A Practical Introduction to Data Science

Part 5

Unsupervised Learning

Gergely Zsombor Haász

haasz.zsombi@gmail.com

Course Agenda

Unsupervised Learning

Clustering

- Customer segmentation

Dimensionality Reduction

- Noise reduction, Visualization, Latent Variables

Anomaly Detection

- Fraud detection, fault detection

Recommendation Systems

- Personalized product/movie/news recommendations

Unsupervised Learning Algorithms

Clustering	Dimensionality Reduction	Anomaly Detection	Recommendation Systems
K-means	PCA	Statistical outlier detection	User-based Collaborative filtering
Hierarchical	Factor Analysis	Isolation forest	Item-based Collaborative filtering
DBSCAN	Manifold learning (e.g. t-SNE, UMAP)	One-class SVM	Content-based filtering
	Autoencoder	Autoencoder	Matrix factorization

Clustering

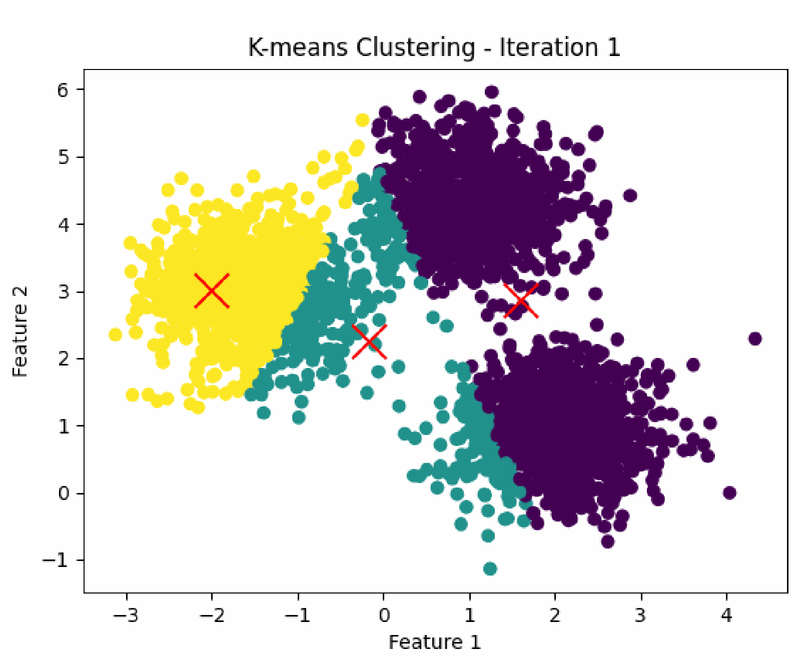
K-means Clustering

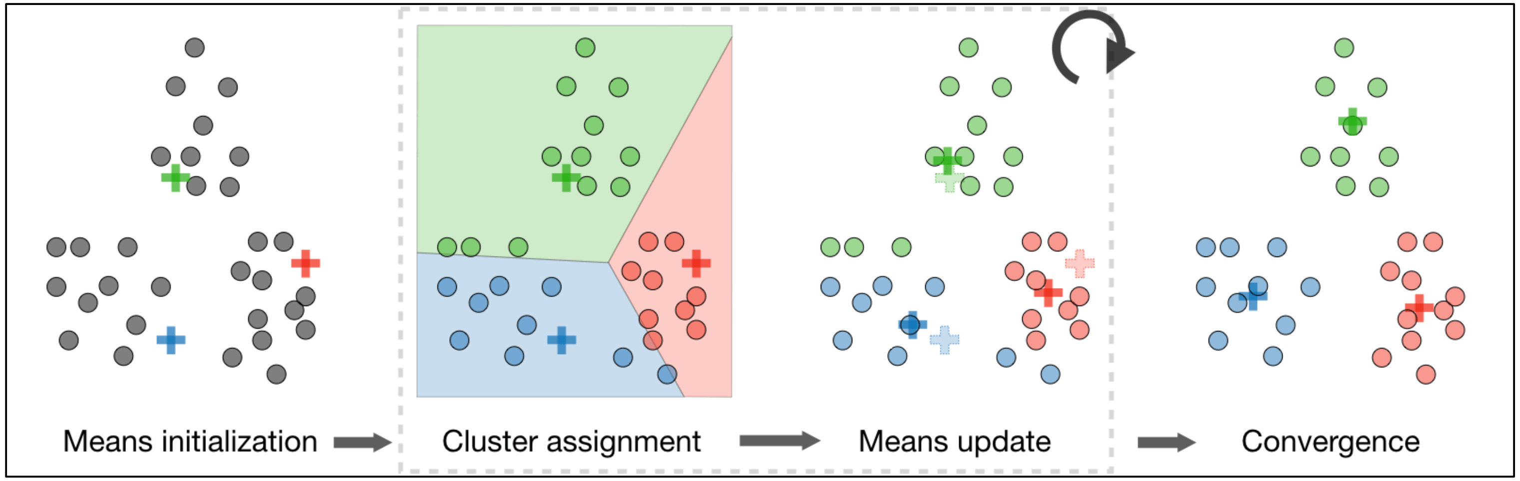
Initial steps:

