Kaggle Seasons #08

QUEEN MARY MACHINE
LEARNING SOCIETY

PCA

QUEEN MARY MACHINE
LEARNING SOCIETY

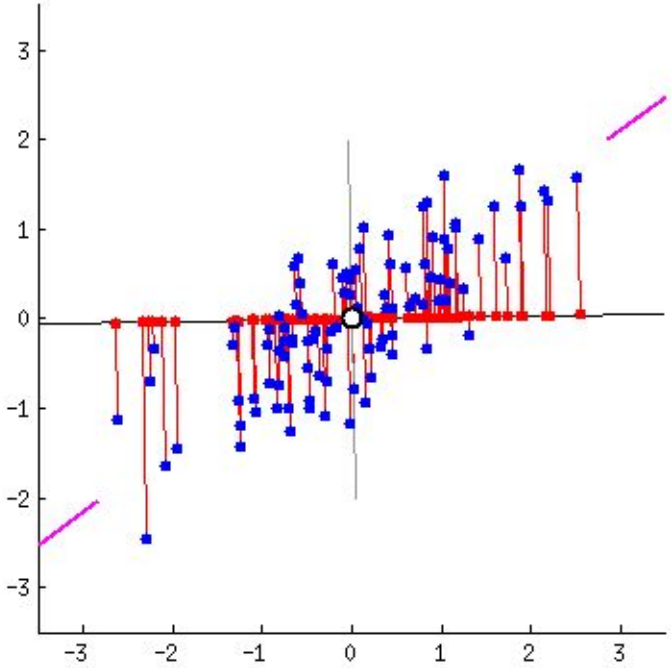
What is PCA?

- PCA is a **dimensionality reduction** technique
  - Reduces the number of features while maintaining information
- It is useful when the data has too many features, especially if many of the features are correlated or uninformative
- Can be used in EDA to understand key relationships in the data
- Is often used in the social sciences to identify influential psychological or sociological factors
  - E.g. IQ tests and personality tests

kaggle

How does PCA work?

PCA works by transforming the original coordinates into a different coordinate system:



builtin.com

How does PCA work?

It does so by computing a linear combination (i.e. weighted sum) of the original coordinates:

$$Z_k = \sum_{i=1}^{d} x_i v_{ik}$$

Where Z_k is the k-th transformed coordinate, or **principal component**

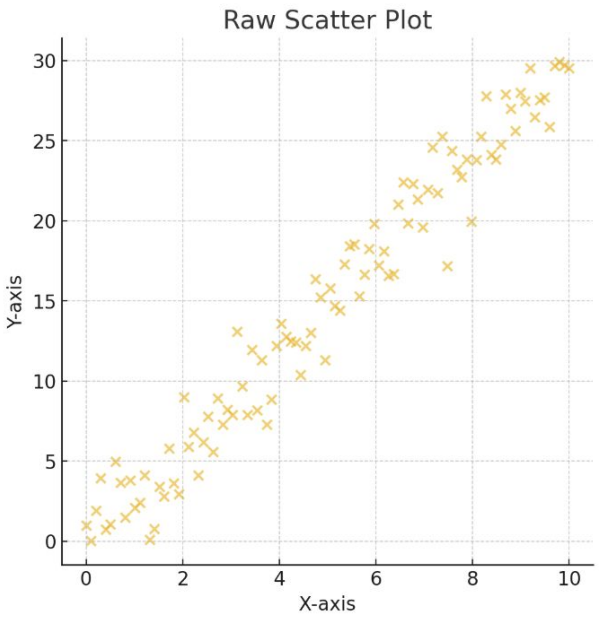
How does PCA work?

The weights x_i are **learnt** such that they **maximise the variance** of the data along the first (then second, then third, etc) principal component while all the components **orthogonal** to each other:

$$Z_k = \sum_{i=1}^{d} x_i v_{ik}$$

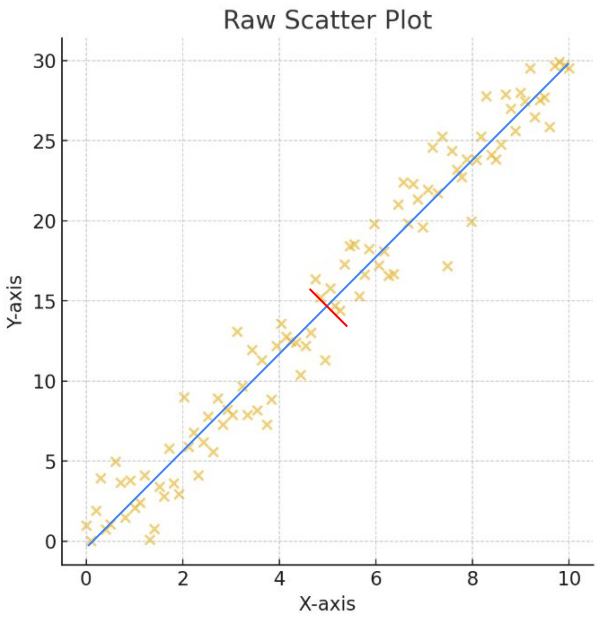
How does PCA work?

The weights x_i are **learnt** such that they **maximise the variance** of the data along the first (then second, then third, etc) principal component while all the components **orthogonal** to each other:



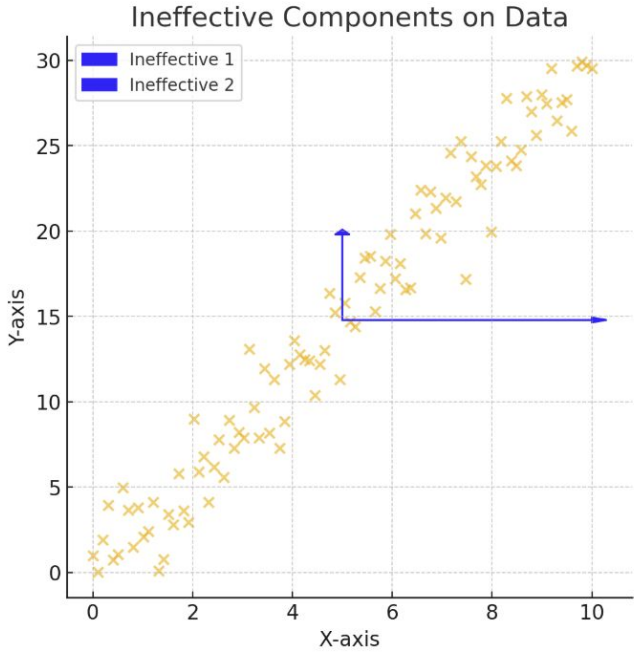
How does PCA work?

The weights x_i are **learnt** such that they **maximise the variance** of the data along the first (then second, then third, etc) principal component while all the components **orthogonal** to each other:

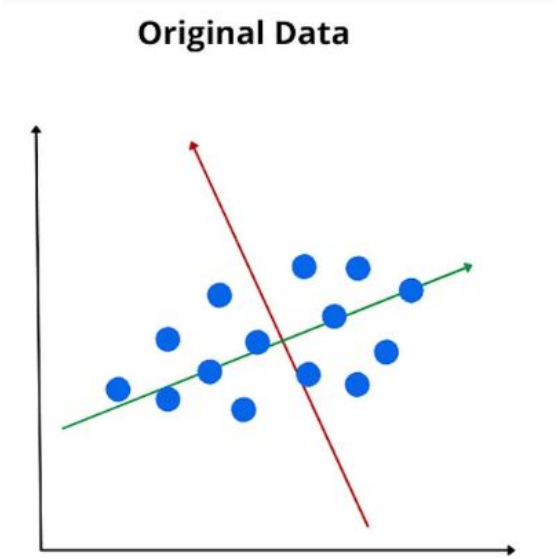


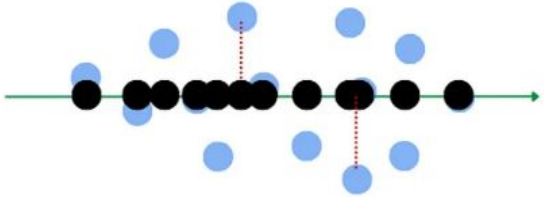
How does PCA work?

