# Trade-offs of Bias with other verticals in Trustworthy AI Cheatsheet

January 2023

# Part I - Regression and Multi-class

## 1 Introduction

### 1.1 Regression

Regression is used to determine the relationship between a dataset's dependent (goal) and independent variables (Y and X respectively). There are many types of regression, including:

- **Linear Regression**. assumes a linear relationship between the dependent variable (Y) and an independent variable (X). Y and X can both be multidimensional.
- **Logistic Regression**. When the independent variable is discrete, We consider this classification in this course, although it uses linear regression as an intermediate step.
- **Polynomial Regression**. Same as linear regression but with a non-linear relationship.

In this class, we consider only Linear Regression.

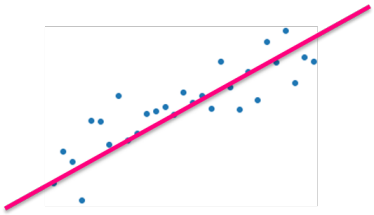


Figure 1: An example of linear regression with x & y in $R$

### 1.2 Multi-class classification

This is similar to binary classification, but with more than two output classes. All binary classification methods can be used using the **one-vs-rest** or **one-vs-one** approach. In the **one-vs-rest** approach, we fit a binary classifier model for each class, where one of the output is the class, and the other the aggregate of all other classes. In the **one-vs-one**, we fit a binary classifier for each pait of class. Then in both case, we combine the results.

Other methods that can be adapted directly to multi-class classifications are: Neural networks, k-nearest neighbours, Naive Bayes, Decision trees, ...

## 2 Explainability

### 2.1 Overview

Explainability methods range from local to global, and from model specific to model agnostic. We give an overview of them with the below figure taken from Koshiyama et al.,2021.

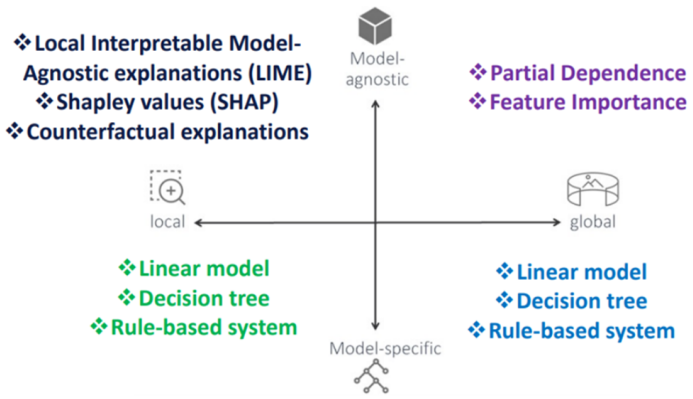



Figure 2: Explainability methods

For details of permutation feature importance, SHAP and LIME, please refer to our first Turing course (Assessing and Mitigating Bias and Discrimination in AI). In this section, we will look at how methods for binary classification differs from regression and multi-class classification, through the example of SHAP.

### 2.2 Motivation

We give two examples of where Explainability is particularly desirable for regression and multi-class classification.

- **Health premium regression**. Predict Health Insurance Premiums using regression. See this paper
- **Cancer diagnosis**. Multiclass cancer diagnosis using tumor gene expression signatures. See this paper.

## 2.3 From binary classification to regression and multi-class

Most explainability methods used in a binary classification context can be used in both a regression and multi-class classification context.

**From binary classification to regression**
As illustrated in figure below, all black-box explainability methods operate with the same logic:

1. Measure a base outcome, such as the global error for permutation feature importance, or the actual prediction for the shapley values.
2. Perturbate the model/feature.
3. Measure the new value of the variable of interest after perturbation.
4. Compare and infer feature importance.

This approach is directly translatable to a regression setting.

