Dimensionality Reduction

Sabya D.
| Lighthouse Labs Instructor, Data Analytics

W06D1

Who am I >

Instructor: Dr. Sabya DG
(sabya.datatech@gmail.com)

Director/Founder, Altius AI Solutions Inc.

-PhD in Theoretical Physics, Germany

-Postdoc: NUS, Singapore / UToronto

Superpowers:

- Microsoft Azure, Data Platform Solutions
- Big data Engineering, MLOPs
- Unsupervised ML / Statistical Methods

LinkedIn: **https://www.linkedin.com/in/sabyadg/**

“If you want to master something, teach it” … Richard P. Feynman

Agenda

- Intro to Dimensional reduction
  - Motivation- Why care ..
  - 2 approaches- (unsupervised) DRTs & (supervised) FS
- Dimensional reduction techniques (DRT)
  - PCA
  - LDA
- Feature selection (FS)
  - Filter methods
  - Wrapper methods
- Demo 1 (FS)
- Demo 2 (DRT: PCA)

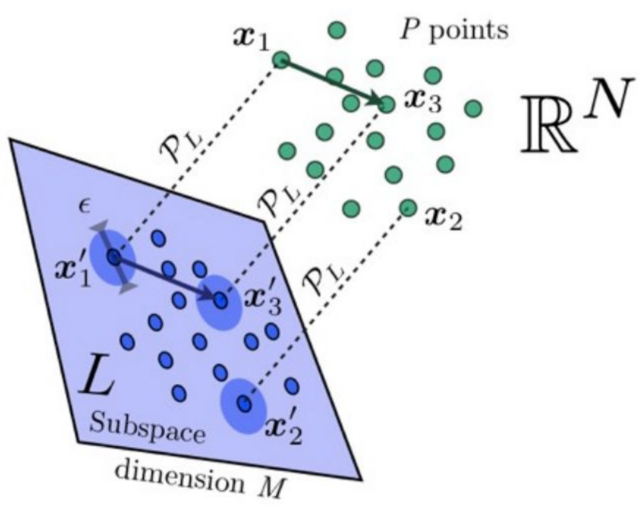
Dimensional reduction techniques : Motivation I

Clustering/group the data according to similarity to help visualize & discover no. of cluster, inter/intra cluster distance/relationships etc.

But we still cannot plot high-dimensional (or non-numeric) data.

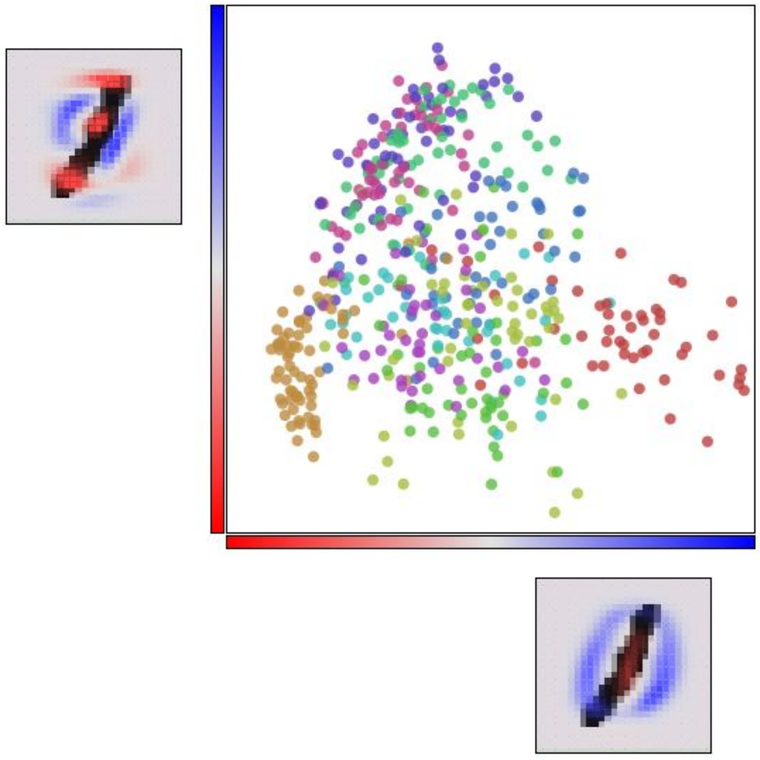
Dimensionality reduction (or embedding) techniques:

- Assign new coordinates, in low-D space (2D/3D for visualization)
- Preserve similarity/distance relationships between input data approx.
- Discover distance relationships more directly.

