Dimensionality reduction: feature extraction

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON
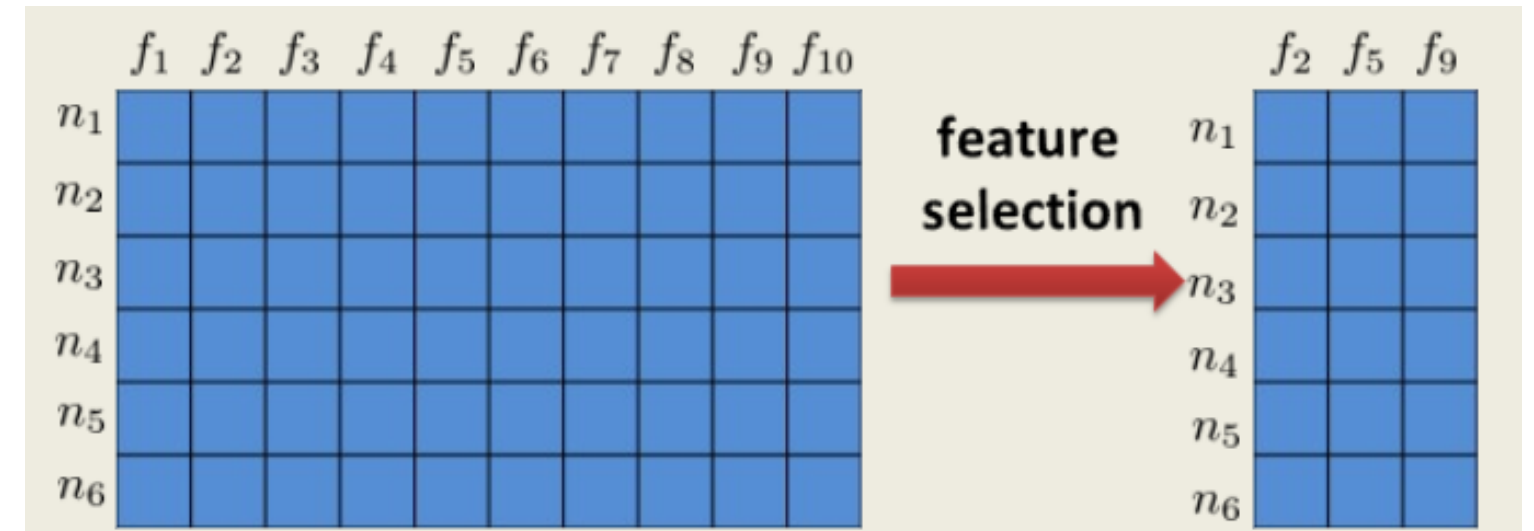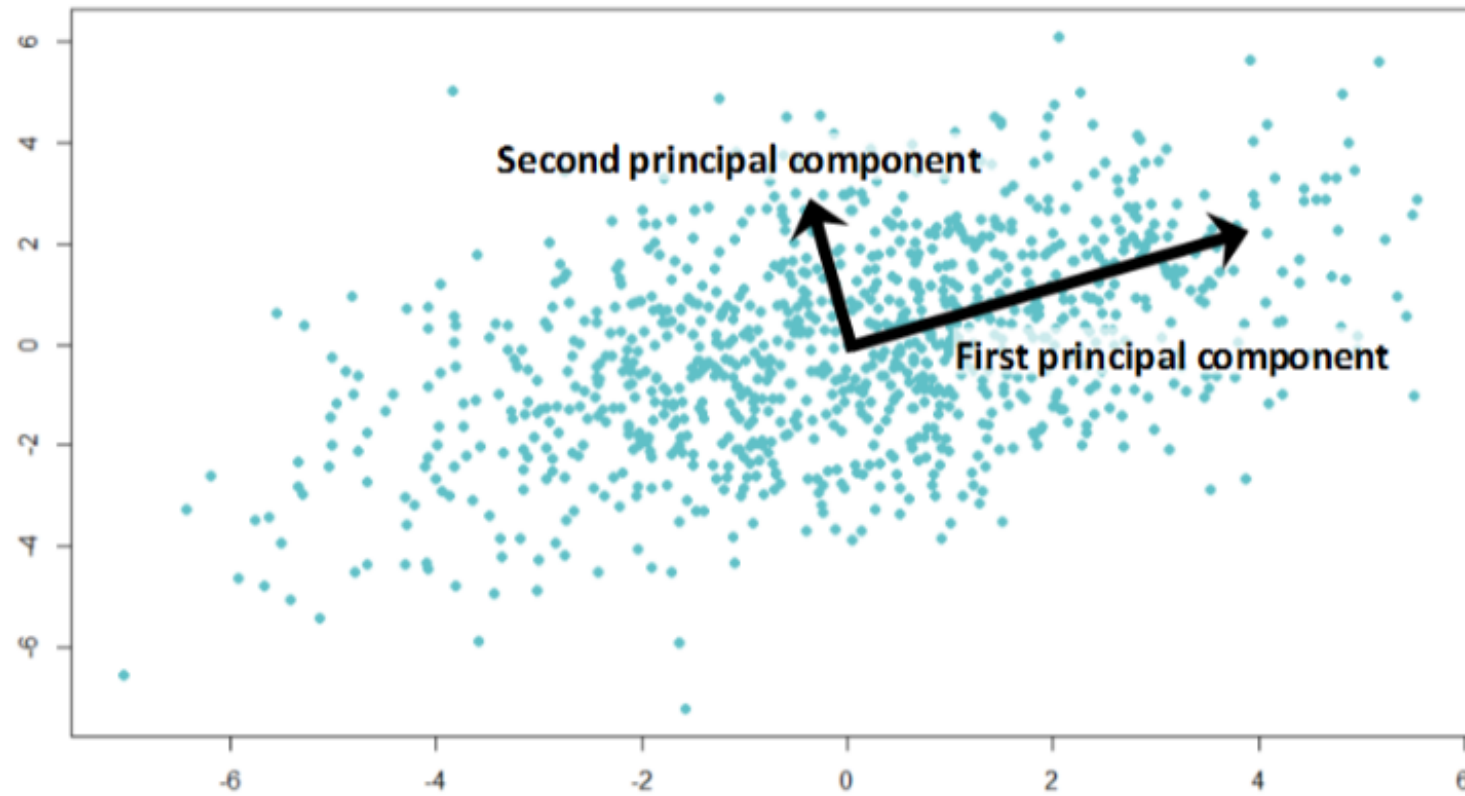
Lisa Stuart
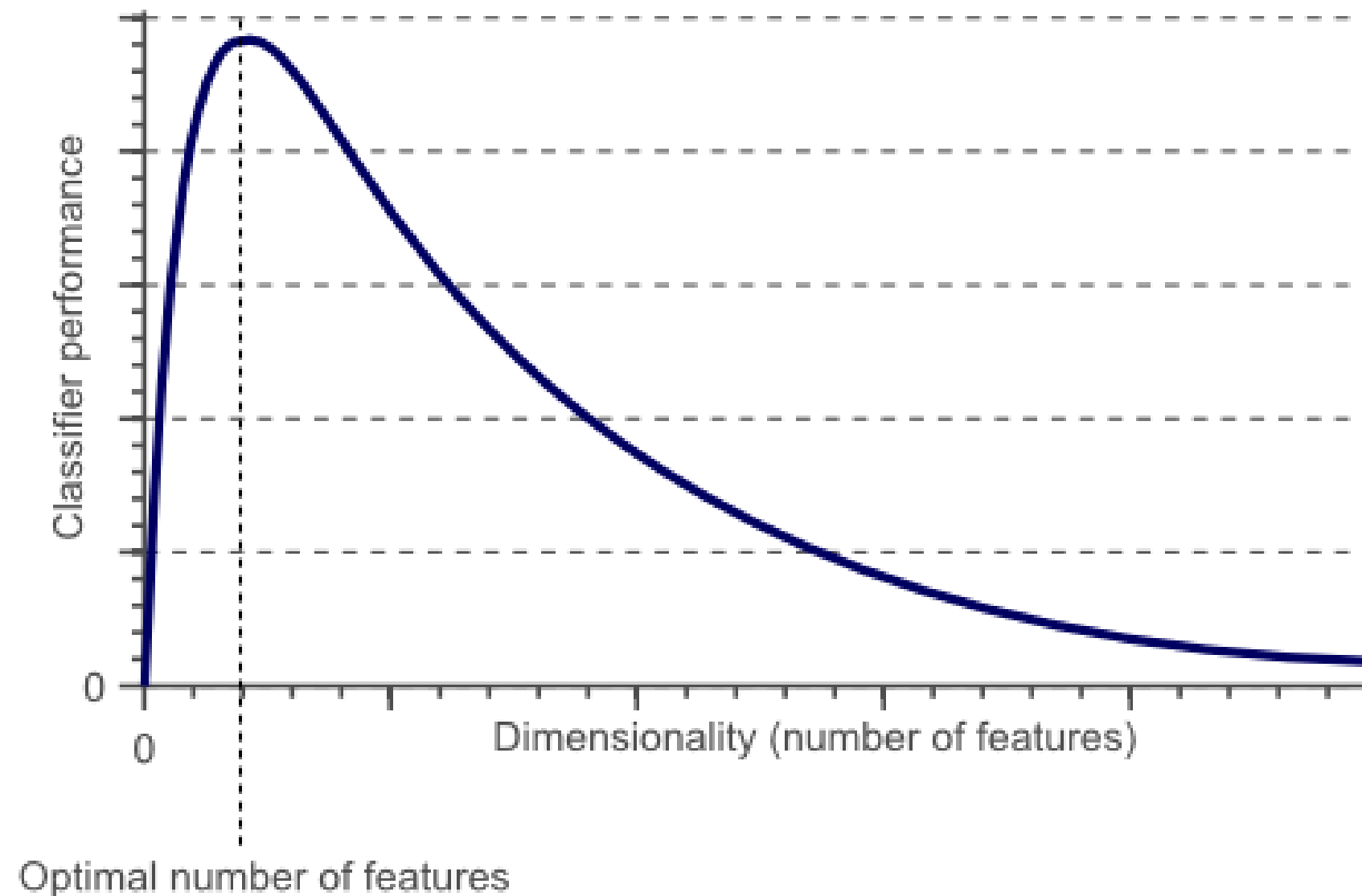Data Scientist

