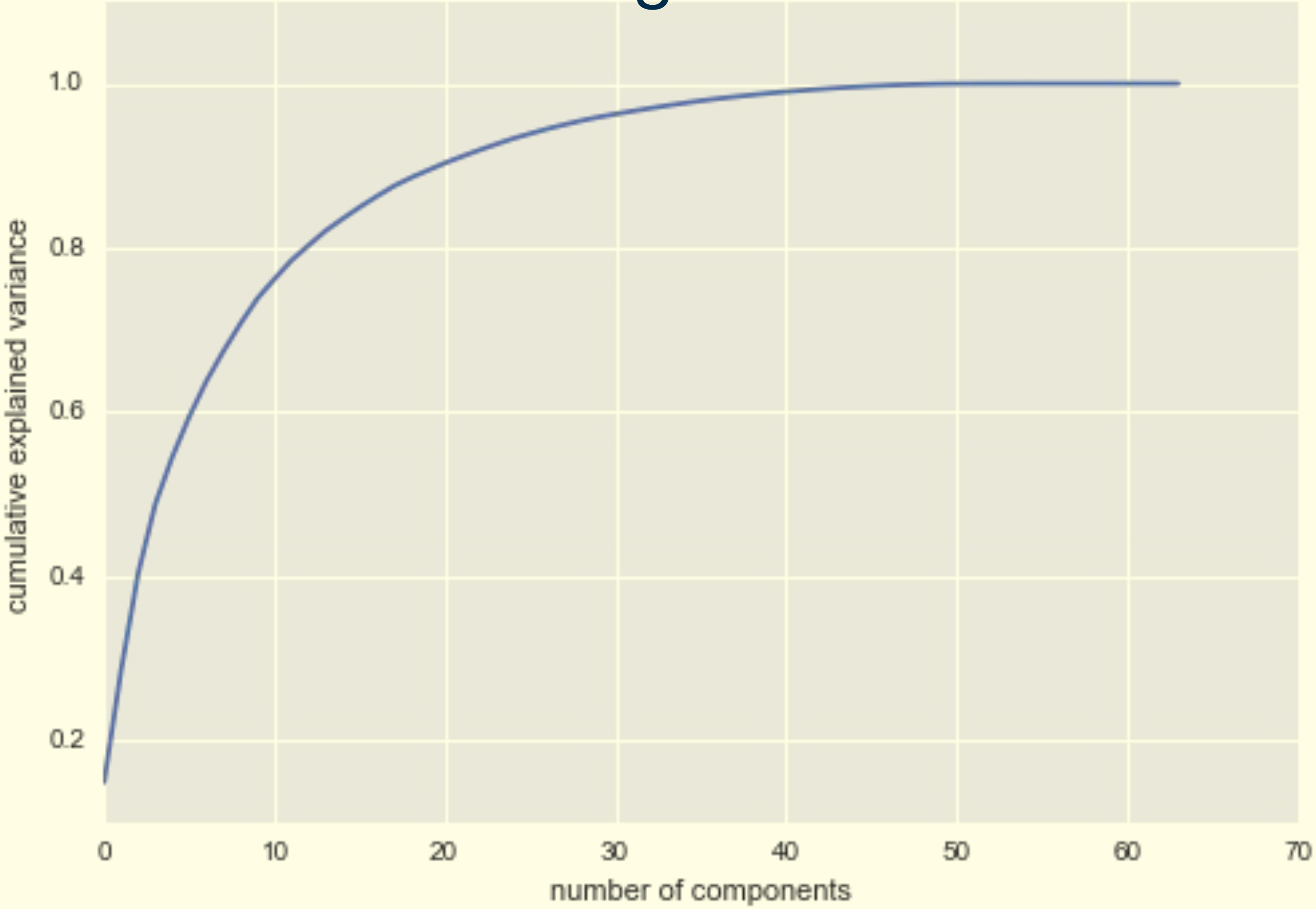
Reducing Dimensionality of Images

- An x,y coordinate counts only as 2D, whereas:
- Each pixel in an image counts as a dimension!
- How do we make the latter avoid the curse of dimensionality?
  - Use the mean + a number of components
  - Reconstruct an approximate image by adding comp



Reducing Dimensionality of Images

- As benefits of having more data (dim) start diminishing, fully utilizing the space become more important than increasing dims.



Result: Less Detail = Good or Bad?

- But visually we lost a lot of information right?
- Yes, but the model does not ‘see’ like we do