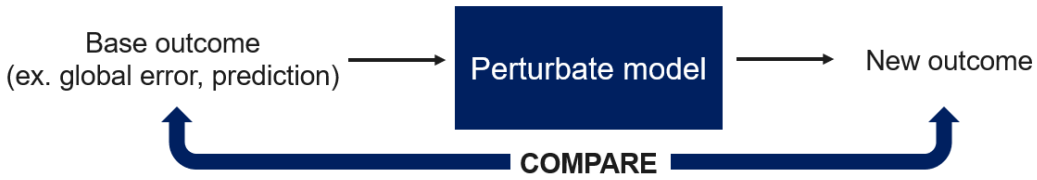



Figure 3: Similarity between binary classification and regression explainability

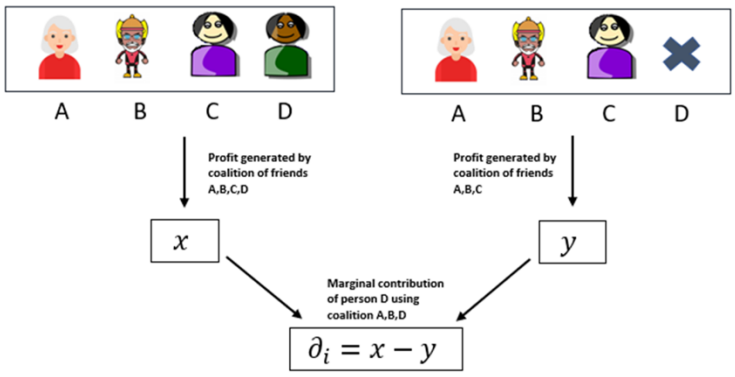
**From binary classification to multi-class**
Binary classification explainability techniques can be used once again by decomposing the multiclass problem in multiple binary one, again using a one-vs-one or one-vs-rest approach.

## 2.4 An example with Shapley values

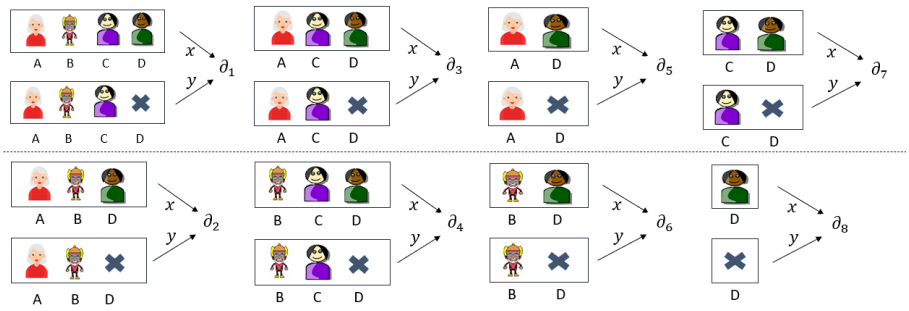
### 2.4.1 Shapley values for Regression

For regression, we can use the Shapley values exactly the same way as in the binary case. In a sentence, the shapley value is: "the average marginal contribution of an instance of a feature among all possible coalitions". It is a concept taken from Game Theory. We will quickly illustrate its principle here. For more detailed information, you can refer to this page which is a section of the book Interpretable Machine Learning. A Guide for Making Black Box Models Explainable by Christoph Molnar.

Imagine we have 4 friends (A,B,C,D) working together for a profit. How can we decide what contribution each person had in order to share the profit fairly? We can see the marginal contribution of person D by calculating the profit with and without them as shown in the figure below.

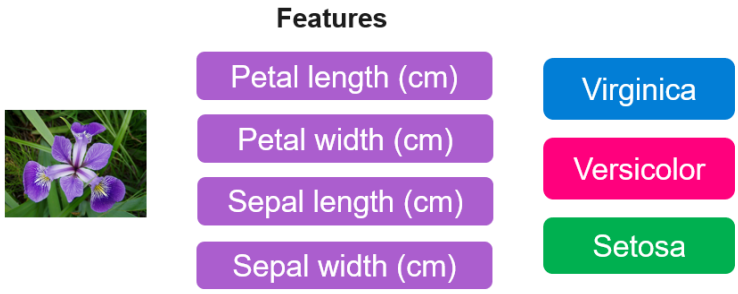



We can do this for "all possible coalitions" as pictured below.