- Choose k
- Choose a distance metric
- Standardize your data

Training steps:

1. Initialize k centroids randomly
2. Assign each data point to the closest centroid
3. Recalculate centroids (cluster means)
4. Repeat 2 and 3 until convergence





Source: [CS 221 - Reflex-based Models Cheatsheet]

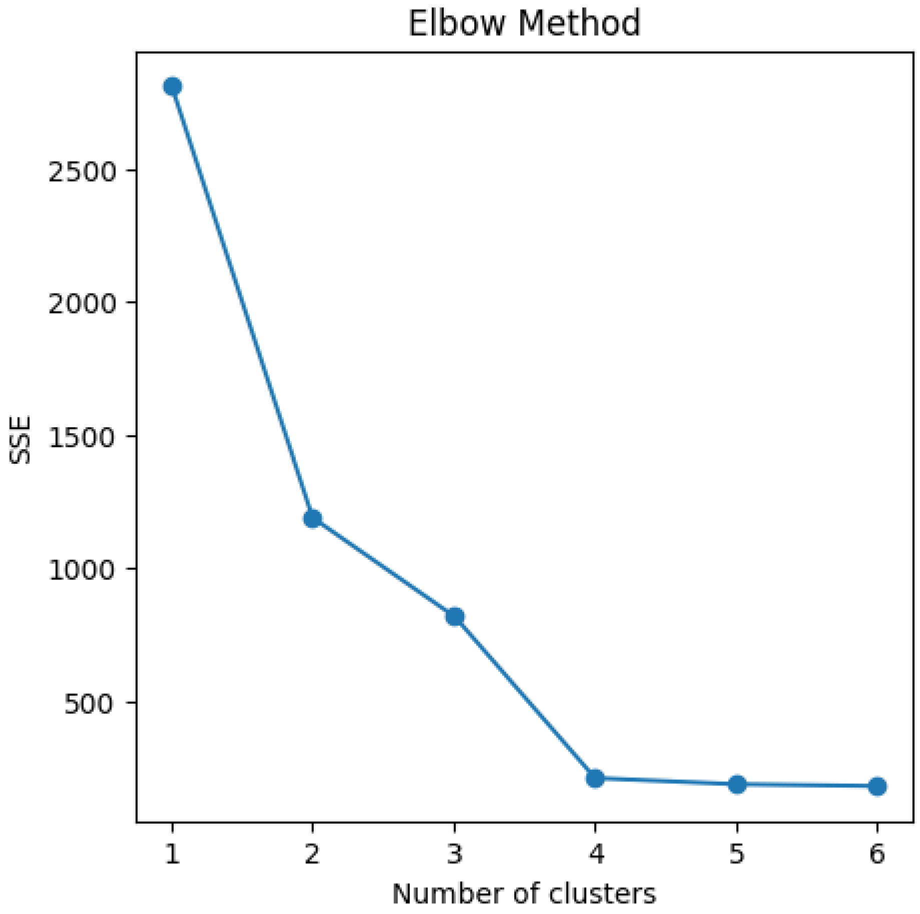
K-means Clustering

Goal:

- High within-cluster similarity: minimize SSE
- Low between-cluster similarity

Choosing the optimal k

- Elbow method
- Silhouette score
  - Measures how similar a data point is to its own cluster compared to other clusters. It ranges from -1 to 1



K-means Clustering

Advantages

- Simple and efficient
- Interpretability of centroids

Disadvantages

- K must be chosen manually
- Sensitive to initial centroid positions
- Sensitive to outliers
- Struggles with varying cluster shapes and densities
- Struggles with high dimensionality