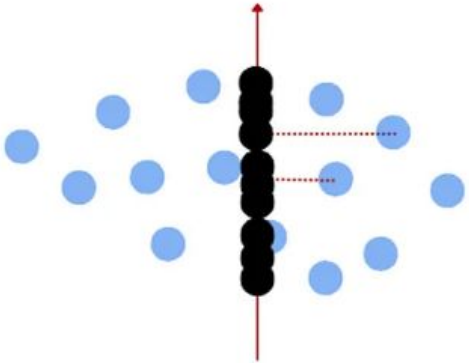
An example of ineffective coordinates would be as follows, as the **variance** of the data along those components isn't **maximised**:



Interactive PCA - Part 1





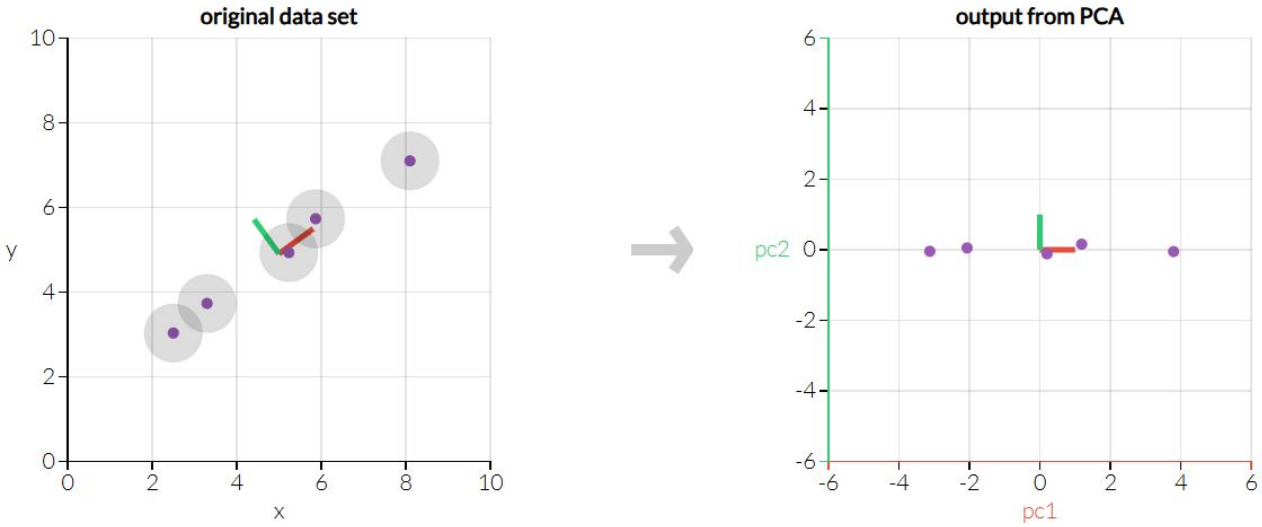


kaggle

https://medium.com/@libertihub

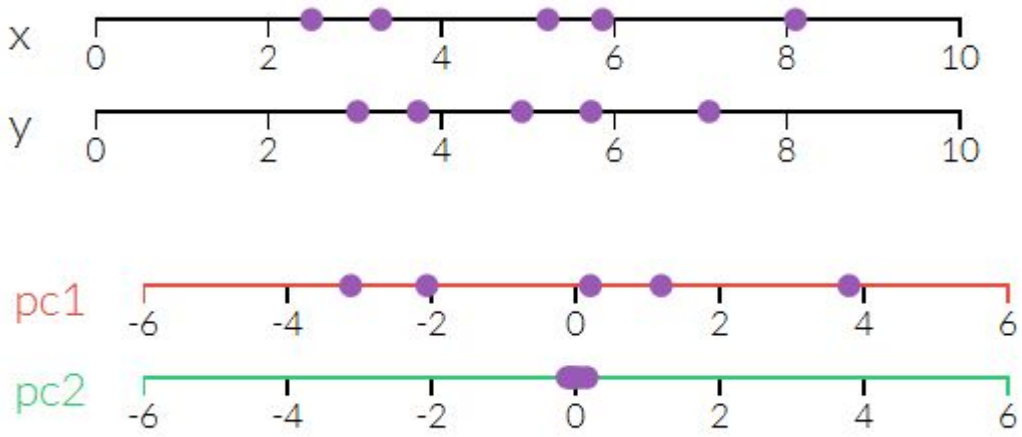
Interactive PCA - Part 2

Interactive Visualization: https://setosa.io/ev/principal-component-analysis/



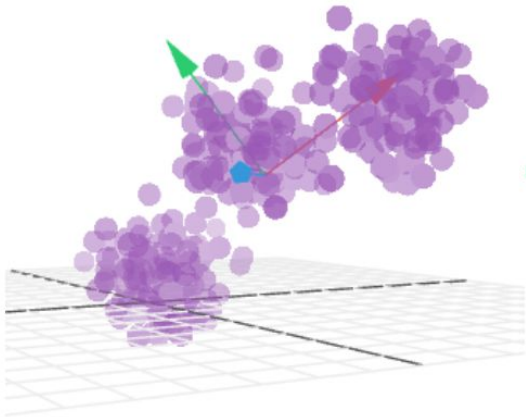
Interactive PCA - Part 3

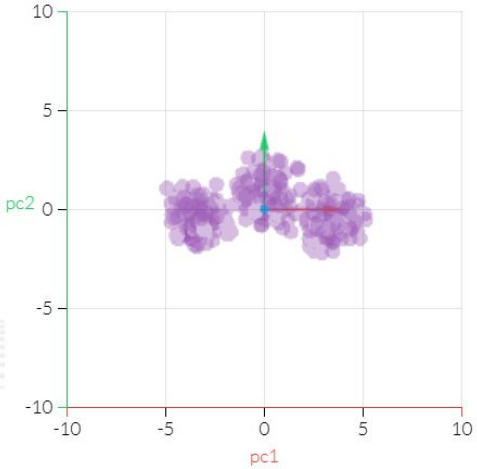
Interactive Visualization: https://setosa.io/ev/principal-component-analysis/

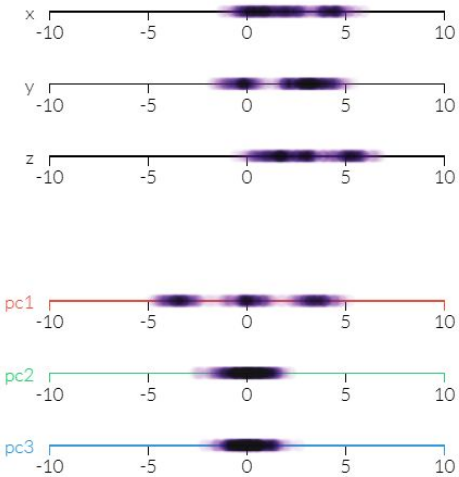


Interactive PCA - Part 4

Interactive Visualization: https://setosa.io/ev/principal-component-analysis/







How to use PCA in Python?

SciKit Learn's library includes PCA. The documentation can be found here: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

Install User Guide API Examples Community More 1.6.1 (stable)

KernelPCA
LatentDirichletAllocation
MiniBatchDictionaryLearning
MiniBatchNMF
MiniBatchSparsePCA
NMF
PCA
SparseCoder
SparsePCA
TruncatedSVD
dict_learning
dict_learning_online
fastica
non_negative_factorization
sparse_encode
sklearn.discriminant_analysis
sklearn.dummy
sklearn.ensemble
sklearn.exceptions
sklearn.experimental
sklearn.feature_extraction
sklearn.feature_selection
sklearn.frozen