Unsupervised learning methods

- Principal component analysis (PCA) --> Lesson 3.1

- Singular value decomposition (SVD) --> Lesson 3.1

- Clustering/grouping --> Lesson 3.3

- Exploratory data mining

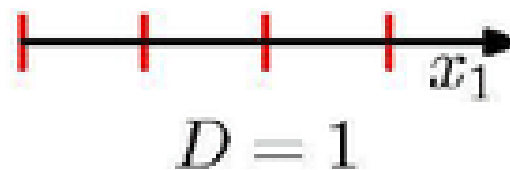# Dimensionality reduction != feature selection

[1] https://slideplayer.com/slide/9699240/ [2] https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/

Curse of dimensionality

1-D search

$D = 1$
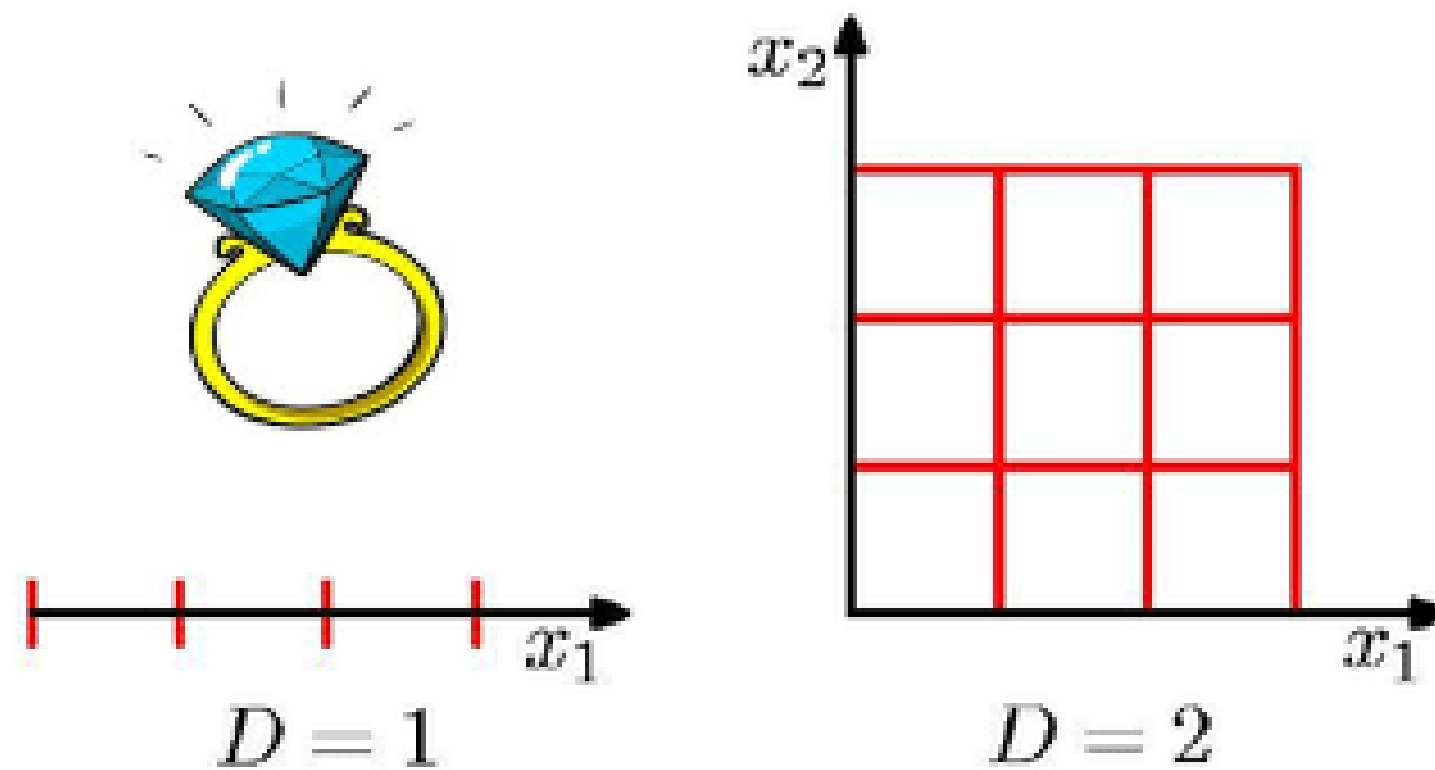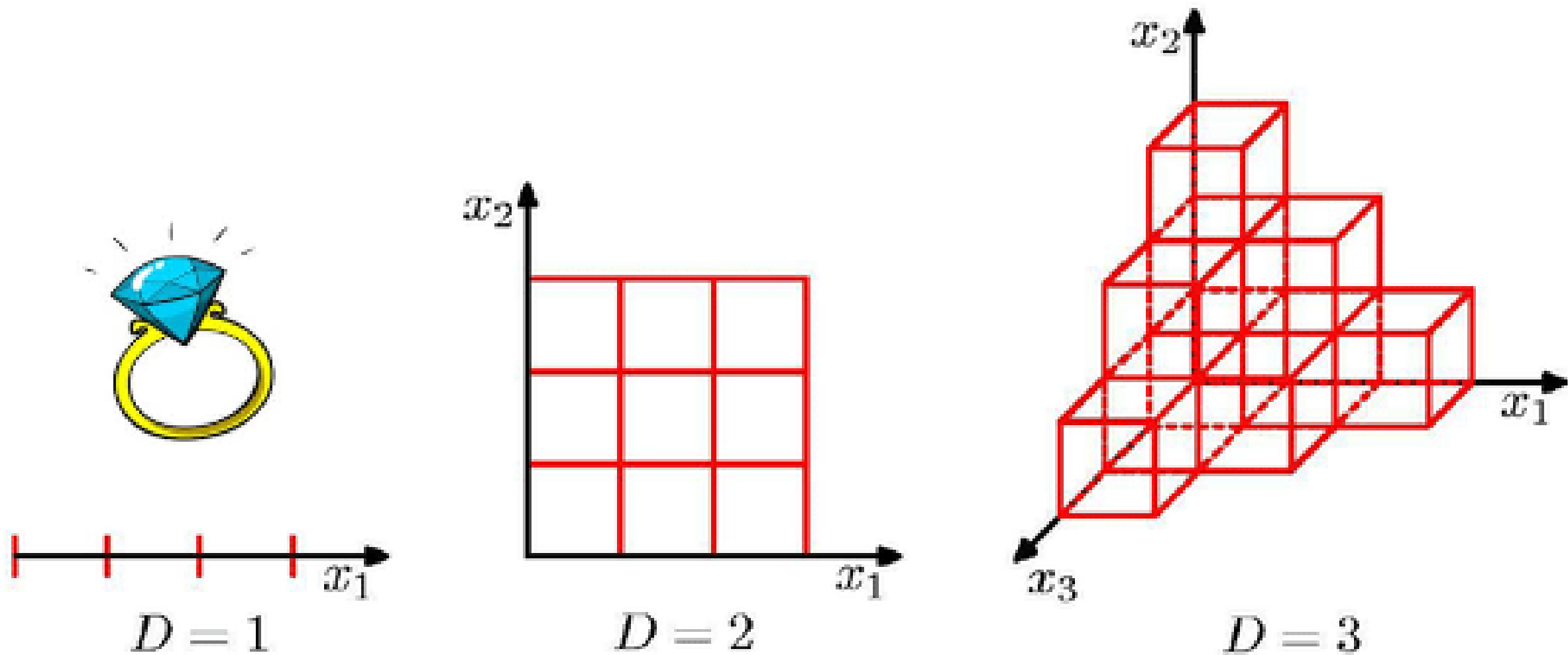
2-D search

