Machine Learning and Deep Learning
Lecture-07

By: Somnath Mazumdar
Assistant Professor
sma.digi@cbs.dk

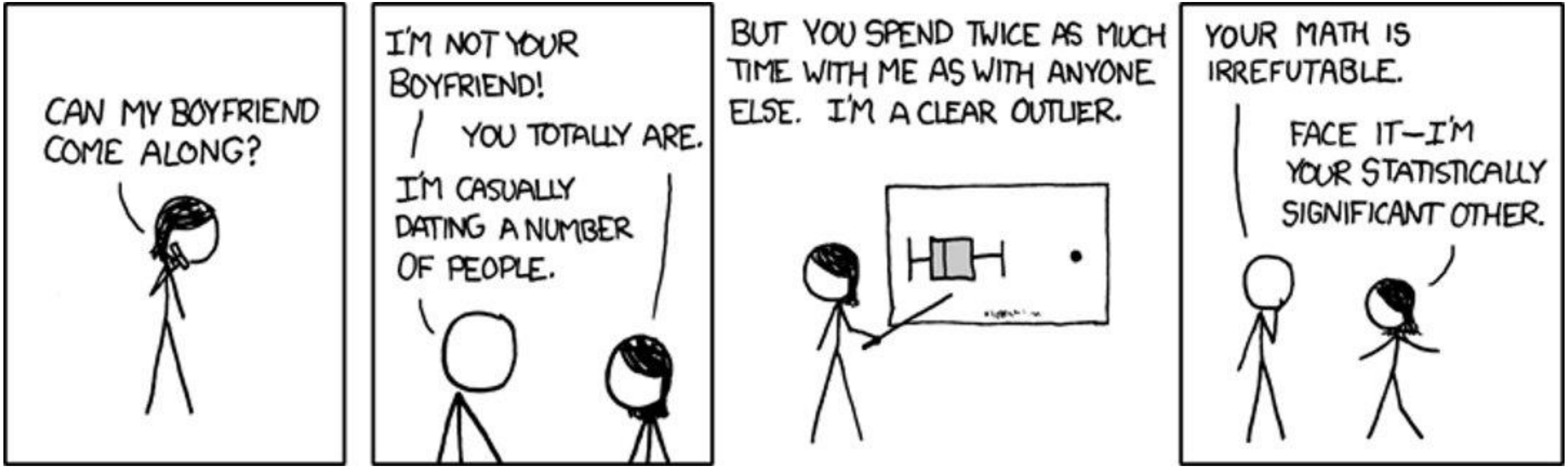
Spring 2025

CDSCO2004U/7.5 ECTS

Overview

- Outlier Detection
- Finding Similar Items
- Recommender Systems
- Class Imbalance

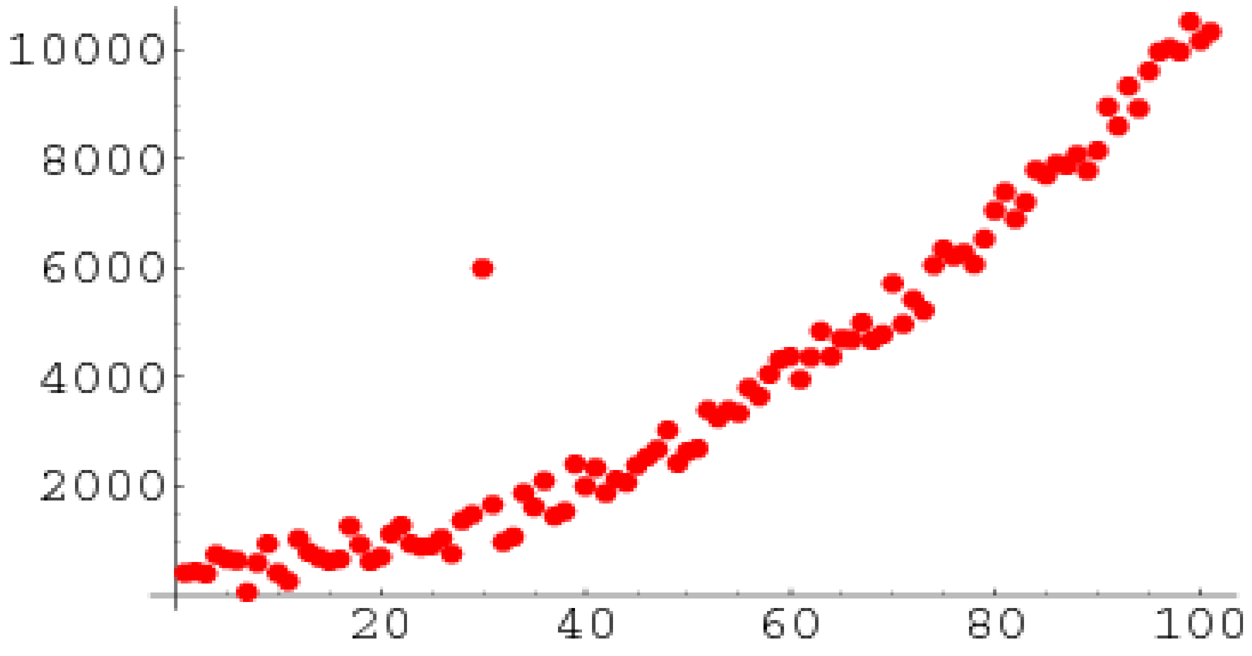
Outlier Detection



https://xkcd.com/539/

Outlier/Anomaly Detection

- Outlier: Data object that deviates significantly from normal objects as if it were generated by a different mechanism.
    - Find observations that are “unusually different” from others.
        - Issue: hard to define precisely.

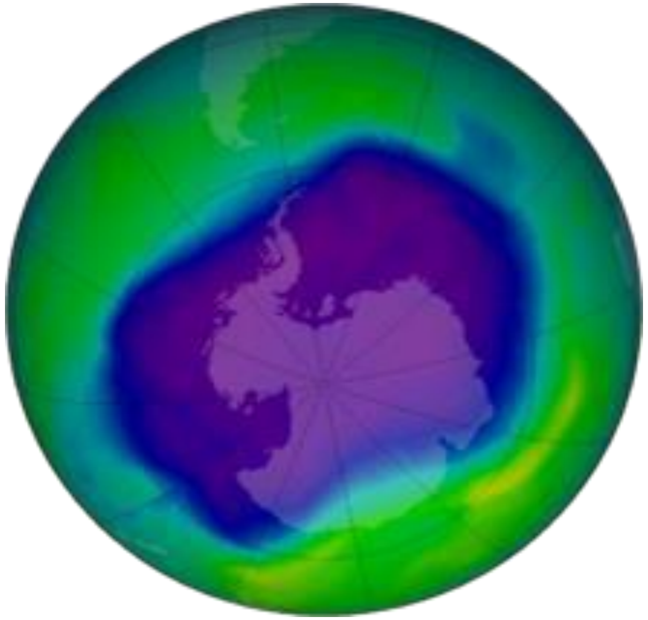


- Sources:
    - Measurement errors
    - Data entry errors
    - Contamination of data from different sources
    - Rare events

http://mathworld.wolfram.com/Outlier.html

Example: Finding Holes in Ozone Layer

- Huge Antarctic ozone hole was “discovered” in 1985.
- It had been in satellite data since 1976
    - But it was flagged and filtered out by quality-control algorithm.



https://en.wikipedia.org/wiki/Ozone_depletion

Outlier Detection

- Application Domain:
  - Data cleaning.
  - Security and fault detection
  - Fraud detection
  - Detecting natural disasters
  - Astronomy
  - Genetics

- Methods to detect outliers:
  - Model-based
  - Graphical approaches
  - Cluster-based
  - Distance-based
  - Supervised-learning

Global vs. Local Outliers

- Is the red point an outlier?

Global vs. Local Outliers

- Is the red point an outlier? What if we add the blue points?

Global vs. Local Outliers

- Is the red point an outlier? What if we add the blue points?

- Red point has the lowest z-score.
  - First case: “global” outlier.
  - Second case: “local” outlier:
    - Within normal data range, but far from other points.

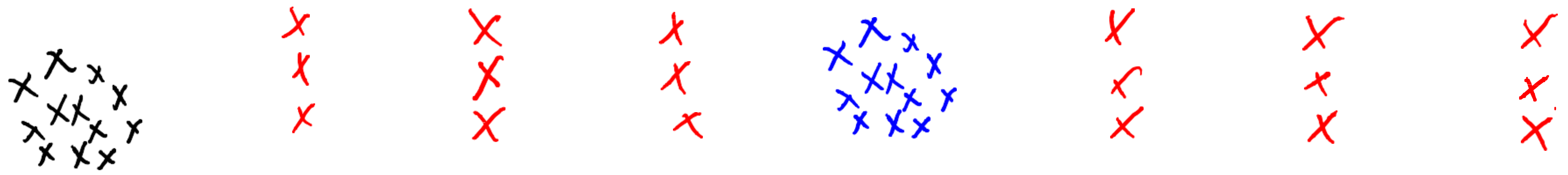
Global vs. Local Outliers

- Is the red point an outlier? What if we add the blue points?

- Red point has the lowest z-score.
  - First case : “global” outlier.
  - Second case : “local” outlier:
    - Within normal data range, but far from other points.
- Can we have outlier groups?

Global vs. Local Outliers

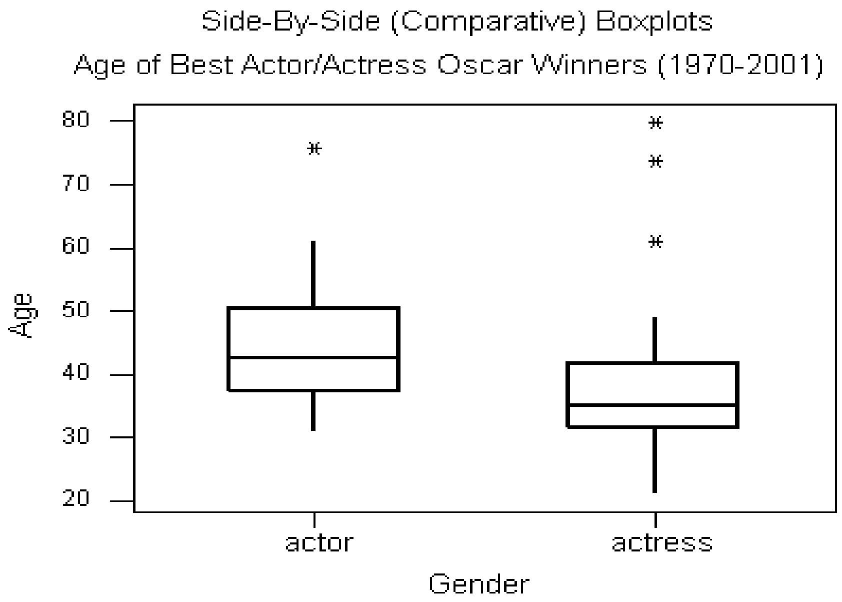
- Is the red point an outlier? What if we add the blue points?

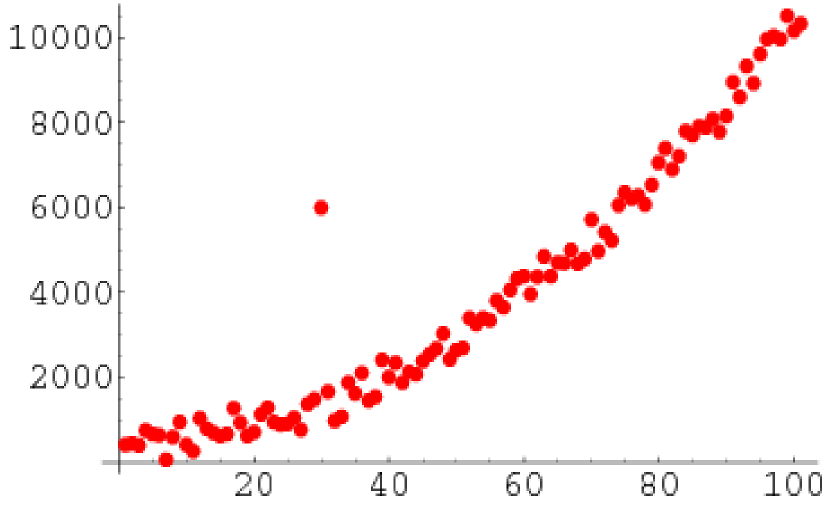


- Red point has the lowest z-score.
  - First case : “global” outlier.
  - Second case : “local” outlier:
    - Within normal data range, but far from other points.
- Can we have outlier groups?
- What about repeating patterns?