### 2.4.2 Shapley values for Mutliclass

For multi-class, imagine we now have the iris classificaion problem where we look to classify Iris flowers in different type based on four features.




For each class, we will do the shapley values as before but considering a one-vs-rest problem:

1. Select a class, for instance "Setosa"
2. Consider the labels "Setosa" (label 1) or "not Setosa" (label 0)
3. Perform "classic" Shapley value calculation in this new binary context.
4. Repeat for all classes

### 2.5 Interaction with fairness

The interactions between Explainability and Fairness are very much the same as the ones detailed in the previous Turing course. As Explainability gives clarity on the factors that led to a decision, it seems intuitive that it can only help to determine if a decision is fair. As an illustration, the following questions can be answered using Explainability techniques:

- Are the most influential factors reasonable? Are they proxy for a protected characteristics?
- Is the model relying too much on one feature?
- In addition, explainability techniques can be used to explain directly the chosen fairness metric rather than the accuracy. For example, we can use permutation feature importance to assess the impact of each feature on Disparate Impact, hence unveiling which features are the most responsible for any potential observed bias.

## 3 Robustness

### 3.1 Overview

According to the European Comission AI ethics guidelines for Trustworthy AI, Robustness and Safety comprises the 4 following subrequirements:

- Resilience to attack and security.

- Fallback plan and general safety.
- Accuracy
- Reliability and Reproducibility

Refer to our first Turing course (Assessing and Mitigating Bias and Discrimination in AI) for a description of what each of these entails. In practice however, most academic papers looking at Robustness care about the following:

- Resistance to outliers
- Does small change in input result in small change in outputs?

## 3.2 Motivation

If a regression/classification output changes significantly when introducing an outlier, this could have very damaging consequences. If we take again the example of the regression used to predict health insurance premium, an outlier could skew the regression and make everyone pay more than what they would have without the outlier.

## 3.3 Example techniques

### 3.3.1 Changing the loss function

In regression, one of the most common objective function used to fit the model parameters to the training data is the least square error.

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where $y_i$ is the true outcome and $\hat{y}_i$ is the predicted outcome. This is particularly sensitive to outliers because the distance between a point and its prediction is squared, hence distant (i.e. outlier) points will weigh a lot and skew significantly the model.

One common strategy to reduce the impact of outliers is to use a L1-norm instead, the Least Absolute Deviation:

$$\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Although outliers that are far enough can still have an unbounded effect on the model fitting.

### 3.3.2 Random Sample Consensus (RANSAC)

RANSAC works as follow:

- Subset data randomly (minimum number of points to find parameters)
- Fit model on subset
- Remaining data points are classified as inliers or outliers depending on threshold distance
- Select highest scoring models and keep inliers

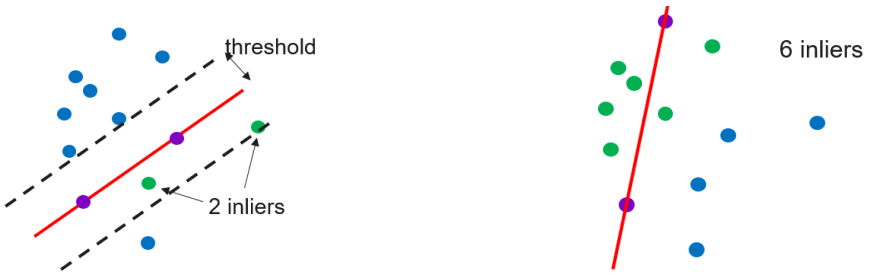



Figure 4: Two RANSAC iterations

### 3.3.3 Distributionnally Robust Optimisation (DRO)

This technique works for any supervised algorithm, and works particularly well when the data-generating distribution $P$ is not representative of the population on which the model will be used. It is an alternative way of performing optimisation than Empirical Risk Minimzation (ERM).