$D = 1$

$D = 2$

3-D search
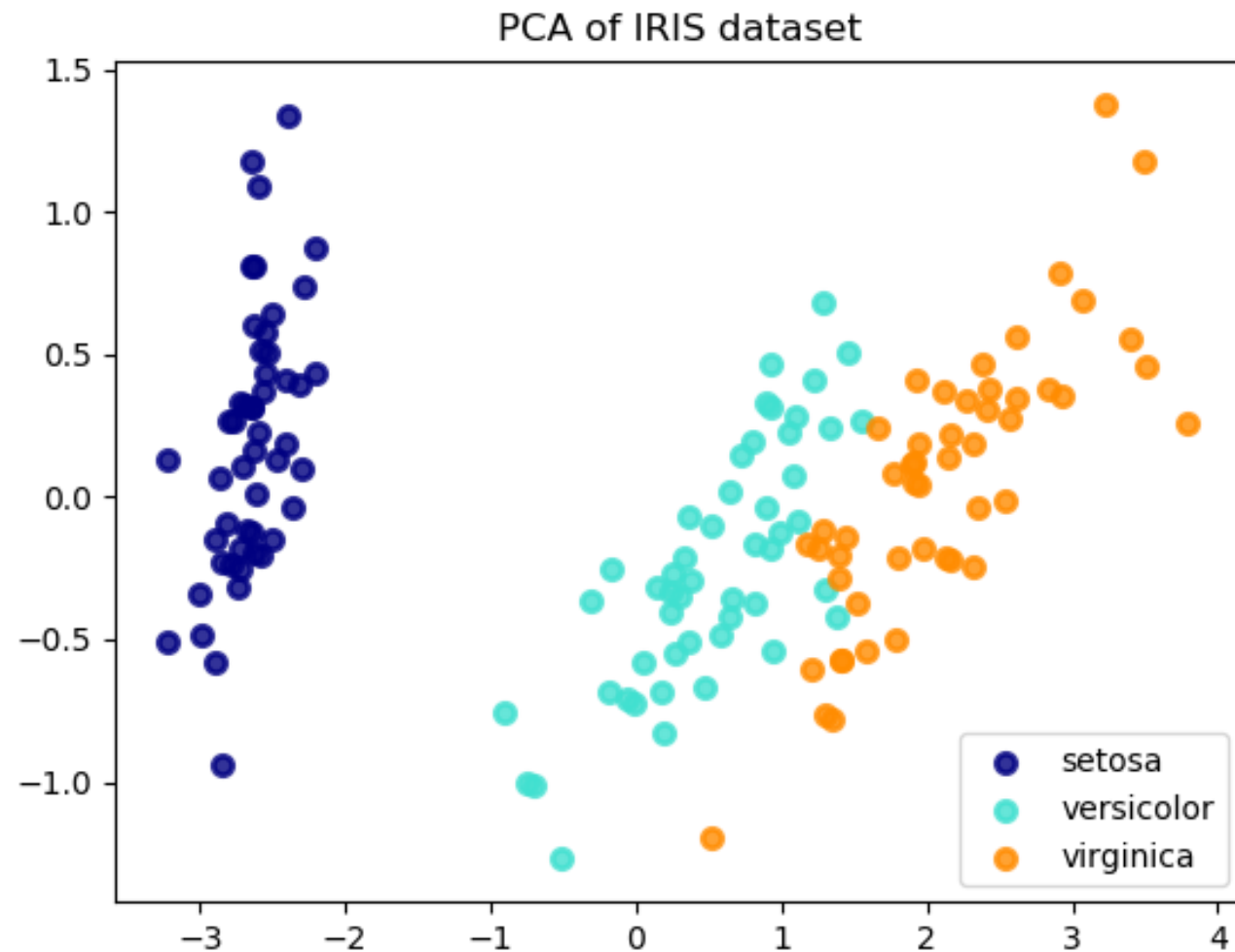

$D = 1$

$D = 2$

$D = 3$

Dimensionality reduction methods
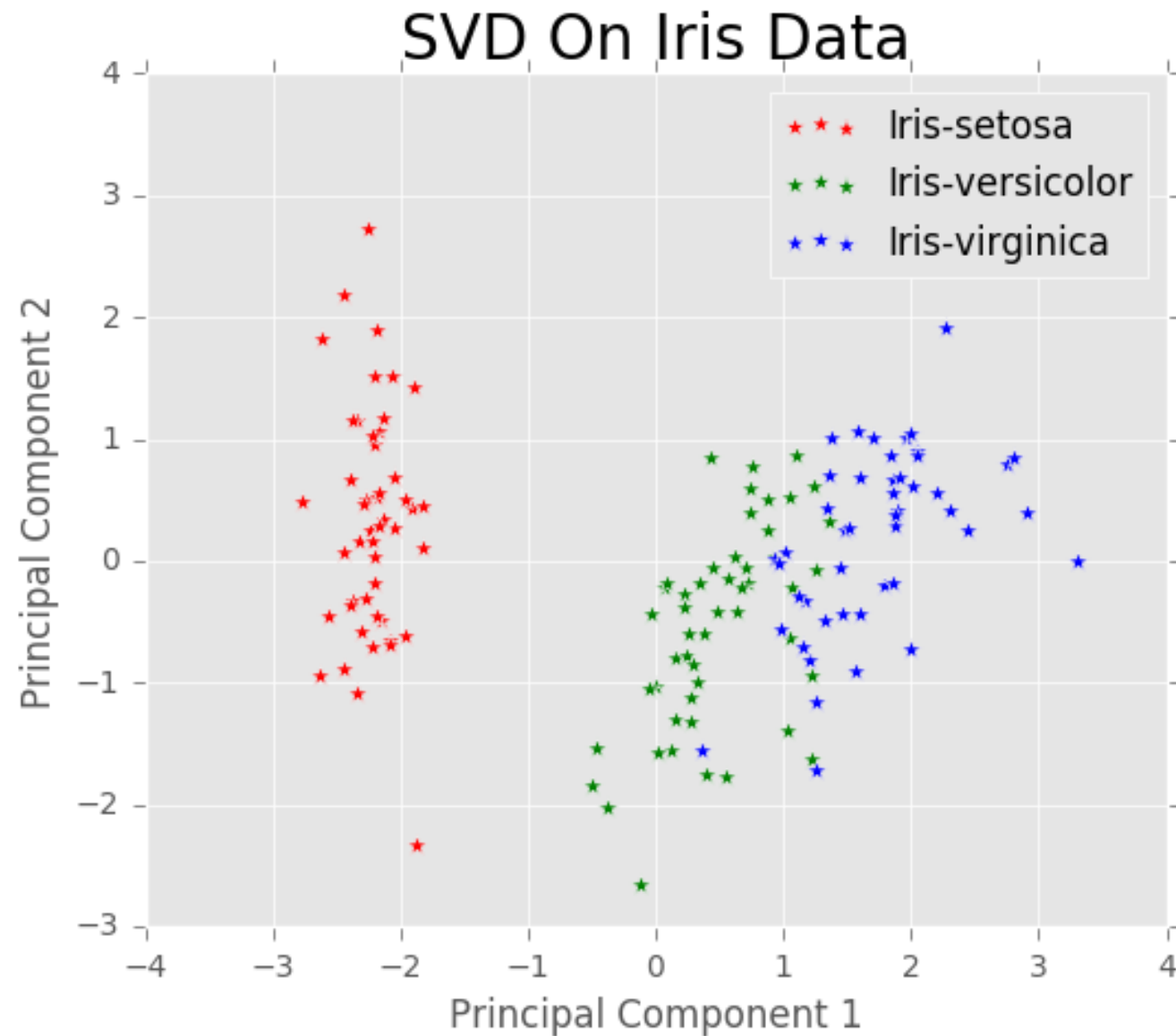
- PCA

- SVD

PCA

PCA of IRIS dataset

- PCA
  - Relationship between X and y
  - Calculated by finding principal axes
  - Translates, rotates and scales
  - Lower-dimensional projection of the data

[1] https://scikit-learn.org/stable/modules/decomposition.html

SVD

- SVD
  - Linear algebra and vector calculus
  - Decomposes data matrix into three matrices
  - Results in 'singular' values
  - Variance in data approximately equals SS of singular values

[1] https://galaxydatatech.com/2018/07/15/singular-value-decomposition/

Dimension reduction functions

| Function/method	returns
`sklearn.decomposition.PCA`	principal component analysis
`sklearn.decomposition.TruncatedSVD`	singular value decomposition
`PCA/SVD.fit_transform(X)`	fits and transforms data
`PCA/SVD.explained_variance_ratio_`	variance explained by PCs

- **Other matrix decomposition algorithms**

Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Dimensionality reduction: visualization techniques

