
# Big Data Visual Analytics (CS 661)

Instructor: Soumya Dutta

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur (IITK)

email: soumyad@cse.iitk.ac.in

# Study Materials for Lecture 11

- Visualizing High-Dimensional Data: Advances in the Past Decade; S. Liu et al., TVCG2016
- t-SNE: https://distill.pub/2016/misread-tsne/
- UMAP: https://pair-code.github.io/understanding-umap/

# Acknowledgements


- Some of the following slides are adapted from the excellent course materials and tutorials made available by:
  - Prof. Klaus Mueller (State University of New York at Stony Brook)
  - Prof. Tamara Munzner (University of British Columbia)
  - Zaur Fataliyev  (Research Scientist – Meta)
  - Andreas Kollegger (neo4j)

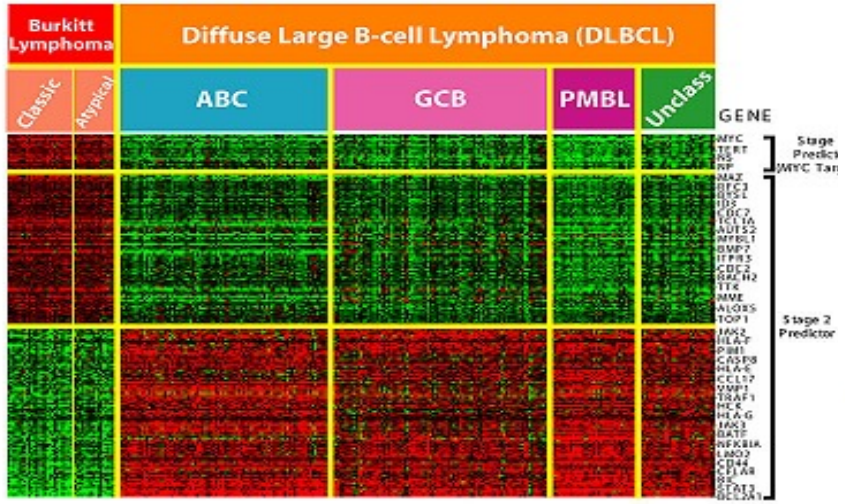
# High Dimensional Data

- In statistics, high dimensional data is a data where the number of attributes (features) are larger than the number of samples
- In practice, often, when a data set has large number of attributes, it is also referred to as high dimensional data
  - Examples: biological data, gene expression data, social media user data, network data, etc.

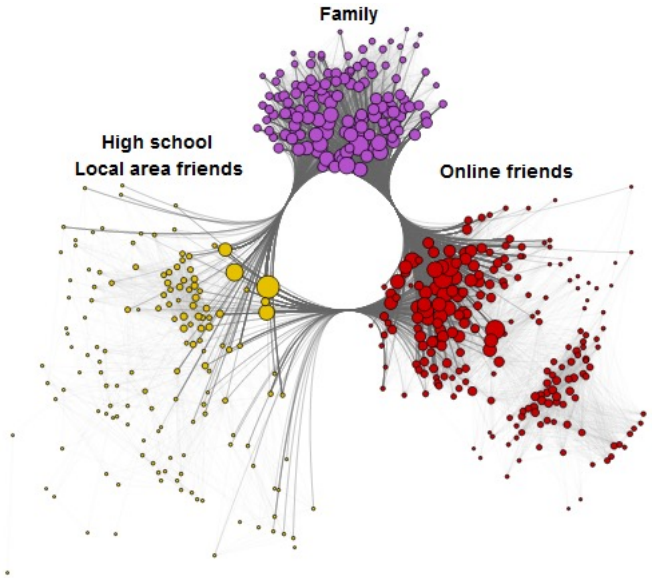
| Dataset | Features | GP | DT | NB | KNNs | SVMs | RF |
|---|---|---|---|---|---|---|---|
| Adenocarcinomas(58) | 50 | 99.83±0.009 | 73.33±1.6 | 87.66±3 | 86±1.3 | 91.33±0.7 | 77.67±1.5 |
| | 100 | 98.46±0.08 | 76.33±3.3 | 90±3.3 | 93.3±1.0 | 91.67±1.2 | 89.67 ± 1.5 |
| | 150 | 97.14±0.9 | 86±2.7 | 90±3.3 | 91.67±1.3 | 95±1.6 | 89.67±1.5 |
| Oral Mucosa(79) | 50 | 99.95±0.002 | 79.82±1.8 | 74.82±3.44 | 57.14±6.6 | 81±1.4 | 78.3±2.2 |
| | 100 | 98.57±0.08 | 70.89±0.16 | 69.46±3.8 | 62.14±2.1 | 77.32±2.6 | 81.25±5.9 |
| | 150 | 96.93±0.17 | 68.39±0.95 | 65.71±2.7 | 64.64±1.3 | 81.75±1.4 | 78.75±6.7 |
| B-Cells(79) | 50 | 99.41±0.03 | 77.32±2.6 | 74.82±3.44 | 82.32±2.2 | 85±4.7 | 82±5.5 |
| | 100 | 97.28±0.15 | 72.32±4.2 | 76.07 ±3.0 | 77.57±2.2 | 88.75±3.5 | 83.75±5.1 |
| | 150 | 96.59±0.19 | 71.25±9 | 78.57±2.2 | 76.25±6.7 | 87.5±3.9 | 83.75±5.1 |
| Placenta(76) | 50 | 99.91±0.005 | 77.49±2.5 | 68.75±9.8 | 70.71±4.7 | 84.28 ± 0.45 | 84.28±0.45 |
| | 100 | 99.30±0.03 | 73.39±5.1 | 67.5±10.7 | 69.46±5.1 | 84.28±0.45 | 84.28±0.45 |
| | 150 | 97.95±0.11 | 70.71±4.2 | 68.75±9.8 | 69.46±5.1 | 84.28±0.45 | 81.6±1.2 |
| Melanoma(83) | 50 | 97.64±0.13 | 88.19±4.1 | 94.16±1.8 | 95.13±2.4 | 91.67±1.3 | 96.52±1.0 |
| | 100 | 97.06±0.16 | 84.3±2.9 | 92.77±1.67 | 95.13±1.53 | 96.3±2.8 | 97.77±0.7 |
| | 150 | 96.37±0.51 | 85.5±3.3 | 95.2±1.4 | 96.3±1.14 | 97.63±0.7 | 97.77±0.7 |
| Breast cancer(97) | 50 | 97.87±0.12 | 52.77±0.8 | 49.44±1.9 | 43.22±0.3 | 54.66±0.2 | 56±0.14 |
| | 100 | 96.75±0.18 | 52.55±2.5 | 49.33±1.9 | 47.22±11 | 51.55±1.2 | 55.88±0.1 |
| | 150 | 96.90±0.17 | 51.67±2.2 | 51.44±1.3 | 52.1±2.4 | 52.55±0.9 | 57±0.45 |
| Skeletal Muscle(110) | 50 | 99.24±0.04 | 65.45±2.2 | 69.09±7.4 | 74.54±0.57 | 89.09±2.2 | 83.63±0.5 |
| | 100 | 98.69±0.07 | 71.81±5.4 | 66.36±2.0 | 83.63±5.1 | 98.18±0.57 | 82.72±0.28 |
| | 150 | 98.27 ±0.09 | 63.36±0.8 | 68.18±1.43 | 85.45±1.72 | 97.27±2.0 | 81.81±2.8 |
| Osteoarthritis(139) | 50 | 99.90±0.005 | 71.97±1.5 | 72.74±3.7 | 78.4±0.46 | 86.97±3.1 | 78.46±1.9 |
| | 100 | 99.23±0.04 | 70±9.4 | 69.12±2.4 | 84.23±2.5 | 90.65±0.52 | 81.31±1.04 |
| | 150 | 98.73±0.07 | 82.03±0.8 | 67.69±2.9 | 83.51±2.7 | 94.23±0.68 | 77.74±2.17 |

