Principal Component Analysis

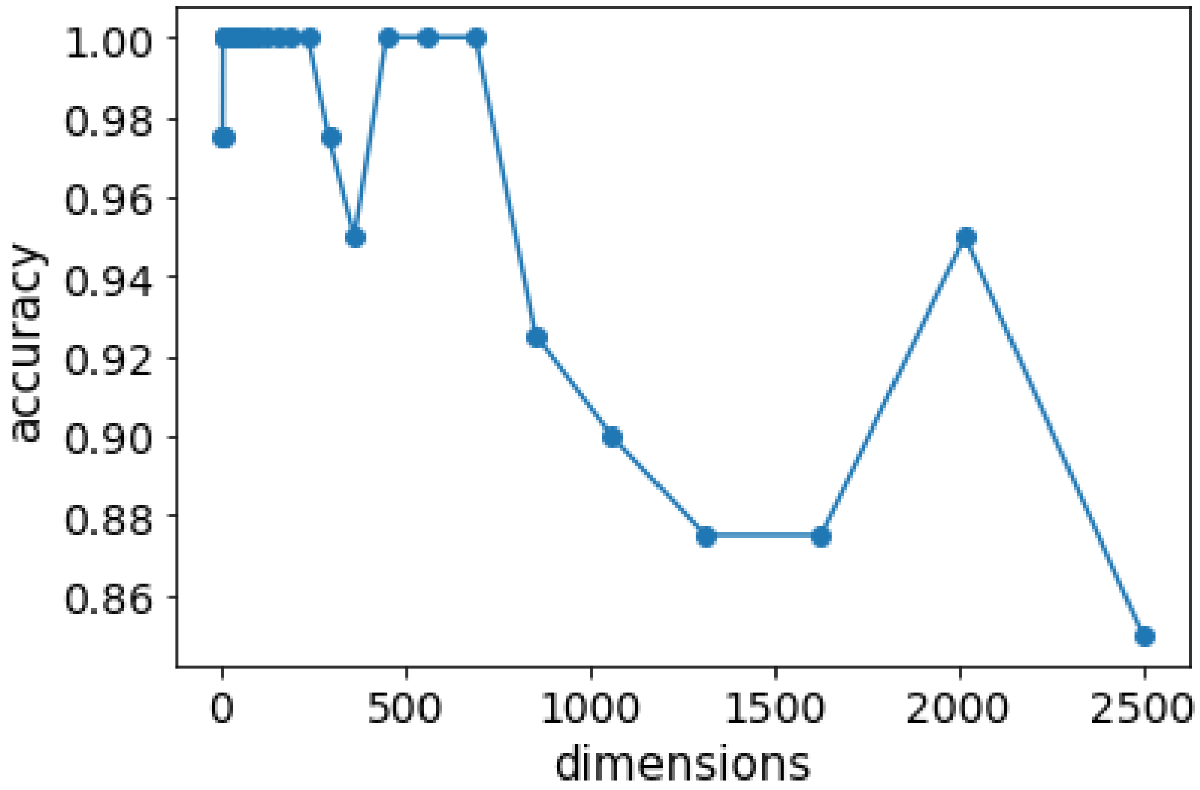
Machine Learning Summer Course 2020

Krishnakant Saboo

25th July 2020

Curse of dimensionality

- Several challenges in dealing with high dimensional data
- Model performance reduces
  - In several cases, #samples < #dimensions
  - All distances become similar in high dimensions
  - There can be noisy features



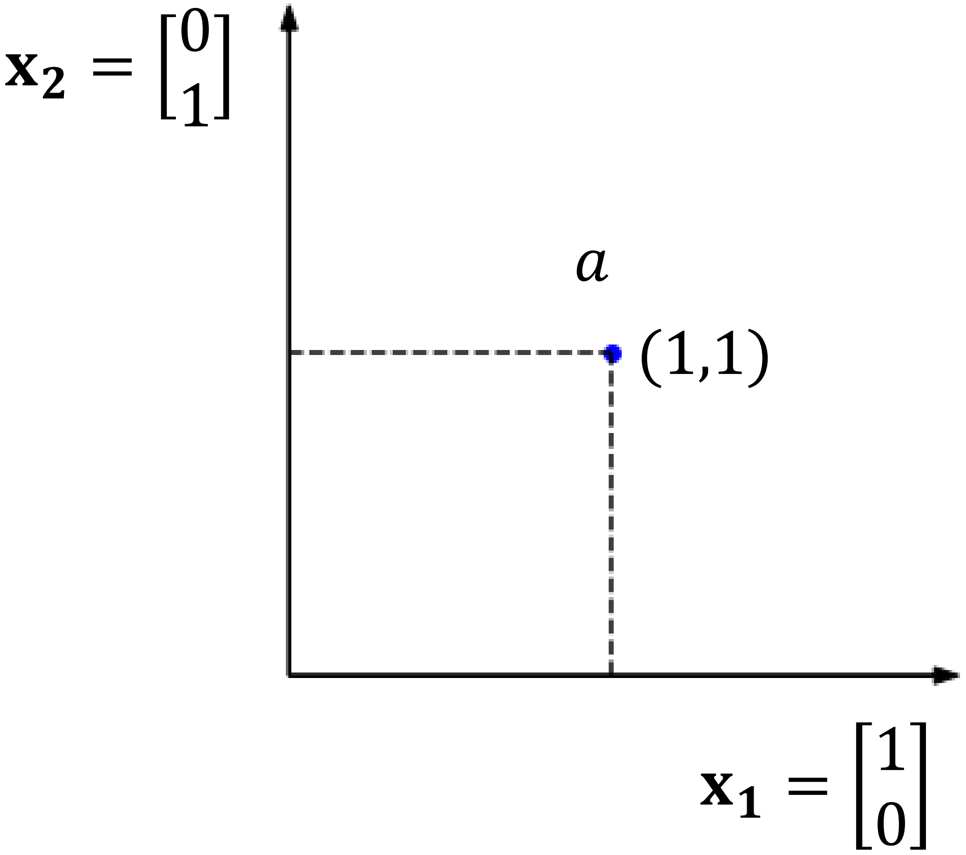
Accuracy decreases as the dimensionality increases. 2 class classification accuracy of SVM classifier applied on 200 samples data (80% training) as the dimensionality increases. Classes are Gaussian with means at 0 and 1 and identity covariance.

Dimensionality reduction

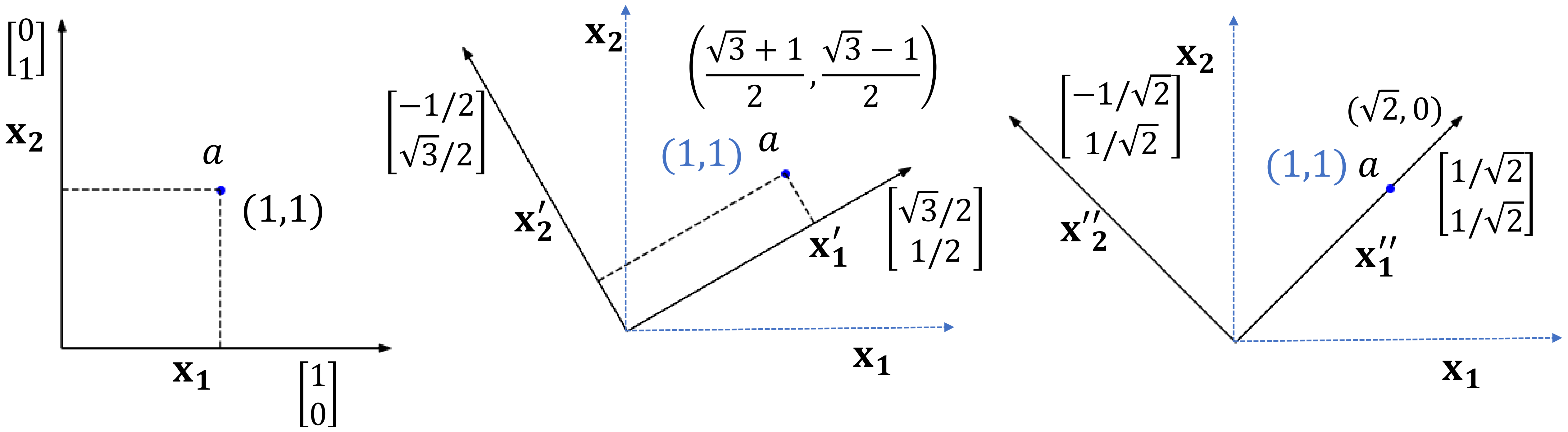
- Solution: Remove some features using domain knowledge
  - Might lose out on useful information
- Another option: Remove dimension that carries lesser information
- Different dimensions have different amount of information
  - Maybe we can remove the dimension which has lesser information?
- **These “dimensions” are inherent in the data and may not always align with the dimensions represented by the features**
- That way, number of dimensions is reduced while minimizing the loss of information

Coordinates recap

- The vector (point) a is in a 2-D space: $a = [1,1]^T$
- Unit vector corresponding to $\mathbf{x_1}$ axis: $\mathbf{x_1} = [1,0]^T$
- Unit vector corresponding to $\mathbf{x_2}$ axis: $\mathbf{x_2} = [0,1]^T$
- Any point in the space can be given as weighted sum of vectors $\mathbf{x_1}$ and $\mathbf{x_2}$