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Lisa Stuart
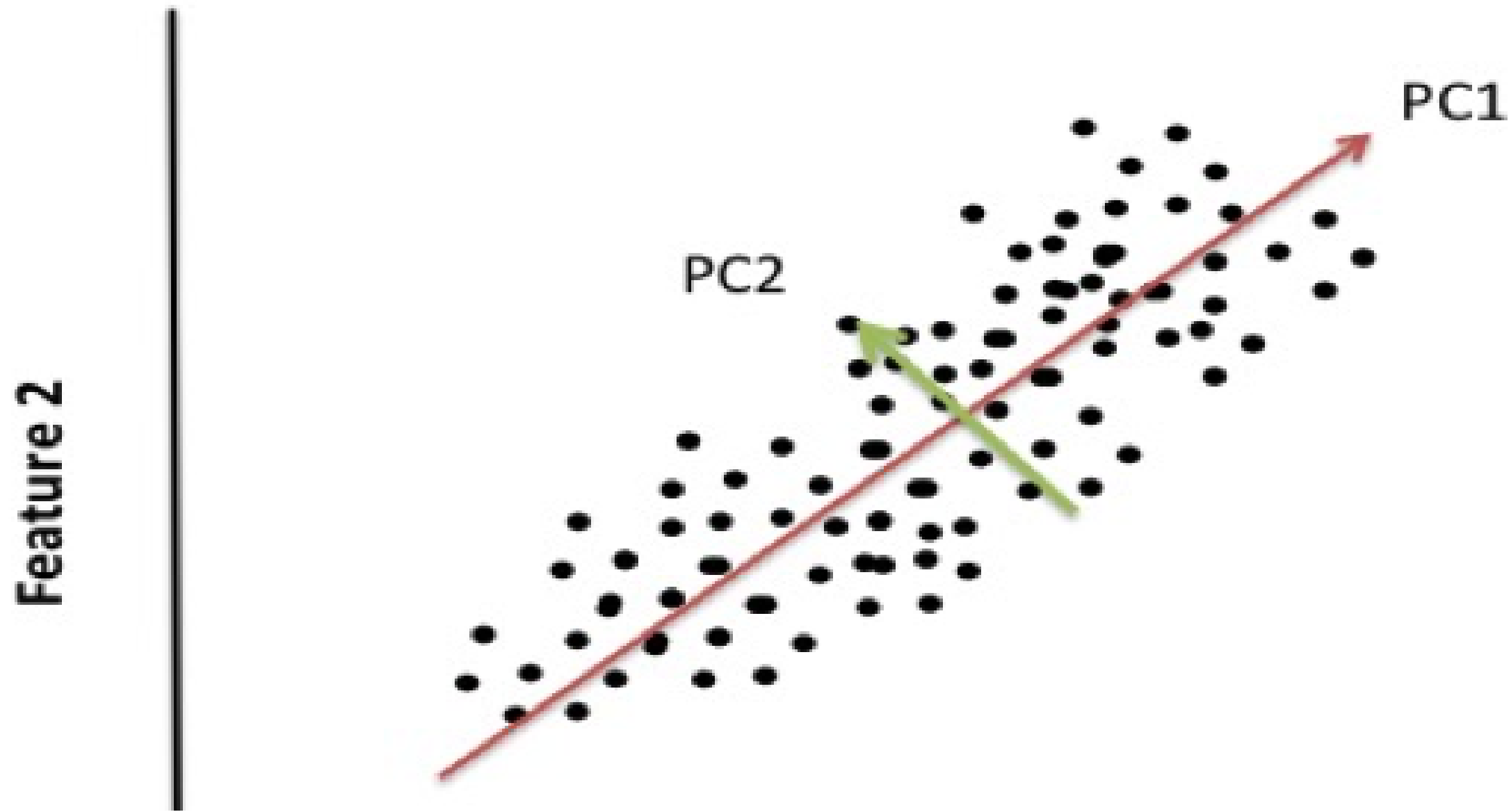Data Scientist

datacamp
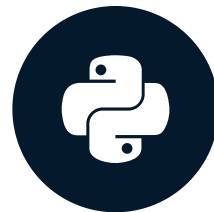
Why dimensionality reduction?