https://doi.org/10.1371/journal.pone.0196385.t004

Tabular data



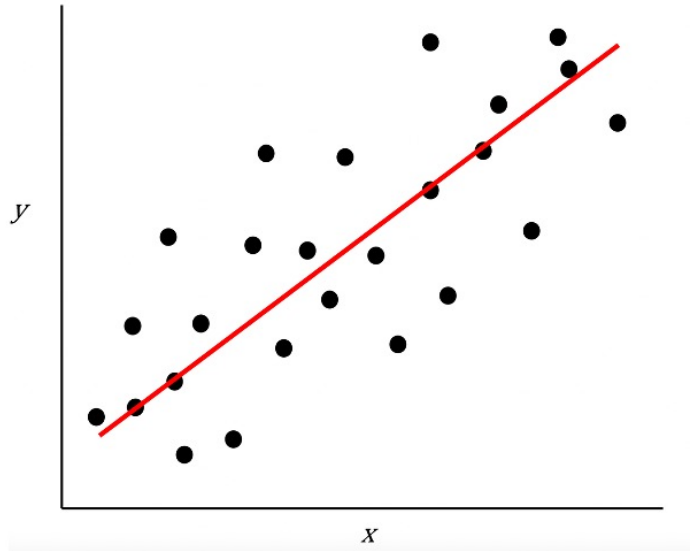

Microarray data

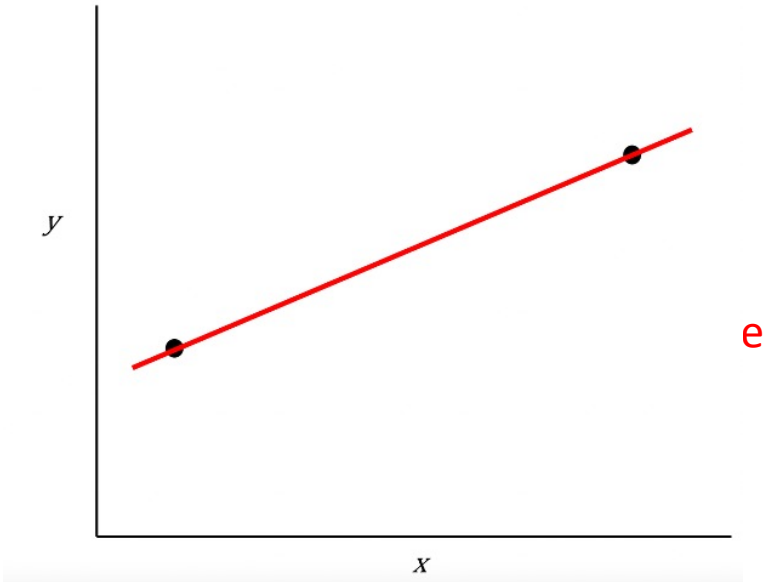



Graph data

# Intuition: Why High-Dimensional Data Can Be A Problem?

- Imagine a situation in which the number of observations and features in a dataset are almost equal
- Effective number of observations per features is low
- Result: Models (Statistical or ML) can overfit and so less generalizable

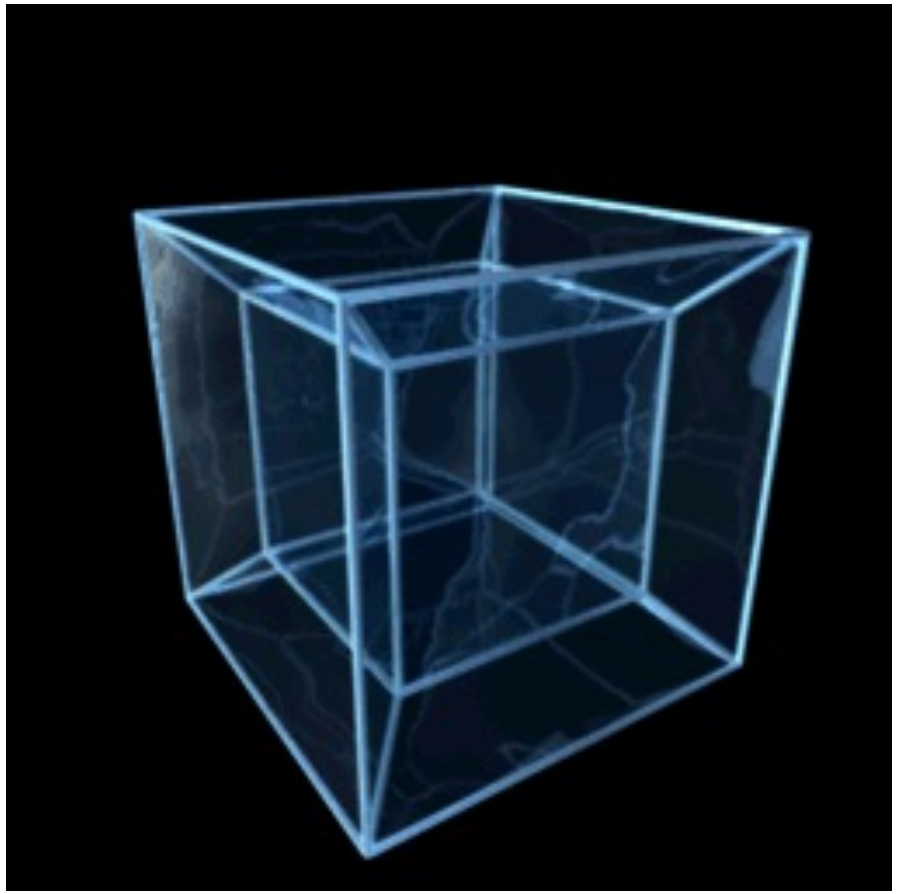



High dim. data, erfitting in regression

# Understanding High-Dimensional Objects

- Feature vectors are typically high dimensional
- We do not understand such vectors well – why?
- Because we don’t learn to see high-dim objects when our vision system develops
- We only perceive 3D world!



3D projection of a 4D cube

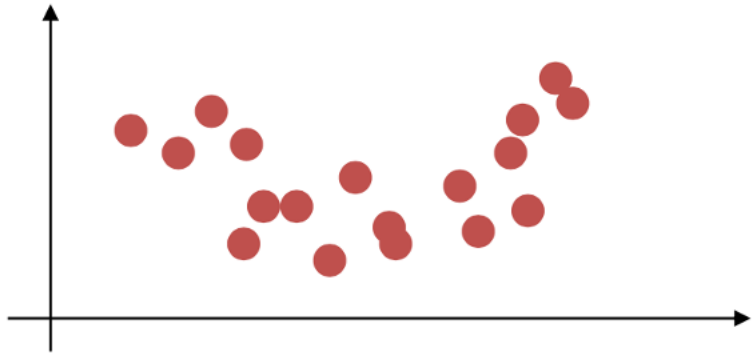
# Curse of Dimensionality

- A phenomenon related with high dimensional data
  - Challenging to identify meaningful patterns while analyzing and visualizing the data
- With increasing dimensionality, the volume of the space increases rapidly, making the data sparse in high dimension
- To obtain a reliable result, the amount of data needed often grows exponentially with the dimensionality
- Distance computation between objects in high dimensional space becomes difficult

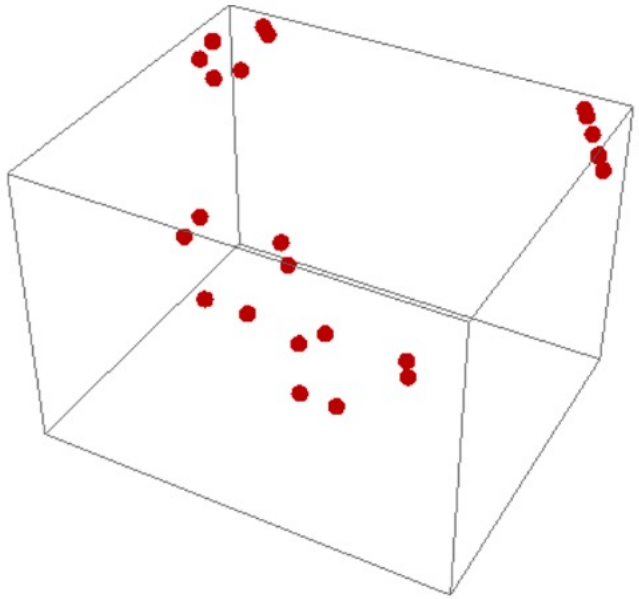
# Sparseness in High Dimensional Space

- Space gets extremely sparse
  - with every extra dimension points get pulled apart further
  - distances become meaningless

1D – points are very close



2D – points spread apart



3D – getting even sparser

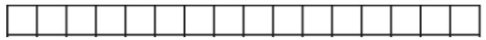
4D, 5D, … – sparseness grows further

# Space and Memory Management

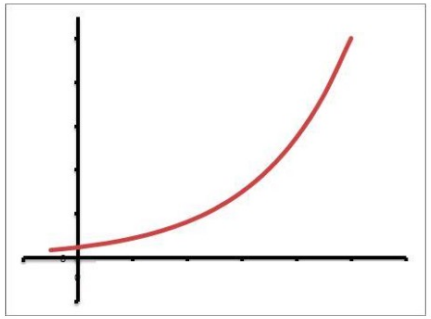
- Indexing (and storage) gets very expensive with increasing dimensionality
  - Exponential growth in the number of dimensions