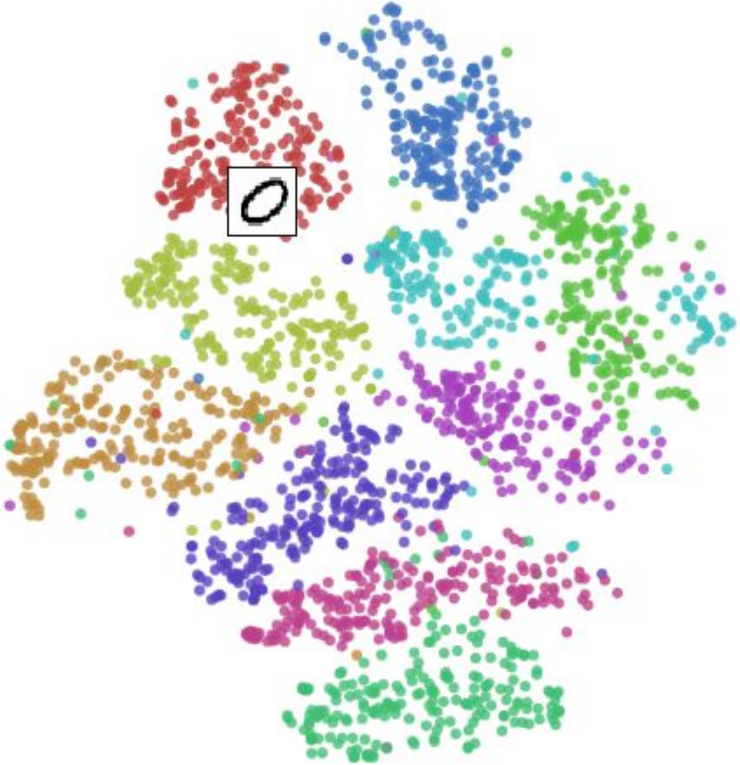


Source: Laurent Jacques

Dimensional reduction : Motivation II



Visualizing MNIST with PCA



A t-SNE plot of MNIST

Dimensional reduction techniques I

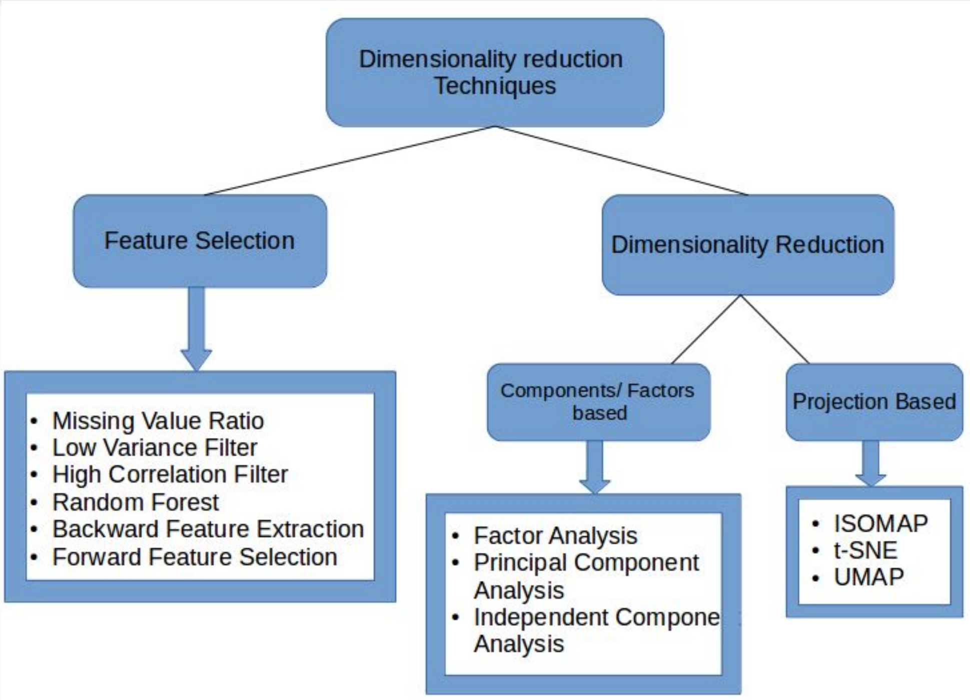
Idea: Transform a problem from

High D -> Low D

Usually, applied on Training Data (x_i)

Benefits:

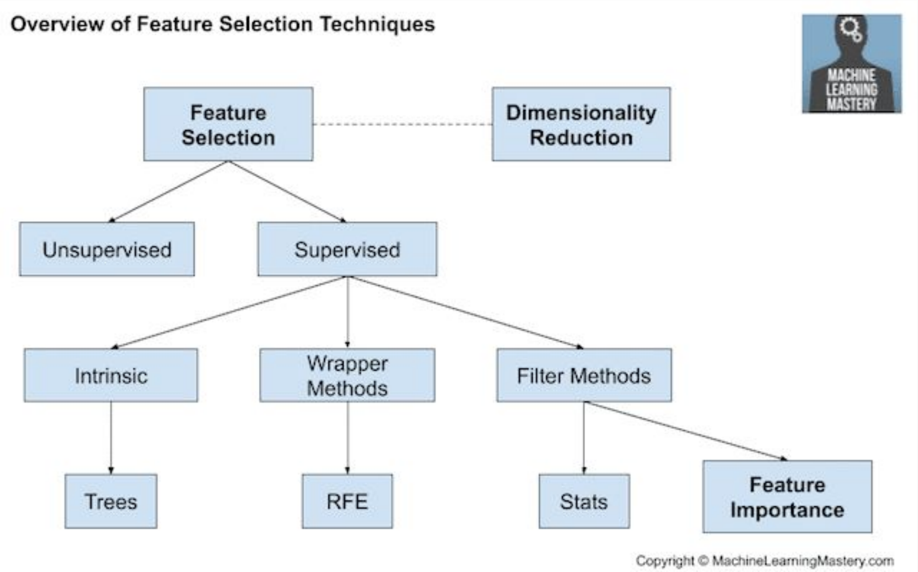
a. Less memory to store (x_i) & model (w)
b. Less time to train model
c. Improving conditions, on the minimization (better accuracy)
d. Unsupervised data modelling
e. Avoid overfitting



Questions ?