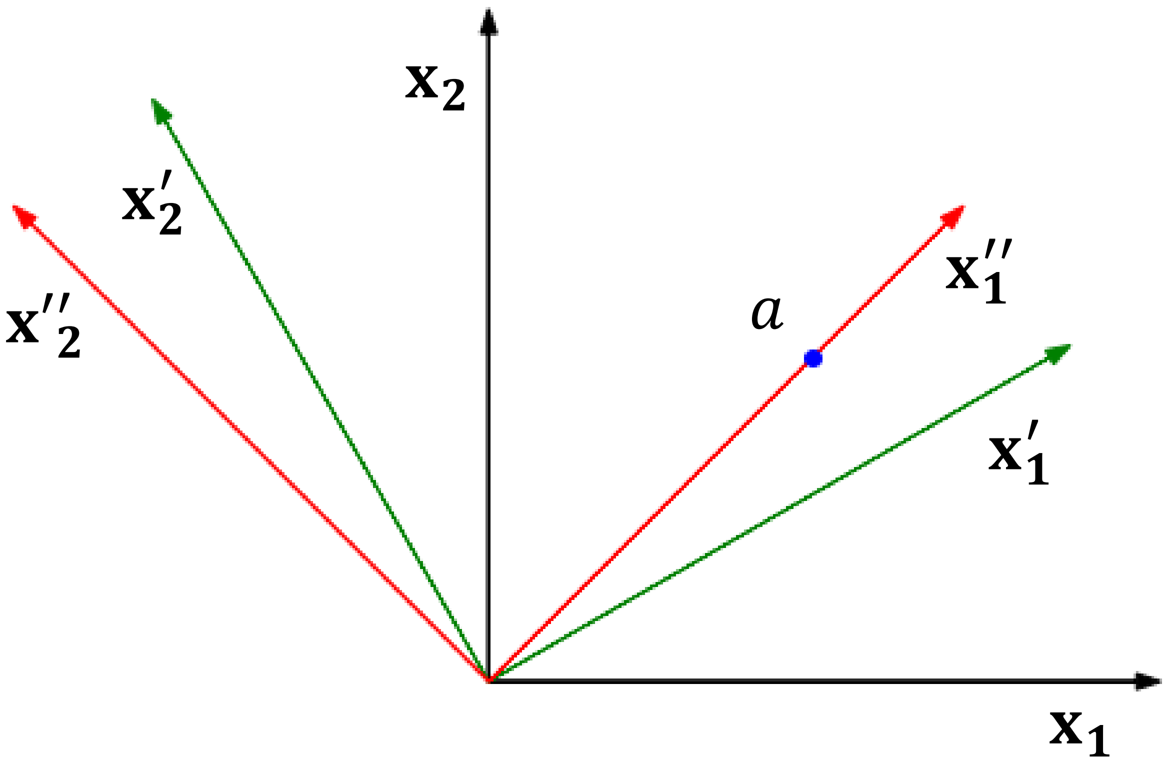


There can be other axes too…



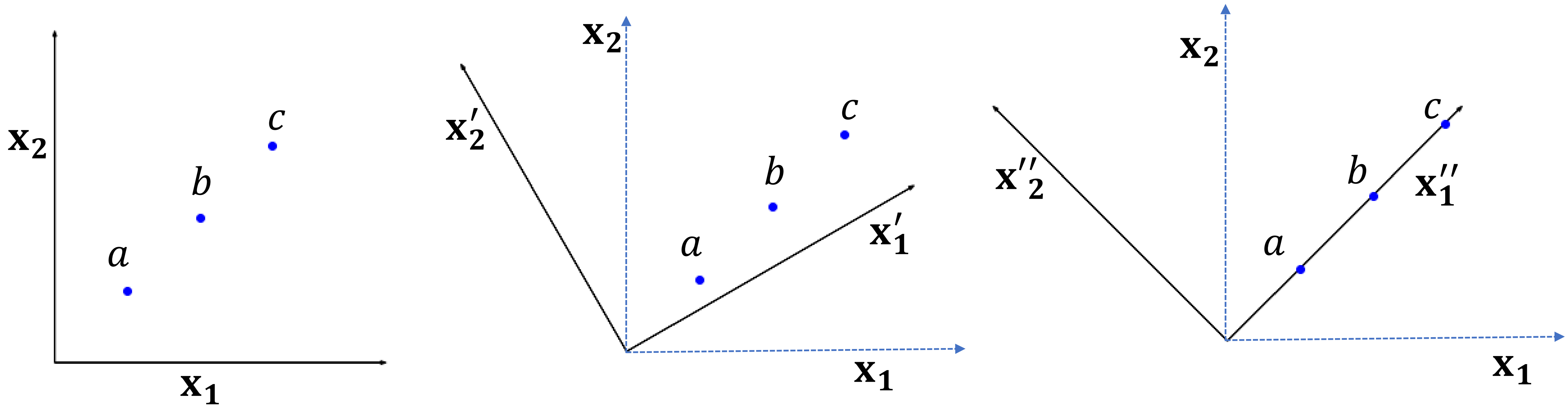
- Point a has an equivalent representation for choice of axes $(\mathbf{x}'_1, \mathbf{x}'_2)$ and $(\mathbf{x}''_1, \mathbf{x}''_2)$
- The other axes are obtained by rotating $(\mathbf{x_1}, \mathbf{x_2})$ around the origin
- All other such axes-pairs obtained by rotation $(\mathbf{x_1}, \mathbf{x_2})$ are valid axes

There can be other axes too…



- The other axes are obtained by rotating $(\mathbf{x}_1, \mathbf{x}_2)$ around the origin
- All other such axes-pairs obtained by rotation $(\mathbf{x}_1, \mathbf{x}_2)$ are valid axes
- The rotated pairs are also valid ‘dimensions’ of the data

Multiple points with rotated axes



All the points can be represented in the 3 axes pairs

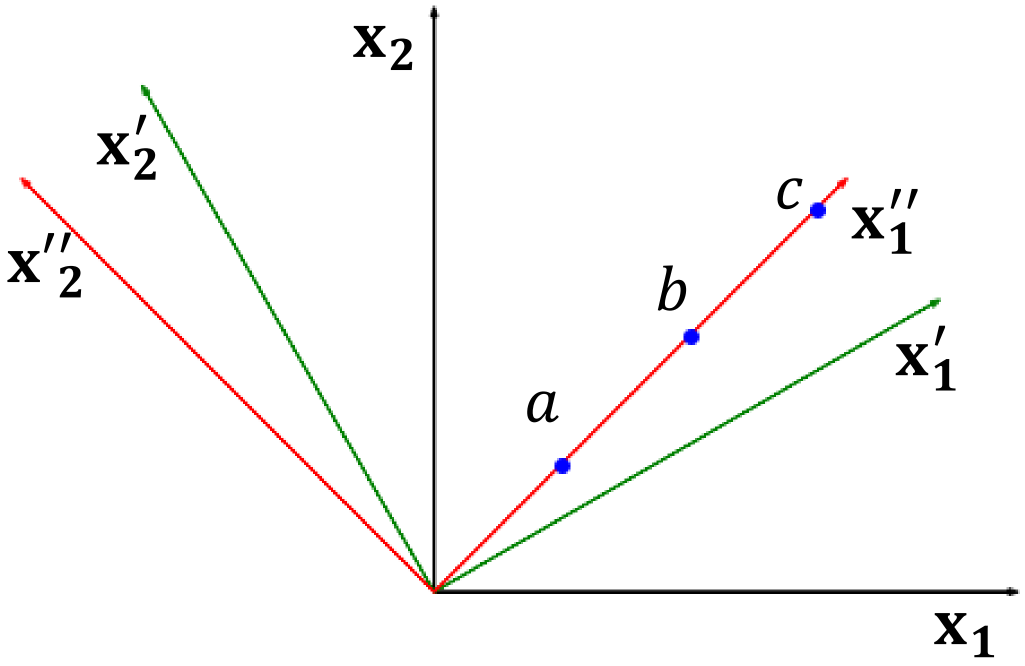
Point	$\mathbf{x}_1$	$\mathbf{x}_2$
a	1	1
b	2	2
c	3	3

Point	$\mathbf{x}'_1$	$\mathbf{x}'_2$
a	$\frac{\sqrt{3}+1}{2}$	$\frac{\sqrt{3}-1}{2}$
b	$\sqrt{3}+1$	2
c	$\frac{3\sqrt{3}+3}{2}$	$\frac{3\sqrt{3}-3}{2}$

Point	$\mathbf{x}''_1$	$\mathbf{x}''_2$
a	$\sqrt{2}$	0
b	$2\sqrt{2}$	0
c	$3\sqrt{2}$	0

Multiple points with rotated axes

- If $(\mathbf{x}_1'', \mathbf{x}_2'')$ is the choice of axes, then the data is essentially one dimensional
- The data here is one dimensional
- For any given set of points, if we can find a axes pair such that few coordinates are needed, then we have achieved **dimensionality reduction**

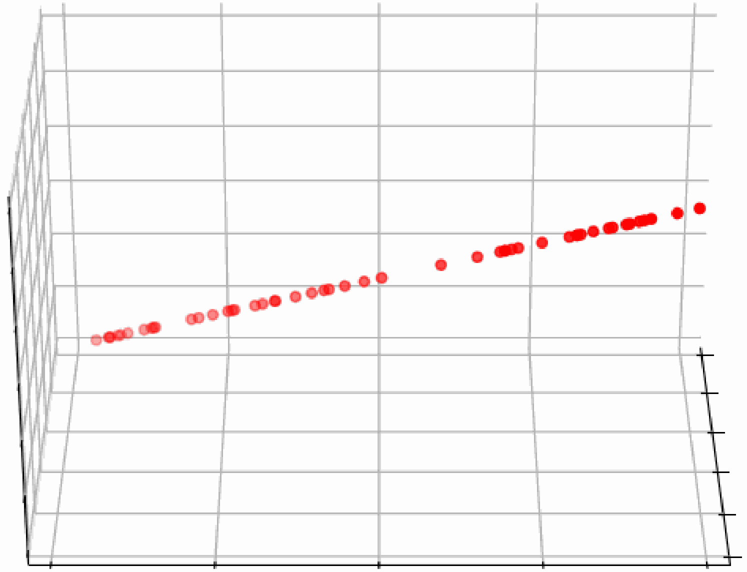


Point	$\mathbf{x}_1$	$\mathbf{x}_2$
a	1	1
b	2	2
c	3	3

Point	$\mathbf{x}_1'$	$\mathbf{x}_2'$
a	$\frac{\sqrt{3}+1}{2}$	$\frac{\sqrt{3}-1}{2}$
b	$\sqrt{3}+1$	2
c	$\frac{3\sqrt{3}+3}{2}$	$\frac{3\sqrt{3}-3}{2}$

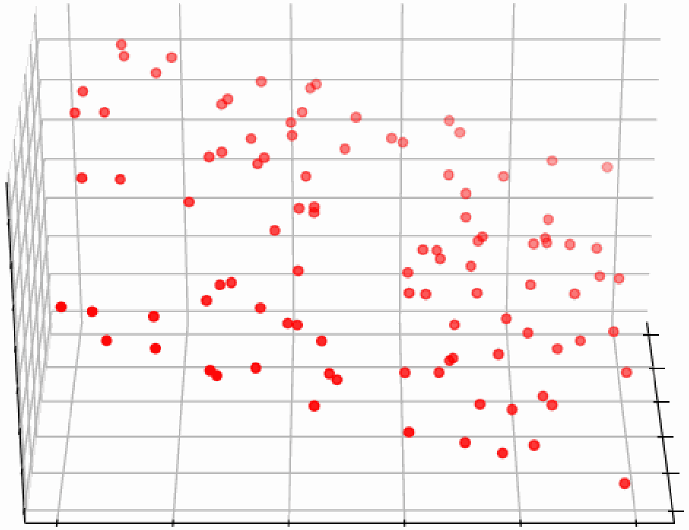
Point	$\mathbf{x}_1''$	$\mathbf{x}_2''$
a	$\sqrt{2}$	0
b	$2\sqrt{2}$	0
c	$3\sqrt{2}$	0

What is the dimensionality of the data here?



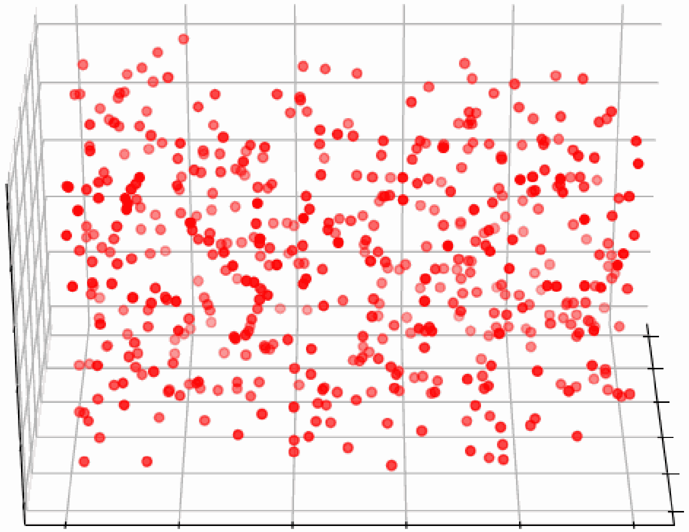
The data shown here is one dimensional

What is the dimensionality of the data here?