Graphical Outlier Detection

- Graphical approach to outlier detection:
  - Plot the data and look for weird points.
  - Human decides if data is an outlier.





- boxplot(x) creates a box plot of data in x.
- If x is a vector--> box.
- If x is a matrix --> one box/column of x.

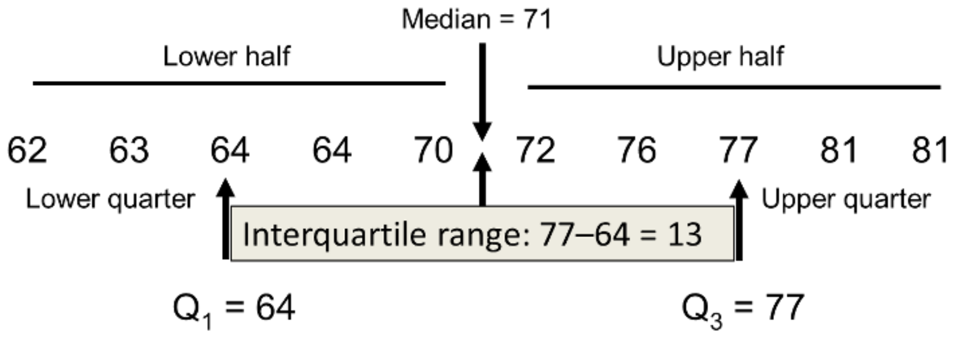
- Scatterplot detects complex patterns.
- Only 2 variables at a time.

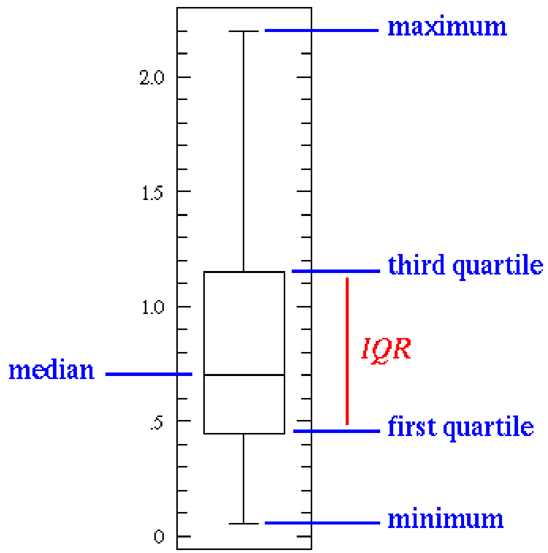
http://bolt.mph.ufl.edu/6050-6052/unit-1/one-quantitative-variable-introduction/boxplot/

http://mathworld.wolfram.com/Outlier.html

Quartiles and Boxplot

- First quartile (Q1): Median of lower half of data set.
  - 25% of numbers in data set lie below Q1 and about 75% lie above Q1.
- Third quartile (Q3): Median of upper half of data set.
  - 75% of numbers in data set lie below Q3 and about 25% lie above Q3.
- Interquartile Range (IQR): When a data set has outliers, variability is often summarized by a statistic called the interquartile range, i.e., Q3-Q1.





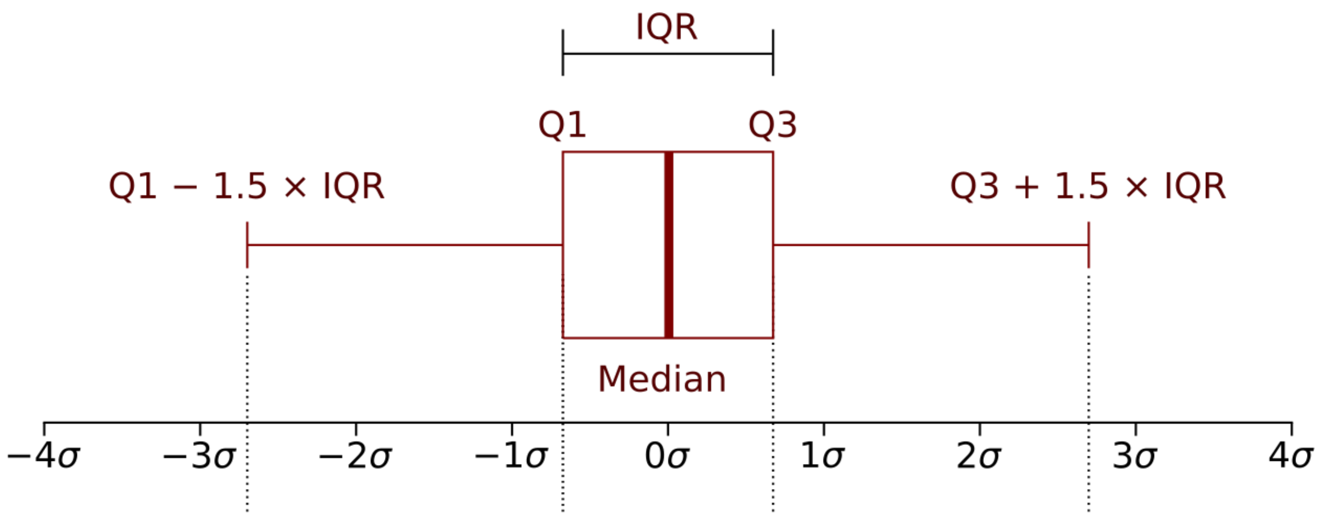
http://www.physics.csbsju.edu/stats/box2.html

- Boxplot: displays five-number summary of a set of data.
- They are minimum, Q1, median, Q3, and maximum.

http://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_summarizingdata/bs704_summarizingdata7.html

Box Plot: 1.5 IQR rule

- Why do we multiply the interquartile range by 1.5 to find the outlier?
- **Short answer:**
  - 1.5 IQR rule: is a rule of thumb used for identifying outliers.
  - It designates outliers based on an upper and lower boundary or “fence” outside of Q1 and Q3.
  - Anything above Q3 +1.5 x IQR is an outlier
  - Anything below Q1- 1.5xIQR is an outlier



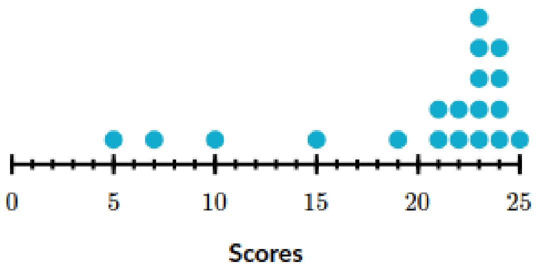
Long Answer

Identifying outliers with the 1.5xIQR rule

Google Classroom

An outlier is a data point that lies outside the overall pattern in a distribution.

The distribution below shows the scores on a driver's test for 19 applicants. How many outliers do you see?

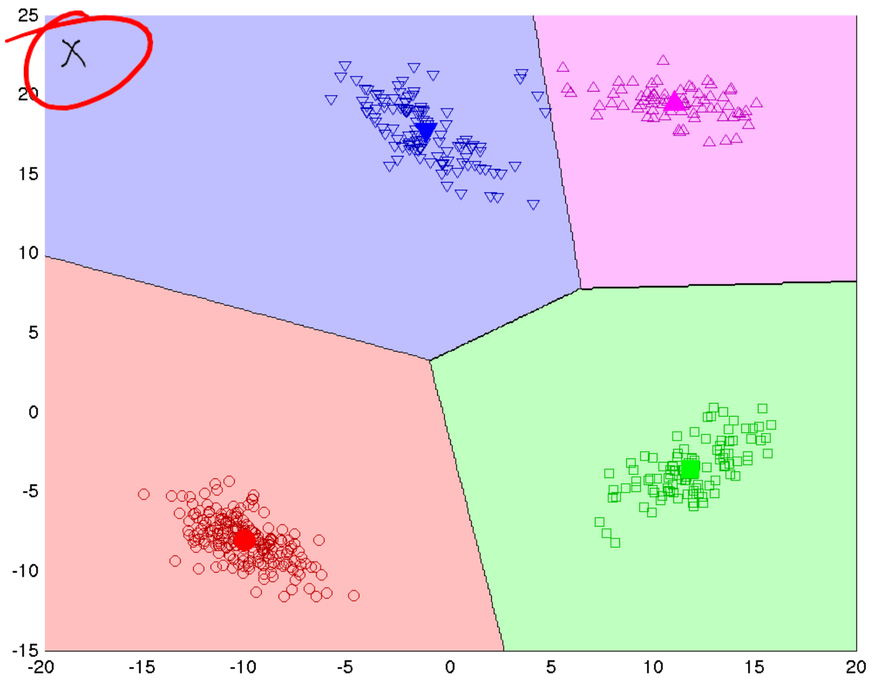


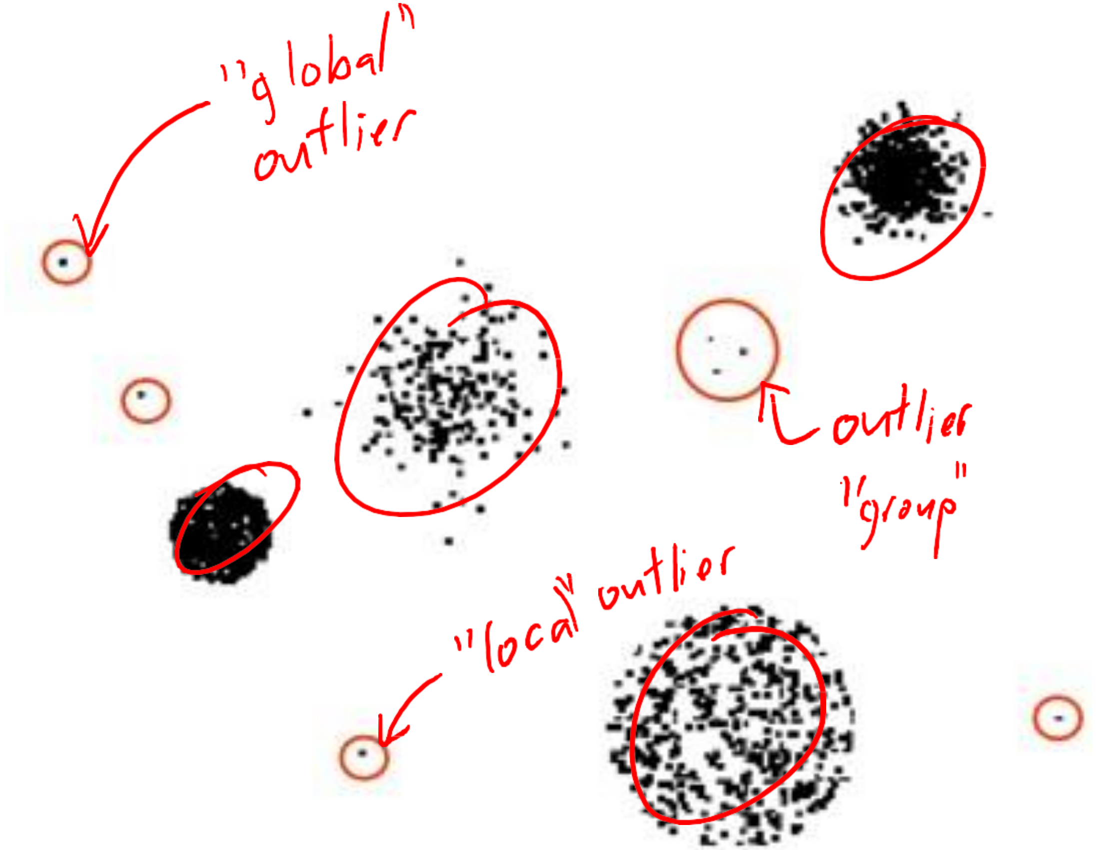
Some people may say there are 5 outliers, but someone else might disagree and say there are 3 or 4 outliers. Statisticians have developed many ways to identify what should and shouldn't be called an outlier.