# Space and Memory Management

- Indexing (and storage) gets very expensive with increasing dimensionality
  - Exponential growth in the number of dimensions

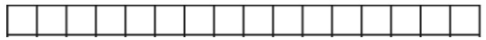


16 cells

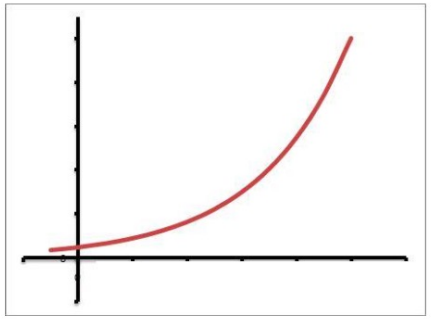


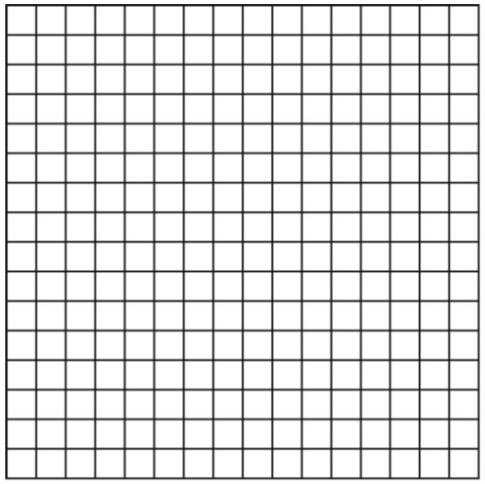
# Space and Memory Management

- Indexing (and storage) gets very expensive with increasing dimensionality
  - Exponential growth in the number of dimensions



16 cells

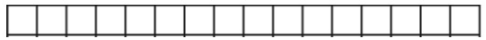




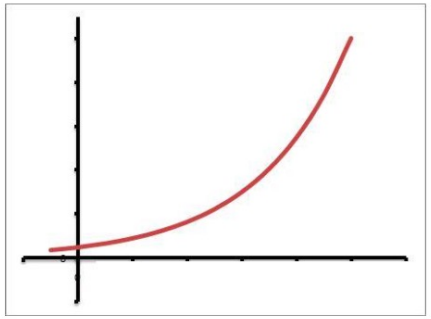
$16^2 = 256$ cells

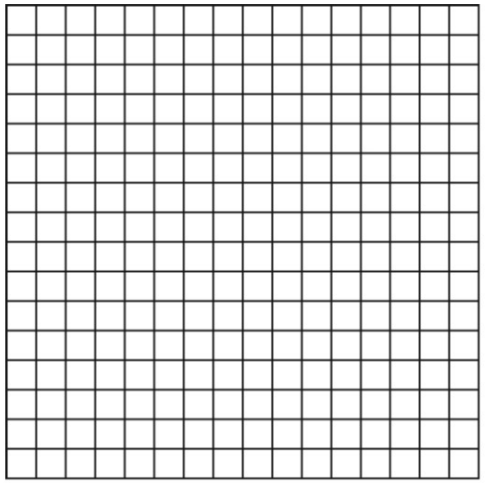
# Space and Memory Management

- Indexing (and storage) gets very expensive with increasing dimensionality
  - Exponential growth in the number of dimensions

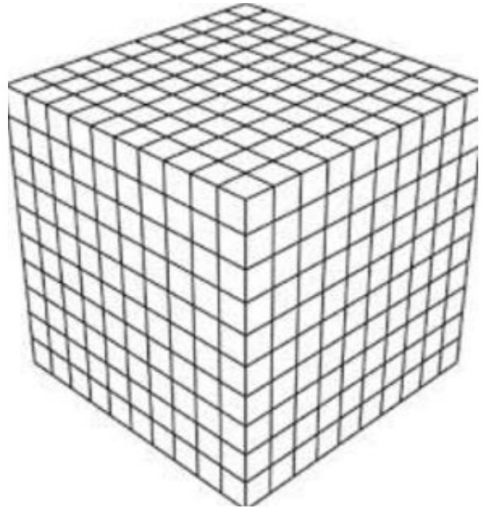


16 cells





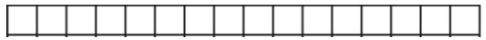
$16^2 = 256$ cells



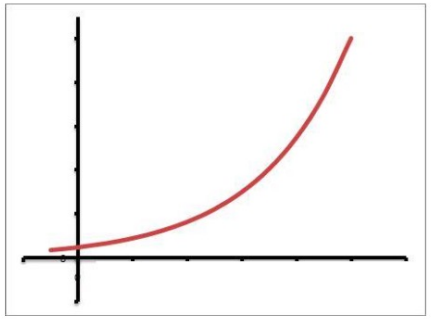
$16^3 = 4{,}096$ cells

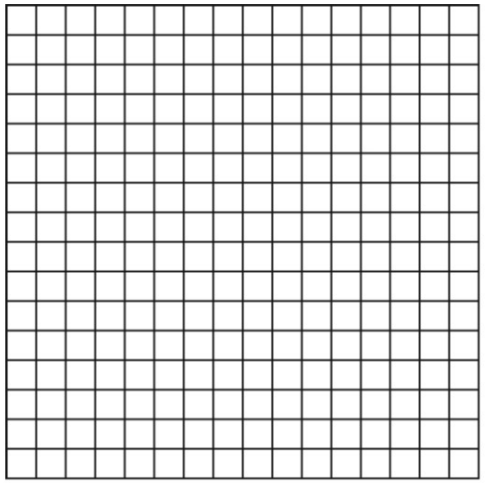
# Space and Memory Management

- Indexing (and storage) gets very expensive with increasing dimensionality
  - Exponential growth in the number of dimensions