1. Speed up ML training

2. Visualization

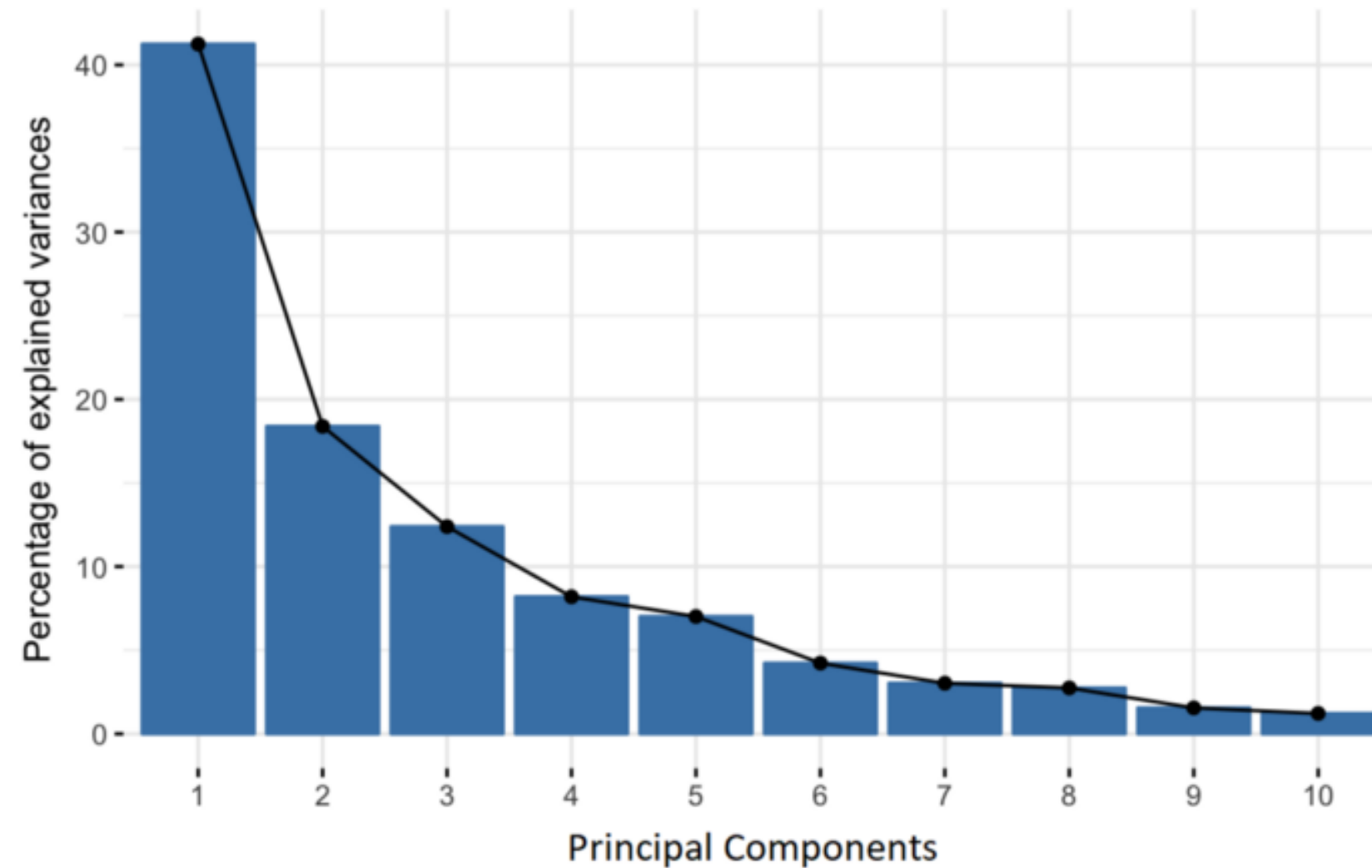3. Improves accuracy

Visualization techniques

- PCA

- t-SNE

Visualizing with PCA

Scree plot

[1] https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-b836fb9c97e2

t-SNE

- Probabilistic

- Pairs of data points

- Low-dimensional embedding

- Plot embeddings

Visualizing with t-SNE

```python
# t-sne with loan data
from sklearn.manifold import TSNE
import seaborn as sns


loans =  pd.read_csv('loans_dataset.csv')


# Feature matrix
X = loans.drop('Loan Status', axis=1)


tsne = TSNE(n_components=2, verbose=1, perplexity=40)
tsne_results = tsne.fit_transform(X)


loans['t-SNE-PC-one'] = tsne_results[:,0]
loans['t-SNE-PC-two'] = tsne_results[:,1]
```
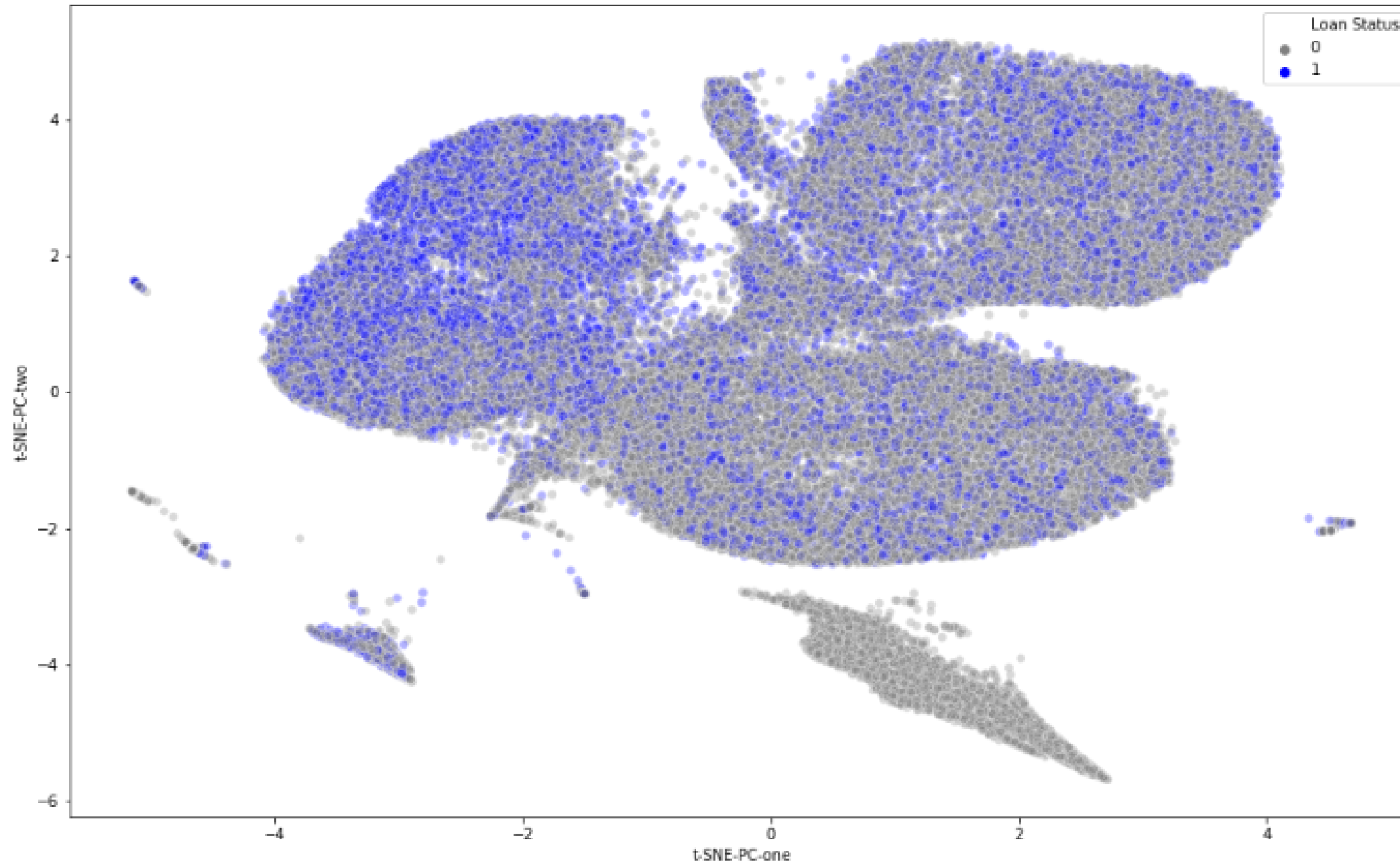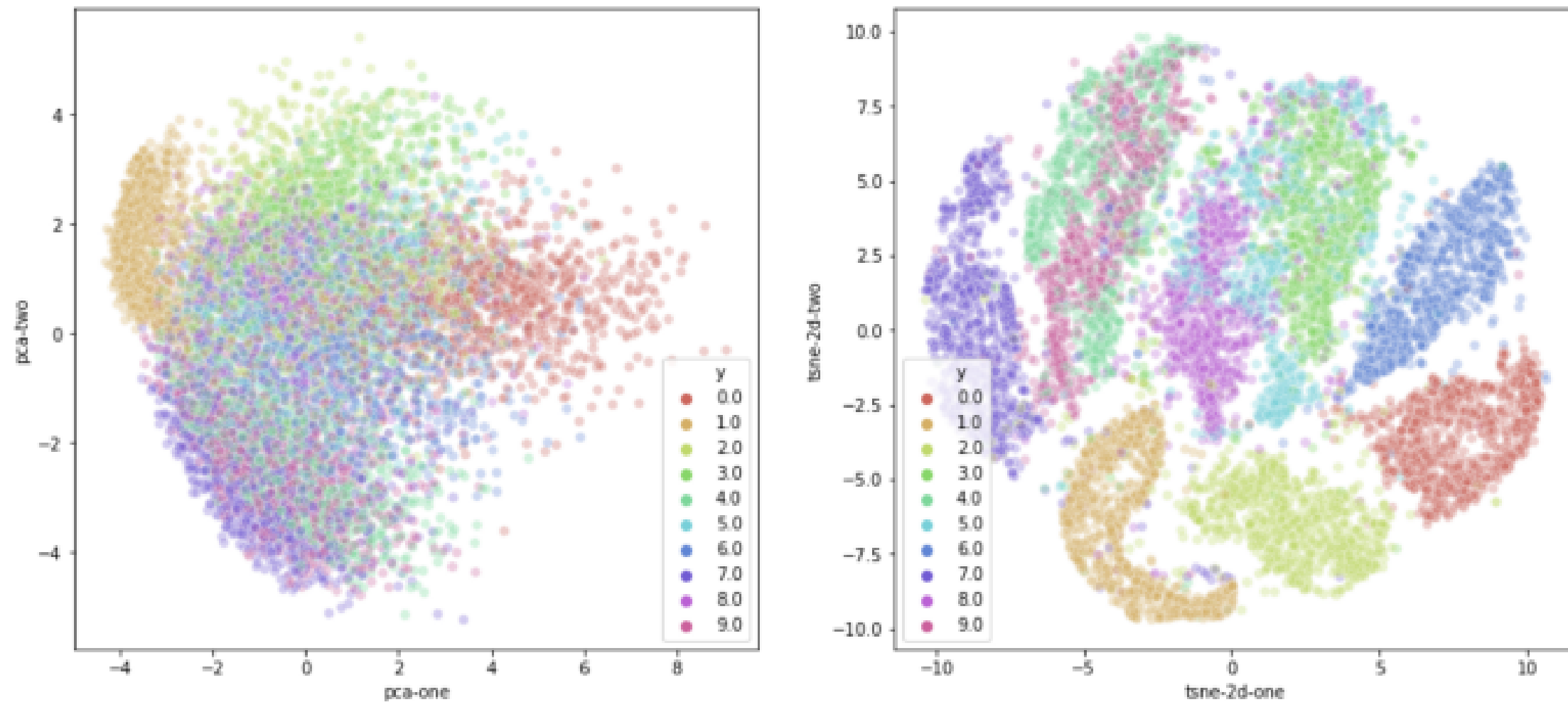
```python
# t-sne viz
plt.figure(figsize=(16,10))
sns.scatterplot(
    x="t-SNE-PC-one", y="t-SNE-PC-two",
    hue="Loan Status",
    palette=sns.color_palette(["grey","blue"]),
    data=loans,
    legend="full",
    alpha=0.3
)
```

[1] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

Visualizing with t-SNE

PCA vs t-SNE digits data

[1] https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b

Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Clustering analysis: selecting the right clustering algorithm