A commonly used rule says that a data point is an outlier if it is more than $1.5 \cdot \text{IQR}$ above the third quartile or below the first quartile. Said differently, low outliers are below $Q_1 - 1.5 \cdot \text{IQR}$ and high outliers are above $Q_3 + 1.5 \cdot \text{IQR}$.

Cluster-Based Outlier Detection

- Detect outliers based on clustering
  - Cluster the data.
  - Find points that don't belong to clusters.





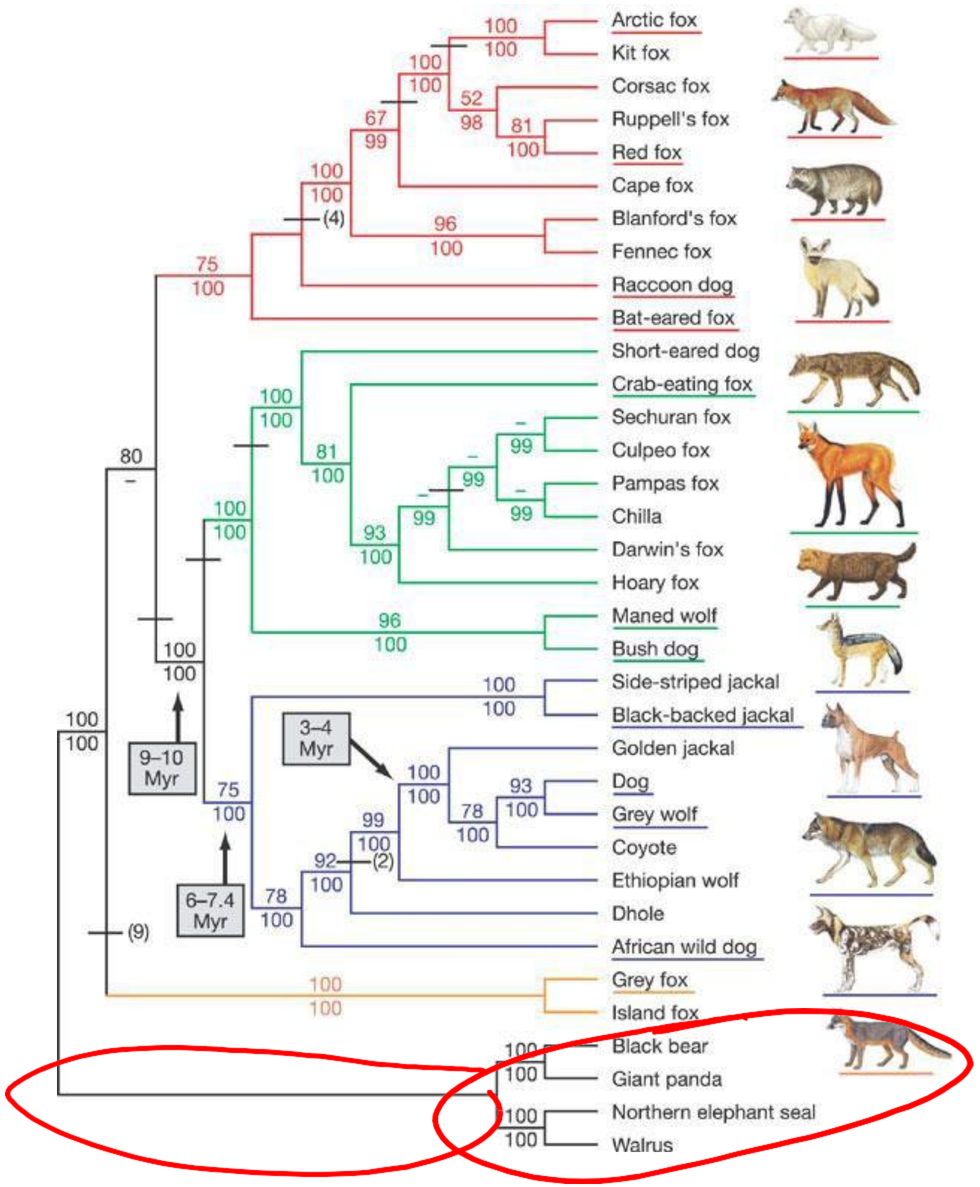
- K-means:
  - Find points that are far away from any mean.
  - Find clusters with a small number of points.

- Density-based clustering:
  - Outliers are points not assigned to cluster.

https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap10_anomaly_detection.pdf

Cluster-Based Outlier Detection

- Hierarchical clustering:
  - Outliers take longer to join other groups.
  - Also good for outlier groups.



http://www.nature.com/nature/journal/v438/n7069/fig_tab/nature04338_F10.html

Distance-Based Outlier Detection

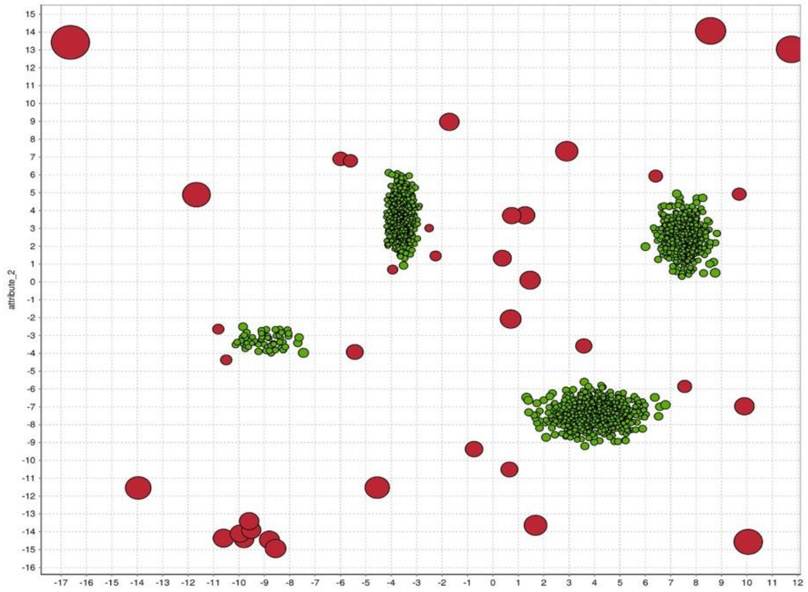
- Most outlier detection approaches are based on distances.
- Can we skip model/plot/clustering and just measure distances?
    - How many points lie in a radius ‘epsilon’?
    - What is distance to k^{th} nearest neighbour?

- Application: KNN best to detect “global” outliers
    - Compared 19 methods on 10 datasets*
    - “Local” outliers best found with local distance-based methods.

*http://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0152173

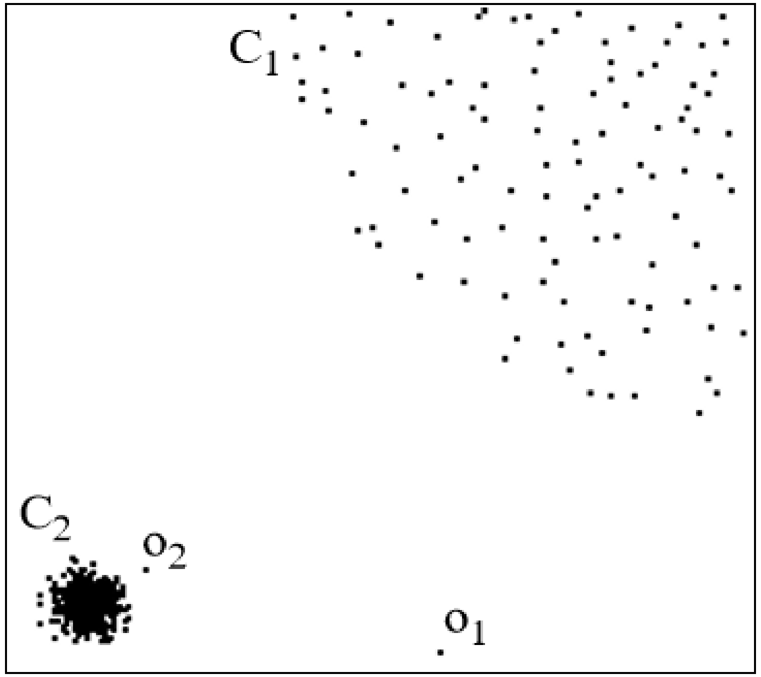
Global Distance-Based Outlier Detection: KNN

- KNN outlier detection:
    - For each point, compute the average distance to its KNN.
    - Sort the set of ‘n’ average distances.
    - Choose the biggest values as outliers.
        - Filter out points that are far from their KNNs.



Local Distance-Based Outlier Detection

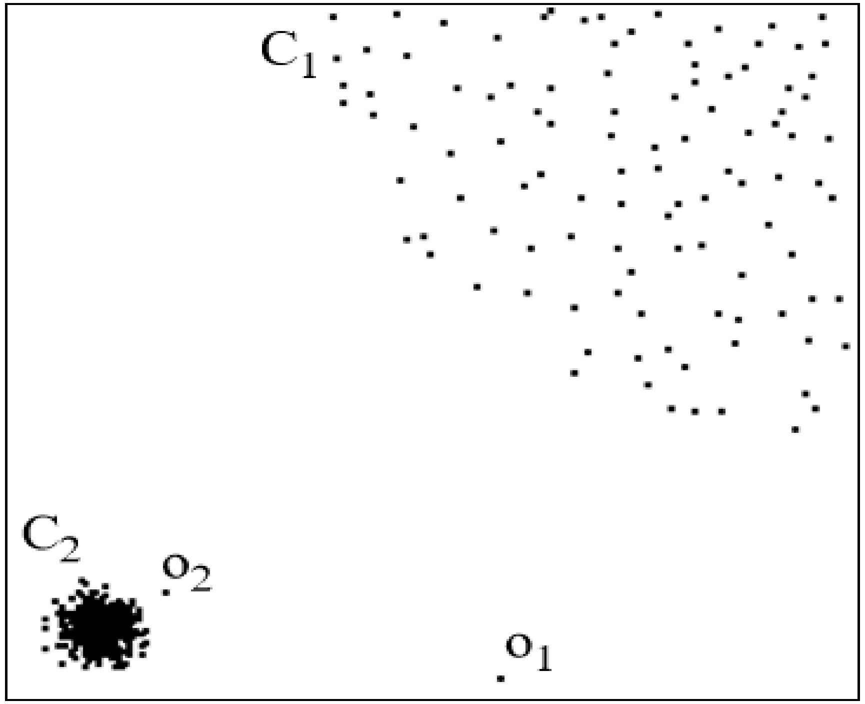
- As with density-based clustering, problem with differing densities
- Outlier o_2 has similar density as elements of cluster C1.
- Basic idea behind local distance-based methods:
  - Outlier o_2 is "relatively" far compared to its neighbors.