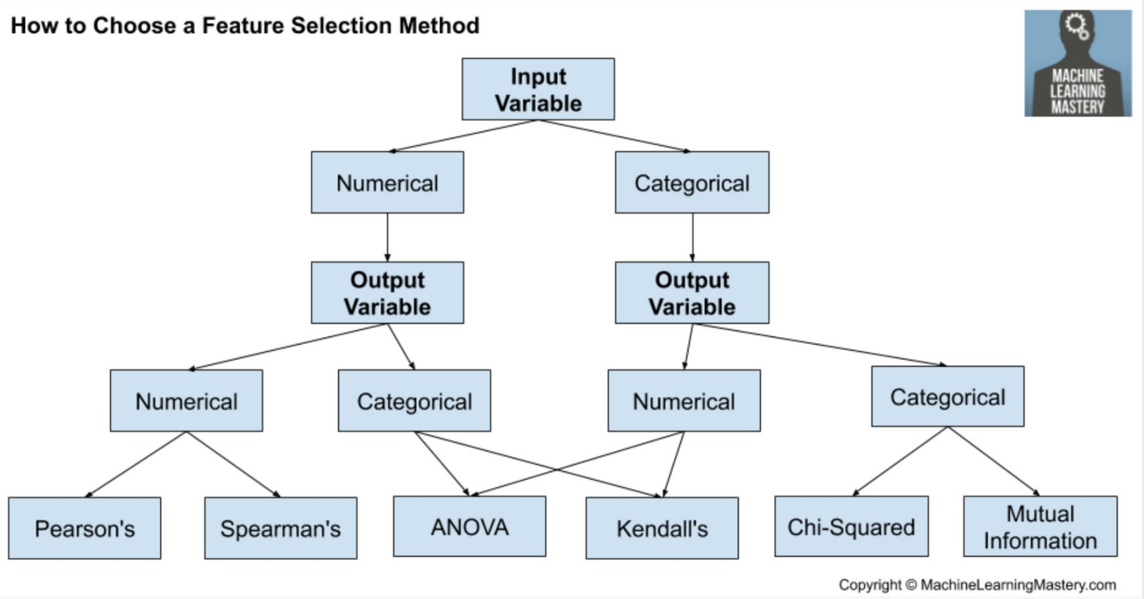
Feature Selection

Feature selection… is the process of selecting a subset of relevant features for use in model construction



Filter methods : Supervised/Statistical approaches

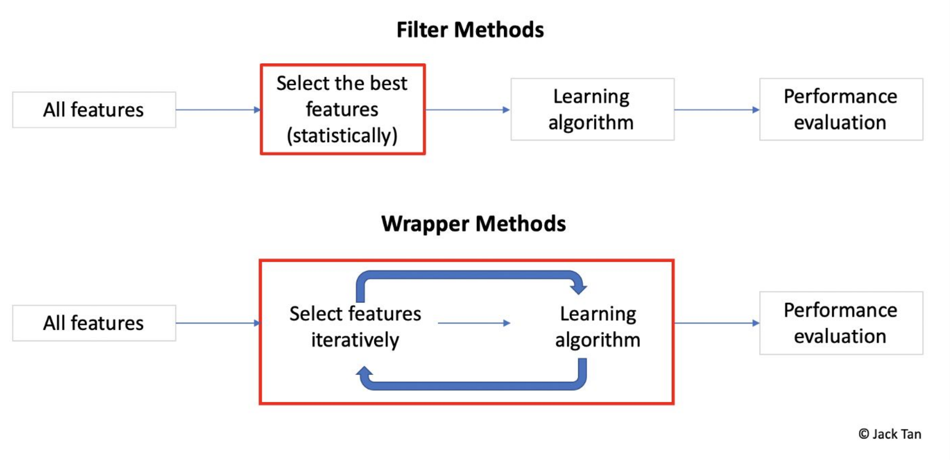
- Give each feature a “score” that represents how “important” it is.
- Scores can be based on:
  - Correlation with target variable
  - High/low variance
  - Feature similarity (correlation between features)
- Keep features with high scores, discard features with low scores.
- Applied before training ML model
- Advantages:
  - Fast—no training involved, just calculations
- Disadvantages:
  - Can ignore feature combinations
  - May keep redundant features



How to Choose Feature Selection Methods For Machine Learning

Wrapper methods: Supervised/Model metric driven

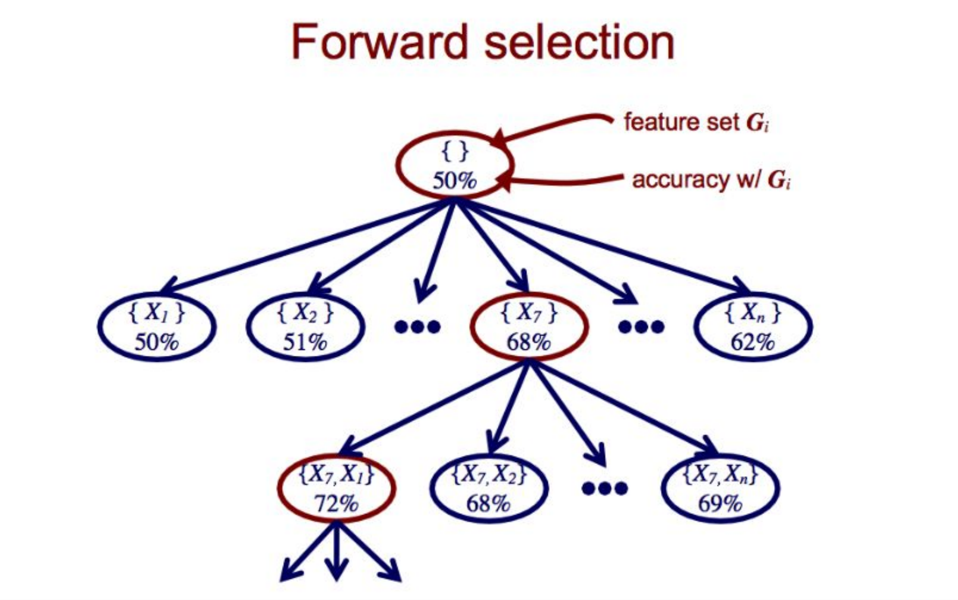
- Iteratively train models with subsets of features.
- Use model metrics to choose best feature set
- Advantages:
  - Features are benchmarked relatively
  - Model metric driven
- Disadvantages:
  - Model-retraining is expensive
- Popular wrapper methods:
  - Forward selection
  - Backward selection
  - Stepwise selection



Wrapper Methods: Forward Selection

1. SelectedFeatures = [ ]
2. Find F in (AllFeatures - SelectedFeatures) that, if added to SelectedFeatures, best improves model performance.
3. If adding F improved performance more than some threshold, permanently add it to SelectedFeatures and go back to 2.