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

**Lisa Stuart**
Data Scientist

datacamp

Clustering algorithms

- Features >> Observations

- Model training more challenging

- Rely on distance calculations

- Most commonly used unsupervised technique

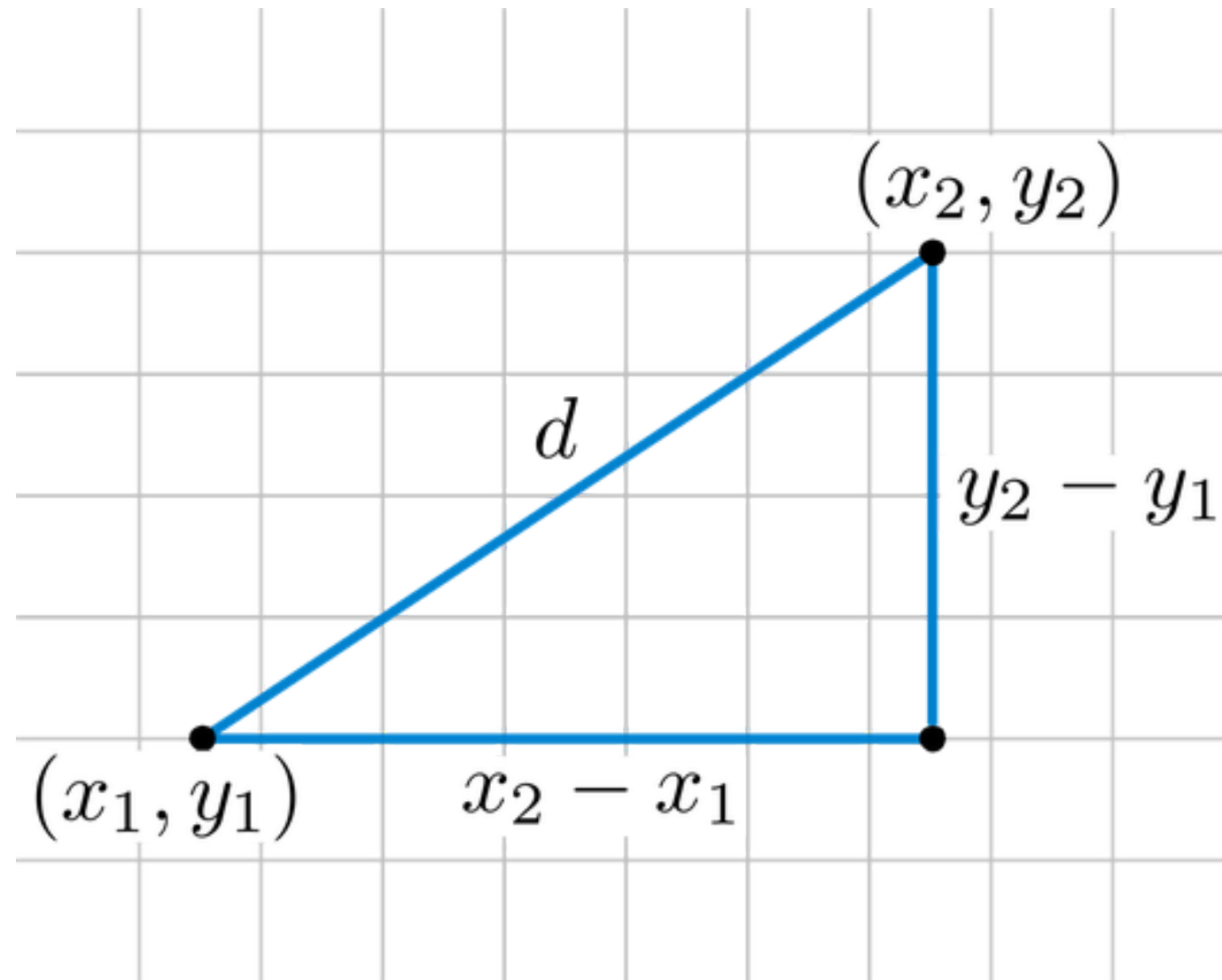# Practical applications of clustering

- Customer segmentation

- Document classification

- Insurance/transaction fraud detection

- Image segmentation

- Anomaly detection

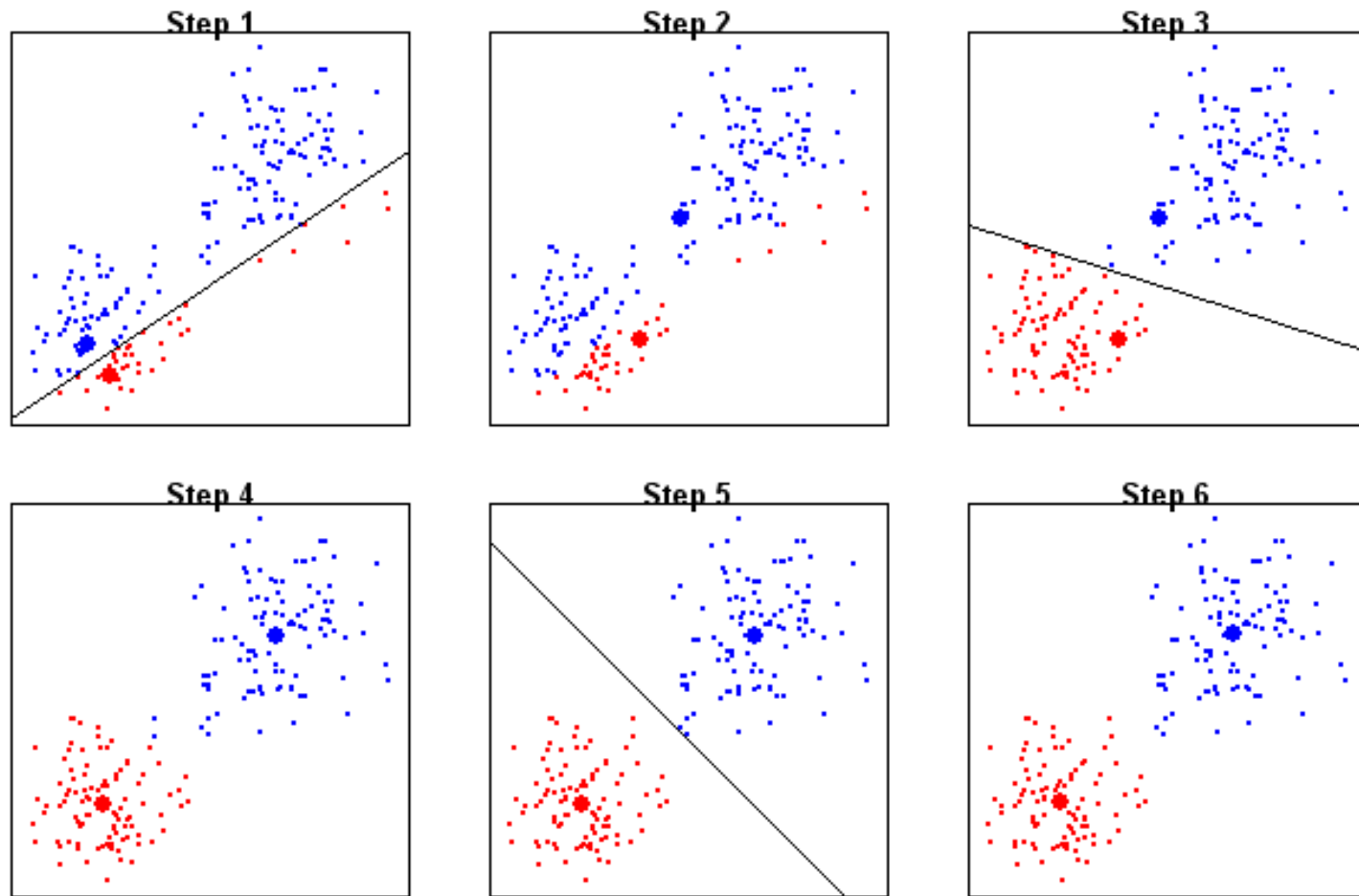- Many more...

Distance metrics: Manhattan (taxicab) distance
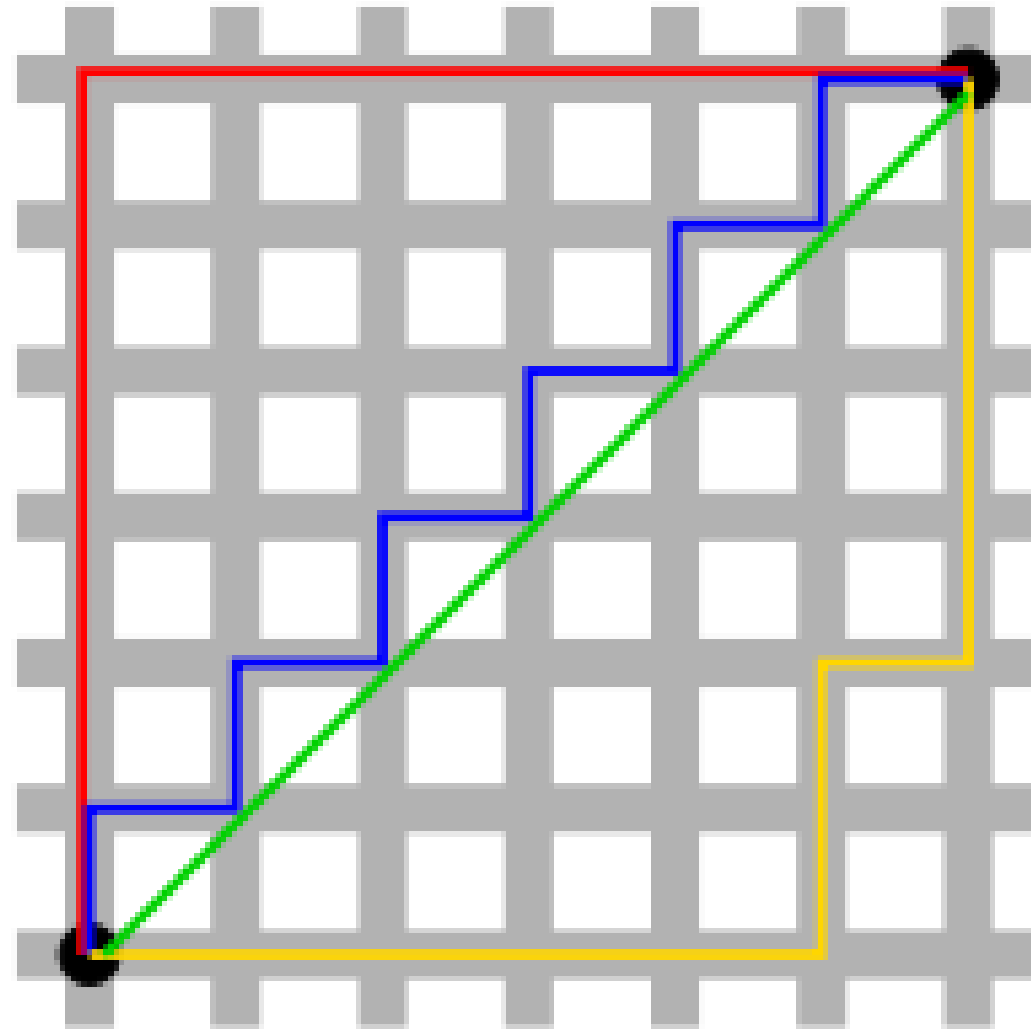
Distance metrics: Euclidian distance