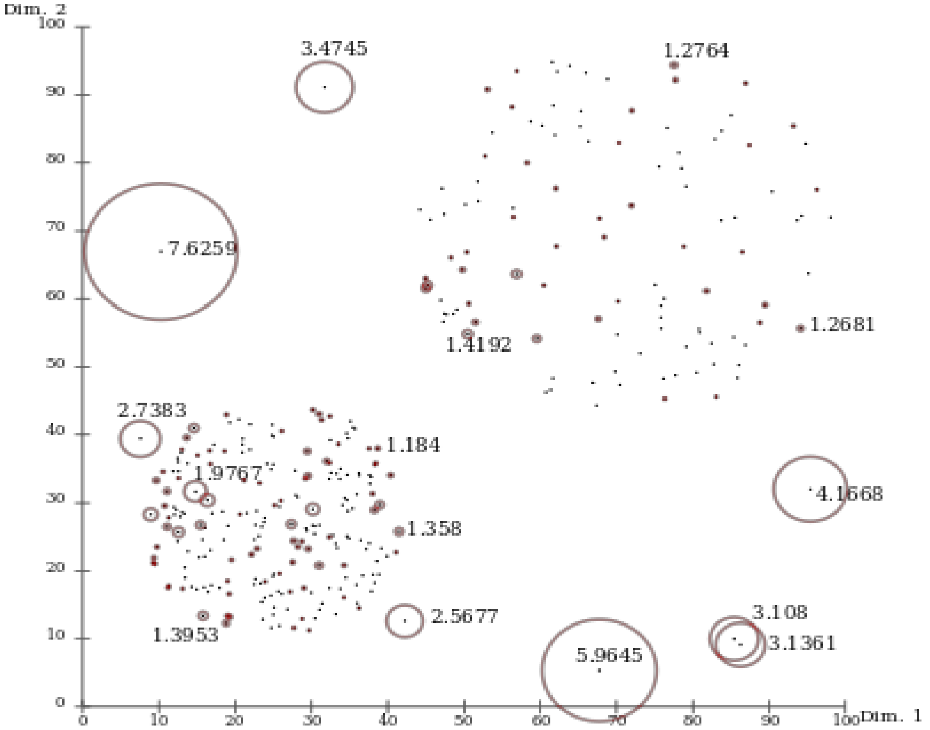


http://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf

Local Distance-Based Outlier Detection

- "Outlierness" ratio of example 'i':
  - (Avg distance of 'i' to its KNNs)/(Avg distance of neighbors of 'i' to their KNNs)
- If outlierness > 1, x is further away from neighbors than expected.

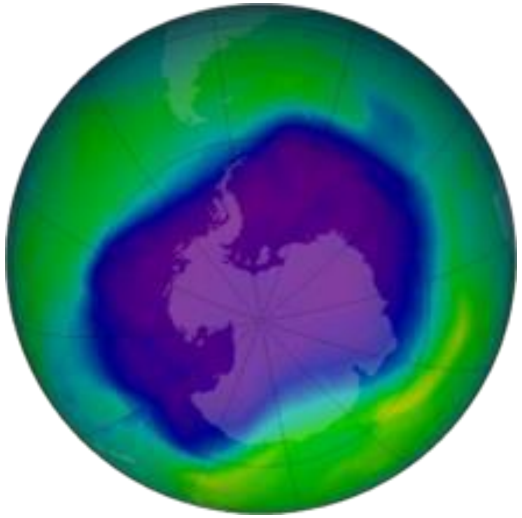




http://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf
https://en.wikipedia.org/wiki/Local_outlier_factor

Problem with Unsupervised Outlier Detection

- Why wasn’t the hole in the ozone layer discovered for 9 years?
- Can be hard to decide when to report an outlier.
  - If you report too many non-outliers, users will turn you off.



https://en.wikipedia.org/wiki/Ozone_depletion

Isolation Forests*

- Isolation Forests are a tree-based method.
- It uses a collection of these trees to calculate anomaly scores for each instance.
  - It has a linear time complexity.
  - It is unsupervised.
  - It requires large and high-dimensional data.
  - It does not perform well with small datasets.
  - It performs random partitioning.
  - It is scalable.
  - It does not make any assumption about a feature's distribution.

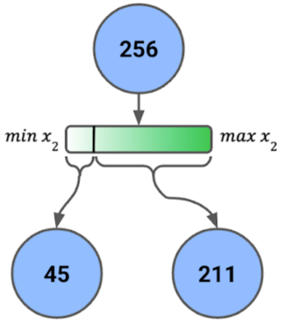
- Can detect anomalies without prior knowledge in large datasets.
- Tells us nothing about why certain instances are anomalies.

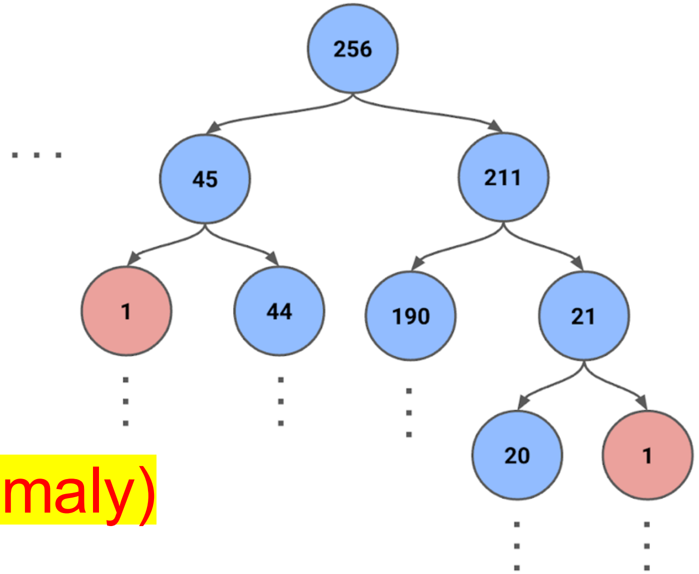
*Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua. "Isolation forest." Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.

Isolation Forests*

- Steps:
  1. Randomly select a feature.
  2. Randomly select a split value within the range of the selected feature for the instances currently in the node.
  3. Partition the data into two child nodes based on the split value.
  4. Repeat the process recursively for each child node until one of the following conditions is met:
     1. Each leaf node has only one instance.
     2. A predefined maximum depth is reached.

- Topic: Fraudulent transactions detection
- Instances: 256 transactions
- Var: amount (x1) and time of day (x2)
- Note: Feature’s range will change for each step.





Here, we use feature's minimum and maximum values for the node's instances.

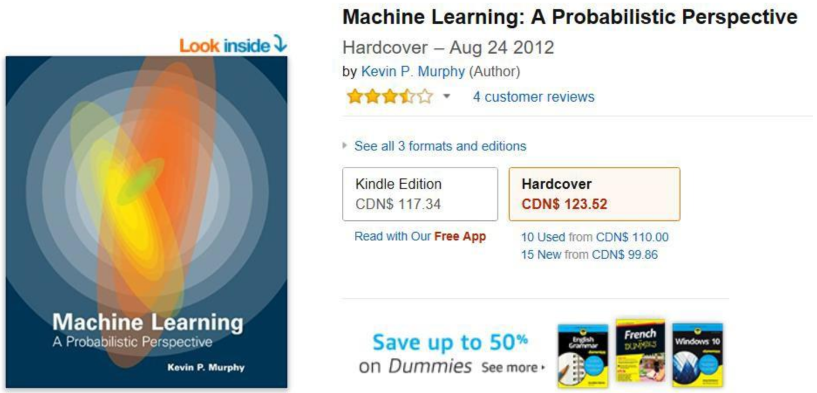
Score range: 0 (not an anomaly) and 1 (an anomaly)

Src Datacamp https://www.datacamp.com/tutorial/isolation-forest

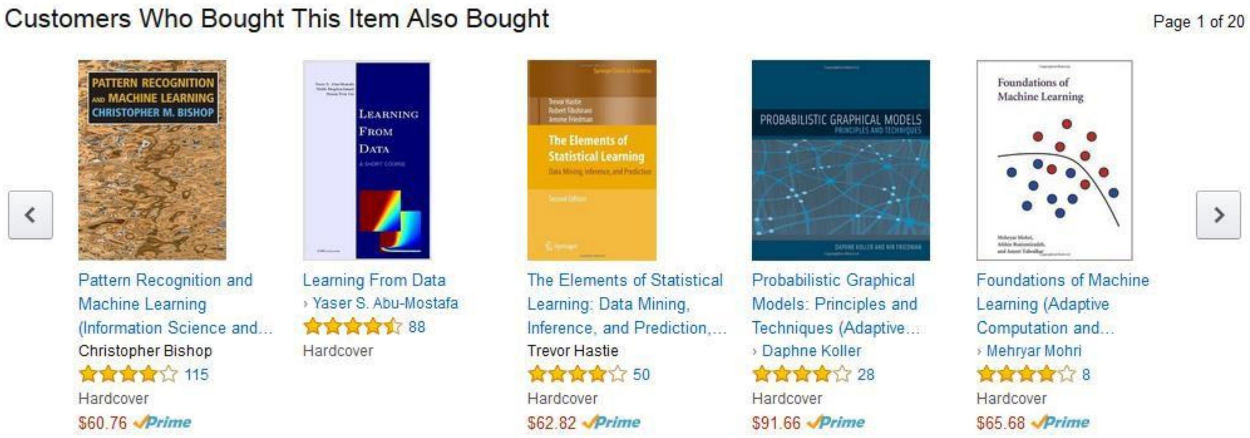
Finding Similar Items

Product Recommendation

- A customer comes to your website looking to buy at item:



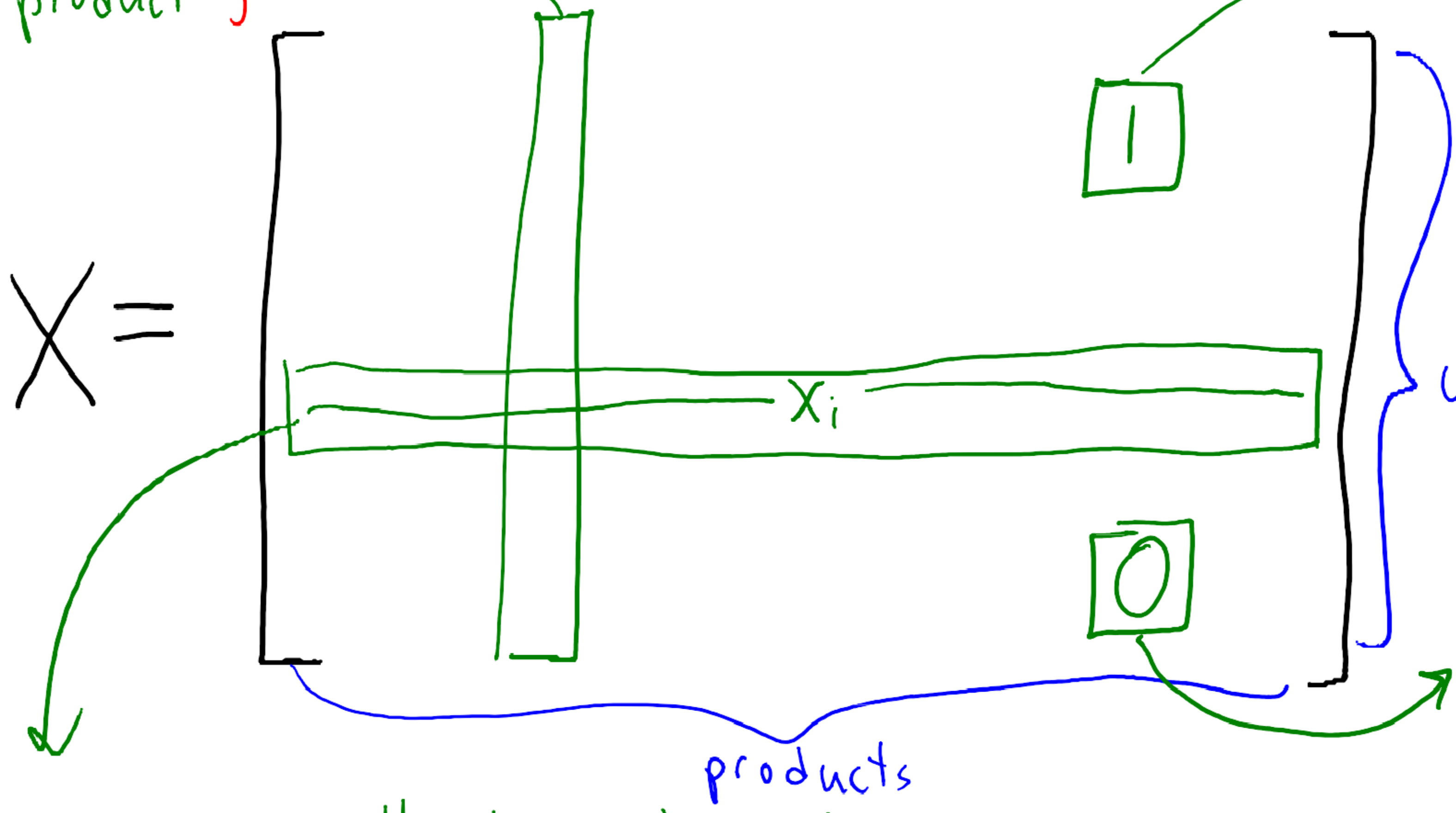
- You want to find similar items that they might also buy:



User-Product Matrix

Column x^j gives <u>all users</u> that bought product 'j'

$X_{ij} = 1$ means user 'i' bought item 'j'!