API Reference > sklearn.decomposition > PCA

PCA

class sklearn.decomposition.PCA(*n_components=None, *, copy=True, whiten=False, svd_solver='auto', tol=0.0, iterated_power='auto', n_oversamples=10, power_iteration_normalizer='auto', random_state=None*) [source]

Principal component analysis (PCA).

Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD.

It uses the LAPACK implementation of the full SVD or a randomized truncated SVD by the method of Halko et al. 2009, depending on the shape of the input data and the number of components to extract.

With sparse inputs, the ARPACK implementation of the truncated SVD can be used (i.e. through `scipy.sparse.linalg.svds`). Alternatively, one may consider `TruncatedSVD` where the data are not centered.

Notice that this class only supports sparse inputs for some solvers such as "arpack" and "covariance_eigh". See `TruncatedSVD` for an alternative with sparse data.

For a usage example, see Principal Component Analysis (PCA) on Iris Dataset

Read more in the User Guide.

Parameters:

n_components : *int, float or 'mle', default=None*

On this page
PCA
fit
fit_transform
get_covariance
get_feature_names_out
get_metadata_routing
get_params
get_precision
inverse_transform
score
score_samples
set_output
set_params
transform
Gallery examples

This Page
- Show Source

Thanks for listening!

QUEEN MARY MACHINE
LEARNING SOCIETY