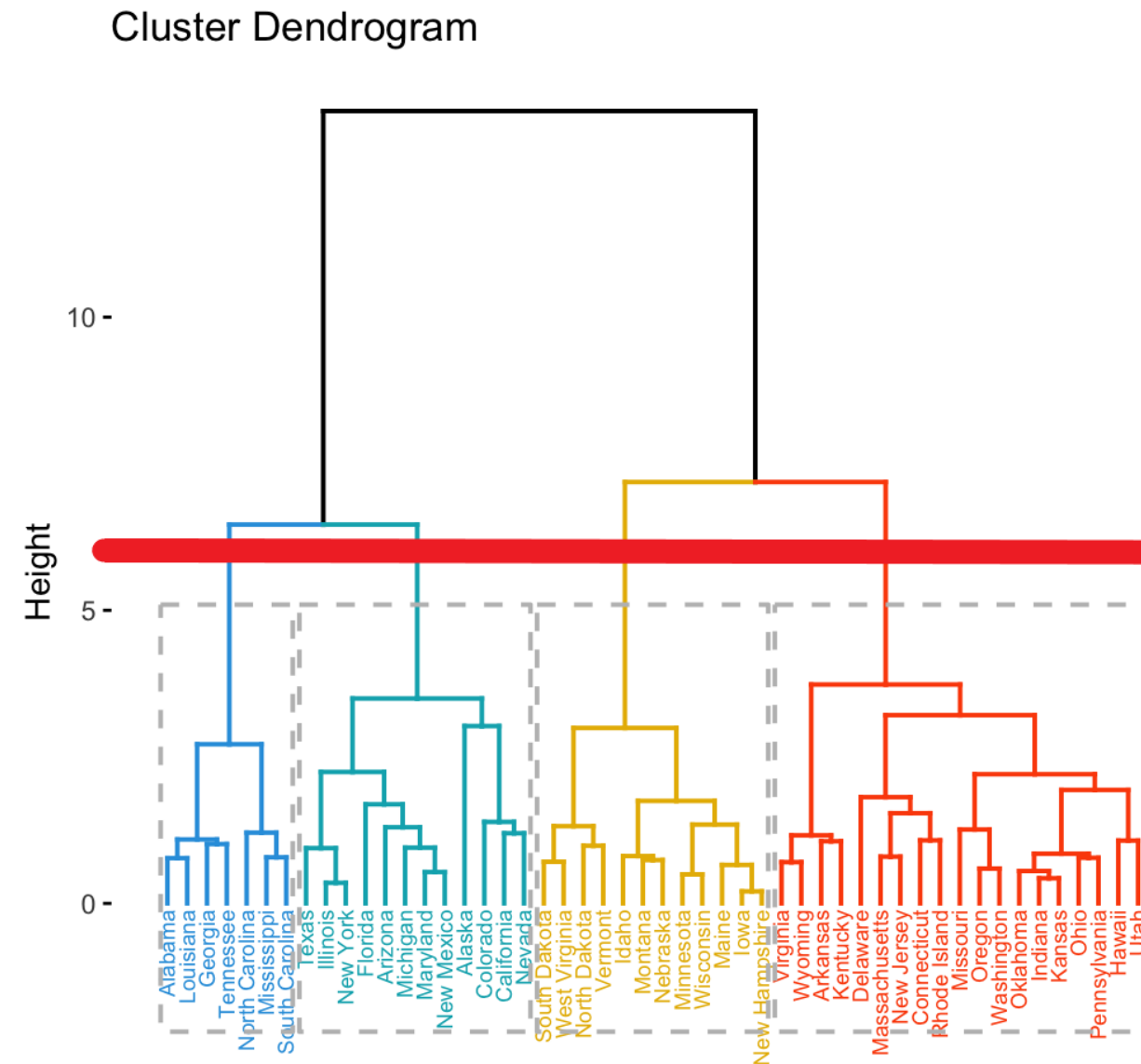# K-means

1. Initial centroids

2. Assign each observation to nearest centroid

3. Create new centroids

4. Repeat steps 2 and 3

[1] http://sherrytowers.com/2013/10/24/k-means-clustering/

Hierarchical agglomerative clustering
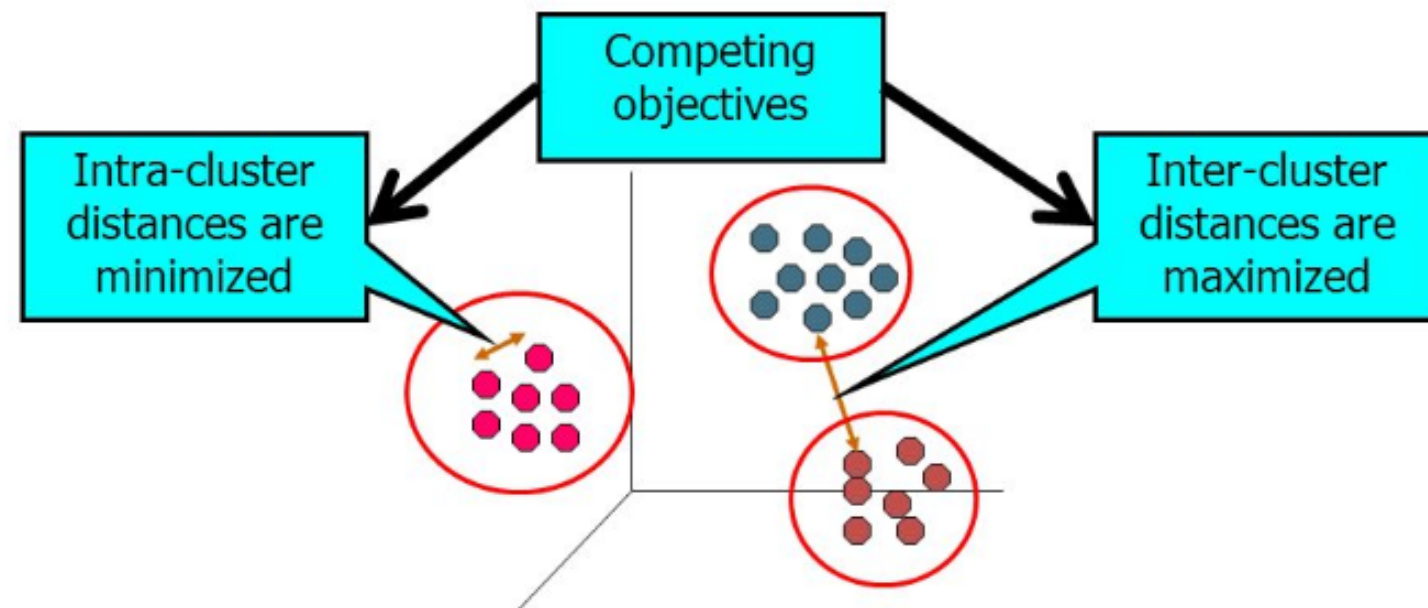

Cluster Dendrogram

Agglomerative clustering linkage

- Ward linkage

- Maximum/complete linkage

- Average linkage

- Single linkage

Selecting a clustering algorithm

- Cluster stability assessment

- K-means and HC use Euclidian distance

- Inter- and intra-cluster distances

"An appropriate dissimilarity measure is far more important in obtaining success with clustering than choice of clustering algorithm." - from **Elements of Statistical Learning**