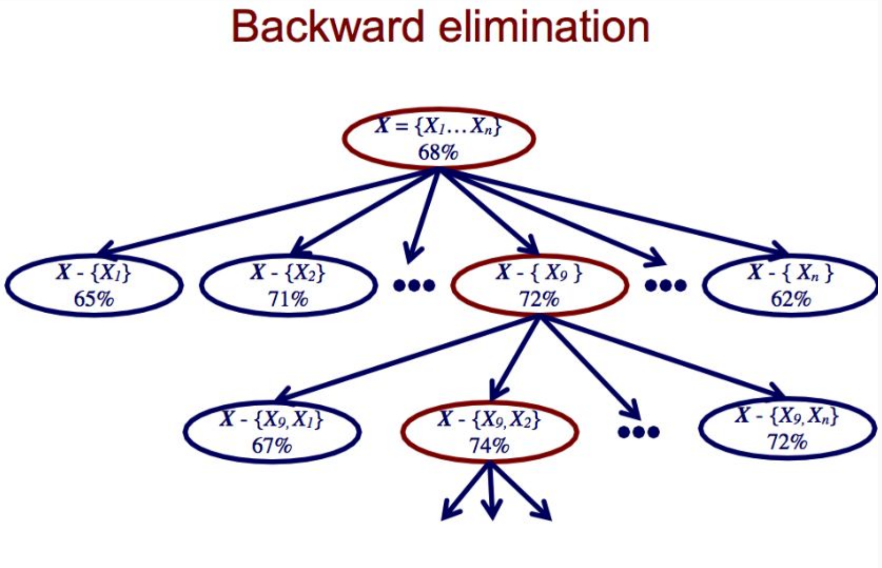
- Efficient for choosing a small subset of features.
- Misses features whose usefulness requires other features (feature synergy).



Wrapper Methods: Backward elimination

1. SelectedFeatures = AllFeatures
2. Find F in SelectedFeatures that, if removed from SelectedFeatures, best improves model performance (or decreases model performance the least).
3. If removing F improved (or decreased) performance more (or less) than some threshold, permanently remove it from SelectedFeatures and go back to 2.

- Efficient for discarding a small subset of features.
- Preserves features whose usefulness requires other features.
- Less efficient for computation.
    - It takes more time to fit models with all features than with one feature.



Other Wrapper Methods: Stepwise Selection, RFE

- Stepwise selection: Combination of Forward and Backward Selection.

1. SelectedFeatures = [ ]
2. Perform Forward Selection
3. Perform Backward Selection
4. Repeat 2. and 3. until a final optimal set of features is obtained.

- Can alternatively start with SelectedFeatures = AllFeatures, or somewhere in between.

RFE is easy to configure and use and effective at selecting those features (columns) in a training dataset that are more or most relevant in predicting the target variable.

2 important configuration options when using RFE:
(i) choice in the number of features to select
(ii) choice of the algorithm

Both of these hyperparameters can be explored, although the performance of the method is not strongly dependent on these hyperparameters being configured well.

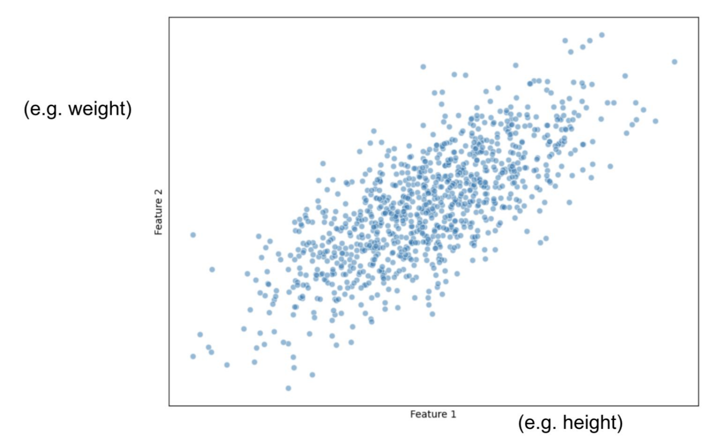
Questions ?

PCA & LDA

See,
[StatQuest: PCA main ideas in only 5 minutes!!! - YouTube]
[StatQuest: Linear Discriminant Analysis (LDA) clearly explained. - YouTube]

PCA

- Principal Component Analysis
  - Most popular (and important)
- Idea: represent many variables with fewer variables while **minimizing** loss of information
  - Simplest case: represent two variables as a single variable
  - How would you do this?



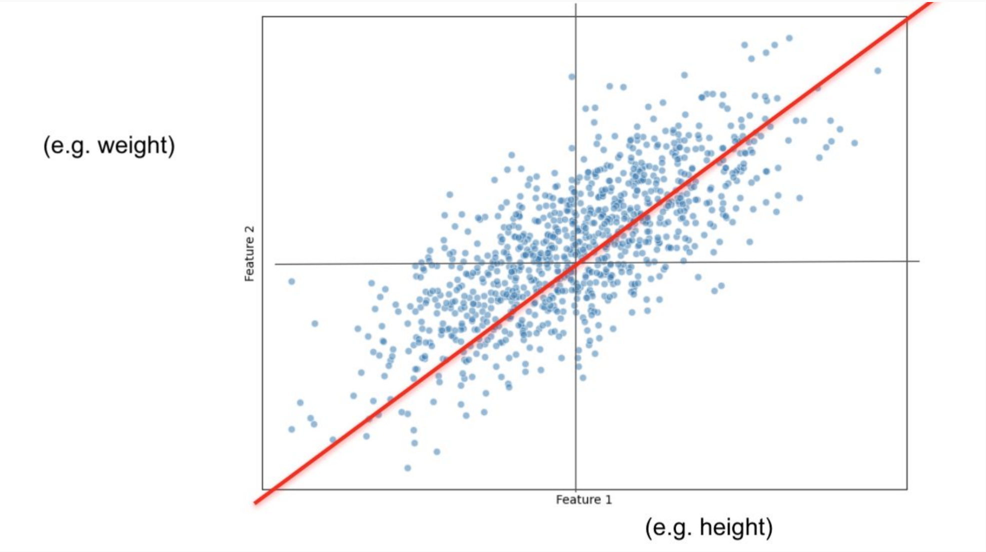
height (h)	weight (w)
170	80
175	81
161	71
182	83
164	76
165	76
185	90