**Empirical Risk Minimization (ERM)**
This is the classic way of doing optimisation, i.e. minimizing a loss function over numerous training samples so we get the best model fit. In this scenario, we have a dataset $X$ composed of $n$ datapoints, and we have the loss $\ell(\theta, X)$.

We assume that $X$ was generated by a probability distribution $P$. Usually, we assume $P$ to be given by the empirical distribution resulting from the observed sample, and when doing the average, each sample is given equal weight. Therefore, we usually minimize the following:

$$\min_{\theta} \mathbb{E}_P(\ell(\theta, X)) = \min_{\theta}(\frac{1}{n}\sum_{i=1}^{n} \ell(\theta, x_i))$$

**Distributionnally Robust Optimisation (DRO)**
In Distributionnally Robust Optimisation, we look at the expected loss over

several "plausible" distribution. For instance, one way of finding the plausible distributions are to consider all distributions that are at a close enough Wasserstein distance to the empirical distribution mentioned above. If we call $\mathbb{Q}$ this ensemble of distribution, we will this time look at the distribution that gives us the worst expected loss over all samples, and then minimize the loss for that distribution:

$$\min_{\theta} \sup_{Q \in \mathbb{Q}} \mathbb{E}_Q(\ell(\theta, X)))$$

### 3.4 Interaction with fairness

If we think of Robustness as accuracy, there is always the known accuracy vs fairness trade-off, where adding fairness constraints on the model may have a negative impact on accuracy. The figure below illustrates this trade-off.

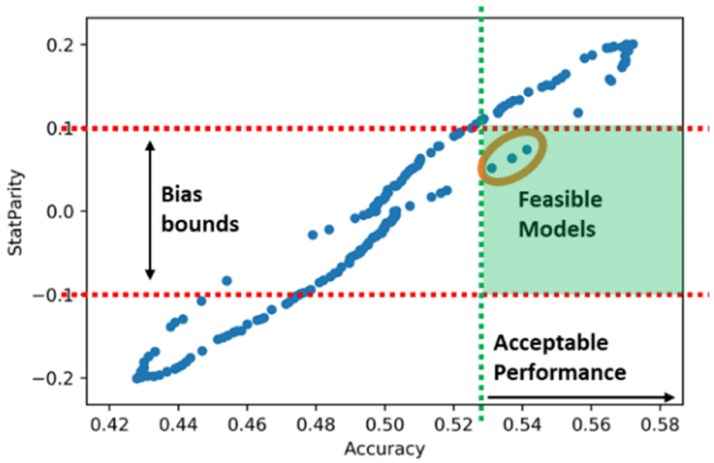



Figure 5: Accuracy vs Fairness

The blue dots correspond to various trained model with different parameters and initial condition, and how hey perform regarding accuracy (x axis) and fairness (y axis). The feasible models will be the one with enough accuracy, and within our fairness boundaries.

We also look at a paper that looks at the trade-off between robustness to adversarial attacks and fairness. The paper To be Robust or to be Fair: Towards Fairness in Adversarial Training, by Xu et al. claims that adversarial training algorithms tend to introduce severe disparity of accuracy and robustness between different groups of data.

# 4 Privacy

## 4.1 Overview

We have looked at many different attacks and defense in our first Turing course (Assessing and Mitigating Bias and Discrimination in AI). For a quick reminder, we can classify different defense depending on the attacker access (see figure below).

- Prevent access to data using access control, hardware/software security, cryptography, etc.
- Minimize the leaked information if attacker gains access to data: synthetic data, data minimisation and anonymisation, etc.
- Prevent the attacker to learn about the data through access to the model (black-box, or sometimes white-box).