[1] https://slideplayer.com/slide/8363774/

Clustering functions

| Function/method	returns
`sklearn.cluster.Kmeans`	K-Means clustering algorithm
`sklearn.cluster.AgglomerativeClustering`	Agglomerative clustering algorithm
`kmeans.inertia_`	SS distances of observations to closest cluster center
`scipy.cluster.hierarchy` as `sch`	Hierachical clustering for dendrograms
`sch.dendrogram()`	Dendrogram function

Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Clustering analysis: choosing the optimal number of clusters

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Lisa Stuart
Data Scientist
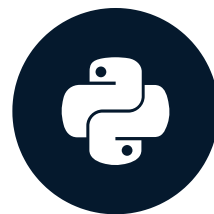
Methods for optimal k

- Silhouette method

- Elbow method

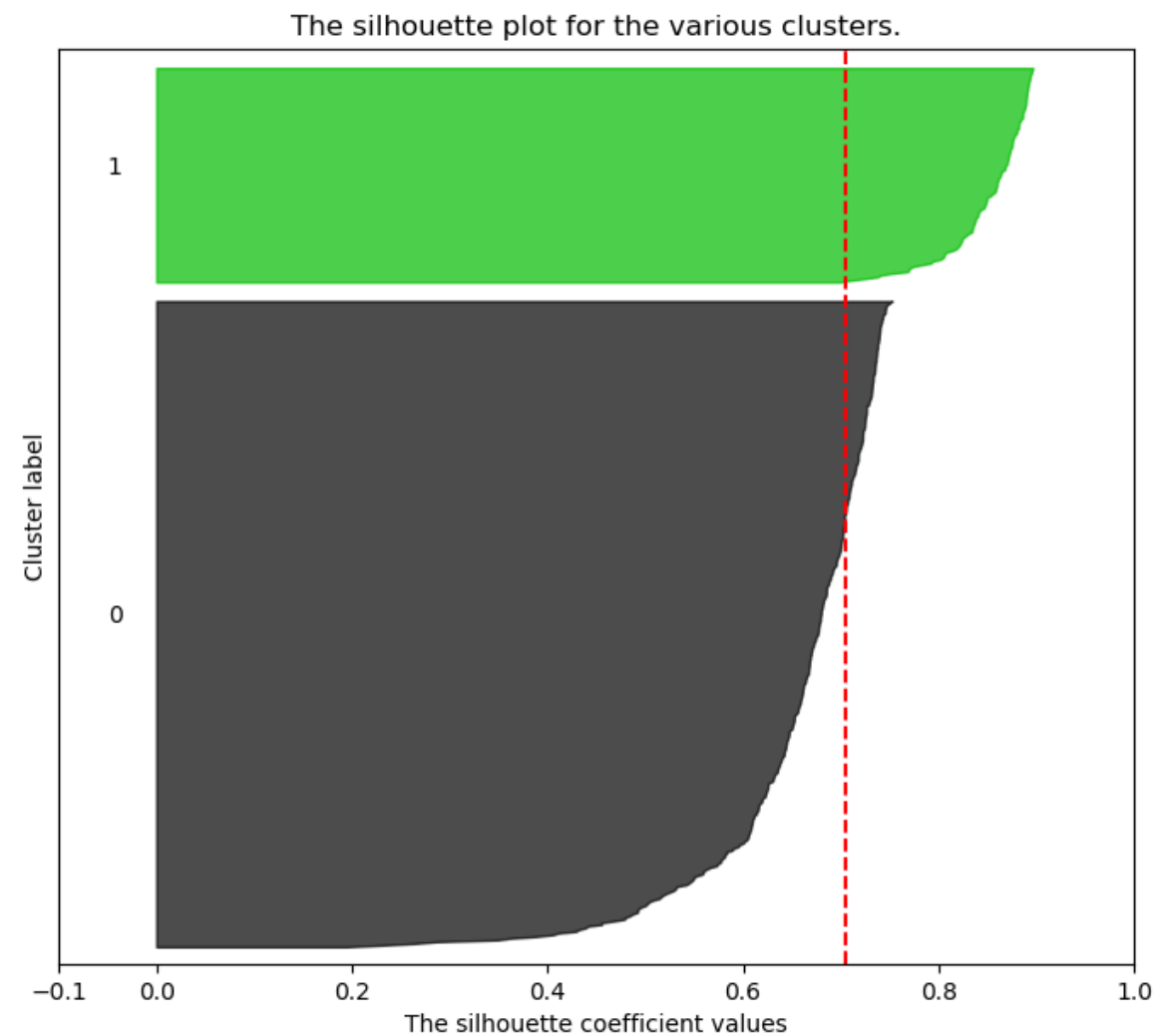# Silhouette coefficient

- Composed of 2 scores
  - Mean distance between each observation and all others:
    - in the same cluster
    - in the nearest cluster

Silhouette coefficient values

- Between -1 and 1
  - 1
    - near others in same cluster
    - very far from others in other clusters
  - -1
    - not near others in same cluster
    - close to others in other clusters
  - 0
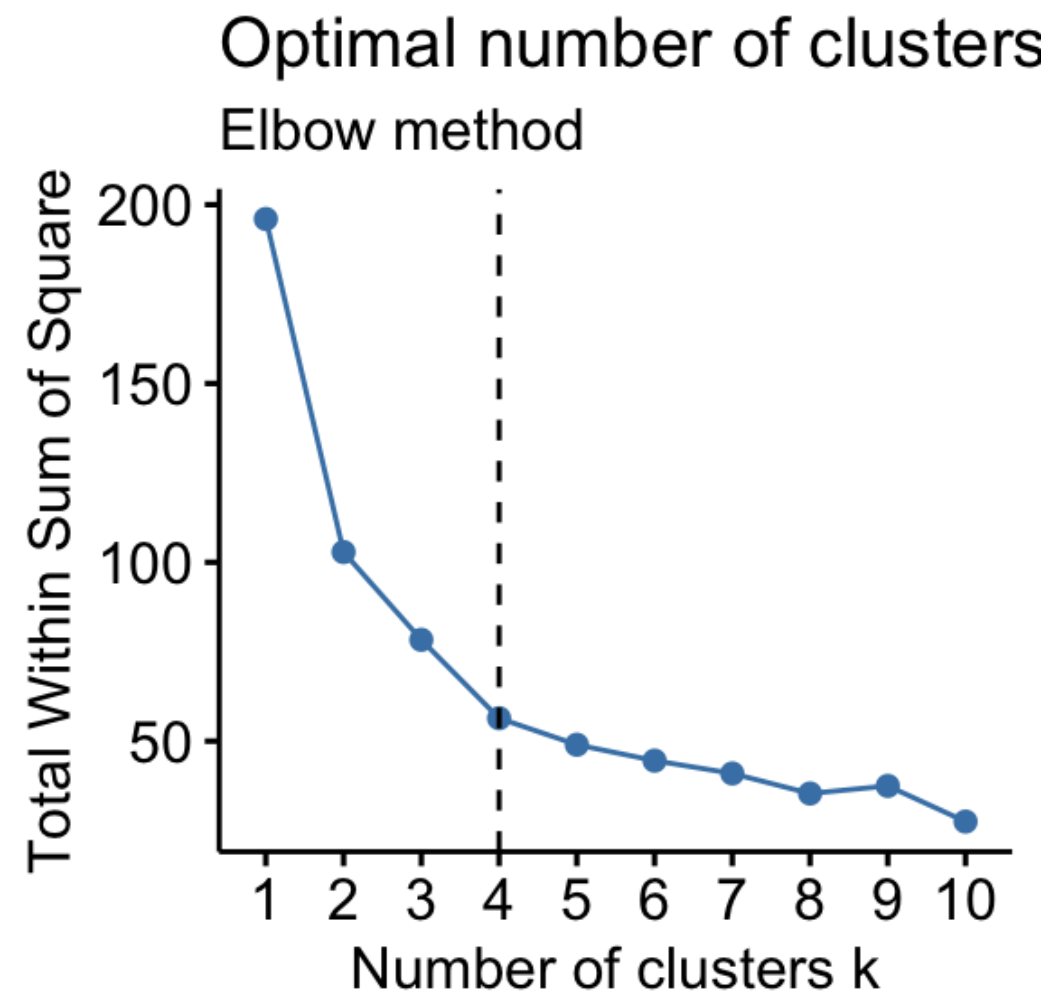   - denotes overlapping clusters

Silhouette score

Silhouette analysis for KMeans clustering on sample data with n_clusters = 2

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON

Elbow method

Optimal number of clusters

Optimal k selection functions

Function/method	returns
`sklearn.cluster.KMeans`	K-Means clustering algorithm
`sklearn.metrics.silhouette_score`	score between -1 and 1 as measure of cluster stability
`kmeans.inertia_`	SS distances of observations to closest cluster center
`range(start, stop)`	list of values beginning with start, up to but not including stop
`list.append(kmeans.inertia_)`	appends inertia value to list

Let's practice!

PRACTICING MACHINE LEARNING INTERVIEW QUESTIONS IN PYTHON