H + w

???
250
256
232
265
240
241
275

PCA: Linear Combinations

- We can use a **linear combination** of our two variables to transform our two variables into a single variable (size).
  - $ah + bw = s$ (line)
  - Can think of it as a "projection"
- How do we choose *a* and *b*?
  - Graphically, we want to create a "best-fit" line that passes through the mean of each variable.
  - By "best-fit", we mean the perpendicular distance from the points to the line is minimized.
  - Can also think of this line as going in the direction of MOST variation.
- This line we found is referred to as **Principal Component 1 (PC1)**.



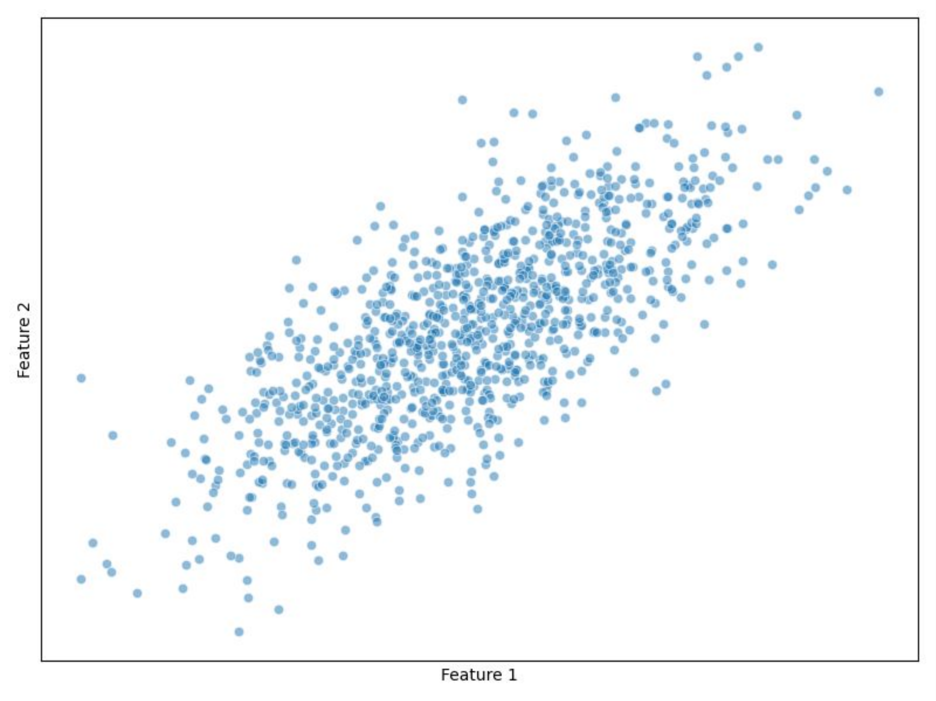
Slide credits : Jeremy Eng (LHL, Instructor)

PCA: Principal Components

- What if we have *d* variables and want to reduce down to *k* variables?
  - Turns out, we just need to find the first *k* principal components!
- Let's discuss principal components more!
- For each dimension (variable/feature) in our data, we also have that many principal components.
  - E.g. if we have two variables, there are two principal components (we just found the first one in the last example).
  - E.g. if we have 100 variables, there are 100 principal components.
- We showed how to find PC1, but how do we find PC2, PC3, etc.?
- Remember that PC1 was a "best-fit" line in the direction of MOST variation.
- PC2 will be in the direction of the next most variation, following a few rules:
  - Must pass through the mean of each variable
  - Must be perpendicular/orthogonal to all other previous principal components.
- Following these rules, PC2, PC3, …, PC*k* can be found in order!
- This way, we are keeping the principal components that have the most variation!

PCA: Change of basis

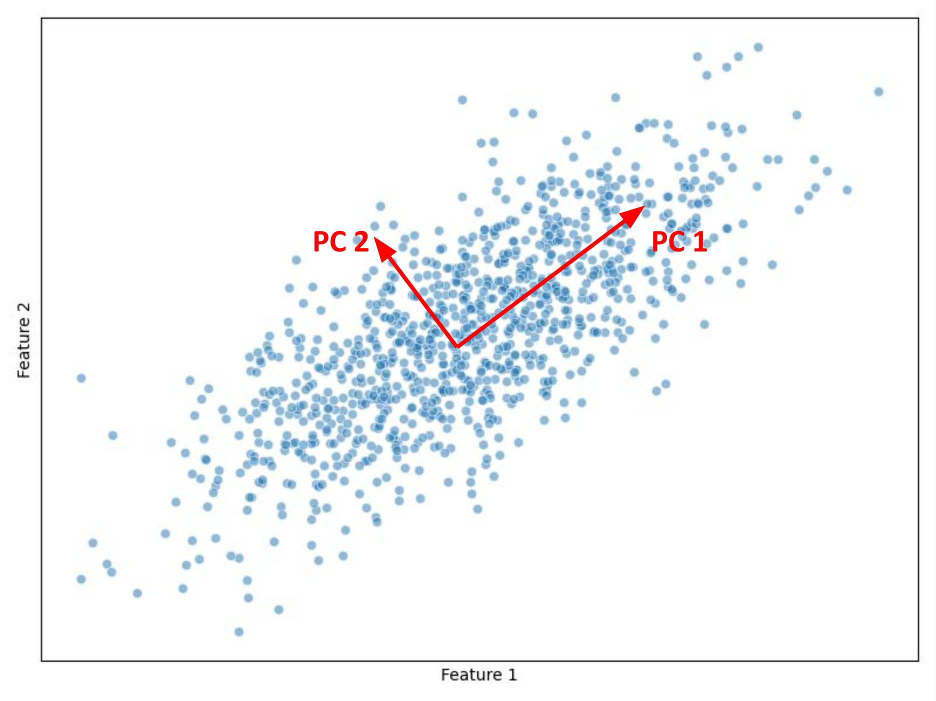
- Another way to think of PCA is performing a **change of basis**