The data shown here is two dimensional

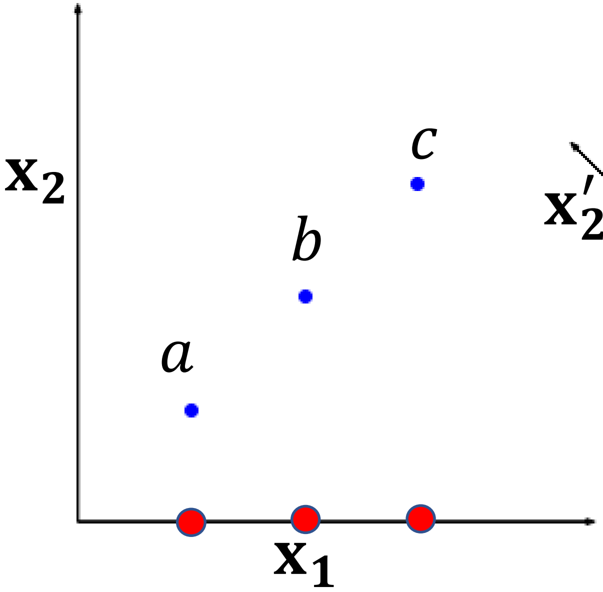
What is the dimensionality of the data here?

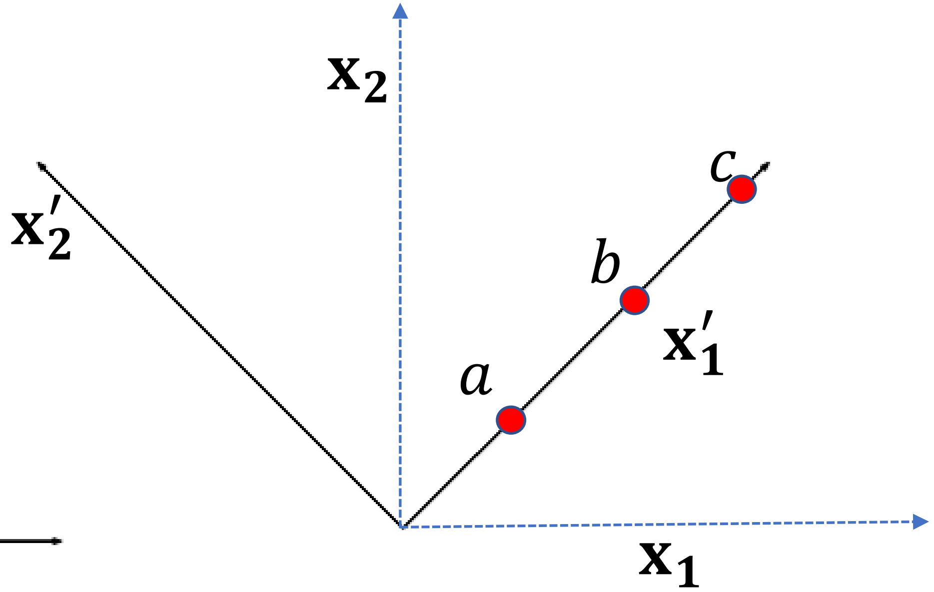


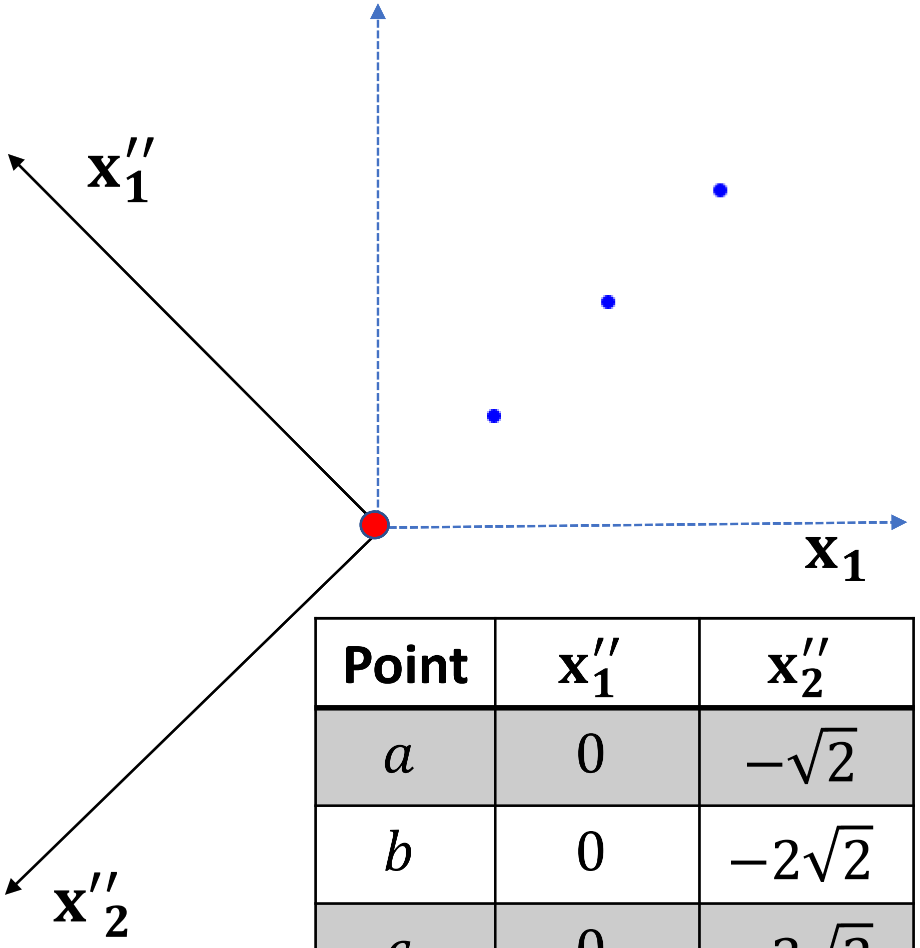
The data shown here is three dimensional

Criteria for selecting axes

Consider the case that after transformation (***projection***), the first axis is kept.
Which of the following is the best axes?







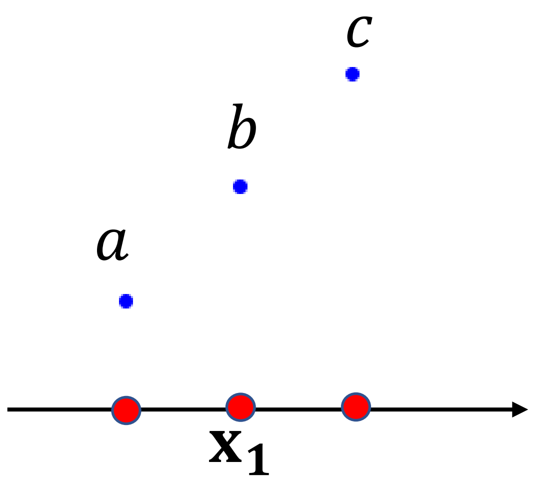
Point	$\mathbf{x}_1$	$\mathbf{x}_2$
a	1	1
b	2	2
c	3	3

Point	$\mathbf{x}'_1$	$\mathbf{x}'_2$
a	$\sqrt{2}$	0
b	$2\sqrt{2}$	0
c	$3\sqrt{2}$	0

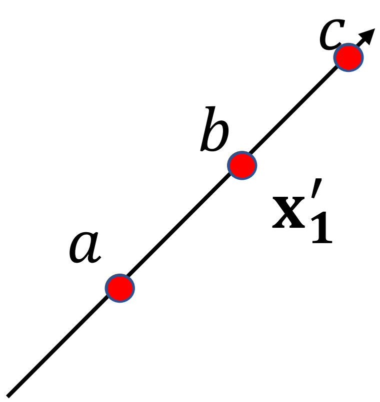
Point	$\mathbf{x}''_1$	$\mathbf{x}''_2$
a	0	$-\sqrt{2}$
b	0	$-2\sqrt{2}$
c	0	$-3\sqrt{2}$

Criteria for selecting axes

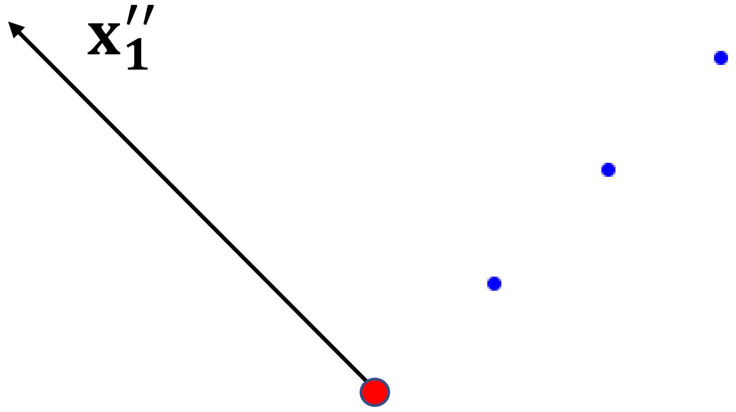
- The second scenario is the best because the entire “spread” of the data is conserved; spread is the variance
- Variance can also be thought of as the information in the data



Point	$\mathbf{x}_1$
a	1
b	2
c	3



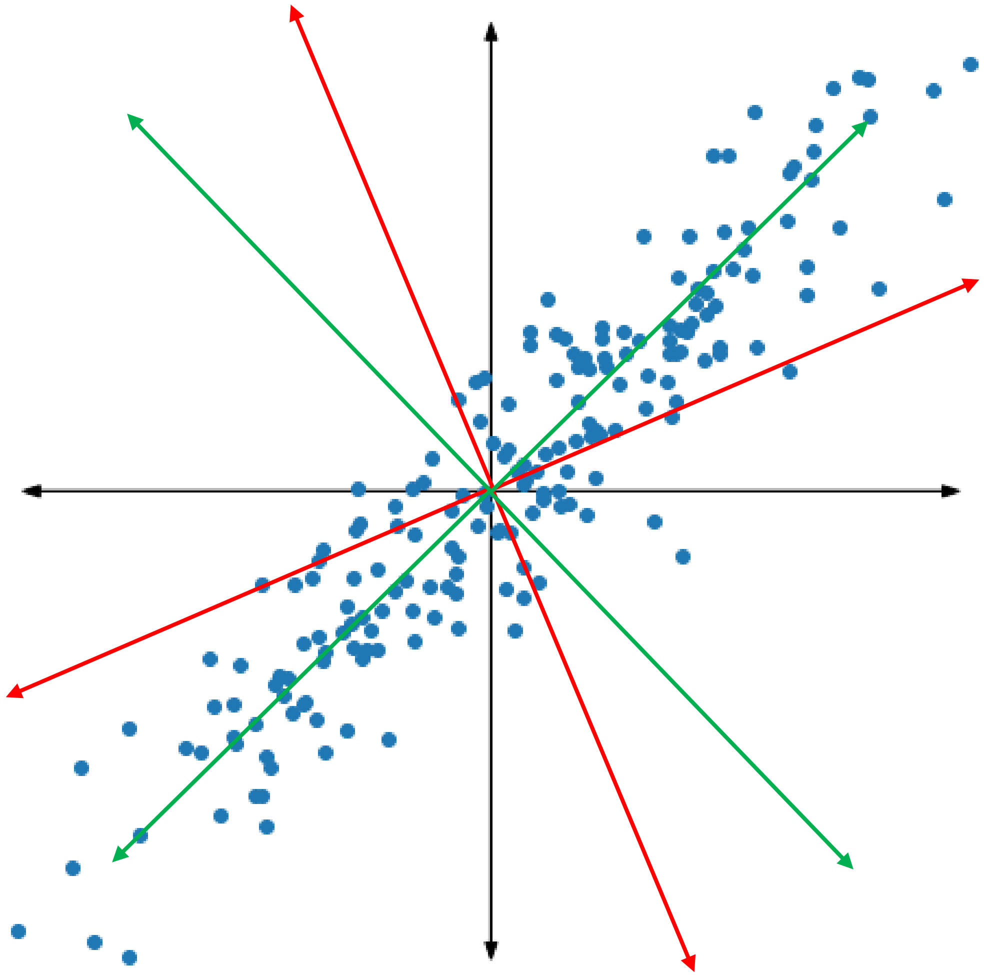
Point	$\mathbf{x}_1'$
a	$\sqrt{2}$
b	$2\sqrt{2}$
c	$3\sqrt{2}$