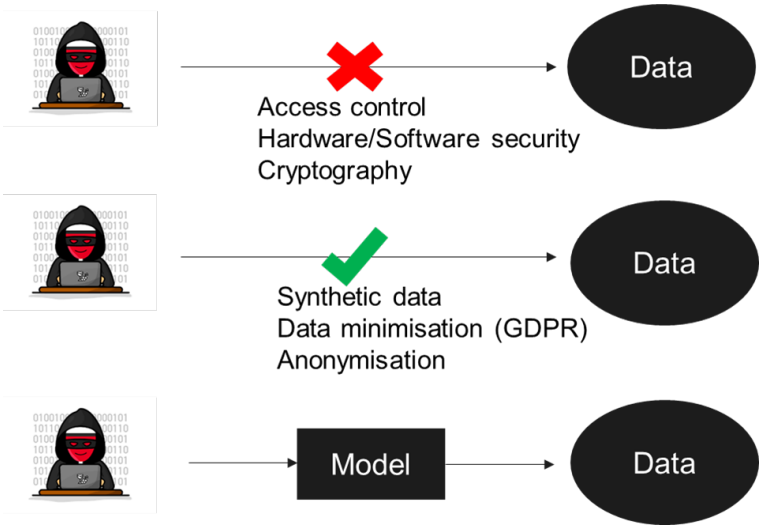



Figure 6: Attacker access

In this course, we focus on Membership Inference and Model extraction, which are two attacks from the third category that can be used both in classification and regression context. As a reminder, membership inference is when an attacker tries to infer whether an individual's data has been used in training. Model extraction is when an attacker tries to create a mock-up model by querying the models for various inputs.

## 4.2 Motivation

- **Membership inference.** Whether an individual is used in the training data or not can be sensitive information. The typical example is when training is in the context of a medical study, for instance about Alzheimer disease. If hospital records are used to train a model for such a study, one could potentially infer if a particular patient has been used in the study, divulging that the patient may have dementia.
- **Model extraction.** If able to query the model a lot, one can create a mock up model, which can then be used for other adversarial attacks.

## 4.3 Which model features favour which attacks?

Shokri et al, 2017 is a good resource for a survey of privacy attacks as well as their causes. Many of the resources cited below are summarised in their paper.

### 4.3.1 Membership inference

The more the model "holds" information about the training data, the more likely it is an attacker could infer some information about the training data. This means that the potential for membership inference is strongly positively correlated with overfitting of the model. This is also correlated with generalisation error: the higher it is, the higher the success of the attack.

Truex et al,2021 ran various experiments on how model type as well as the the number of classes impacted membership inference. They found that the more classes there are in multi-class classification models, the more signals about the internal state of the model are available to the attacker. They also found that some model types such as Naive Bayes are less susceptible to membership inference attacks than decision trees or neural networks.

### 4.3.2 Model extraction

The situation is here opposite than for membership inference. The more complex the model is, the less easy it is to reconstruct. Some model types such as linear regression or classifiers are easier to "reverse engineer" than for instance deep neural networks. Here, overfitting prevents the attack, and higher number of classes may lead to worse attack performance (Liu et al, 2021).

## 4.4 Interaction with fairness

As noted in our previous Turing course, sensitive data such as sex, gender, religion, ethnicity strongly overlaps with information required to measure/mitigate group fairness. There is a clear trade-off between the two in that sense.

**Bonus:** We look here at the interaction of privacy and robustness. Raghunatha et al, 2019 showed that adversarial training made models more susceptible to membership inference attacks as it increases the generalization error.

# Part II - Clustering

# 5 Introduction

As a reminder, clustering is an unsupervised learning method that consists in grouping unlabelled data points together in clusters, such that data points in the same groups are more "similar" to each other than to those in other groups. There are tow main categories of clustering algorithms:

- **Partitional.** Determine all the groups at once (e.g. k-means algorithm).
- **Hierarchichal.** Successively identify groups from a bottom-up (agglomerative) or top-down (divisive) approach.
  - **Agglomerative.** Each data point starts in its own cluster and clusters are successively joined together until we obtain the desired level of clustering.
  - **Divisive.** All data points are in one big cluster that is successively divided in smaller clusters.