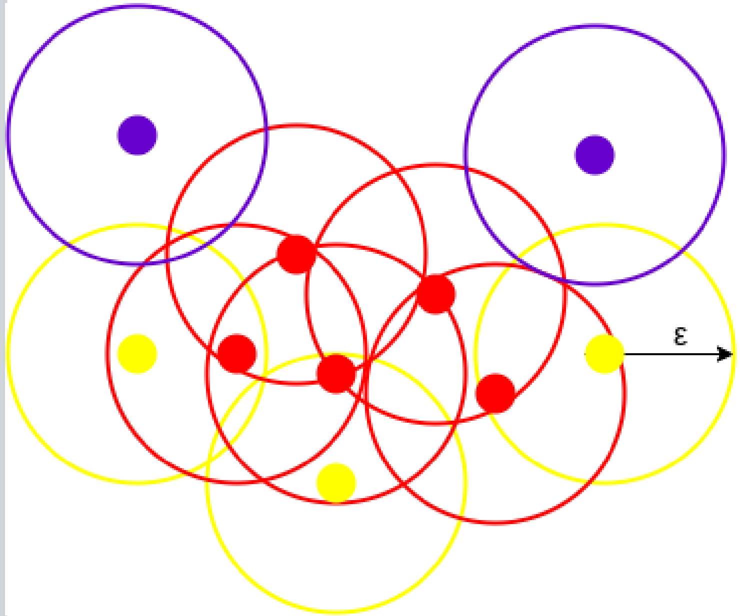
Density based clustering – DBSCAN

Core points: points with at least *minpoints* neighbours within *eps* radius

Border points: neighbours of core points within r radius but not core points

Outliers: neither core nor border points

1. Choose *eps* and *minpoints*
2. Start with a random point.
    a. If it is not a core point, then mark it as noise. (It can still become a border point later)
    b. If it is a core point, then start forming a cluster by adding neighbours to the cluster
    c. Assess neighbours the same way and extend the cluster until it is complete
3. Move on to another unvisited point and repeat the process until all points have been assigned to a cluster or marked as noise.



Source: DBSCAN Algorithm for Fraud Detection & Outlier Detection in a Data set | by Dilip Valeti | Medium

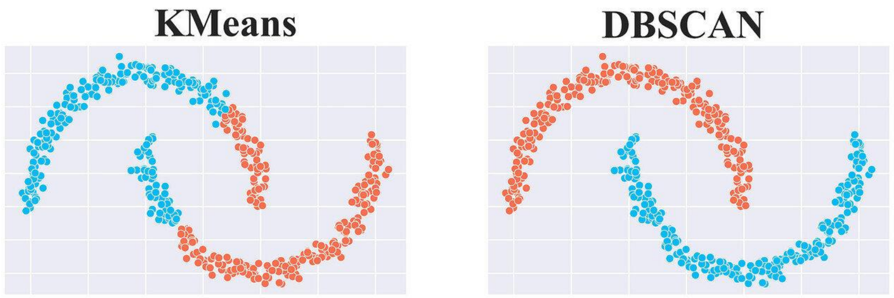
Density based clustering – DBSCAN

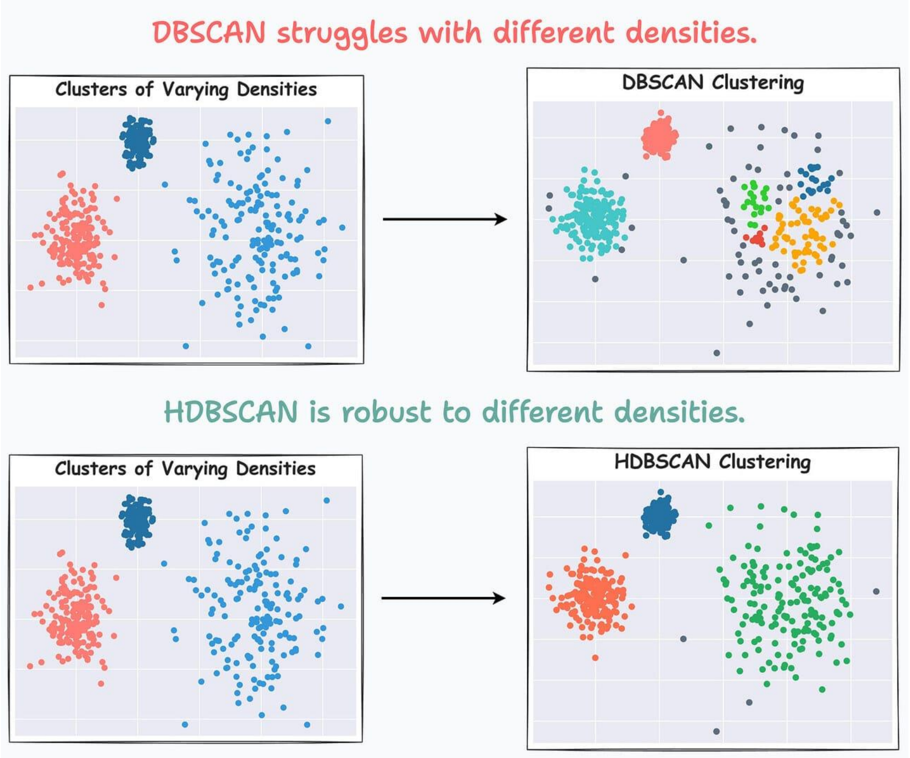
Advantages:

- Well suited for datasets with outliers
- Don’t need to specify number of clusters in advance

Disadvantages:

- Sensitive to parameters (*eps* and *minpoints*)
- Struggles with varying densities





Source: [HDBSCAN vs. DBSCAN]

Hierarchical Clustering

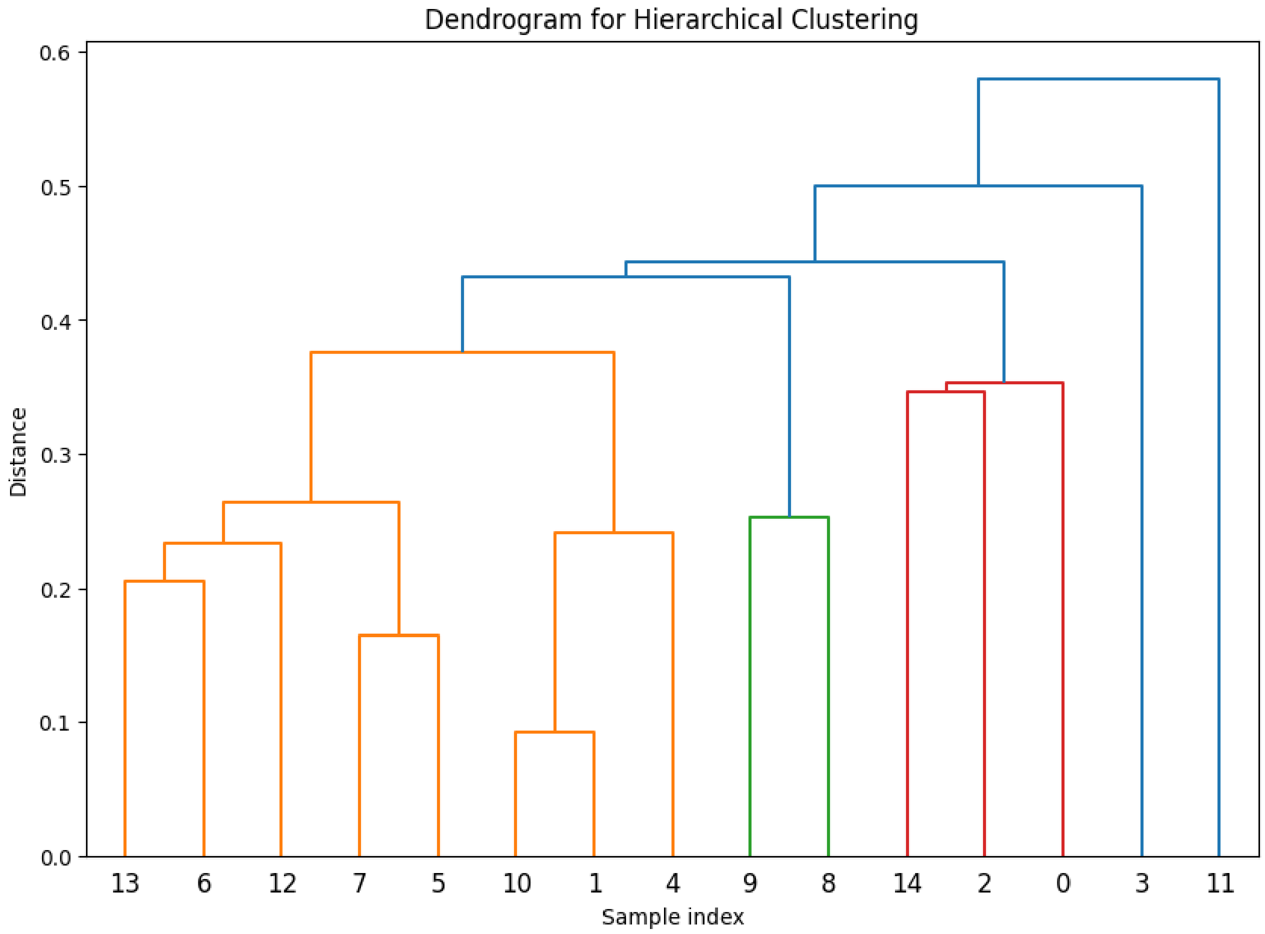
- Agglomerative or Divisive
- Choose a distance metric
- Choose a linkage method

Advantages:

- No need to specify number of clusters
- Easy to interpret (dendrogram)
- Good for small data

Disadvantages:

- Computationally expensive with large datasets
- Dependent on distance and linkage
- Struggles with high dimensionality