Slide credits : Jeremy Eng (LHL, Instructor)

PCA: Change of basis

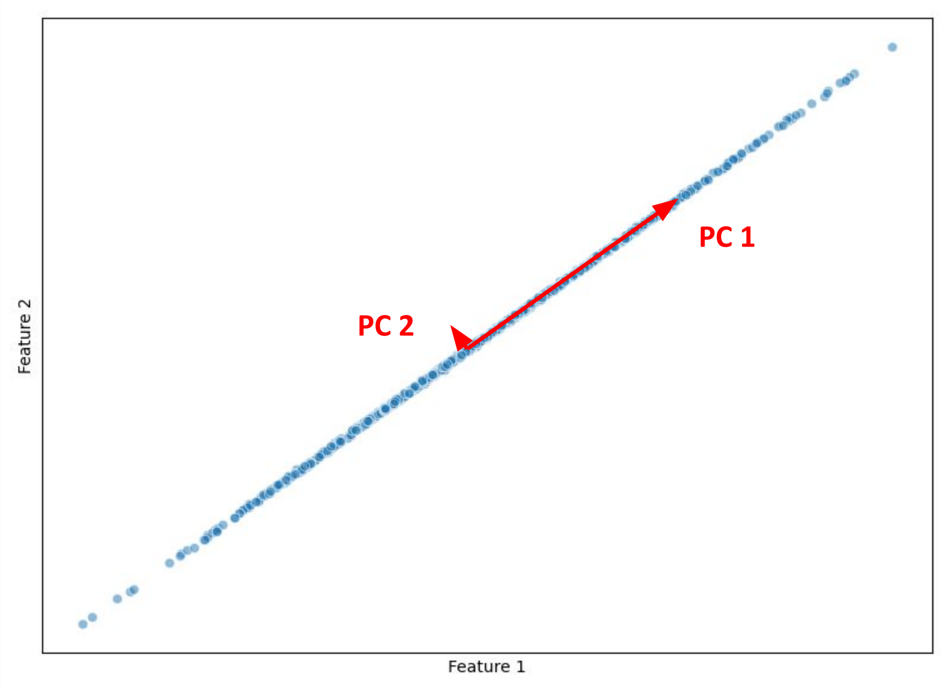
- Another way to think of PCA is performing a **change of basis**



Slide credits : Jeremy Eng (LHL, Instructor)

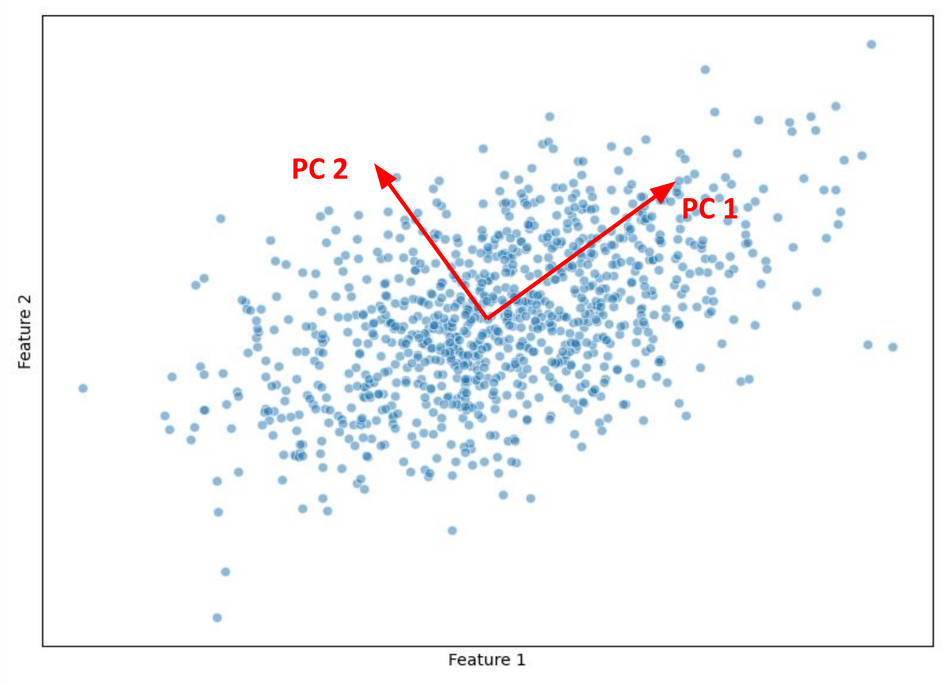
PCA: Extreme Cases

- Extreme case where PCA is very useful (very little information lost)