Point	$\mathbf{x}_1''$
a	0
b	0
c	0

Data may not be collinear

- Goal is to rotate the axes and then keep data of only one axis
- Which orientation of axes pairs to choose and which of the two axis to keep?
- Criteria of maximizing variance can be applied here too
  - We want to minimize the information loss



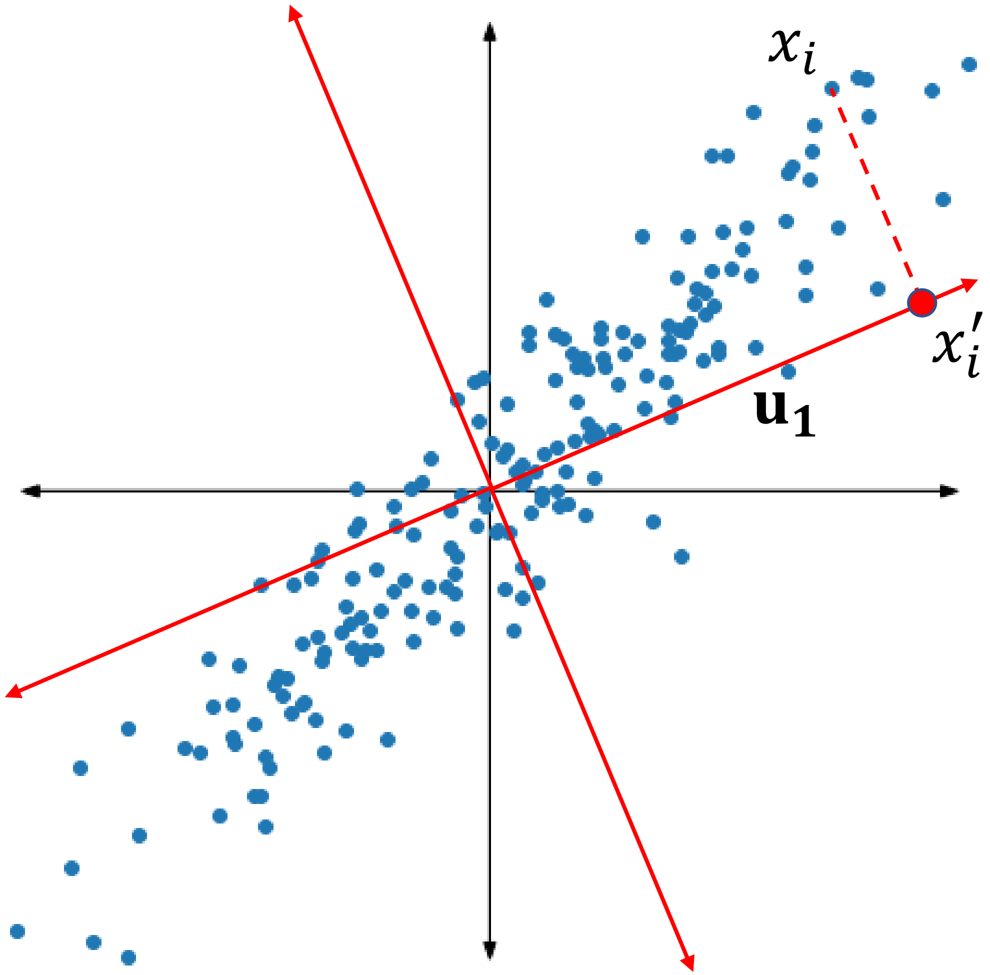
Optimization formulation

- Let the data be $x_1, x_2, \ldots, x_N$ where $x_i = [x_{i1}, x_{i2}]^T$
- Let $\mathbf{u_1}$ be the unit vector corresponding to the axis that is retained after dimensionality reduction
- x_i' is the projection of x_i on $\mathbf{u_1}$

$$x_i' = \mathbf{u_1}^T x_i$$

- Variance: $\frac{1}{N}\sum_{i=1}^{N}(x_i' - \overline{x'})^2$

Mean of all projections



Optimization formulation

- Variance: $\frac{1}{N}\sum_{i=1}^{N}(x_i' - \overline{x'})^2$
- Substituting:

$$x_i' = \mathbf{u}_1^T x_i$$

we get,

Mean of data

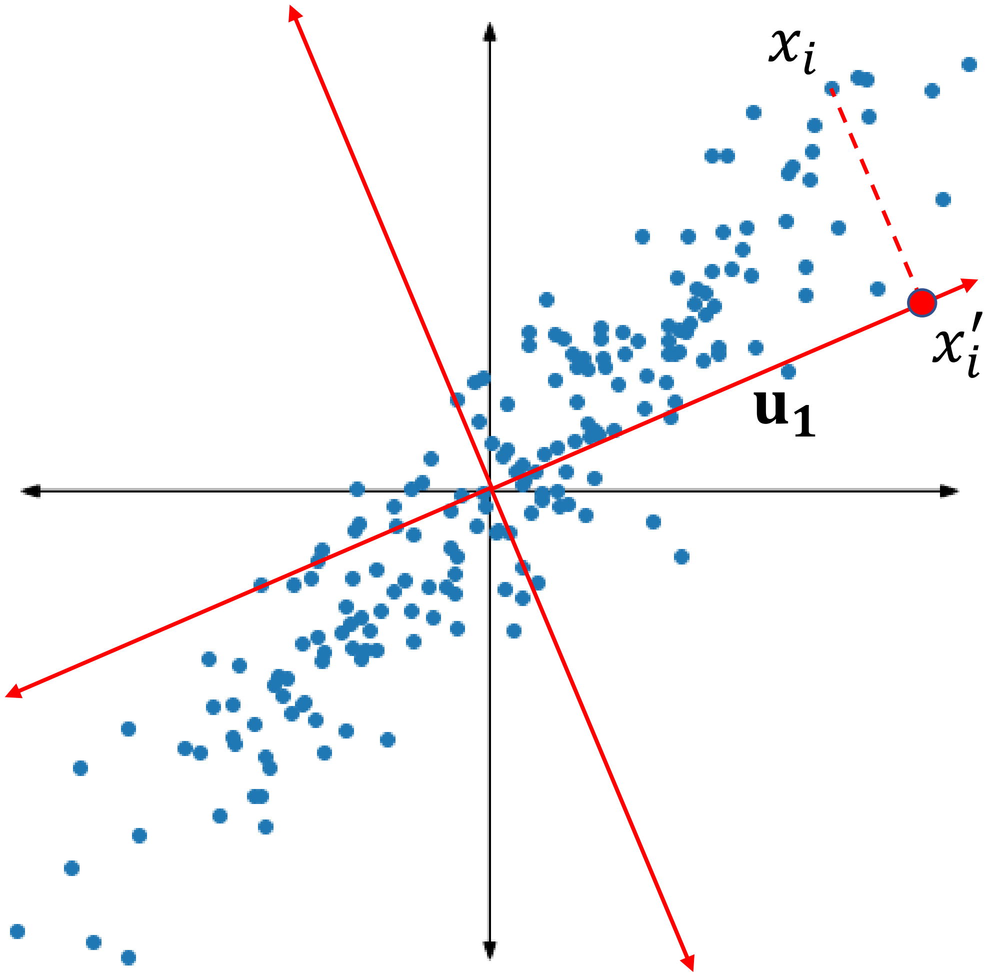
$$\frac{1}{N}\sum_{i=1}^{N}(\mathbf{u}_1^T x_i - \mathbf{u}_1^T \bar{x})^2$$

$$= \mathbf{u}_1^T S \mathbf{u}_1$$

Covariance matrix

where,

$$S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T$$

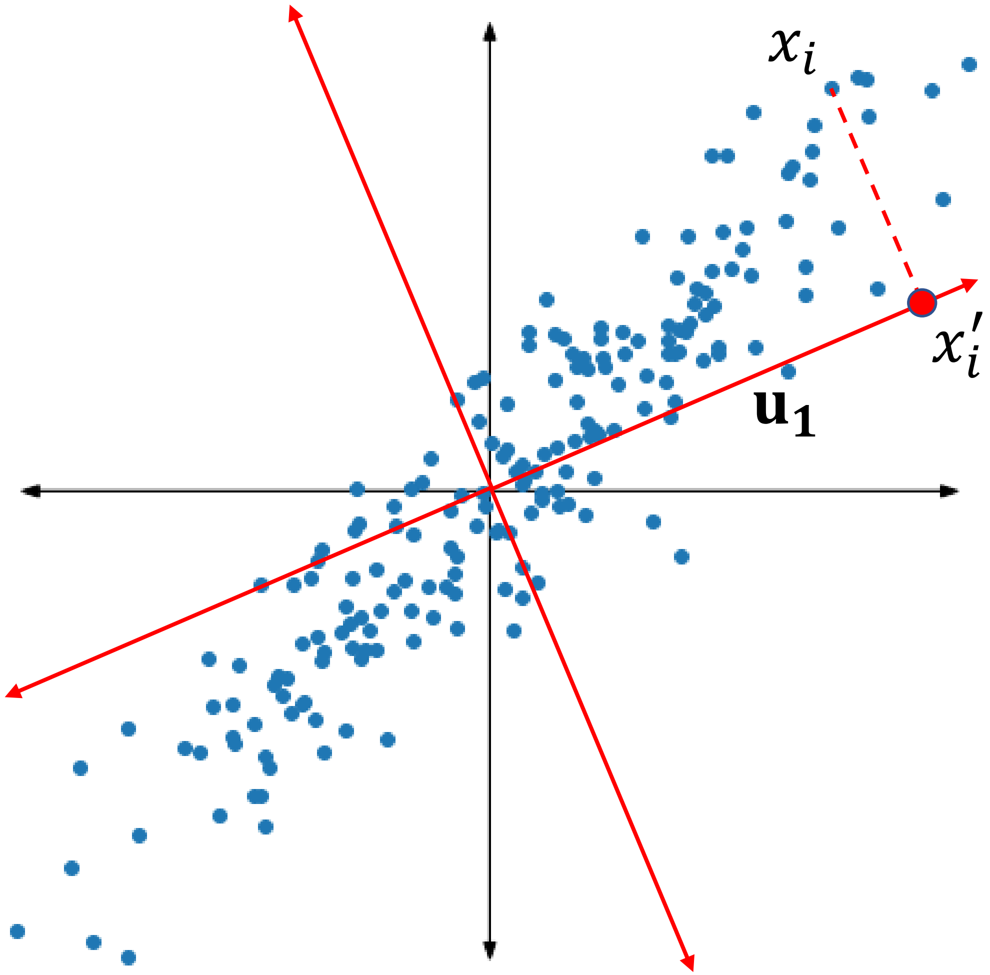


Optimization problem

- Variance: $\mathbf{u_1}^T S \mathbf{u_1}$
- To find best $\mathbf{u_1}$, maximize the variance