Dimensionality Reduction

Principal Component Analysis

- Principal components are uncorrelated linear combinations of the original vectors
- By selecting the first K Principal Components, we can reduce dimensionality (from N to K), while keeping the maximum variance (information) possible in the data

Steps:

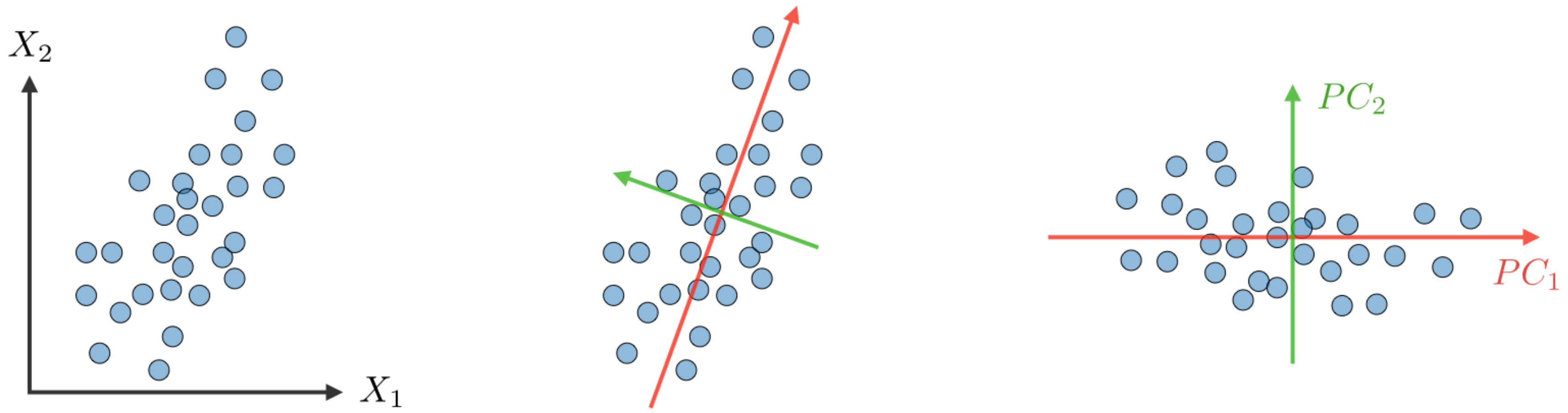
1. Standardize the data
2. Calculate the covariance matrix
3. Compute **eigenvectors** and **eigenvalues** (linear algebra)

$$Av = \lambda v$$

   - The **eigenvectors** (v) of the covariance matrix are the directions of the maximum variance
   - The corresponding **eigenvalues** (λ) represent the magnitude of that variance

Principal Component Analysis

4. Sort the eigenvectors by their corresponding eigenvalues descending.
5. Select the first K eigenvectors (e.g. until $\lambda > 1$)
6. Project the data onto the subspace spanned by the selected eigenvectors.
   In other words, we obtain the principal components by multiplying the original data matrix by the matrix of eigenvectors.



Data in feature space → Find principal components → Data in **p**rincipal **c**omponents space

Source: [CS 221 - Reflex-based Models Cheatsheet]

Principal Component Analysis

Advantages:

- Produces uncorrelated features
- Simple, fast, no hyperparameters
- Data gets smaller: easier to handle, explore and visualize
- Noise reduction and feature extraction

Limitations:

- Assumes strong linear correlation between variables
- Linear model, cannot detect complex patterns
- Does not necessarily preserve local structure of data
- Not suitable for categorical data
- Sensitive to Outliers