# 6 Explainability

## 6.1 Overview

There are two types of explainability in clustering.

- **By design.** The clusters are built using tree-like algorithms which are interpretable. Such methods include DReaM, CLTrees and CUBT.
- **Post-processing explainability.** The clusters are built using any clustering method, and then used to fit a supervised model which is explainable (such as an interpretable tree or using another "classical" explainability method such as SHAP). Methods using trees include IMM and ExKMC

## 6.2 Motivation

We give two examples demonstrating why explainaility in clustering is important.

- Clustering candidates. How can it explain a candidate belonging to a particular cluster? This is particularly important as wrongly clustering a candidate could cost them a job if they were grouped with "less qualified" candidates.
- Clustering patients based on their scans from medical imaging techniques in order to identify cancer, tumors or other conditions. This is obviously important as a wrong diagnosis could have significant consequence on the health of the patient.

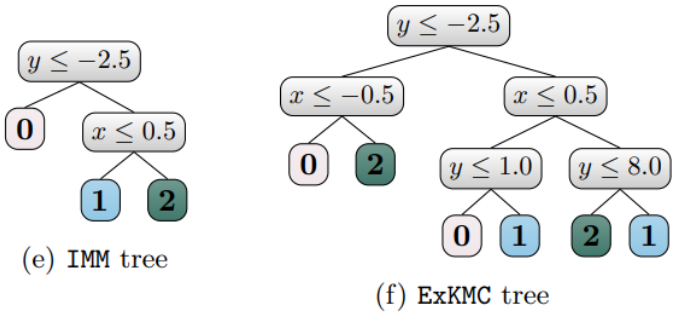
## 6.3 Methods

### 6.3.1 IMM and ExKMC

IMM was introduced by Moshkovitz,Dasgupta et al. in 2020. ExKMC was introduced as an extension to IMM by Frost, Moshkovitz et al. also in 2020. Both methods function with this basic principle:

1. Finding a clustering solution using some non-explainable clustering algorithm (like K-means)
2. Labelling each example according to its cluster
3. Calling a supervised algorithm that learns a decision tree

The only difference is that in IMM, the tree has the same number of leaves as the number of clusters $k$, whereas in ExKMC, there is an additional parameter $k'$ which represents the number of leaves of the tree, and which can be equal or greater than $k$.




(e) `IMM` tree

(f) `ExKMC` tree

Figure 7: Tree size (explanation complexity) vs. k-means clustering quality(from ExKMC)

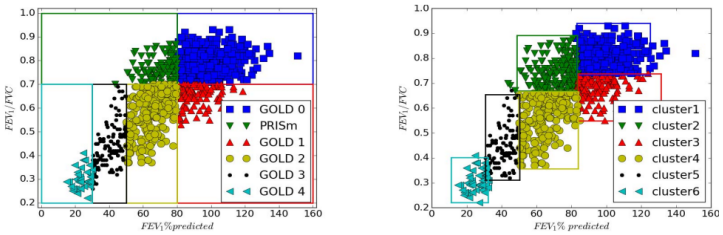
### 6.3.2 Discriminative Rectangle Mixture (DReaM)

DReaM is an interpretable clustering method that incorporate prior domain knowledge. The method splits the feature set in:

- Rule-generating features. Used in combination with informative prior distributions (domain-knowledge) to define soft thresholds to define cluster boundaries.
- Cluster-preserving features. clusters structure's preservation.

We will walk through it using an example from the orignal paper, for clustering Chronic Obstructive Pulmonary Disease (COPD) patient in various severity stages.

The Global initiative for Chronic obstructive Lung Disease (GOLD) proposes to classify the patients with COPD into stages according to spirometry, which is a measure of lung function. Patients are classified into mild, moderate, severe, or very severe airflow limitation using a set of thresholding rules on two variables: the FEV1 (forced expiratory volume in 1 second, % of predicted normal) and the FEV1 to forced vital capacity ratio. The rules used are illustrated by rectangles on the left of Figure 7.