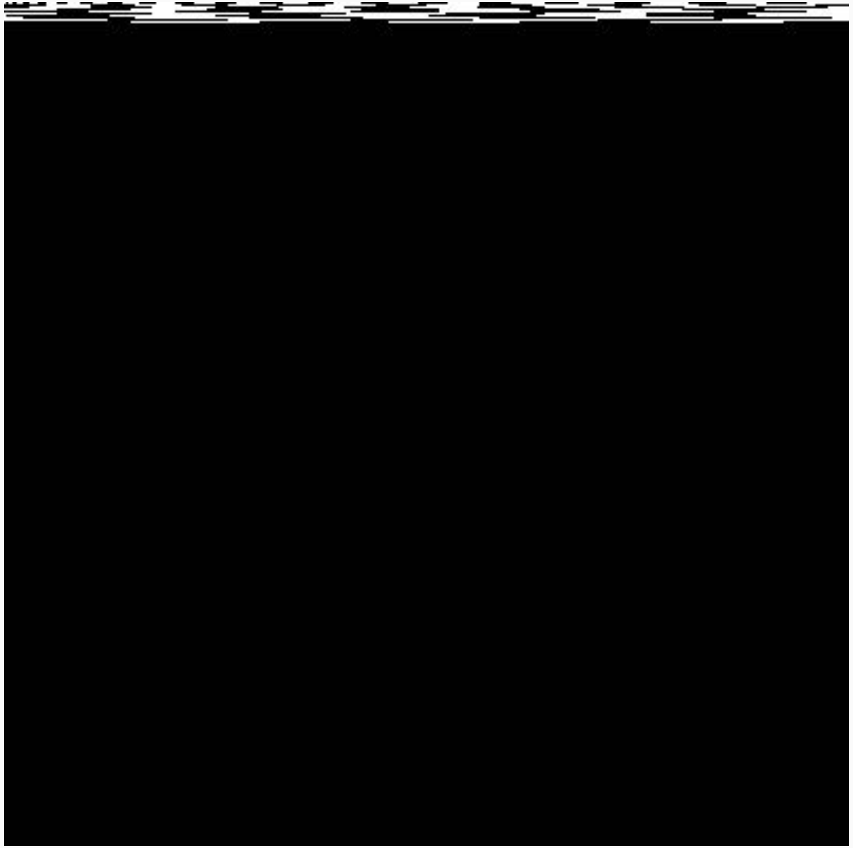
PCA: Extreme Cases

- Other extreme case where PCA is NOT very useful (PC2 explains as much variance as PC1)



Slide credits : Jeremy Eng (LHL, Instructor)

PCA: Visualized in 3 Dimensions



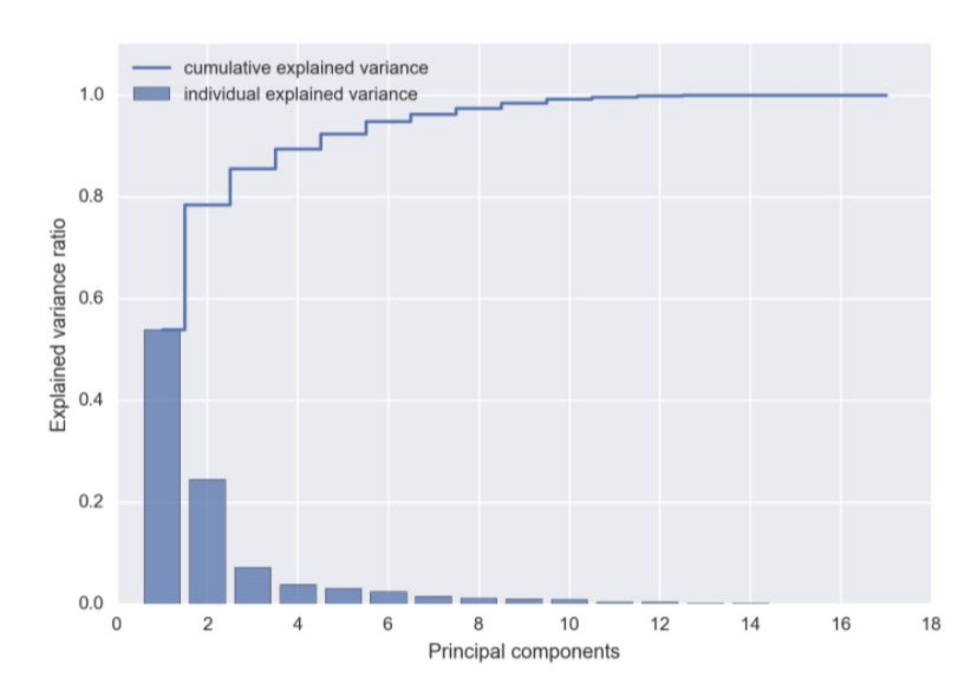
- Source

PCA: The Math (yay!)

- Suppose we have *n* observations, *d* original dimensions (variables/features), and we want to reduce down to *k* dimensions.
- Original data: $X_{(nxd)}$
- Reduced data: $Z_{(nxk)}$
- Then: $Z_{(nxk)} = X_{(nxd)}A_{(dxk)}$
- Where: The columns of $A_{(dxk)}$ are the eigenvectors corresponding to *k* largest eigenvalues from the covariance matrix of X ($C = X^TX/(n\text{-}1)$ for zero-mean data).
- For details on the math behind PCA, see here.
- Luckily we have Python's sklearn library to do these calculations for us ☺️

PCA: Choosing the New Dimension

- We can plot the **cumulative explained variance** to choose an optimal new dimension (i.e. number of principal components to keep).
  - Shows how much variance is explained by each PC.
- Strategy: Keep number of PCs up to a certain % of total variance explained.
- Strategy: Elbow method



Slide credits : Jeremy Eng (LHL, Instructor)