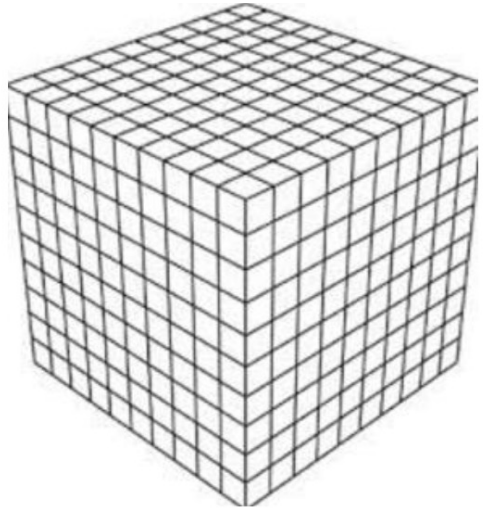


16 cells





$16^2 = 256$ cells

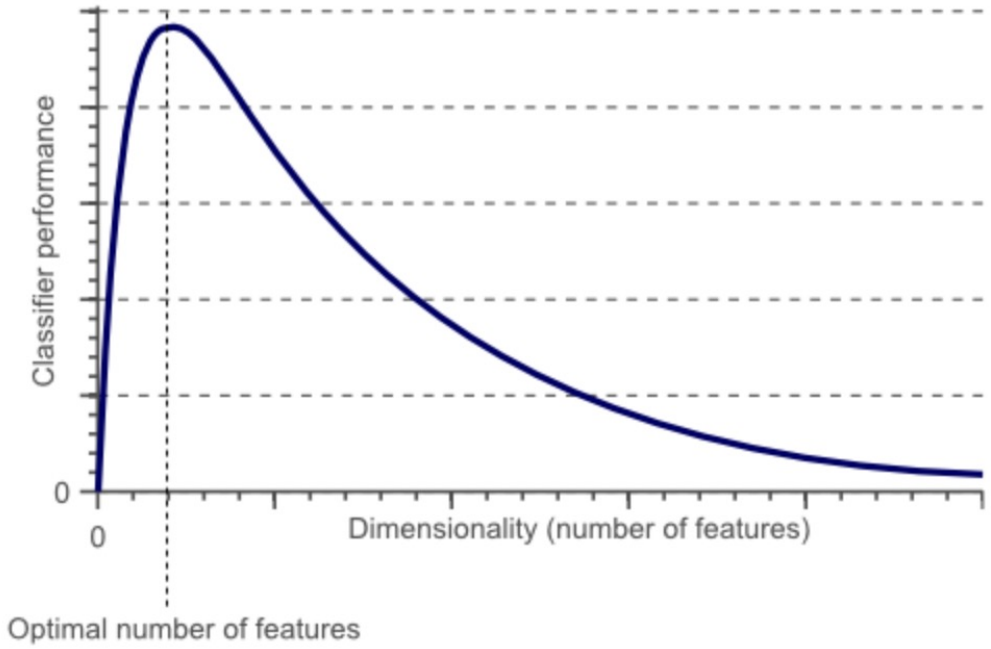


$16^3 = 4{,}096$ cells

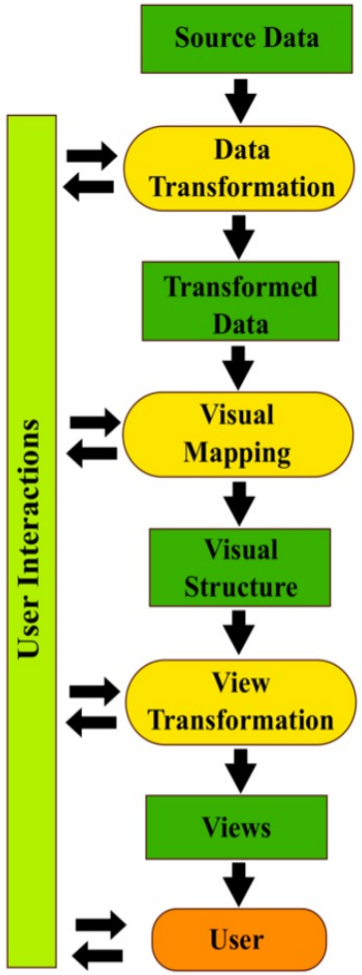
- 4D: 65k cells 5D: 1M cells 6D: 16M cells 7D: 268M cells

# High Dimension: In Machine Learning

- Hughes' Phenomenon: With a fixed number of training samples, the average (expected) predictive power of a classifier or regressor first increases as the number of dimensions or features used is increased but beyond a certain dimensionality it starts deteriorating instead of improving steadily



# High Dimensional Data Analysis and Visualization




*Data Transformation*

| Dimension Reduction | Subspace Clustering | Regression Analysis | Topological Data Analysis |
|---|---|---|---|
| linear projection [21], [22], | Dimension Space Exploration | Optimization | Morse-Smale Complex |
| non-linear DR [23], [24], | [2], [31], [32], | Design Steering | [42], [43], [44], [45] |
| Control Points Projection [25], [26], | Subset of Dimension [33], [34], | [38], [39], [40], | Reeb Graph [46], [47], [48] |
| Distance Metric [27], [28], | Non-Axis-Parallel Subspace | Structural Summaries | Contour Tree [49], [50], |
| Precision Measures [29], [30] | [35], [36], [37] | [5], [41] | Topological Features [51], [52] |

*Visual Mapping*

| Axis Based | Glyphs | Pixel-Oriented | Hierarchy Based | Animation | Evaluation |
|---|---|---|---|---|---|
| Scatterplot Matrix [53], | Per-Element Glyphs | Jigsaw Map [67], | Dimension | GGobi [76], | Scatterplot Guideline |
| Parallel Coordinate [54], | [60], [61] | Pixel Bar Charts [68], | Hierarchy [71], | TripAdvisor$^{ND}$ | [79], [80], |
| Radial Layout [55], | [62], [63], | Circle Segment [69], | Topology-based | [77], | PCPs Effectiveness [81], |
| Hybrid Construction | Multi-Object Glyphs | Value & Relation | Hierarchy [72], [73], | Rolling the | Radical Layout [82], |
| [56], [57], [58], [59] | [64], [65], [66] | Display [70] | Others [74], [75] | Dice [78] | Animation [83] |

*View Transformation*

| Illustrative Rendering | Continuous Visual Representation | Accurate Color Blending | Image Space Metrics |
|---|---|---|---|
| Illustrative PCP [84], | Continuous Scatterplot [89], [90], | Hue-Preserving | Clutter Reduction |
| Illuminated 3D Scatterplot [85], | Continuous Parallel Coordiante [91], [92], | Blending [95], | [97], [98], |
| PCP Transfer Function [86] | Splatterplots [93] | Weaving vs. Blending | Pargnostics [99], |
| Magic Lens [87], [88] | Splatting for PCPs [94] | [96] | Pixnostic [100] |

Visualizing High-Dimensional Data: Advances in the Past Decade

# Dimensionality Reduction: Why?

- Produce embedding of high dimensional data into low dimensional space
- Visual analysis of high dimensional data
- Useful for feature engineering in ML techniques
- Helps in finding redundant features from large scale data

# Dimensionality Reduction Techniques

## Linear methods

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

## Non-linear methods

- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection **(**UMAP)
- Multidimensional Scaling (MDS)
- ISOMAP
- Locally linear embedding (LLE)
- Laplacian Eigenmap (LE)

# Dimensionality Reduction Techniques

## Linear methods

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

## Non-linear methods

- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)
- Multidimensional Scaling (MDS)
- ISOMAP
- Locally linear embedding (LLE)
- Laplacian Eigenmap (LE)

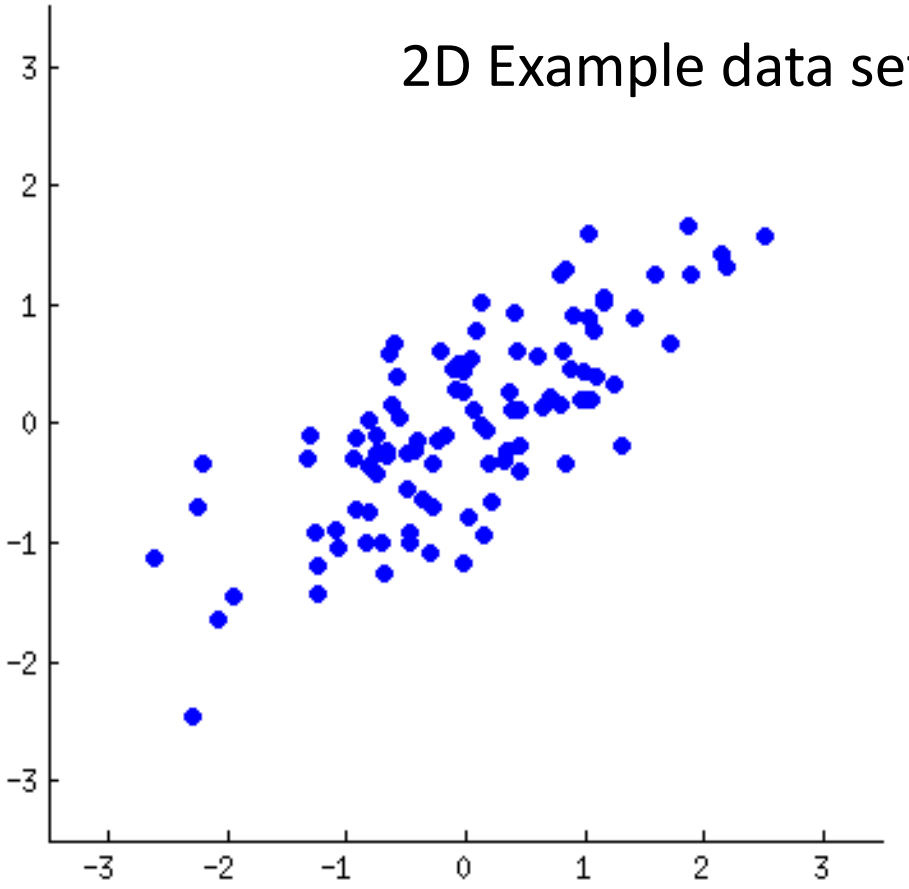
# Principal Component Analysis (PCA)

- Unsupervised technique for extracting variance structure from high dimensional data to represent data in a lower dimensional space
  - An orthogonal (linear) projection of the data into a subspace so that the variance of the projected data is maximized
- Useful for:
  - Visualization
  - Further processing by machine learning algorithms
  - More efficient use of resources (e.g., time, memory, space)
  - In general: fewer dimensions → better generalization and modeling

# Principal Component Analysis

- A linear transformation that chooses a new coordinate system for the data such that greatest variance by any projection of the data comes to lie on the first axis (first principal component), the second greatest variance on the second axis, and so on….
- How to reduce data dimension with PCA?
  - Principal components are sorted in the order of their explained variance in the data
  - Eliminating later principal components