$$\max_{\mathbf{u_1}} \mathbf{u_1}^T S \mathbf{u_1}$$

$$s.t.\ \mathbf{u_1}^T \mathbf{u_1} = 1$$



Solution to the optimization problem

- To find best $\mathbf{u_1}$, maximize the variance

$$\max_{\mathbf{u_1}} \ \mathbf{u_1}^T S \mathbf{u_1}$$

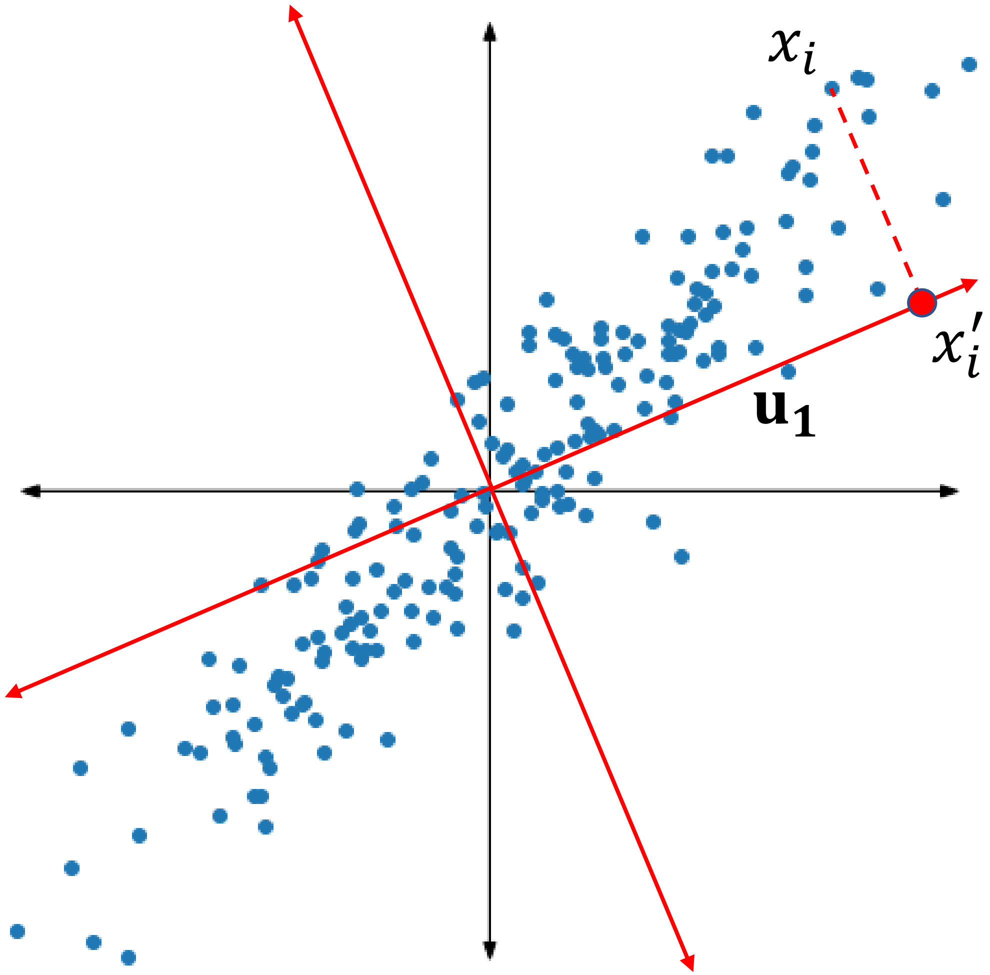
$$s.t. \ \mathbf{u_1}^T \mathbf{u_1} = 1$$

- Solution: $\mathbf{u_1}$ is the first eigenvector of covariance matrix S, i.e.,

$$S\mathbf{u_1} = \lambda_1 \mathbf{u_1}$$

where λ_1 is the largest eigenvalue of S.

- Variance explained by $\mathbf{u_1}$ is λ_1



To retain more than one dimension…

- For data with d dimensions, we might be interested in the $k < d$ axes $\mathbf{u_1}, \mathbf{u_2}, \dots, \mathbf{u_k}$, such that the variance of the projected data is maximized
- A similar optimization problem as above can be setup
- Solution is to choose the axes as the first k eigenvectors of S, i.e.,

$$\mathrm{S}\mathbf{u_j} = \lambda_j \mathbf{u_j} \quad \text{for } j = 1, \dots, k$$

- Variance explained by $\mathbf{u_j}$ is λ_j; $\mathbf{u_j}$ is the jth principal component
- Variance explained by $\mathbf{u_1}, \dots, \mathbf{u_k}$ is $\lambda_1 + \lambda_2 + \cdots + \lambda_k$
- Total variance is the original data is sum of all eigenvalues $\lambda_1 + \lambda_2 + \cdots + \lambda_d$
- In practice, k might not be known to begin with, so all eigenvectors and eigenvalues are computed and then then k is decided

PCA applied to Iris data

- 150 samples with 3 classes of flowers
- 4 dimensions: petal width, petal length, sepal width, sepal length
- % variance explained by j^{th} component = $\frac{\lambda_j}{\lambda_1+\cdots+\lambda_d}$
- 92% of the variance is explained by first principal component (PC)

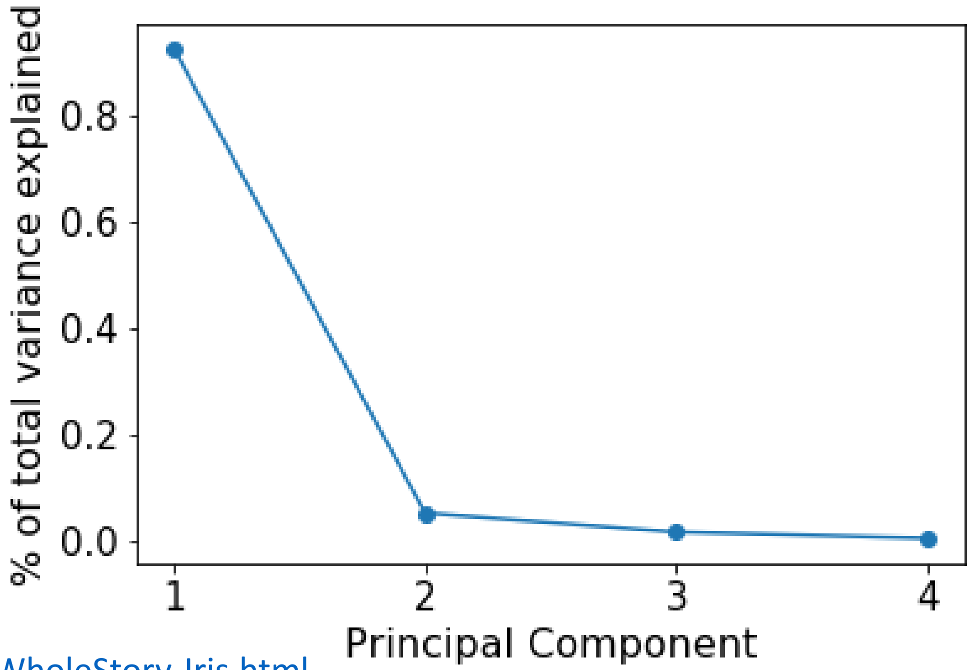
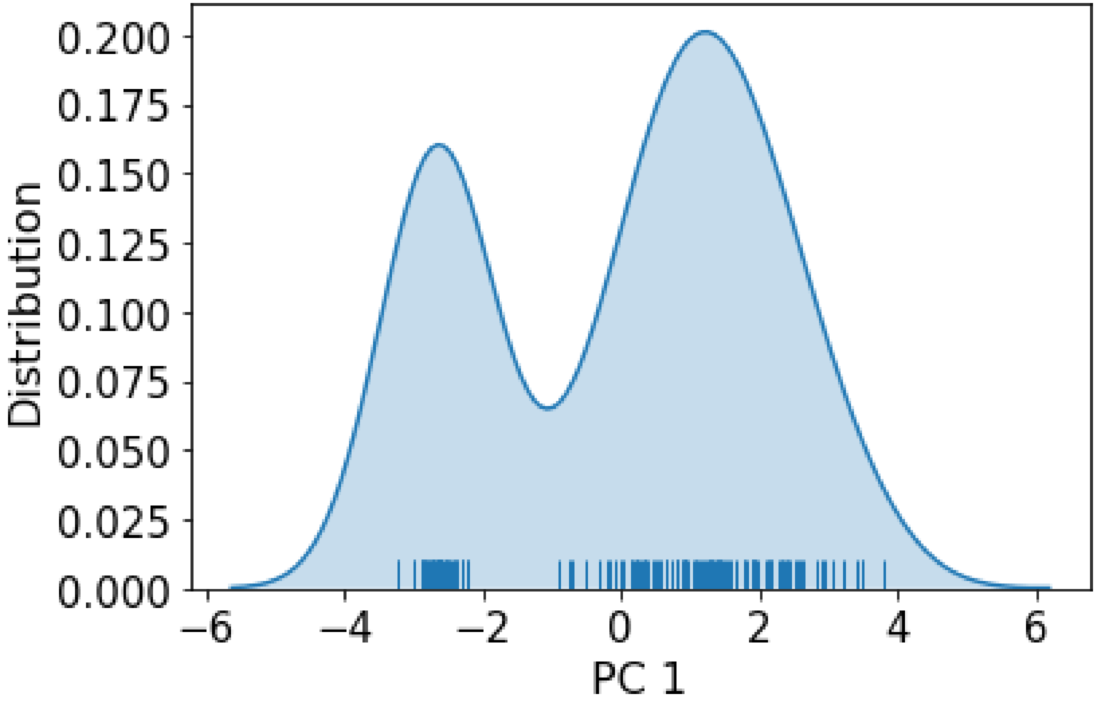
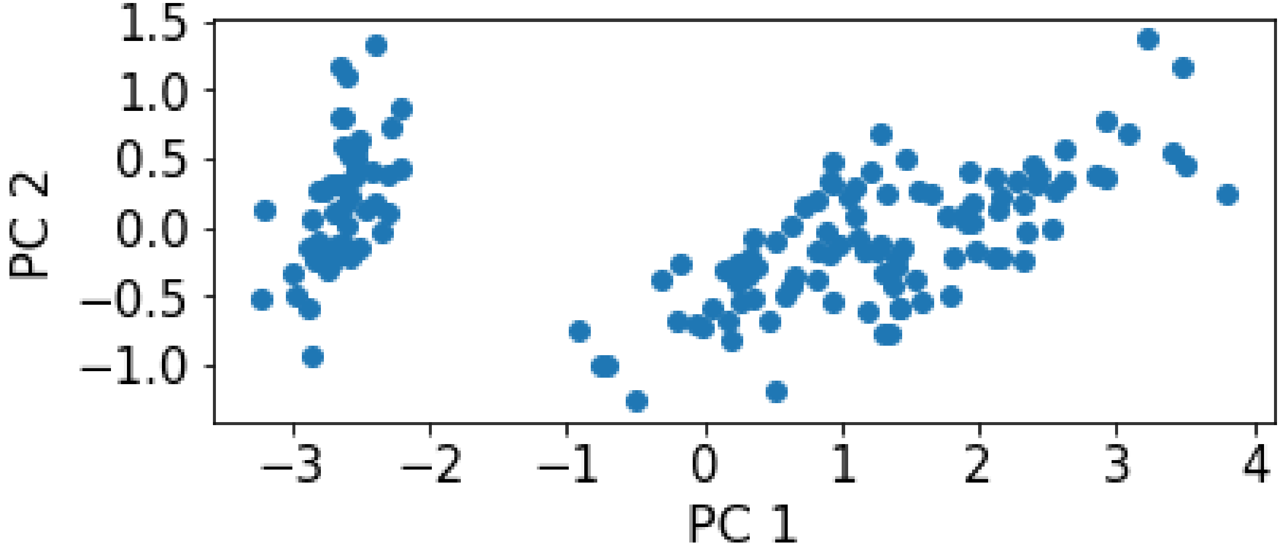