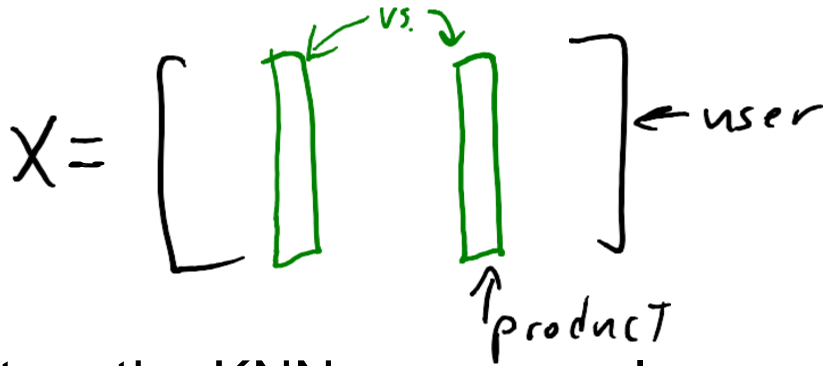


$x_{ij} = 0$ means user 'i' has <u>not</u> buy item 'j'

Row x_i gives <u>all items</u> bought by user 'i'.

Product Recommendation: Amazon

- Amazon product recommendation method:



- Return the KNNs across columns.
  - Find ‘j’ values minimizing $||x^i - x^j||$.
  - Products that were bought by similar users.

- But first divide each column by its norm, $x^i/||x^i||$.
  - This is called normalization.
  - Reflects whether product is bought by many people or few people.

Product Recommendation: Amazon

- Consider this user-item matrix:

$$X = \begin{array}{cc} & \begin{array}{cccccc} \text{Product 1} & \text{Product 2} & \text{Product 3} & \text{Product 4} & \text{Product 5} & \text{Product 6} \end{array} \\ \begin{array}{c} \text{John} \\ \text{Paul} \\ \text{George} \\ \text{Ringo} \\ \text{Yoko} \end{array} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

- Product 1 is most like Product 3 (bought by lots of people).
- Product 2 is most like Product 4 (also bought by John and Yoko).
- Product 3 is equally like Products 1, 5, and 6.
  - Does not consider that Product 1 is more popular than 5 and 6.

Product Recommendation: Amazon

- Consider this user-item matrix (normalized):

$$
X = \begin{array}{c} \\ \text{John} \\ \text{Paul} \\ \text{George} \\ \text{Ringo} \\ \text{Yoko} \end{array}
\begin{array}{c}
\begin{array}{cccccc} \text{Product 1} & \text{Product 2} & \text{Product 3} & \text{Product 4} & \text{Product 5} & \text{Product 6} \end{array} \\
\left[\begin{array}{cccccc}
1/\sqrt{5} & 1/\sqrt{2} & 1/\sqrt{4} & 1/\sqrt{2} & 0 & 1/\sqrt{3} \\
1/\sqrt{5} & 0 & 1/\sqrt{4} & 0 & 1/\sqrt{3} & 0 \\
1/\sqrt{5} & 0 & 1/\sqrt{4} & 0 & 1/\sqrt{3} & 1/\sqrt{3} \\
1/\sqrt{5} & 0 & 1/\sqrt{4} & 0 & 1/\sqrt{3} & 1/\sqrt{3} \\
1/\sqrt{5} & 1/\sqrt{2} & 0 & 1/\sqrt{2} & 0 & 0
\end{array}\right]
\end{array}
$$

- Product 1 is most like Product 3 (bought by lots of people).
- Product 2 is most like Product 4 (also bought by John and Yoko).
- Product 3 is most like Product 1.
  - Normalization means it prefers the popular items.

Cost of Finding Nearest Neighbors

- With ‘n’ users and ‘d’ products, finding KNNs cost is O(nd).
  - Not feasible if ‘n’ and ‘d’ are in the millions.
- It’s faster if the user-product matrix is sparse.
  - But data set is still enormous in the Amazon example.
- “Closest point” problem exists in:
  - KNN classification.
  - K-means clustering.
  - Density-based clustering.
  - Amazon product recommendation.

Recommender Systems

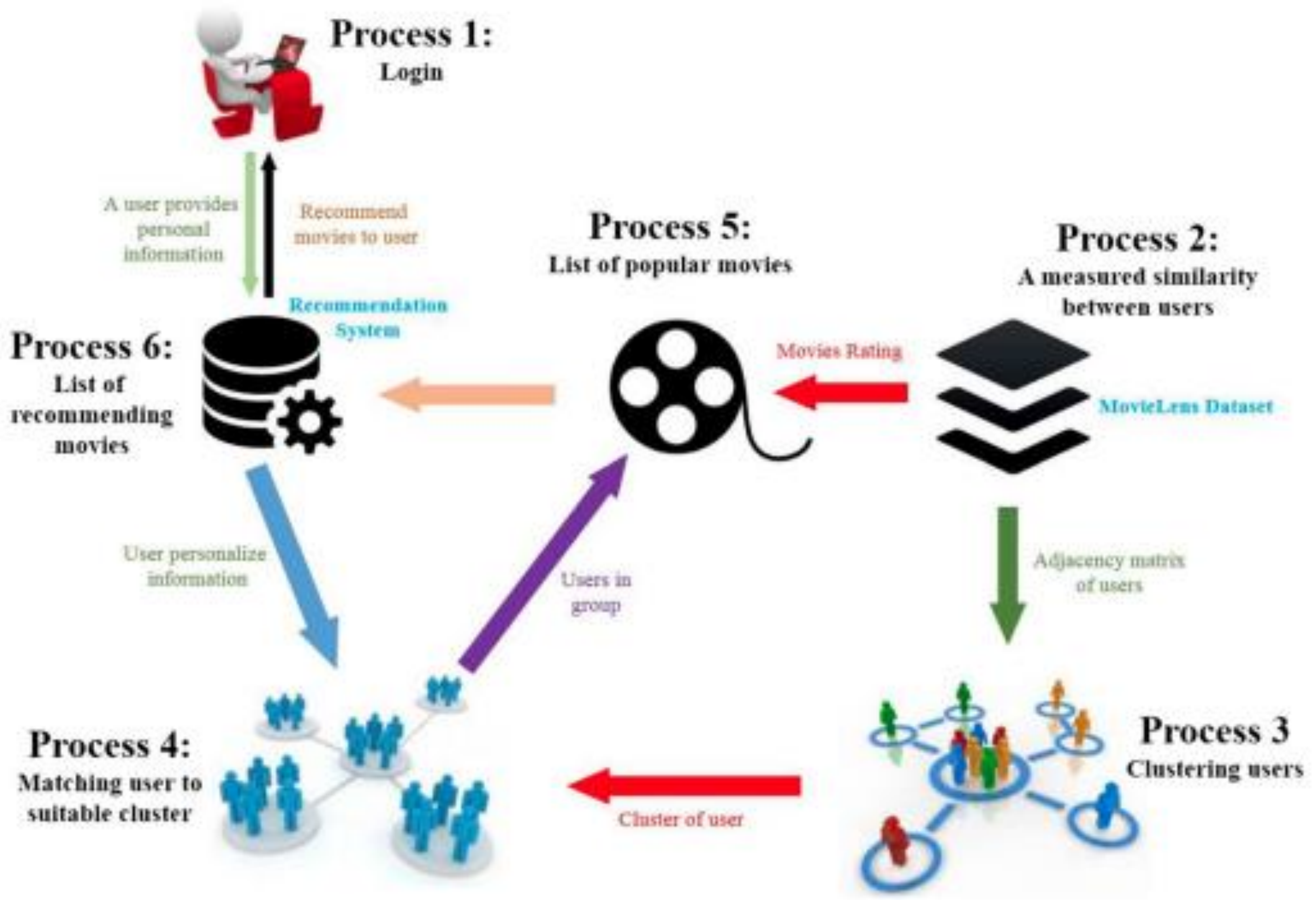
Recommender Systems

- Recommender Systems: maximize profit by recommendation
  - sells items --> collects ratings from users.

- Recommendation Scenarios:
  - Recommend items given an item: Amazon product recommendation.
  - Recommend items given a user: Amazon/Netflix homepage.
  - Or a combination (personalized item-based recommendation).

- Users only rate a small number of items (matrix is sparse)
  - Leads to *predicting missing ratings*.

Sample Recommendation System



Vilakone, P., Park, D. S., Xinchang, K., & Hao, F. (2018). An efficient movie recommendation algorithm based on improved k-clique. *Human-centric Computing and Information Sciences*, *8*(1), 1-15.

Types of Recommender Systems

- Two Types of Recommender Systems:
  - Content Filtering: Assumes access to side information about items
    - Example: Pandora

- Supervised learning: Extract features x_i of users and items, building model to predict rating y_i given x_i.
- Apply model to predict new users/items.
  - Example: Gmail's "important messages"

Types of Recommender Systems

- Collaborative Filtering: Does not assume access to side information about items
    - Example: Netflix
    - Personal tastes are correlated:
    - If Alice and Bob both like X and Alice likes Y then Bob is more likely to like Y.

- "Unsupervised" learning (have label matrix 'Y' but no features)
  - Have labels y_{ij} (rating of user 'i' for movie 'j').
  - Collaborative filtering does not predict well for new users/movies.

Collaborative Filtering Problem

- Collaborative filtering is ‘filling in’ the user-item matrix:

$$Y = \text{user}\left\{\underbrace{\begin{bmatrix} ? & 4 & 3 & 2 & 3 & 3 \\ 2 & 1 & ? & 5 & ? & 5 \\ ? & 1 & ? & 5 & 5 & 5 \\ \vdots & & \vdots & & & \\ 2 & 3 & 3 & ? & ? & ? \end{bmatrix}}_{\text{movie}}\right.$$

"Green Lantern"

"Ryan Reynolds"

- We have some ratings available with values {1,2,3,4,5}.
- We want to predict ratings “?” by looking at available ratings.

Collaborative Filtering Problem

$$Y = \text{user}\left\{\underbrace{\begin{bmatrix} ? & 4 & 3 & 2 & 3 & 3 \\ 2 & 1 & ? & 5 & ? & 5 \\ ? & 1 & ? & 5 & 5 & 5 \\ 2 & 3 & 3 & ? & ? & ? \end{bmatrix}}_{\text{movie}}\right.$$

"Green Lantern"

"Ryan Reynolds"

Similar user

Similar movie

- What rating would "Ryan Reynolds" give to "Green Lantern"?
  - Why is this not completely crazy?
  - We may have similar users and movies.

- Two Types of Collaborative filtering methods:
  - Neighborhood
  - Latent Factor

Neighborhood Method

- Algorithm:
  - Find neighbors based on similarity of movie preferences.
  - Recommend movies that those neighbors watched.