(a) Applying the GOLD criterion (b) Applying rules discovered

Figure 8: Applying GOLD criterion and the rules discovered to the COPD test data. The rectangles represent the decision rules. (from DReaM)

If we apply DReaM to this problem, we have:

- rules-generating features as described above: $FEV1\%predicted$ and $FEV1/FVC$.
- new cluster-preserving features: change in FEV1 (denoted as $\Delta FEV1$ ) and the change in FEV1 % predicted (denoted as $\Delta FEV1\%predicted$ ) after 5 years.

DReaM is trained using the GOLD criterion described above as the prior knowledge. Results are presented on the right of Figure 7.

### 6.4 Interaction with fairness

As always, greater explainability go hand-in-hand with greater fairness as it allows to identify potential problematic features used in the clustering, as well as difference of feature importance across groups.

Using post-processing explanability techniques, it is also possible to combine fairness and explainability, for instance using ExKMC on a fairlet clustering method.

## 7 Robustness

### 7.1 Overview

In clustering, robustness is often measured using the **breakdown point**. This is thee minimum fraction of data points which if significantly modified will also

significantly modify the output of the estimator. Large breakdown point means better robustness.

We illustrate how typical k-means clustering has a breakdown point of zero and why this could be problematic.

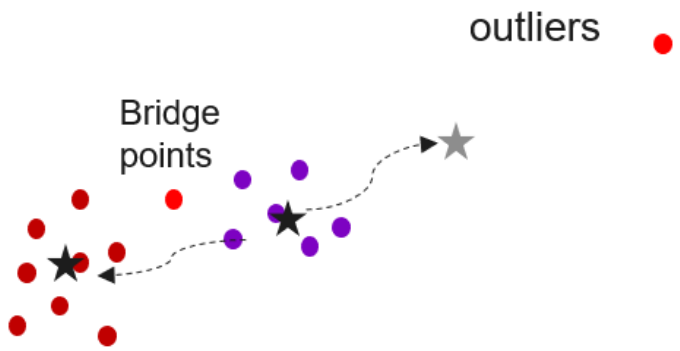



Figure 9: outliers and bridge points in k-means clustering (k=2)

In the illustration above, we can see that the presence of the outlier will move the mean of the purple cluster to the right, and that some purple points on the left will be re-assigned to the red clusters. In similar ways, bridge points can "spoil" the clustering. k-means effectively has a breakdown point of zero.

## 7.2 Motivation

We give an example to illustrate how robustness in clustering can be important. In a college, an algorithm is used to decide who should repeat the year by grouping students into clusters. One of the student has the maximum grade whilst all others are more towards the average grade. This will skew the clusters so that more students will be sent to repeat their year than if that outstanding student was not here.

## 7.3 Methods

### 7.3.1 Changing the loss

Similarly as what we've seen for regression, k-means uses a L2 norm to measure distance:

$$\sum_{i=1}^{n}(x_i - m)^2$$

where $x_i$s are the data points of the cluster with mean $m$. This is particularly sensitive to outliers because the distance between a point and its mean is squared, hence distant (i.e. outlier) points will weigh a lot and skew significantly the model.

One common strategy to reduce the impact of outliers is to use a L1-norm instead:

$$\sum_{i=1}^{n} |x_i - m|$$

However, this strategy still has a breakdown point of zero as a far enough outliers will still spoil the clustering.

### 7.3.2 Trimmed k-means

Another strategy is to "trim" the data. In the illustration below, we show what trimming mean in the context of simply fitting a mean to a set of scalar. We remove the sane number of points at each extremity (here 1), and then fit the means to the remaining points. This can help get rid of some outliers.