Image source: http://www.lac.inpe.br/~rafael.santos/Docs/CAP394/WholeStory-Iris.html

Visualization of data in PC space



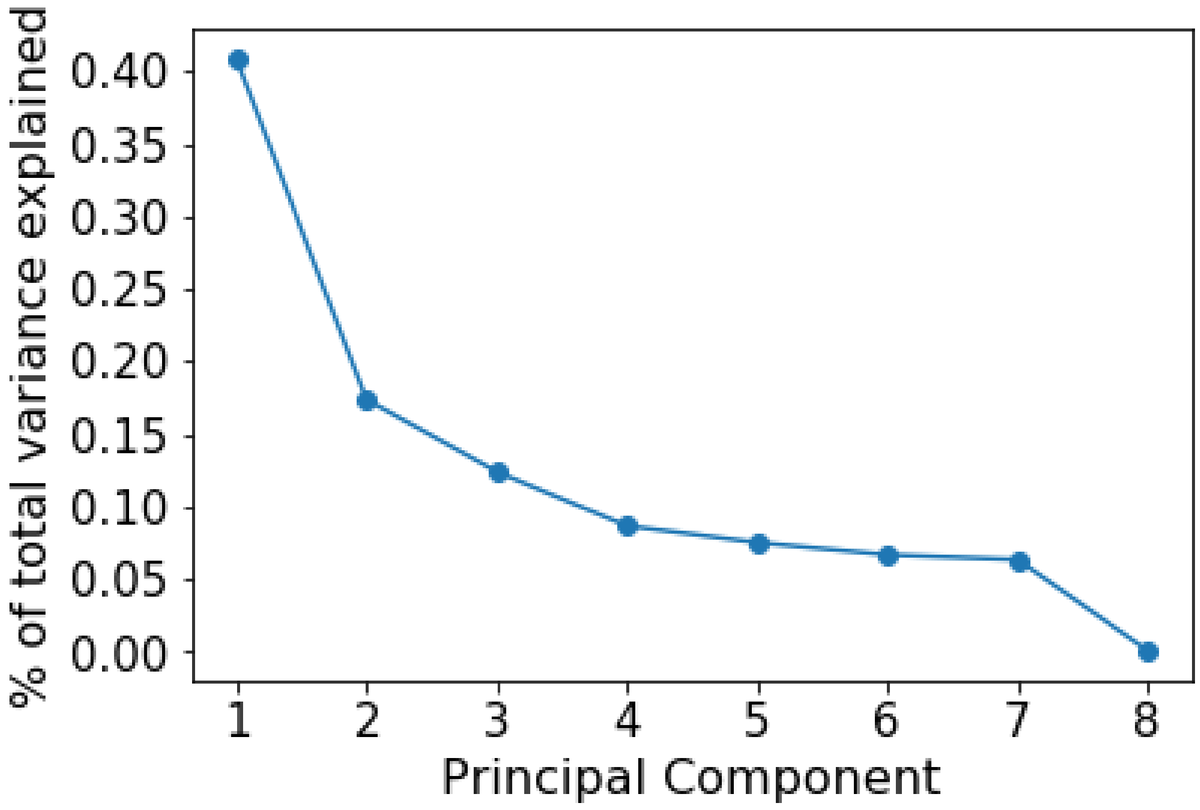
$k = 1$; projection on only the first PC (92% variance)



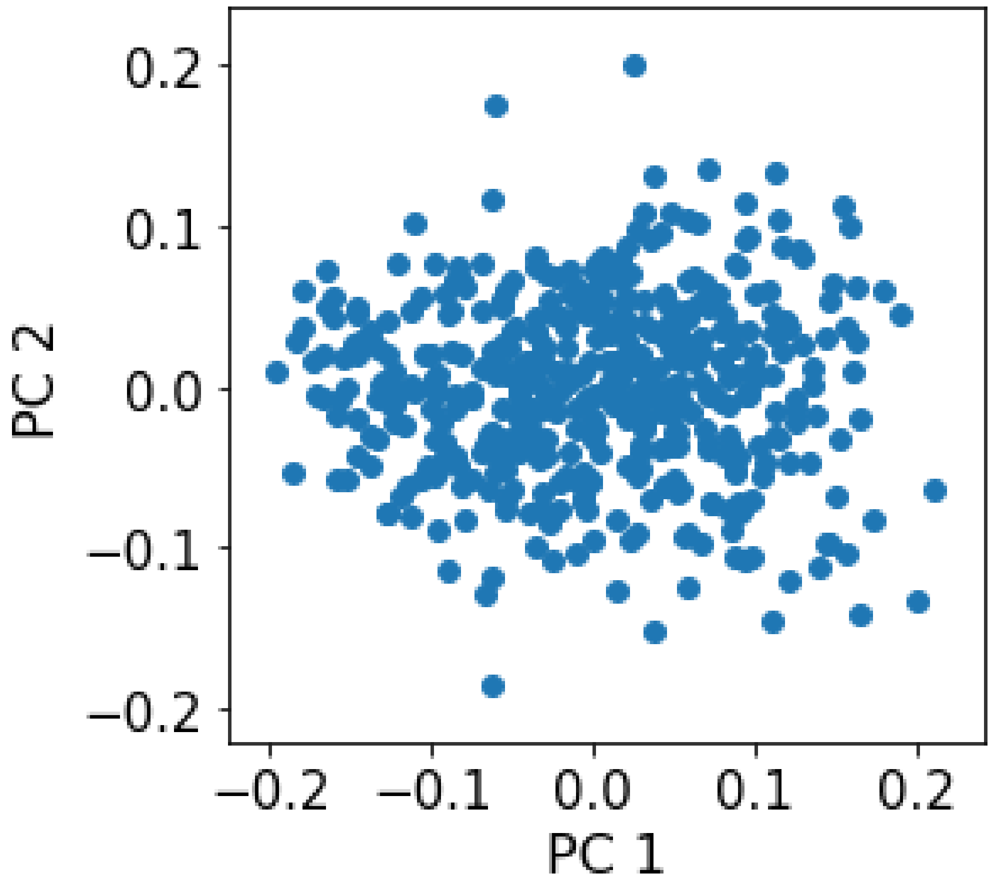
$k = 2$; projection on the first two PCs (98% variance)

PCA applied to Diabetes data

- 442 diabetic individuals with information on one-year progression of disease
- 8 dimensions: age, body mass index, average blood pressure, and five blood serum measurements
- 41% of the variance is explained by first principal component (PC)
- Number of components to retain
  - Rule of thumb: 80%
  - Elbow



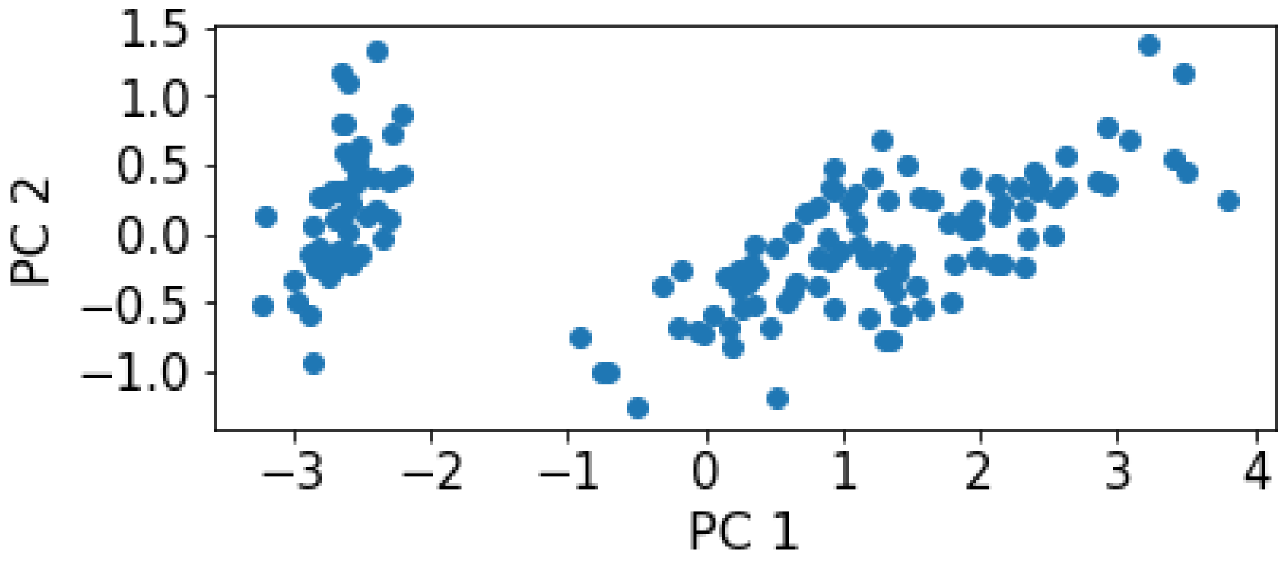
Visualization of data in PC space



$k = 2$; projection on the first two PCs (58% variance)

Interpreting the PCs

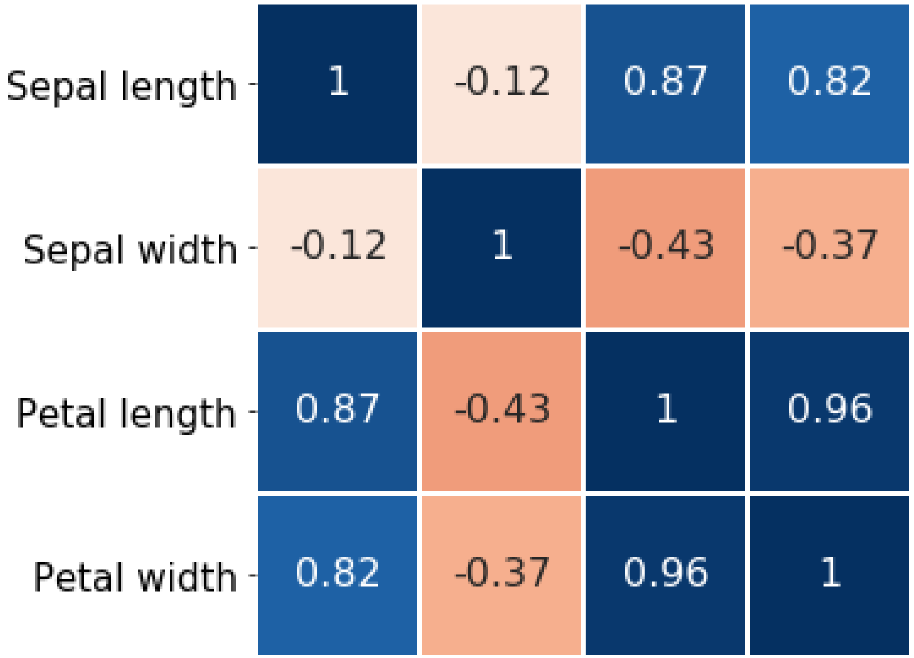
	PC 1	PC 2
Sepal length	0.36	0.65
Sepal width	-0.08	0.71
Petal length	0.86	-0.17
Petal width	0.36	-0.07