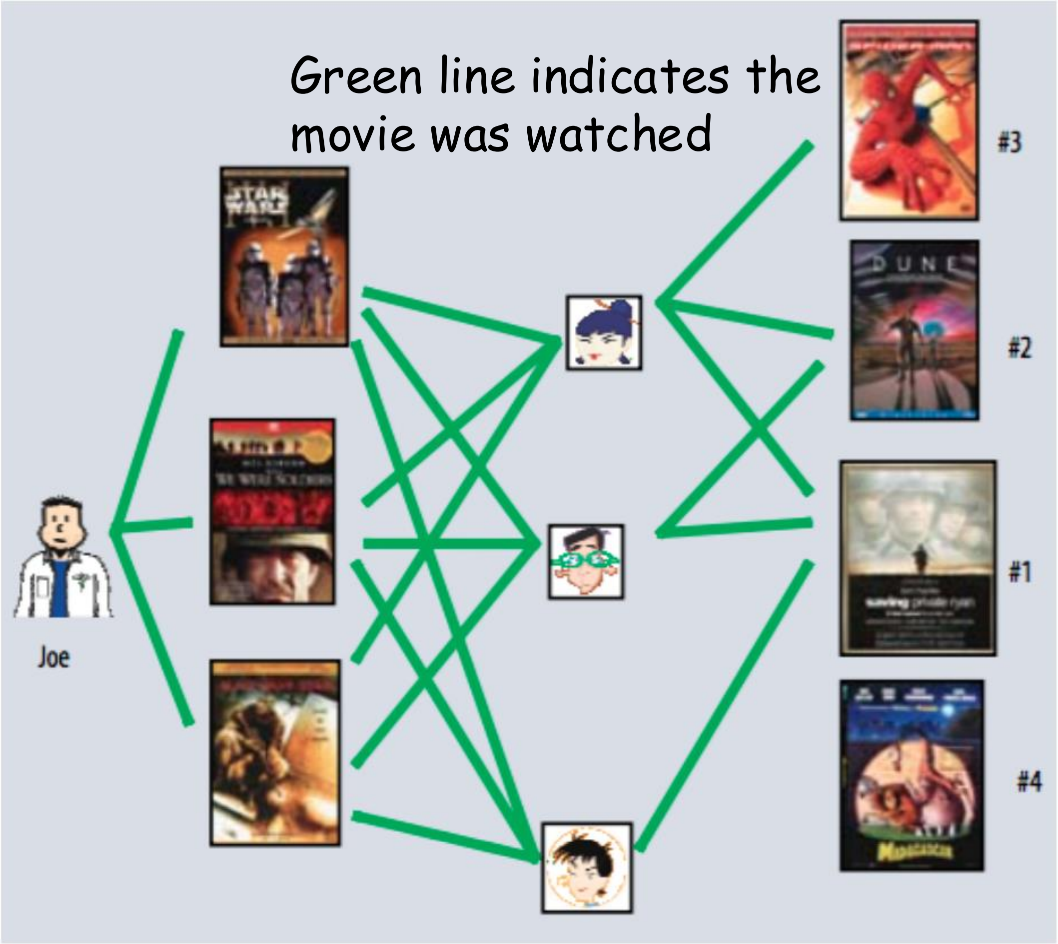


Figure from Koren et al. (2009)

Latent Factor Method

- Assume that both movies and users live in some low-dimensional space describing their properties.

- Recommend a movie based on its proximity to the user in the latent space (where features lie.).

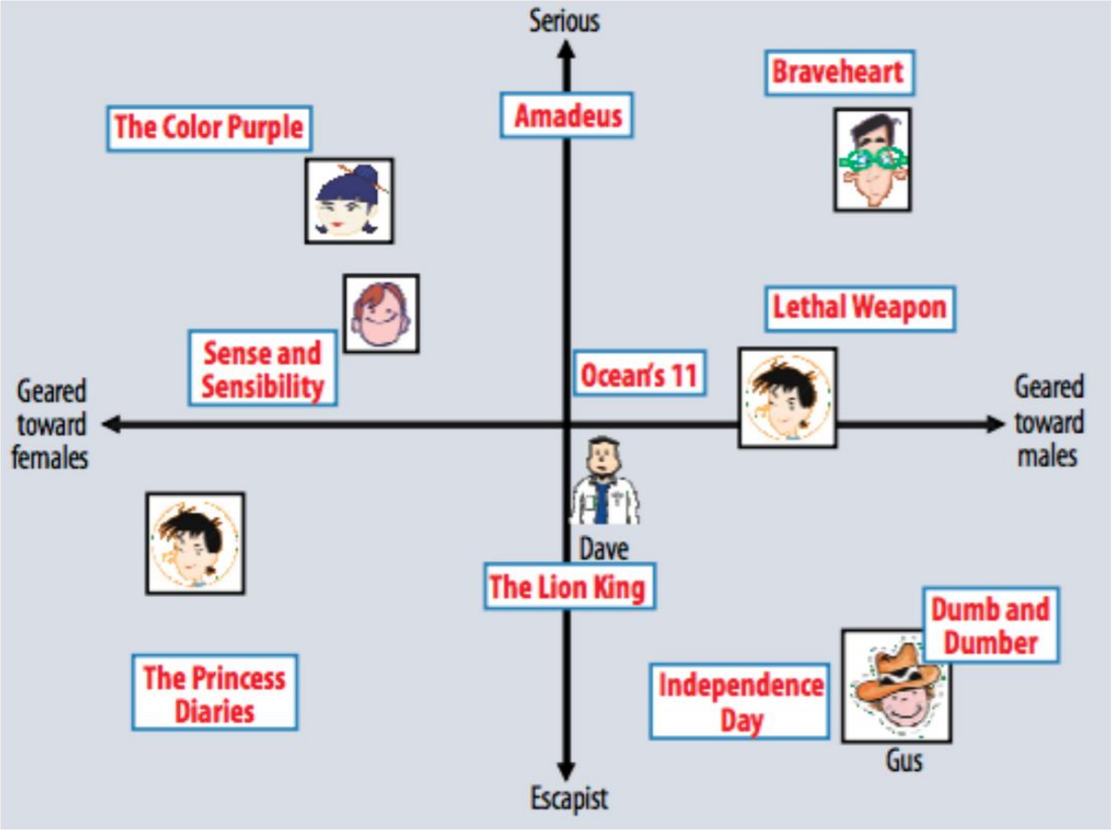


Figure from Koren et al. (2009)

Matrix Factorization (MF)

- Collaborative filtering by MF is an efficient and effective approach

- MF is a way to define model + objective function
    - Also optimizes with stochastic gradient descent.

- Classes of MF:
    - Unconstrained Matrix Factorization
    - Singular Value Decomposition/SVD
    - Non-negative Matrix Factorization
        - Facebook has a “Like” button, but no “Dislike” button
        - Pinterest pins

Beyond Accuracy in Recommender Systems

- Issues:
  - Diversity: How different are the recommendations?
  - Persistence: How long should recommendations last?
  - Trust: Tell user why you made a recommendation.
    - Quora gives explanations for recommendations.
  - Social recommendation: What did your friends watch?
  - Freshness: people tend to get more excited about new/surprising things.

Content-Based Recommender

```python
import pandas as pd

metadata = pd.read_csv('movies_metadata.csv', low_memory=False)

# Print the first three rows
metadata.head(3)
```

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_id	original_language	original_title	overview	...	release_d
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt0114709	en	Toy Story	Led by Woody, Andy's toys live happily in his ...	...	1995-10
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt0113497	en	Jumanji	When siblings Judy and Peter discover an encha...	...	1995-12
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collect...	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	NaN	15602	tt0113228	en	Grumpier Old Men	A family wedding reignites the ancient feud be...	...	1995-12

```python
# Calculate C
C = metadata['vote_average'].mean()

# Calculate the minimum number of votes required to be in the chart, m
m = metadata['vote_count'].quantile(0.90)

# Filter out all qualified movies into a new DataFrame
q_movies = metadata.copy().loc[metadata['vote_count'] >= m]
q_movies.shape
```

```
(4555, 24)
```

```python
# Function that computes the weighted rating of each movie
def weighted_rating(x, m=m, C=C):
    v = x['vote_count']
    R = x['vote_average']
    # Calculation based on the IMDB formula
    return (v/(v+m) * R) + (m/(m+v) * C)

# Define a new feature 'score' and calculate its value with `weighted_rating()`
q_movies['score'] = q_movies.apply(weighted_rating, axis=1)
#Sort movies based on score calculated above
q_movies = q_movies.sort_values('score', ascending=False)

#Print the top 15 movies
q_movies[['title', 'vote_count', 'vote_average', 'score']].head(15)
```

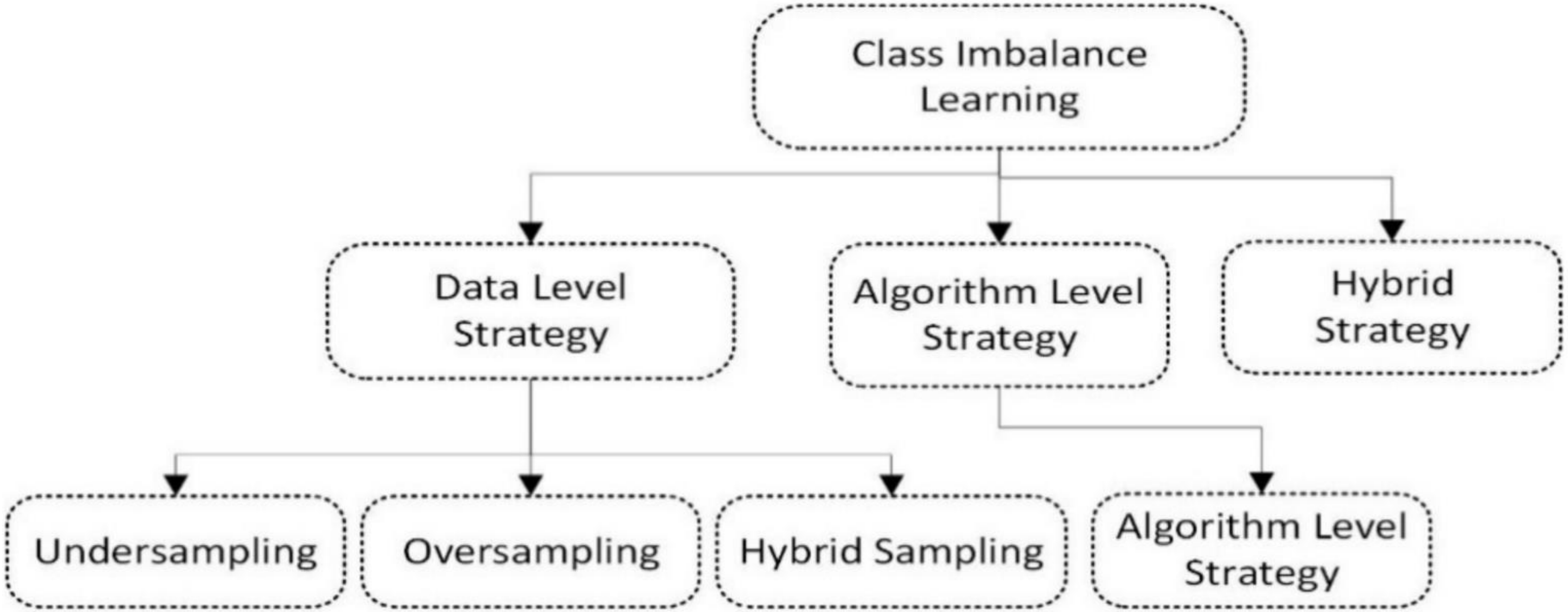
	title	vote_count	vote_average	score
314	The Shawshank Redemption	8358.0	8.5	8.445869
834	The Godfather	6024.0	8.5	8.425439
10309	Dilwale Dulhania Le Jayenge	661.0	9.1	8.421453
12481	The Dark Knight	12269.0	8.3	8.265477
2843	Fight Club	9678.0	8.3	8.256385
292	Pulp Fiction	8670.0	8.3	8.251406

```python
#Print plot overviews of the first 5 movies.
metadata['overview'].head(3)
```

```
0    Led by Woody, Andy's toys live happily in his ...
1    When siblings Judy and Peter discover an encha...
2    A family wedding reignites the ancient feud be...
Name: overview, dtype: object
```