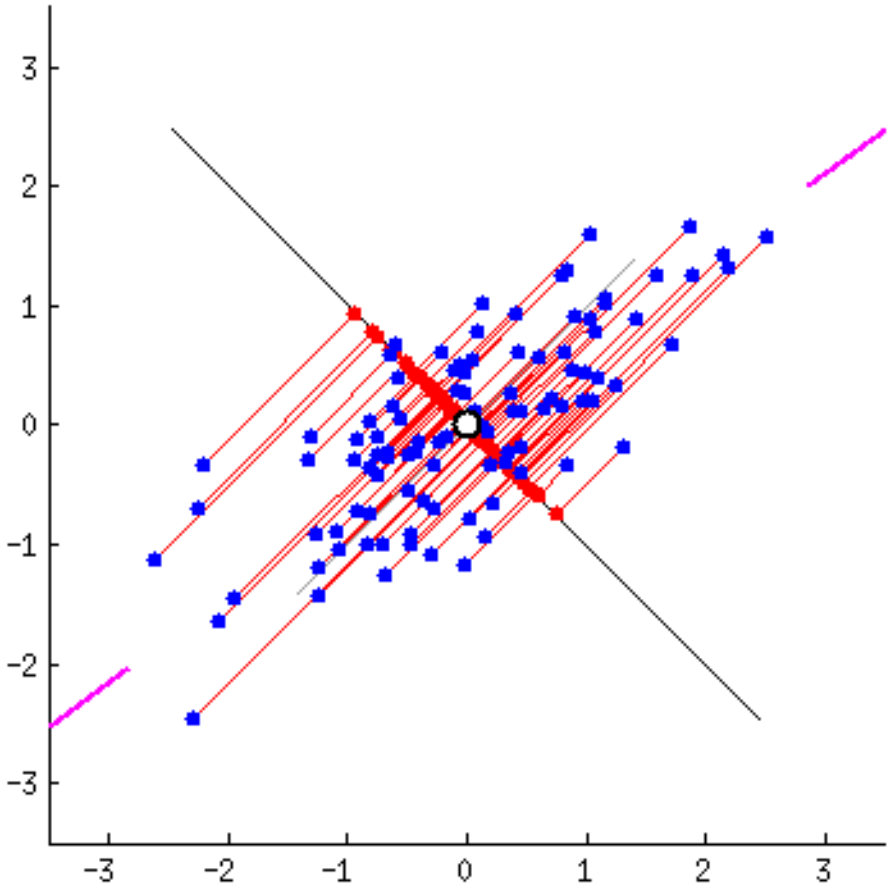
# Principal Component Analysis





https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579

# Principal Component Analysis




Animated view of how PCA works conceptually


https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579

# Steps of PCA Algorithm

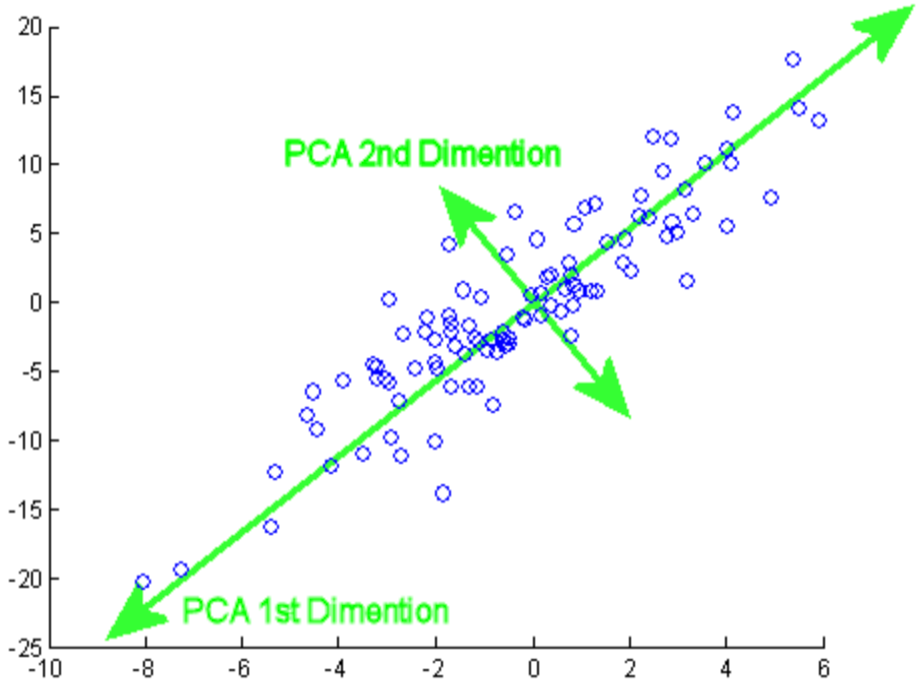
- Suppose we have a dataset with *n* records and *m* features for each record, i.e., data has *n* rows and *m* columns

1. Standardize the data: $x_{new} = \frac{x - \mu}{\sigma}$

2. Calculate the covariance matrix of the standardized data matrix
3. Calculate the eigen decomposition of the covariance matrix
   - Results in a list of eigenvalues and a list of eigenvectors
4. Sort eigenvalues in descending order to get a ranking for the eigenvectors (principal components) or axes of the new subspace

# How to Project Data into a Lower Dimension?

- A total of k components (k < m) can be selected to create a projection subspace.
- The k eigenvectors are called principal components, that have the k largest eigenvalues
- Data can be projected into the k-dimensional subspace via matrix multiplication



# PCA: Explained Variance

- Explained variance is a statistical measure of how much variation in a dataset can be attributed to each of the principal components
  - How much of the total variance is "explained" by each component
- Ordered (large to small) eigenvalues can help
- Explained variance for $i^{th}$ principal component:

$$\frac{\lambda_i}{\lambda_1+\lambda_2+\ldots+\lambda_n}$$

$\lambda_i$ : $i^{th}$ eigen value of covariance matrix

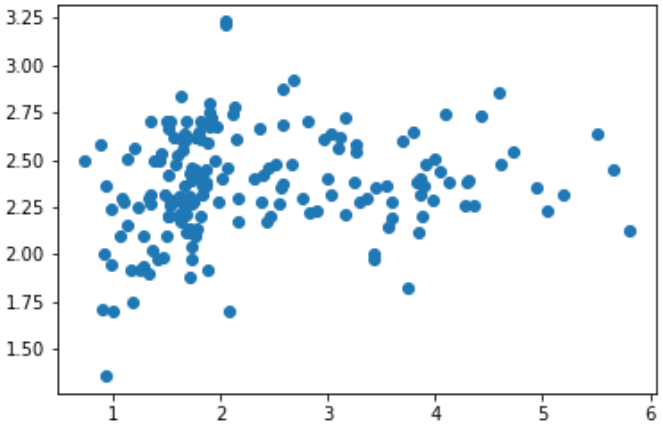
# Visualization of Data using PCA

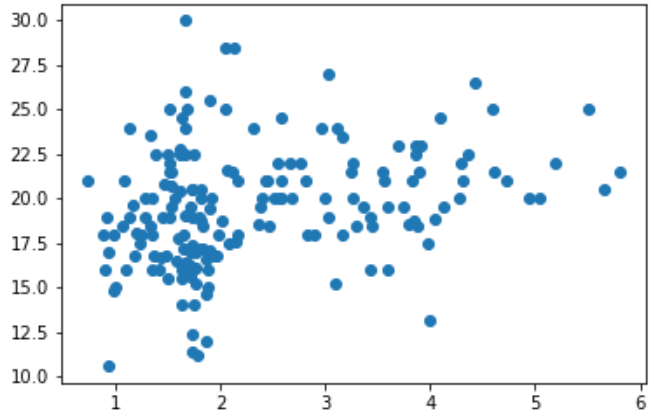
- Wine data set
  - 178 records, 13 features for each record