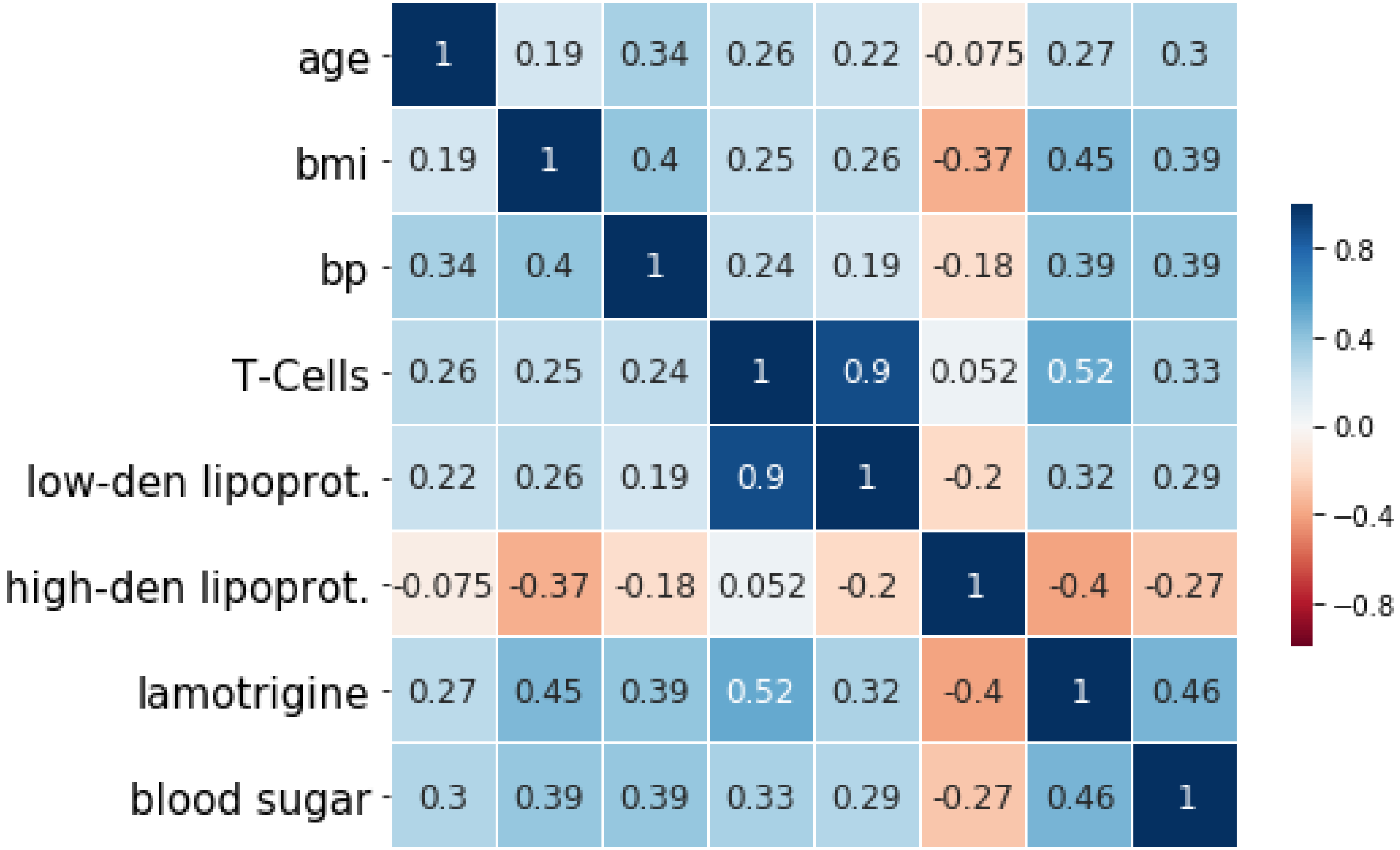


- PC1 is mainly driven by petal length
  - High value of PC1 suggests flower has long petal
  - Note that the projected data has a zero mean
- PC2 is mainly driven by septal width and length
  - High value of PC2 suggests that a flower has large sepals

Relation of eigenvalues to covariance matrix

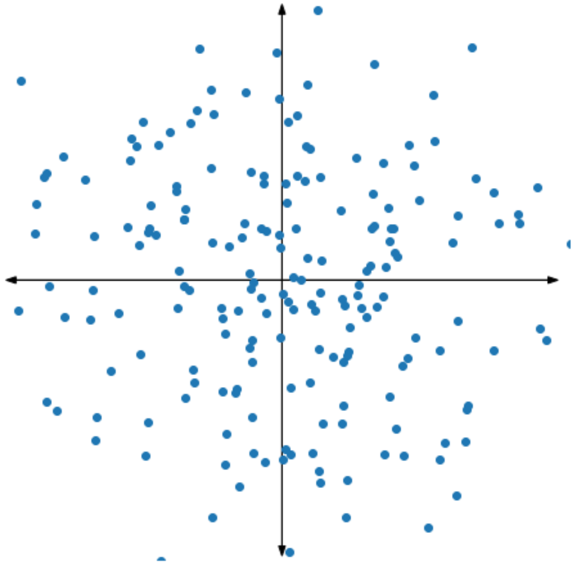
- Why was the % variance explained by first component so different in the two datasets?





- Correlation matrix (related to covariance matrix) for the two datasets
- More high correlation between variables in Iris dataset

Relation between covariance matrix and eigenvalues

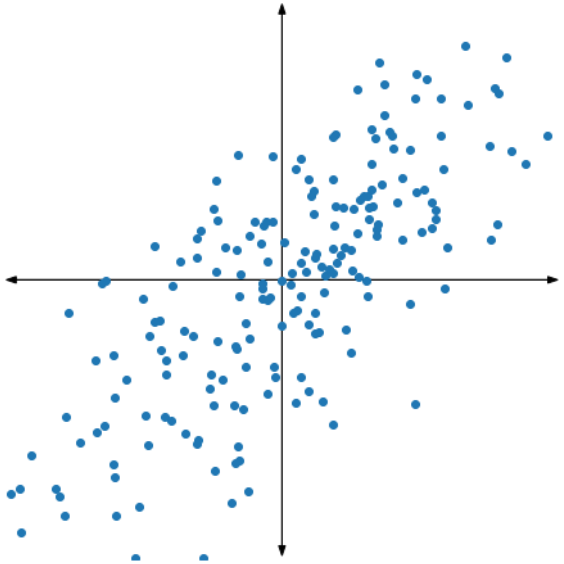


Cov. mat:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Tot. var. = 2

$\lambda_1 = 1$

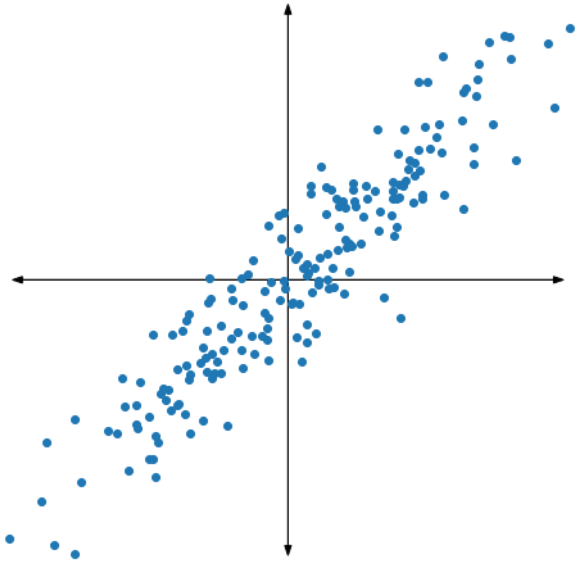


Cov. mat:

$$\begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Tot. var. = 2

$\lambda_1 = 1.8$



Cov. mat:

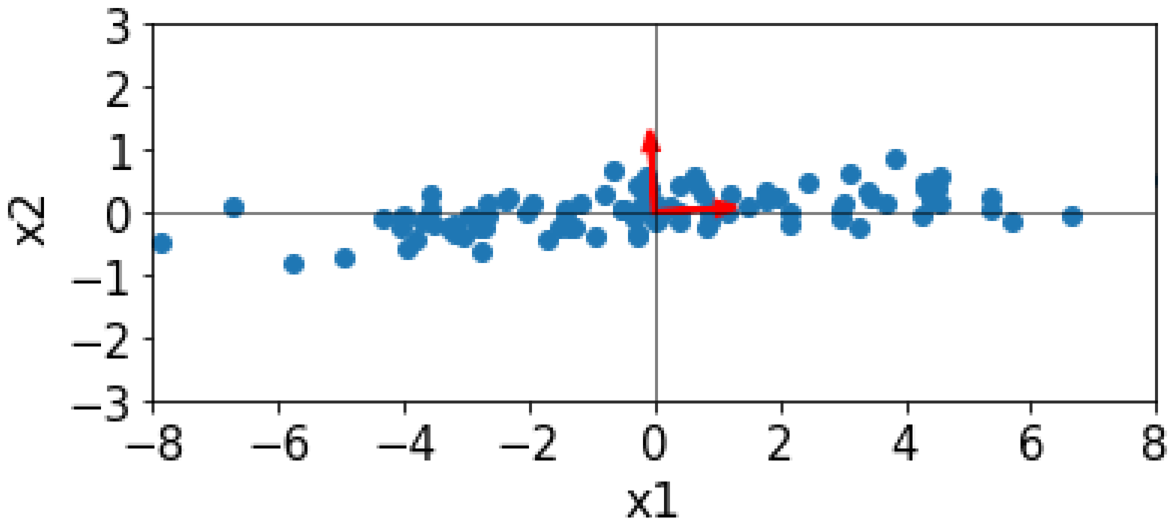
$$\begin{bmatrix} 1 & 0.95 \\ 0.95 & 1 \end{bmatrix}$$

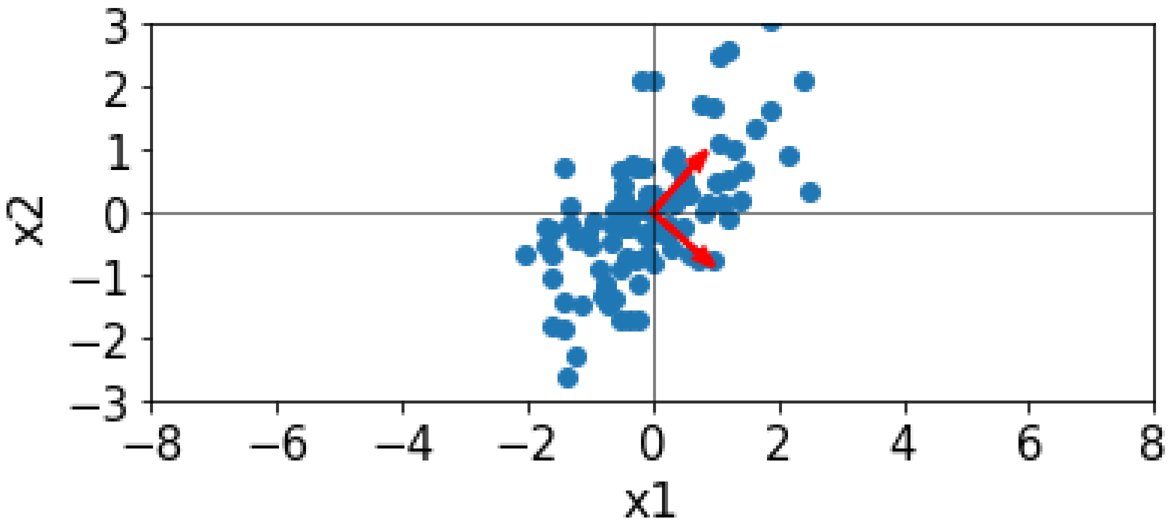
Tot. var. = 2

$\lambda_1 = 1.95$

- As the covariance increases, first eigenvalues increases
- Consequently, % variance explained by first PC will also increase