Class Imbalance

Categorization of class imbalance



https://doi.org/10.3390/healthcare10071293

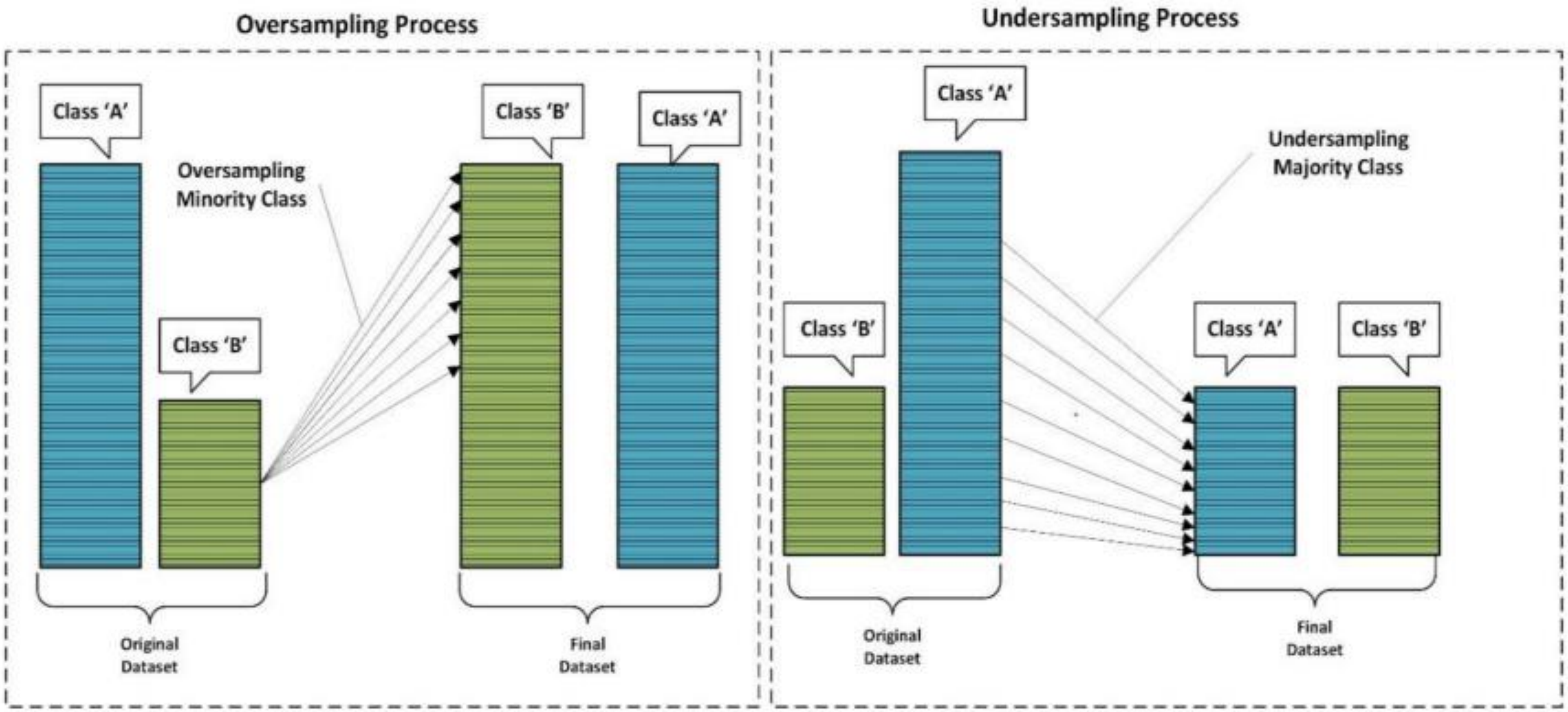
Sampling

- Sampling: A statistical process
  - Predetermined number of observations are taken from a larger population.

- Issues: Oversampling and Undersampling
  - Adjust class distribution of a data set.

- Oversampling techniques for classification problems: SMOTE, ADASYN
- Undersampling techniques for classification problems: Cluster

Undersampling and oversampling



https://doi.org/10.3390/healthcare10071293

Examples of Class Imbalance

SL	Name	Data Types	Default Task	Attribute Types	#Instances	Class Distribution	#Attributes	Imbalance Ratio
1	Breast Cancer	Multivariate	Classification	Categorical	286	**0**:201,**1**: 85	9	2.36
2	Breast Cancer Wisconsin (Original)	Multivariate	Classification	Integer	699	**0**: 458, **1**:241	10	1.9
3	Breast Cancer Wisconsin (Prognostic)	Multivariate	Classification, Regression	Real	198	**0**:151, **1**:47	34	3.21
4	Breast Cancer Wisconsin (Diagnostic)	Multivariate	Classification	Real	569	**0**:357, **1**:212	32	1.69
5	Heart Disease	Multivariate	Classification	Categorical, Integer, Real	303	**0**:164,**1**:55,**2**:36,**3**:35,**4**:13	75	–
6	Hepatitis	Multivariate	Classification	Categorical, Integer, Real	155	**0**:133, **1**:32	19	4.15
7	Pima Indians Diabetes Database	Multivariate	Classification	Integer	768	**0**: 500, **1**:268	8	1.9
8	Liver Disorders	Multivariate	Classification	Categorical, Integer, Real	345	**0**:145,**1**:200	7	1.37
9	Lung Cancer	Multivariate	Classification	Integer	32	**0**:23, **1**:9	56	2.55
10	SPECT Heart	Multivariate	Classification	Categorical	267	**0**:55,**1**: 212	22	3.85
11	SPECTF Heart	Multivariate	Classification	Integer	267	**0**:55,**1**:212	44	3.85
12	Thyroid Disease	Multivariate, Domain-Theory	Classification	Categorical, Real	7200	**1**:166, **2**:368, 3:6666	21	–
13	Breast Tissue	Multivariate	Classification	Real	106	**Car**:21 **Fad**:15 **Mas**:8, **Gla**:16, **Con**:14, **Adi**:22	10	–
14	Fertility	Multivariate	Classification, Regression	Real	100	**N**:88, **O**:12	10	7.33
15	Diabetic Retinopathy Debrecen Dataset	Multivariate	Classification	Integer, Real	1151	**0**:540, **1**:611	20	1.131

Imbalance Ratio = (No. of samples in Majority Class)/(No. of samples in Minority Class)

https://doi.org/10.3390/healthcare10071293

Degree of Class Imbalance

Class Imbalance Degree	Proportion of Minority Class
Extreme	<1% of the dataset
Moderate	1–20% of the dataset
Mild	20–40% of the dataset

Dataset	#Instances	#Attributes	Class	IR	Minority Class (%)	Degree of Imbalanced

https://doi.org/10.3390/healthcare10071293

Synthetic Minority Oversampling Technique (SMOTE)

- Primarily analyze under-represented class.
- Increase number of cases in dataset in a balanced way.
- Works by generating new instances from existing minority cases (input).

- Reasons of imbalanced: rare value, difficulty in collecting data.

- Oversamples minority class by practicing each minority class sample and including synthetic examples along line segments joining any/all of k minority class nearest neighbors.

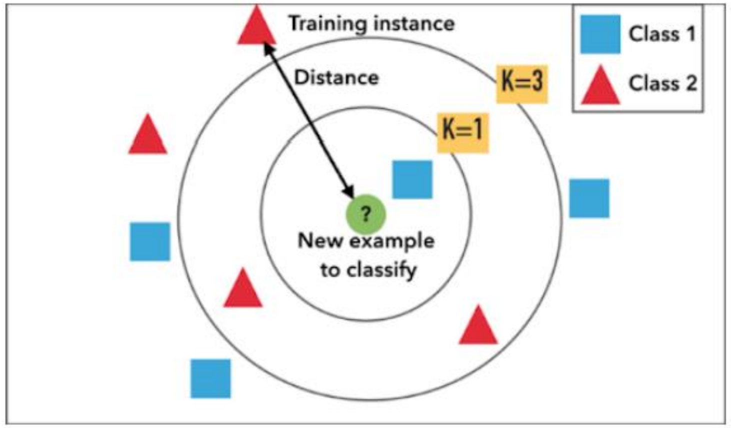
SMOTE

- It takes samples of *feature space* for each target class and its nearest neighbors.
- Then generates new examples that combine features of target case with features of its neighbors.
- It increases features available to each class and makes the samples more general.
- SMOTE takes entire dataset as an input, but it increases percentage of only minority cases.

SMOTE

- Neighbors from k-nearest neighbors are taken randomly (depends on required oversampling)

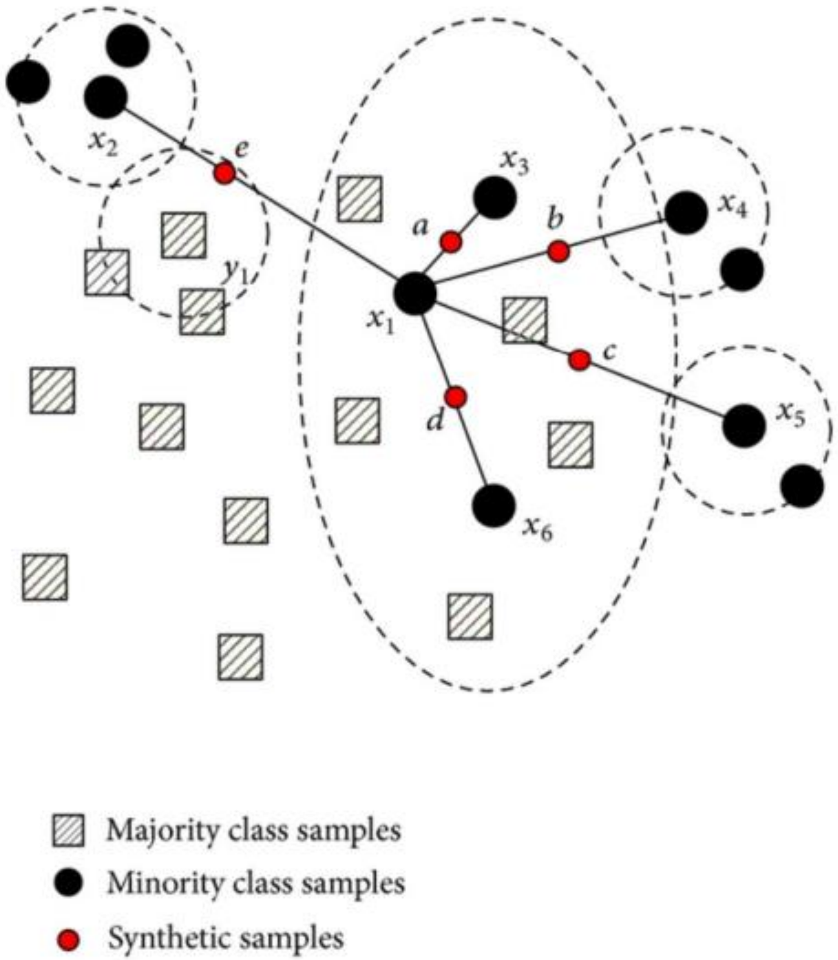
- SMOTE algorithm:
  - Find k-nearest neighbors for each sample.
  - Select samples randomly from a k-nearest neighbor.
  - Find new samples = {original samples + difference * gap (0,1)}.
  - Add new samples to minority.
  - Finally, a new dataset is created.