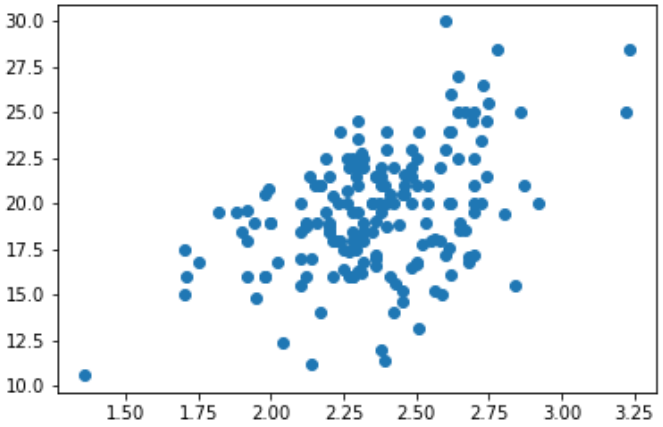
```
[[1.423e+01 1.710e+00 2.430e+00 ... 1.040e+00 3.920e+00 1.065e+03]
 [1.320e+01 1.780e+00 2.140e+00 ... 1.050e+00 3.400e+00 1.050e+03]
 [1.316e+01 2.360e+00 2.670e+00 ... 1.030e+00 3.170e+00 1.185e+03]
 ...
 [1.327e+01 4.280e+00 2.260e+00 ... 5.900e-01 1.560e+00 8.350e+02]
 [1.317e+01 2.590e+00 2.370e+00 ... 6.000e-01 1.620e+00 8.400e+02]
 [1.413e+01 4.100e+00 2.740e+00 ... 6.100e-01 1.600e+00 5.600e+02]]
```

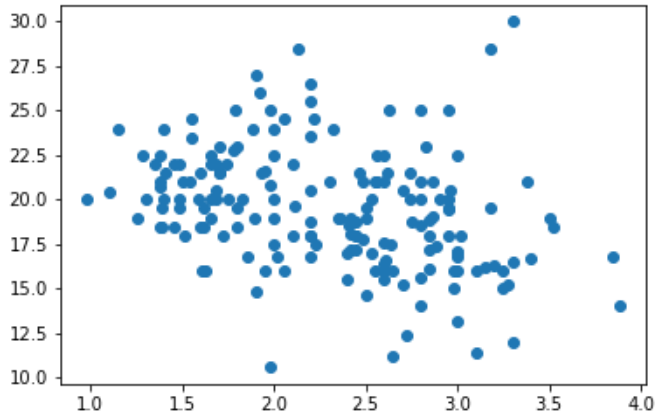
# Visualization of Data using PCA

- Scatter plot for selected pairwise features (attributes)



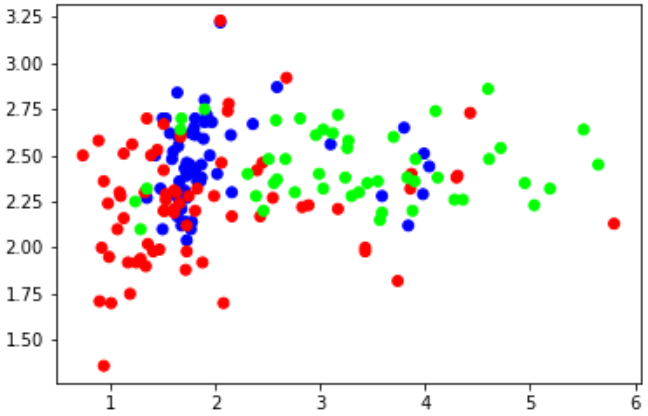



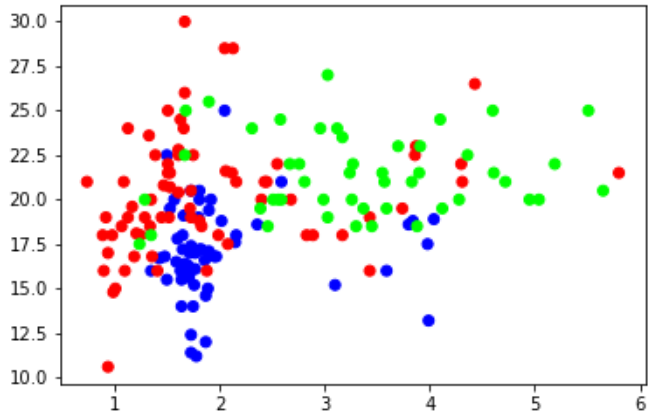



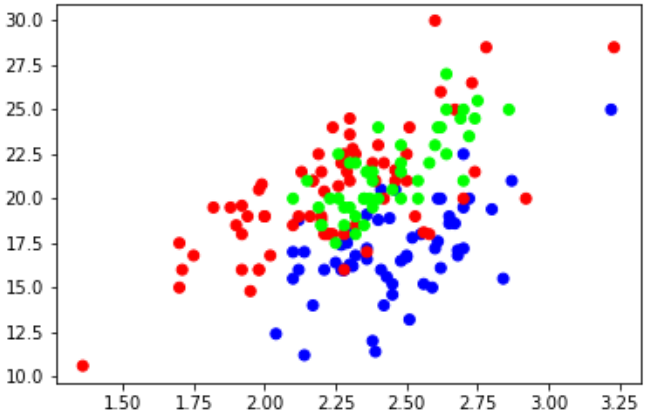



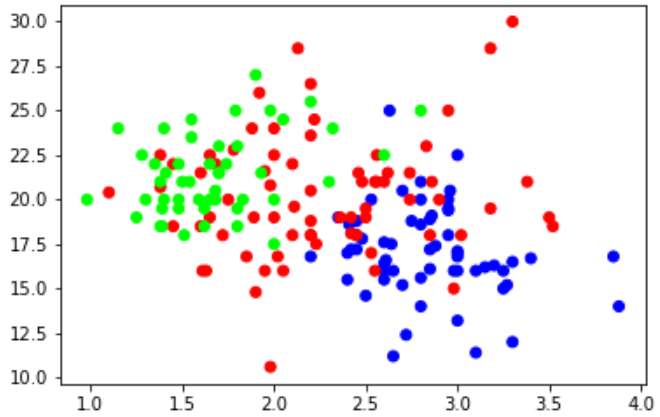
# Visualization of Data using PCA

- Scatter plot for selected pairwise features (attributes)
  - Color points by class label