Anomaly Detection

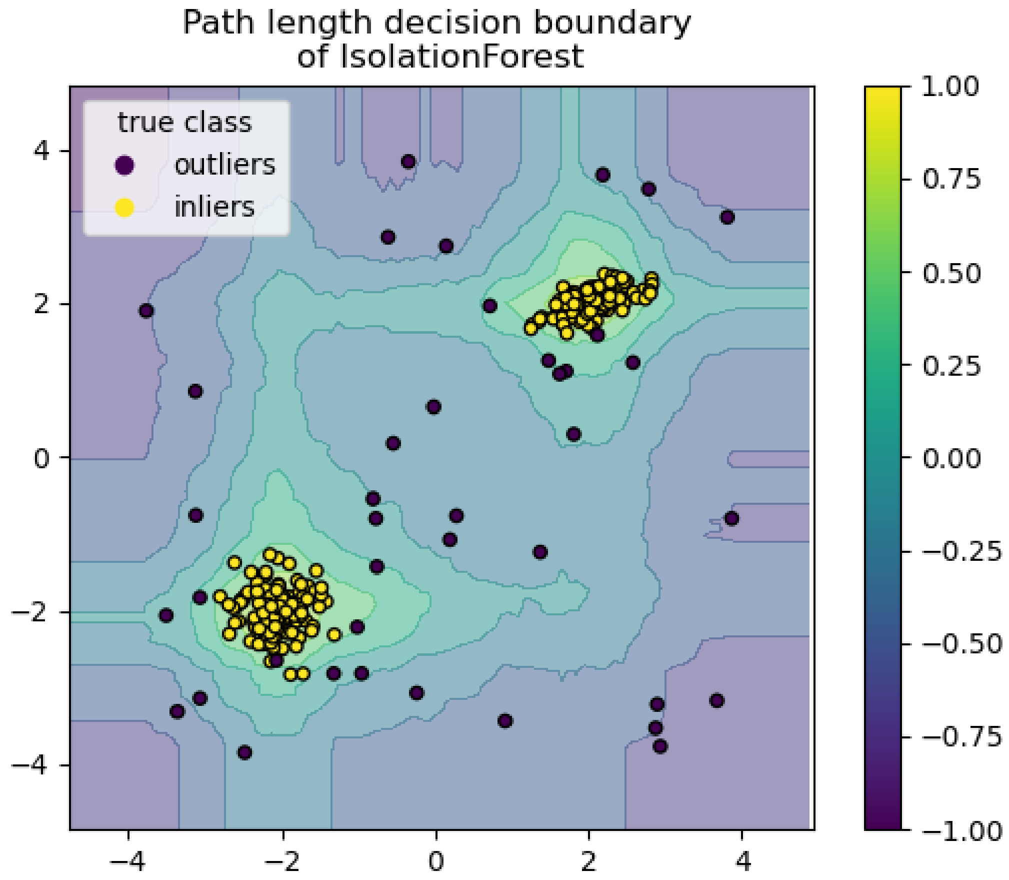
Univariate Anomaly Detection

- Plot the distribution – histogram and boxplot
- Z-score – the distance from the mean in units of standard deviations
- Median Absolute Deviation (MAD)
- Percentiles of the distribution
- Time series: deviations from short-term normal behaviour
    - Single outlier
    - Shift
    - Trend change
    - Increased short-term variance

Multivariate Anomaly Detection

Isolation Forest

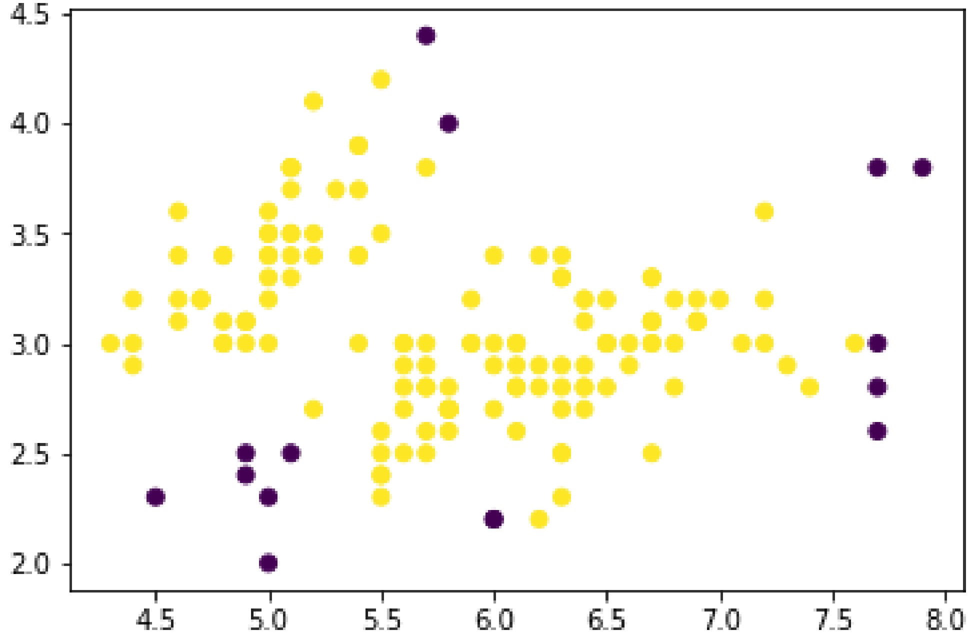
- An ensemble of isolation trees that isolate observations by recursive random partitioning, which can be represented by a tree structure.
- The number of splits required to isolate a sample is lower for outliers and higher for inliers.