Hussein, A. S., Li, T., Yohannese, C. W., & Bashir, K. (2019). A-SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE. *International Journal of Computational Intelligence Systems*, *12*(2), 1412-1422.

SMOTE

- SMOTE tries to avoid the risk of overfitting.
  1. First, it chooses a random minority observation x_i
  2. Next, randomly selects an instance x_u among k nearest minority class neighbours of x_i.
  3. Finally, a new random sample s_i is generated by interpolating the two samples: $s_i = x_i + w \times (x_u - xi)$, where w is a random weight in [0, 1].
- Does not specifically enforce decision boundary.



https://doi.org/10.48550/arXiv.2012.15231

SMOTE Example

- In an imbalanced dataset where just 1% of cases have target value A (minority class), and 99% of cases have value B.
  - To increase % of minority cases to 2x the previous percentage, you would enter **200** for **SMOTE percentage** in the component's properties.

	Class 0	**Class 1**	**total**
Original dataset (equivalent to **SMOTE percentage = 0**)	570 76%	178 24%	748
SMOTE percentage = 100	570 62%	356 38%	926
SMOTE percentage = 200	570 52%	534 48%	1,104
SMOTE percentage = 300	570 44%	712 56%	1,282

ADASYN

ADASYN (extension of SMOTE)

- ADASYN: Adaptive Synthetic Sampling

- SMOTE generates an arbitrary number of synthetic minority examples to shift the classifier learning bias toward the minority class.

- ADASYN improve class balance by synthetically creating new examples from minority class via linear interpolation between existing minority class.

- Creates more examples in vicinity of boundary between two classes than in interior of minority class.

He, Haibo, et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008.

ADASYN (extension of SMOTE)

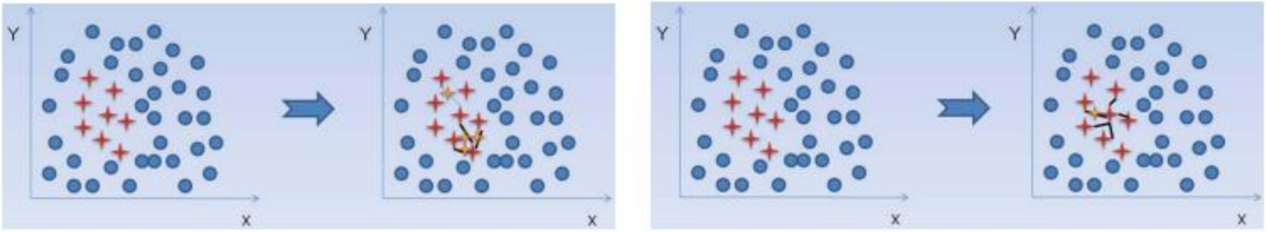
- Shift classifier decision boundary to be more focused on those difficult to learn.
  - Improves learning performance.

- ADASYN adaptively generate synthetic data samples for minority class to reduce bias introduced by imbalanced data distribution.
  - Dynamic adjustment of weights
  - Adaptive learning procedure

- Use a density distribution as a criterion to decide number of synthetic samples that need to be generated for each minority data example.

ADASYN (extension of SMOTE)

- Sample generation technique is similar to SMOTE
  - But this time for each minority class data example x_i, $g_i = r_i \times g$ interpolated samples are generated
    - where g is the total number of synthetic data samples needed
    - r_i is proportional to the number of majority class samples between the k nearest neighbours of x_i
      - It is highest near the class boundaries.



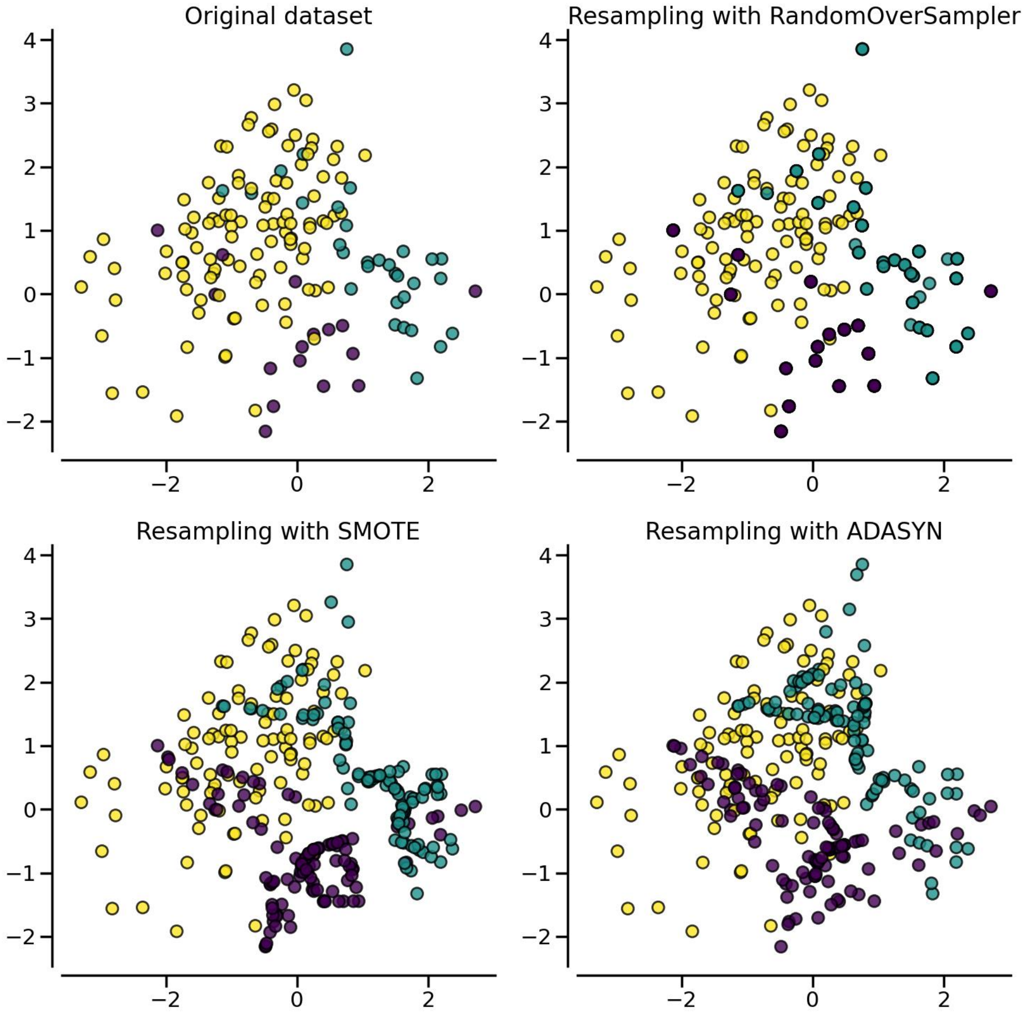
(a) ADASYN (b) SMOTE

https://doi.org/10.48550/arXiv.2012.15231

SMOTE vs ADASYN

```
from imblearn import FunctionSampler  # to use a idendity sampler
from imblearn.over_sampling import ADASYN, SMOTE

X, y = create_dataset(n_samples=150, weights=(0.1, 0.2, 0.7))
fig, axs = plt.subplots(nrows=2, ncols=2, figsize=(15, 15))
samplers = [
    FunctionSampler(),
    RandomOverSampler(random_state=0),
    SMOTE(random_state=0),
    ADASYN(random_state=0),
]
for ax, sampler in zip(axs.ravel(), samplers):
    title = "Original dataset" if isinstance(sampler, FunctionSampler) else None
    plot_resampling(X, y, sampler, ax, title=title)
fig.tight_layout()
```



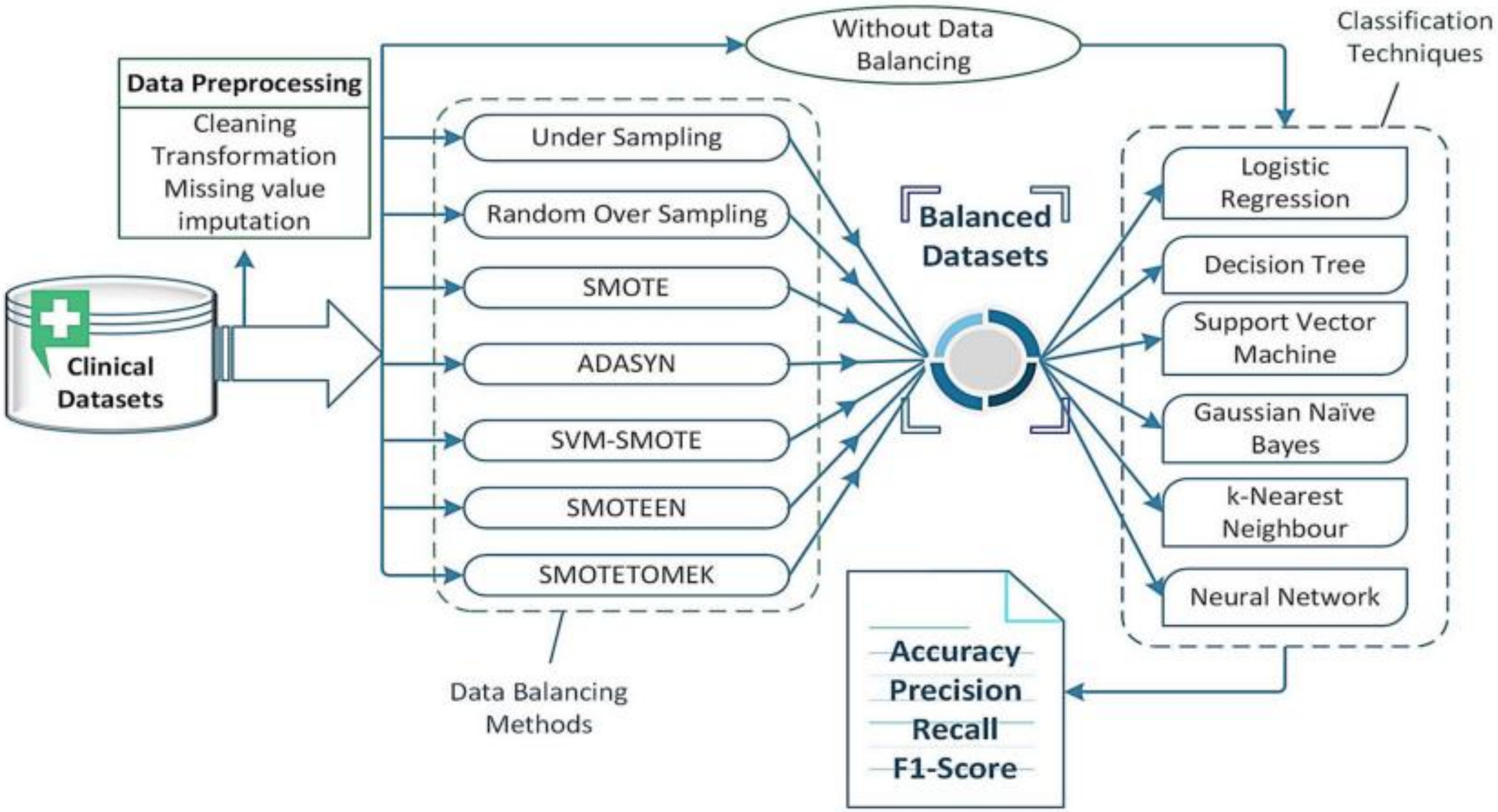
- SMOTE will not make any distinction.
- Focus on samples which are difficult to classify with a nearest-neighbors rule.

https://imbalanced-learn.org/stable/auto_examples/over-sampling/plot_comparison_over_sampling.html#sphx-glr-auto-examples-over-sampling-plot-comparison-over-sampling-py

Disadvantage

Assume that space between any two minority class samples belongs to minority class, which may not be always true when data is not linearly separable.

Evaluation of classifiers using SMOTE/ADASYN



https://doi.org/10.3390/healthcare10071293