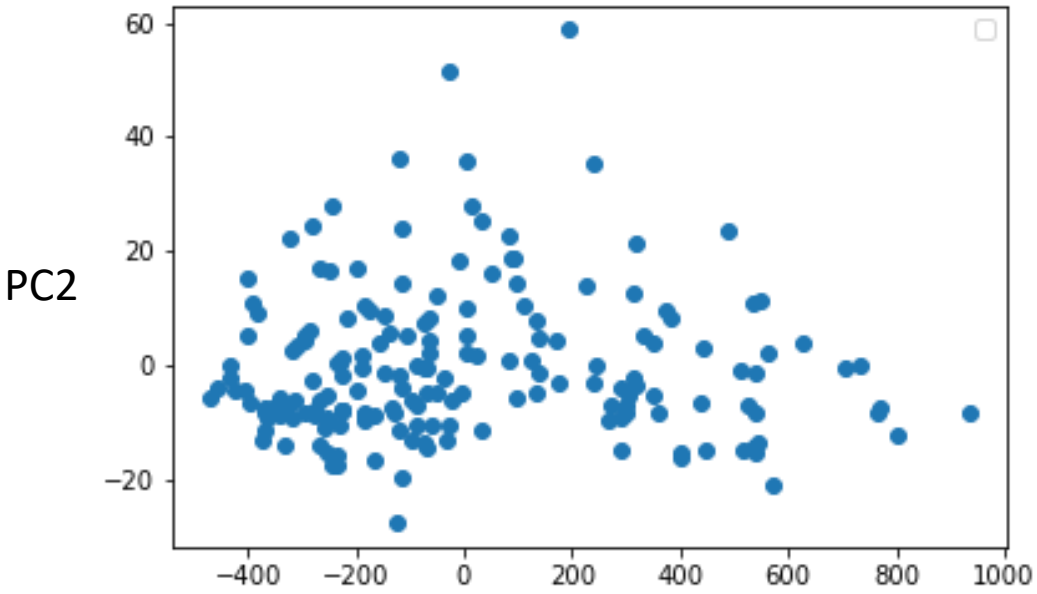



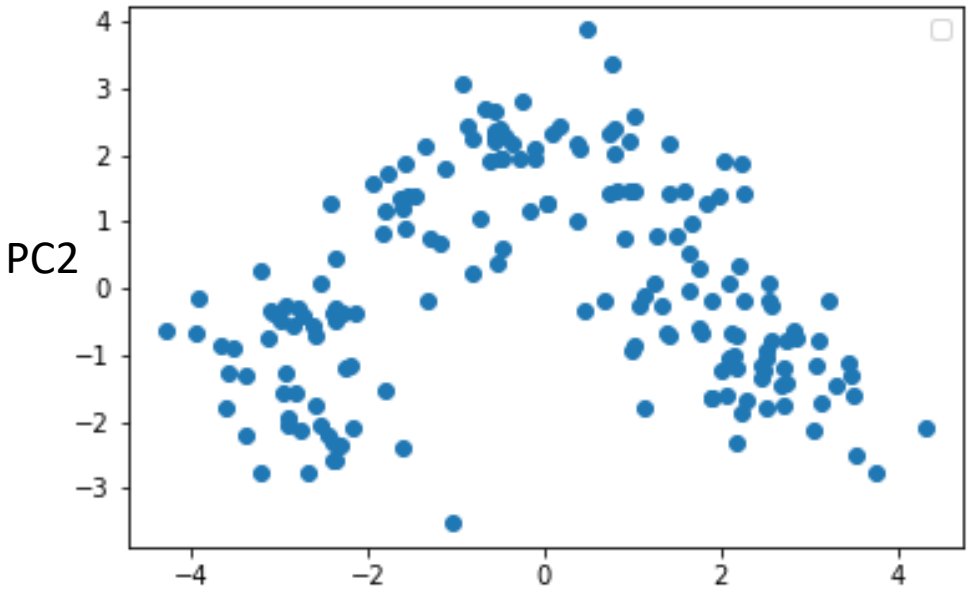



# Visualization of Data using PCA

- Scatter plot using two first principal components as axes after PCA is applied



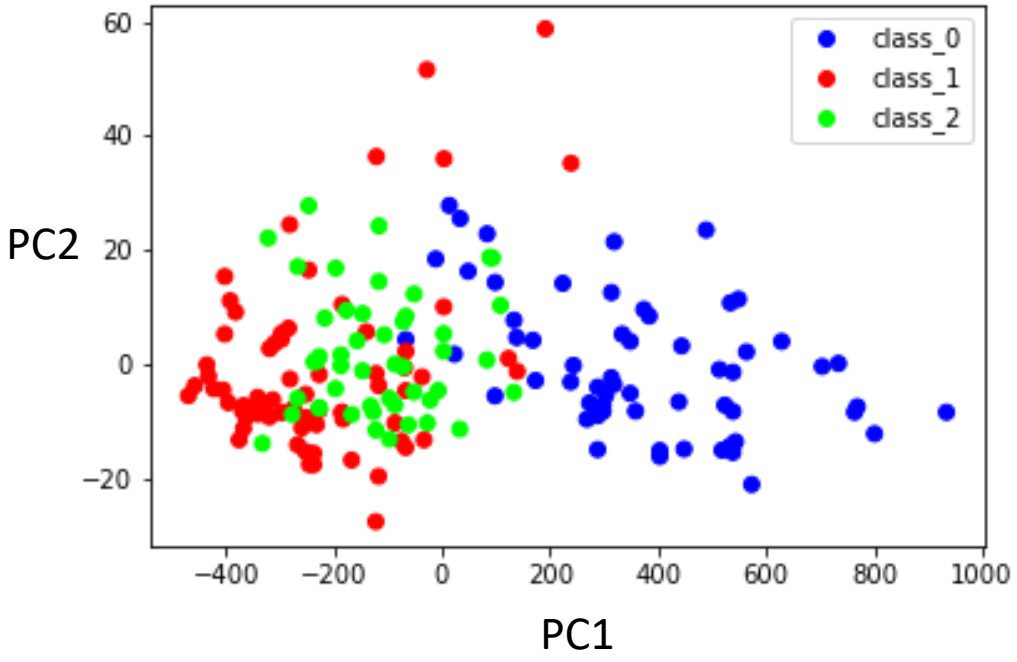

PCA with no standardization



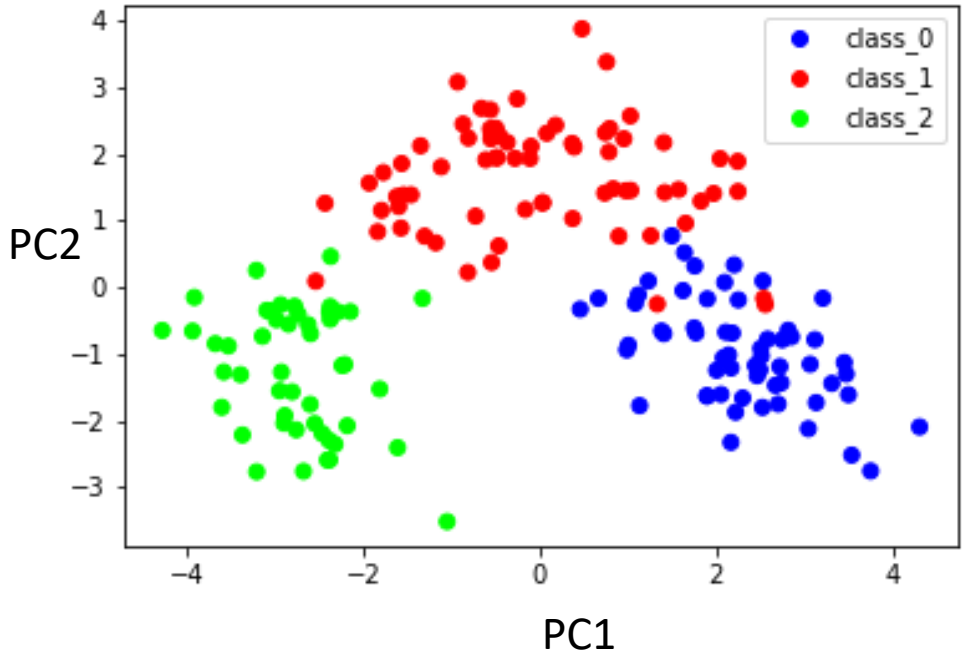

PCA with standardization

# Visualization of Data using PCA

- Scatter plot using two first principal components as axes after PCA is applied
  - Color points by class label