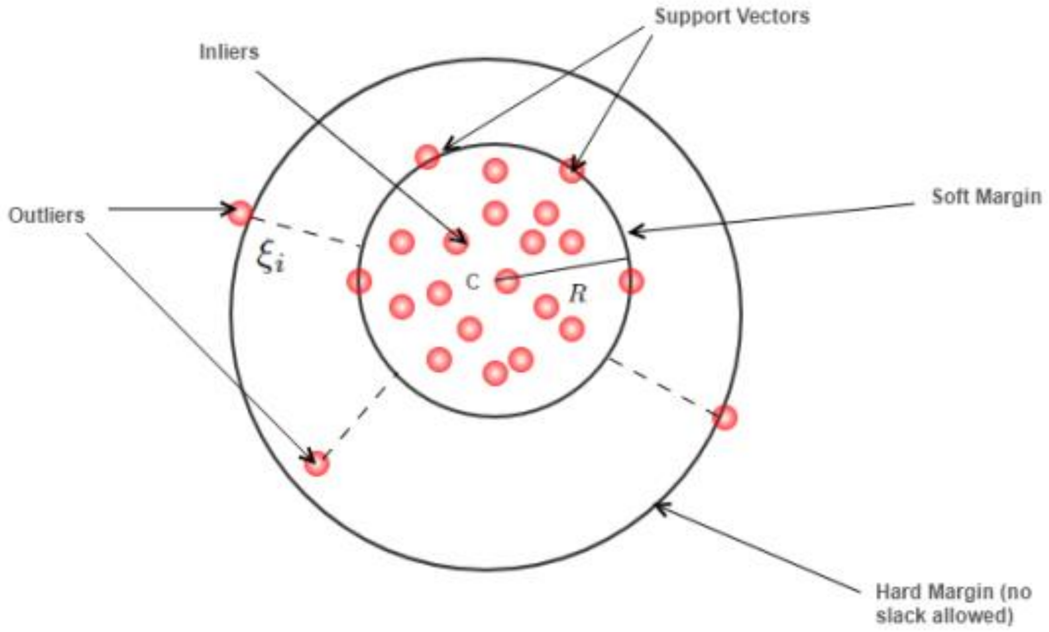
Source: IsolationForest example — scikit-learn 1.6.1 documentation

Multivariate Anomaly Detection

DBSCAN
Outliers are not part of the main cluster



One-Class SVM
Decision boundary around normal points



Source: [DBSCAN Algorithm for Fraud Detection & Outlier Detection in a Data set | by Dilip Valeti | Medium]

Source: [One Class SVM(OC-SVM). We all know that SVM has been a very… | by Garima. | Medium]

Recommendation Systems

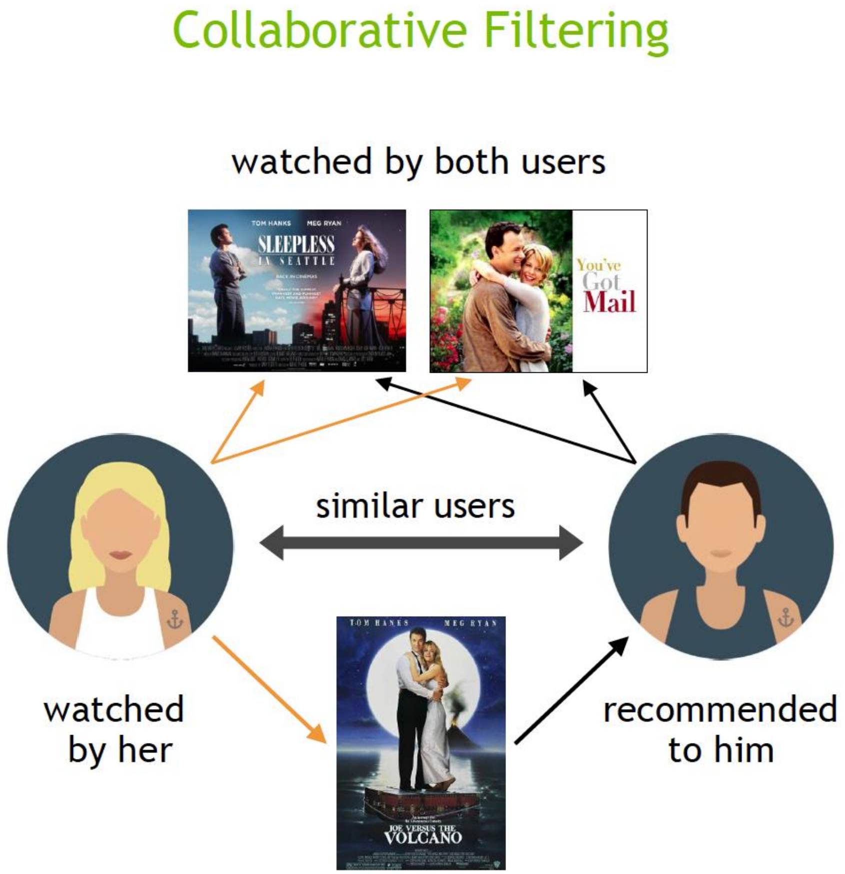
Recommendation Systems

User-User Collaborative Filtering

- Find users who are similar to the target user and recommend items that those similar users have liked