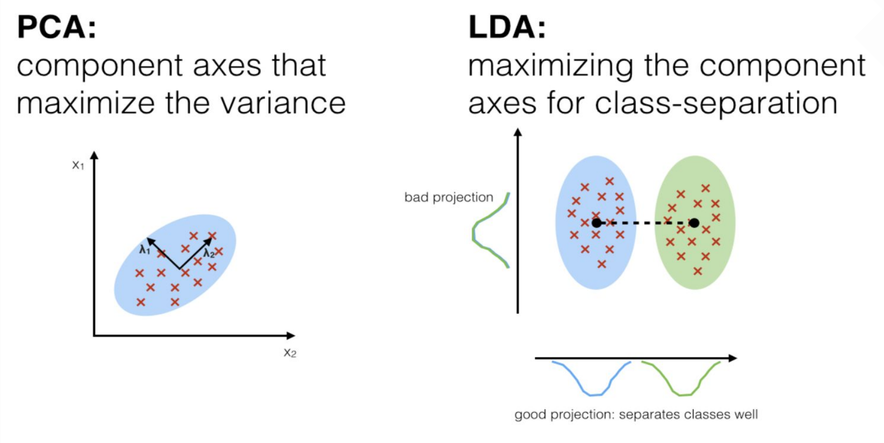
PCA: Summary

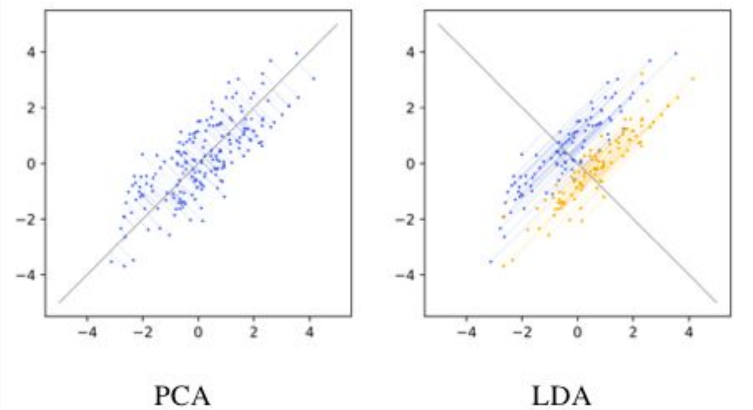
- We start with data consisting of *n* observations and *d* dimensions (X_{nxd})
  - Want to reduce down to *n* observations and *k* (< *d*) dimensions (Z_{nxk})
- PCA creates principal components (PCs)
  - *d* vectors that represent:
    - A series of orthogonal “best-fit” vectors
    - A series of orthogonal vectors that point in the direction of most variance
  - When data is projected onto a PC, it gives one number.
  - Keep first *k* PCs
- Done mathematically through matrix multiplication (python: sklearn)
  - Eigenvalues of covariance matrix: importance of PC
  - Eigenvectors of covariance matrix: direction of PC
- Important to scale data prior to PCA (since it’s based on variance)
  - StandardScaler

Slide credits : Jeremy Eng (LHL, Instructor)

LDA

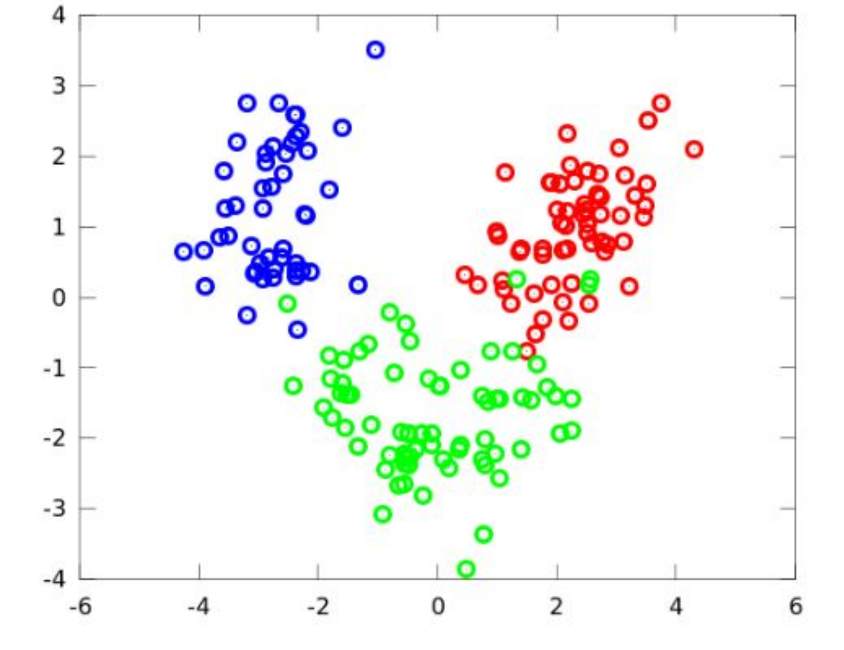
- Linear Discriminant Analysis
- Difference: uses the class label when choosing “PCs” (PC equivalents)
  - LDA is supervised, PCA is unsupervised
- LDA aims to:
  - Minimize “intra-class” variance
  - Maximize “inter-class” variance
- LDA can only be done for classification
  - Target variable is categorical





LDA: Multi-Class Example

- Two dimensions, three classes.



PCA vs LDA

- Both DRT's
- Create new feature dimensions using **linear combinations** of original dimensions
  - Create new basis in a way that minimizes "lost information"
- PCA is unsupervised, LDA is supervised
  - LDA further requires target variable to be categorical
- PCA creates successive PCs in directions of MOST variance in training data
- LDA creates successive components that
  - Minimize "intra-class" variance
  - Maximize "inter-class" variance

Slide credits : Jeremy Eng (LHL, Instructor)

Demo

Questions ?

Resources

Thanks!