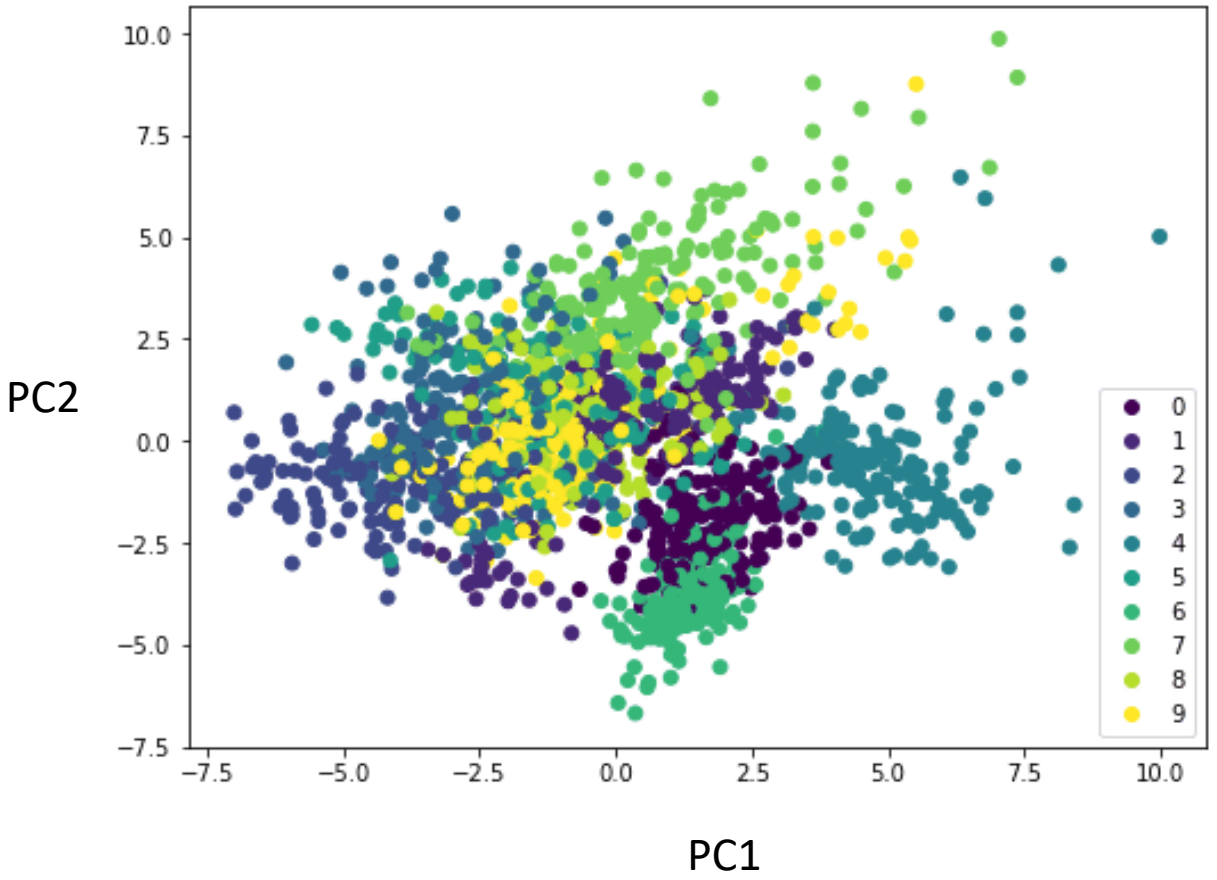

PCA with no standardization




PCA with standardization

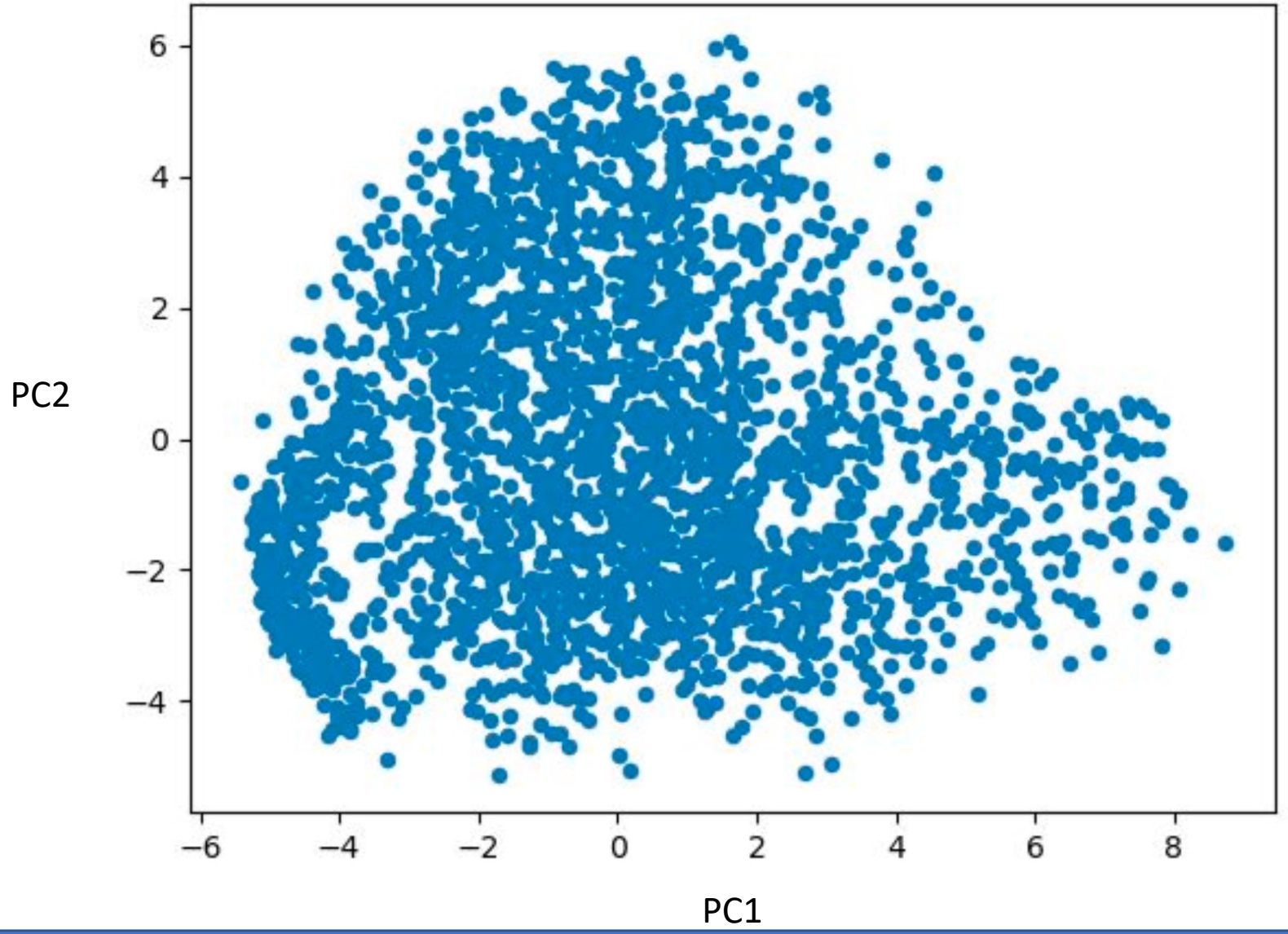
# Visualization of Data using PCA

- 2D projection plot of hand-written digit data set with 10 classes
  - Not easily separable

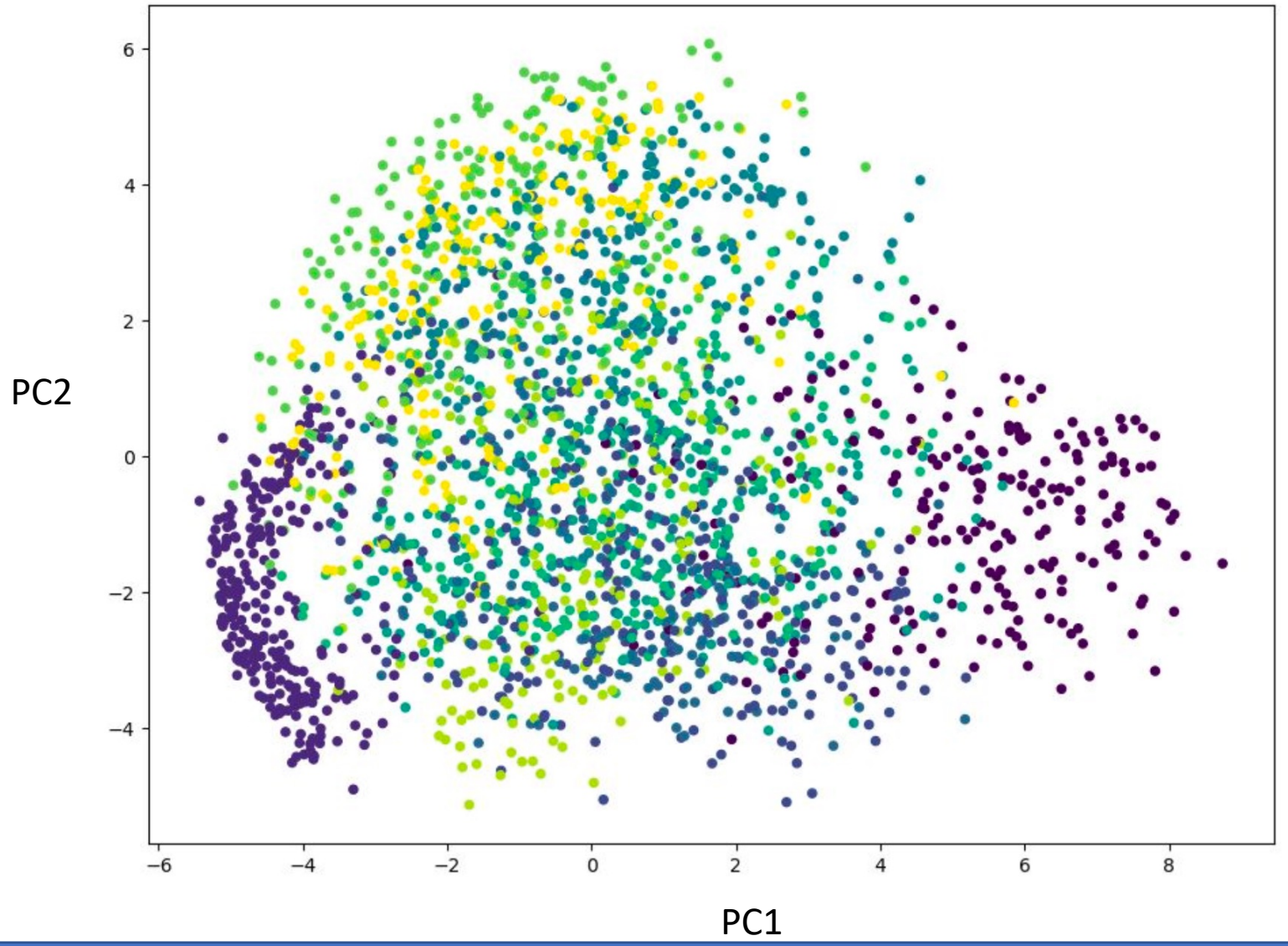


# PCA on MNIST data

# PCA on MNIST data



# PCA on MNIST data




PCA is not able to separate classes clearly