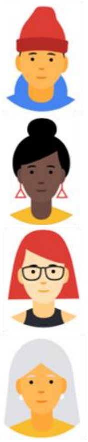
Item-Item Collaborative Filtering

- Fing items similar to the ones the target user has liked and recommend those items



Source: [What is a Recommendation System? | Data Science | NVIDIA Glossary]

Matrix Factorization



	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
	✓		✓	✓	
		✓			✓
	✓	✓	✓		
				✓	✓

≈

1	.1
-1	0
.2	-1
.1	1

.9	-1	1	1	-.9
-.2	-.8	-1	.9	1

.88	-1.08	0.9	1.09	-0.8
-0.9	1.0	-1.0	-1.0	0.9
0.38	0.6	1.2	-0.7	-1.18
-0.11	-0.9	-0.9	1.0	0.91

Source: Matrix factorization | Machine Learning | Google for Developers

Recommendation Systems

Advantages:

- No need for item/user features, only the user-item matrix
- Can recommend different items from previous ones (e.g. what others liked)

Challenges:

- Scalability (computational cost)
- Cold Start (new users or items)
- Data Sparsity
- Popularity and User bias (subtract the mean rating of each user/item)
- Matrix Factorization:
  - Bad interpretability compared to item-item/user-user methods
  - Overfitting

Recommendation Systems

Content-based Filtering

- Uses item features to recommend items similar to what the user likes

Advantages:

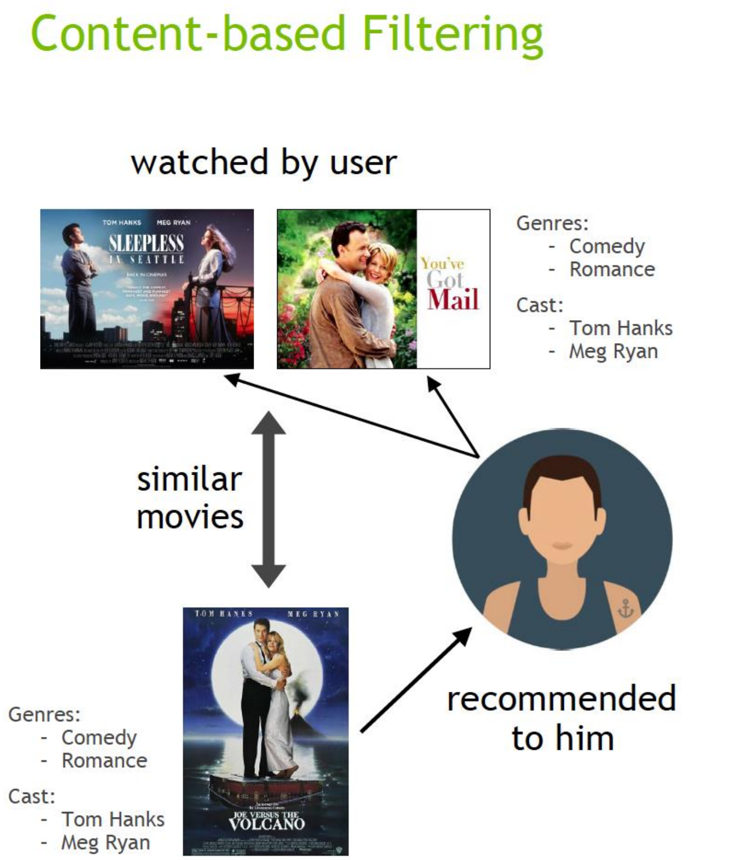
- No cold start problem for items
- Explainability

Disadvantages

- Limited Novelty (only similar items)
- Cold start for users

Tips:

- Hybrid methods
- Frequent data and model updates



Source: [What is a Recommendation System? | Data Science | NVIDIA Glossary]

Unsupervised Learning Algorithms

Clustering	Dimensionality Reduction	Anomaly Detection	Recommendation Systems
☐ K-means	☐ PCA	☐ Statistical outlier detection	☐ User-based Collaborative filtering
☐ Hierarchical	☐ Factor Analysis	☐ Isolation forest	☐ Item-based Collaborative filtering
☐ DBSCAN	☐ Manifold learning (e.g. t-SNE, UMAP)	☐ One-class SVM	☐ Content-based filtering
	☐ Autoencoder	☐ Autoencoder	☐ Matrix factorization

Thank you for your attention!

Your feedback would be much appreciated:

Any Questions?

Gergely Zsombor Haász

haasz.zsombi@gmail.com