Scale of the features affects PCA





Cov. mat:

$$\begin{bmatrix} 10 & 0.5 \\ 0.5 & 0.1 \end{bmatrix}$$

Tot. var. = 10.1

$$\lambda_1 = 10.02$$

PC1 $\begin{bmatrix} 0.99 \\ 0.06 \end{bmatrix}$ PC2 $\begin{bmatrix} -0.06 \\ 0.99 \end{bmatrix}$

Cov. mat:

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

Tot. var. = 2

$$\lambda_1 = 1.5$$

PC1 $\begin{bmatrix} 0.66 \\ 0.75 \end{bmatrix}$ PC2 $\begin{bmatrix} -0.75 \\ 0.66 \end{bmatrix}$

- In the first case, >90% of the variance is explained by PC1 but PC1 is mainly driven by the first feature (since it has a relatively larger variance)
- In the second case, 75% of the variance is explained by PC1 and it has similar contribution of both the features
- The correlation between the two features is the same in both cases

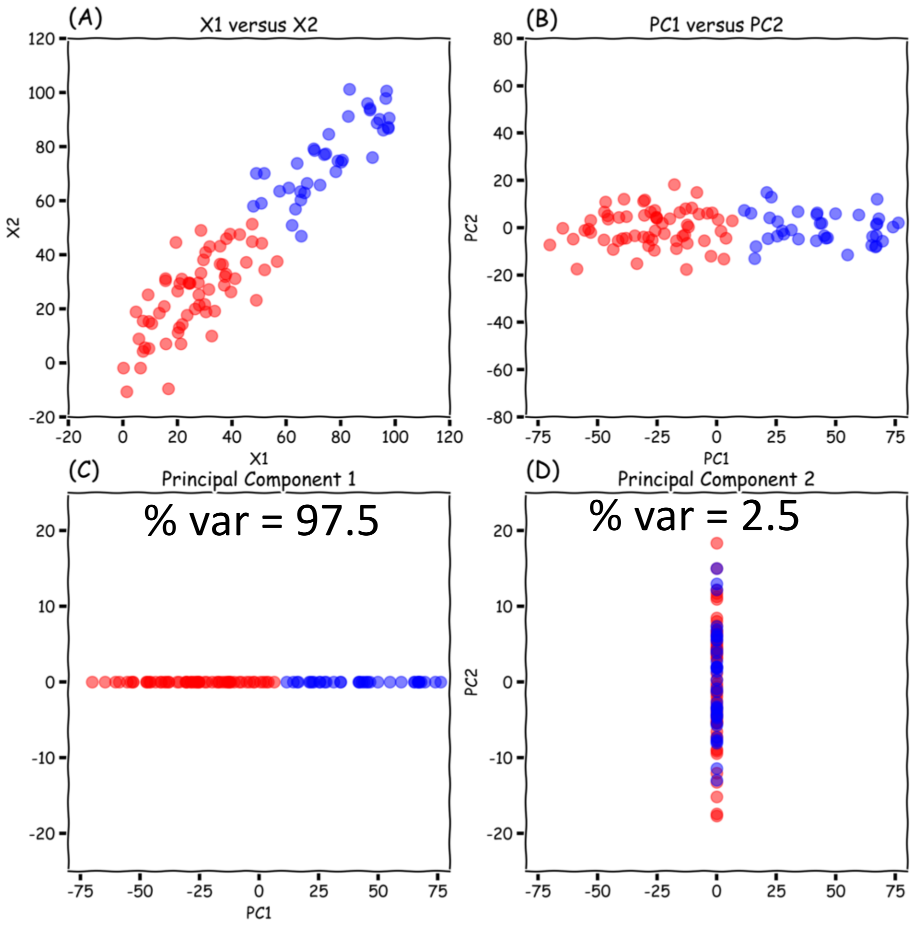
Scale of features affects PCA

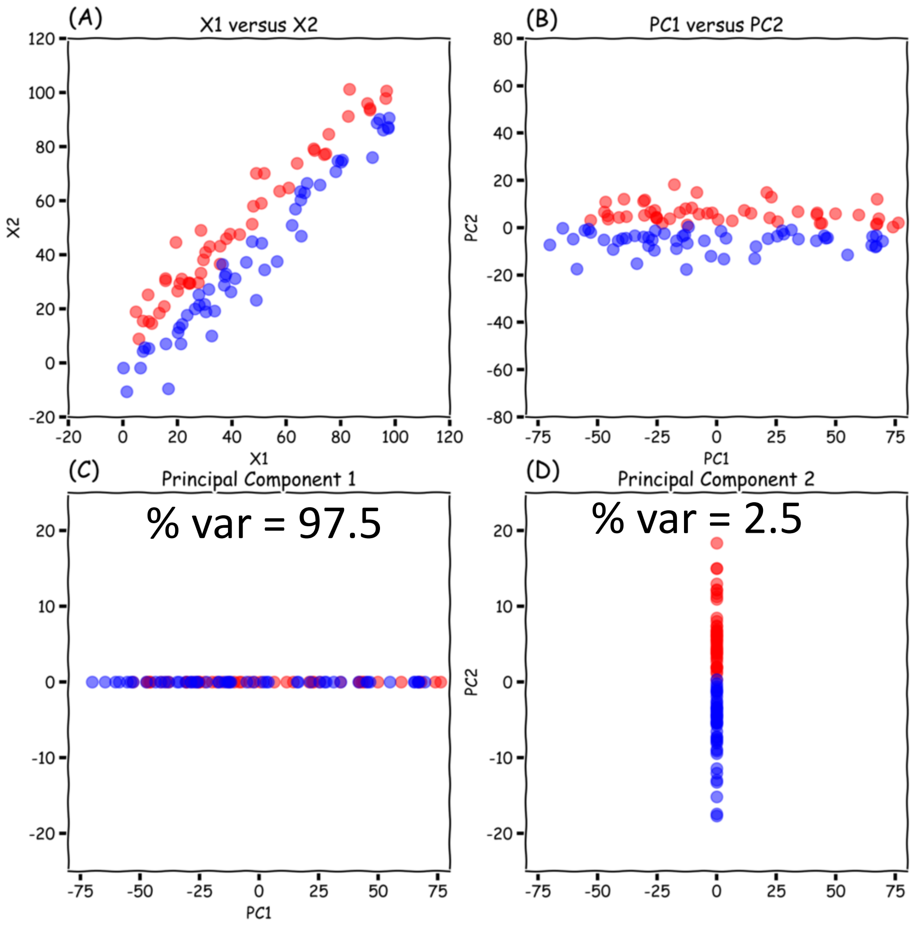
- Features with larger variance dominate PCs and may result in loss of useful information
  - Example: Analysing COVID-19 data with features of age (range 20-80), blood oxygen level (range 90-98), body temperature (range 97-104)
  - Most important features relation to severity might be oxygen level but it has a smaller variance compared to others
- Solution: Standardizing features (making them zero mean and unit variance) before PCA computations
  - Equivalent to using correlation matrix for analysis

Covariance matrix = $\begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$ → Correlation matrix= $\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$

- For a given application of PCA, should correlation matrix be used, or covariance matrix be used?
  - Depends on the application

Loss of information relevant for classification





- Inherent assumption is that variance between clusters/classes would be more than variance within clusters/classes
- Removing low variance PC might result in loss of information relevant to classification

Image source: https://www.robertoreif.com/blog/2018/1/9/pca

Good references for PCA

- Bishop book on pattern recognition
- http://www.cse.psu.edu/~rtc12/CSE586Spring2010/lectures/pcaLectureShort.pdf
- https://www.cs.cmu.edu/~mgormley/courses/10701-f16/slides/lecture14-pca.pdf

Miscellaneous

Which method to use?

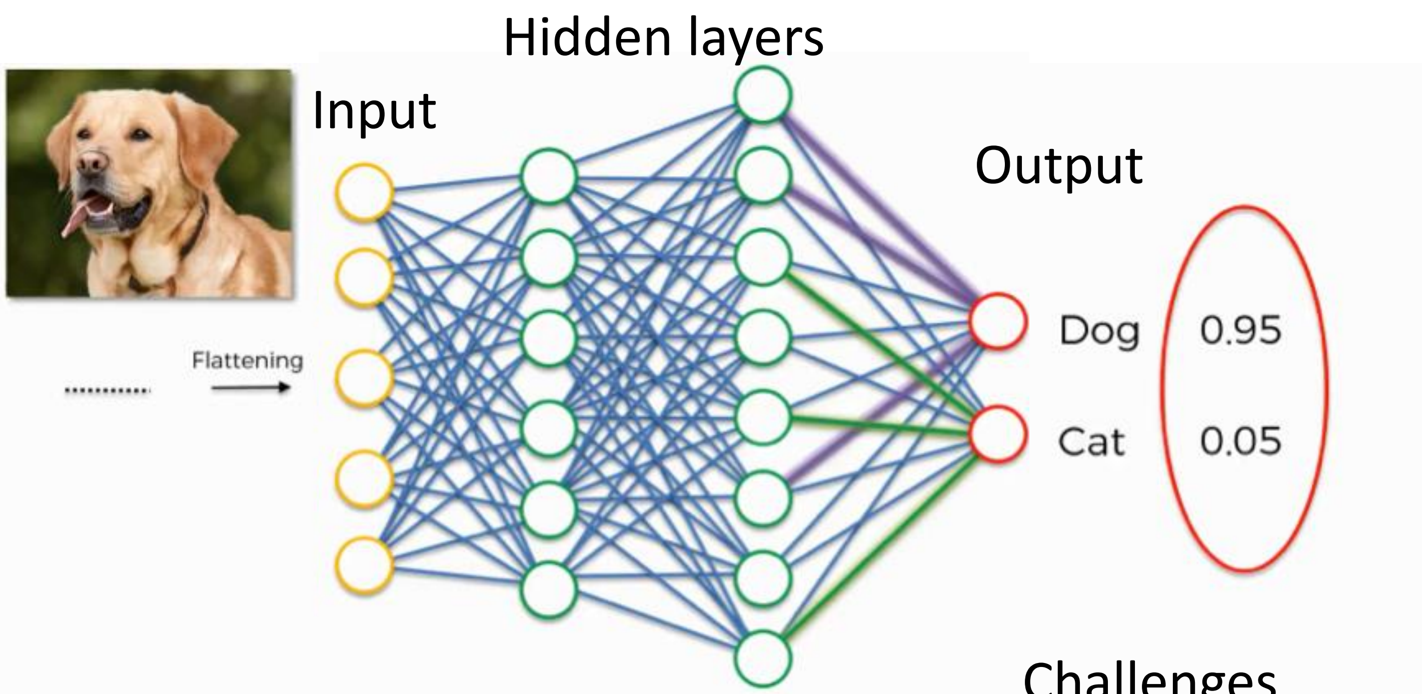
Depends on the dataset!

	Logistic Regression	SVM	Random Forest
Decision Boundary	Linear	Non-linear (w/ kernel)	Non-linear
Provides probability of class	Yes	No; but there are ways of estimating	No; but there are ways of estimating
Interpretability	Yes	Yes	Lesser than decision trees and other methods
Handles large dimensionality	No	Yes	Yes
Handles large number of samples	Yes	Slow for >10k samples	Yes
Handles categorical features	Yes if few	No	Yes
Features with different scales	Yes	No ("distance" may not be meaningful)	Yes
Handles missing data	No	No	Yes

Visual resource: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Neural Networks

Learns non-linear decision boundary by combining input data non-linearly



Challenges

- Large amounts of training data
- Training is computationally heavy
- Low interpretability

Advantages

- Non-linear decision boundary
- Learns features from the data

Image source: https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-softmax-crossentropy

Questions?