Figure 10: Trimmed mean example

In the context of clustering, we will do the same and perform the clustering on a subset of the data of size $[n(1-\alpha)]$ where $n$ is the total number of data points. As the concept of "extremity" is not as clear in clustering as in our simple example above, the set of points to remove is decided during the optimisation, performed through the double minimization problem below:

$$\arg\min_{Y} \min_{m_1,..,m_k} \sum_{x_i \in Y} \min_{j=1,..,k} ||x_i - m_j||^2$$

If we decompose what happens, we have:

- through all subset $Y$ of the data of size $[n(1-\alpha)]$
- Find the cluster means $m_1, ..., m_k$ that minimize the loss
- The loss is defined as the sum of the L2 distance of the data points to the mean closest to them.

Then we select the subset $Y$ with the minimum loss.

## 7.4 Interaction with Fairness

Robust Fair Clustering by Chhabra et al, 2023 looks at the robustness of fairness. The authors invent an attack that significantly decrease fairness, and propose a way to make clustering fairness more robust.

### 7.4.1 The attack

By changing a small portion of individuals' protected group memberships, the authors notices that fairness was significantly impacted on existing technique for fair clustering algorithms.

### 7.4.2 The defense

The authors propose Consensus Fair Clustering (CFC), which utilizes consensus clustering along with fairness constraints to output robust and fair clusters. It consists in sampling subsets of the training data and running the cluster analysis a number of times (let's say $n$). Since the attacked samples are a tiny portion of the whole training data, the probability of these being selected into the subset is also small, which decreases their negative impact. This creates $n$ basic partitions of the data. CFC then fuses the basic partitions using consensus clustering with a fairness constraint. Refer to this blog for more information about consensus clustering, which is also a method for robust clustering.

# 8 Privacy

## 8.1 Overview and Motivation

In clustering, the main focus on privacy is to be able to join several datasets to form more robust clusters, but without revealing information about the individual clusters as these could be owned by various parties and contain sensitive information. This is for instance often the case in bioinformatics where the data sets are owned by separate organizations, who do not want to reveal their individual data sets.

## 8.2 Distributed k-means

Assuming two parties A and B have access to a trusted third-party (TTP), the distributed k-means algorithm has the following steps:

1. TTP gives means $\mu_1, ..\mu_k$ for all clusters.
2. A & B assign data points to cluster
3. A sends to TTP the pair: $(s_i^A, n_i^A)$ where $s_i^A$ is the sum of samples in cluster $i$ for party $A$ and $n_i^A$ the number of samples in cluster $i$ for party A. B does the same.
4. TTP calculates the new means: $\mu_i = \frac{s_i^A + s_i^B}{n_i^A + n_i^B}$.
5. TTP sends the new means to A & B

And this is repeated a number of times. This way, no other data than the cluster means is shared between A and B.

## 8.3 Privacy preserving k-means

Privacy-preserving k-means algorithm presented by (Jha et al., 2005) does exactly the same, but the third-party replaced by a privacy-preserving protocol. The only thing parties A and B needs to agree on is a set of initial cluster means.

## 8.4 Interaction with fairness

Because little information is share, the fairness metrics can only be calculated on the individual datasets, which gives less insights into fairness and less statistical power.

# Part III - Recommender Systems

# 9 Introduction

## 9.1 Recommender Systems

Recommender Systems are a subclass of information filtering systems, that seek to predict user ratings from past interactions. The predicted user ratings are used to recommend new items to each user, they are likely to enjoy.


Figure 11: Interaction Matrix.

One important particularity of recommendation, is that the data takes a different form from traditional AI tasks. Since the data is made of interactions between users and items, we often display it as a matrix (Figure 11). Wherever a user has rated an item, we insert the rating within the matrix. User item pairs that do not have a past interaction contain $NaN$ values, we also sometimes use 0 if it is clear it is not a rating.

## 9.2 Importance

Recommender systems are used in a wide range of sectors: e.g., e-commerce, social networks, search engines, news portals, hiring platforms, intelligent assistants, smart home, smart city services, healthcare, financial applications, etc. More importantly, they are one of the big points of contact between AI and humans. That is why we say Recommender systems are at the forefront of human-centered AI research.

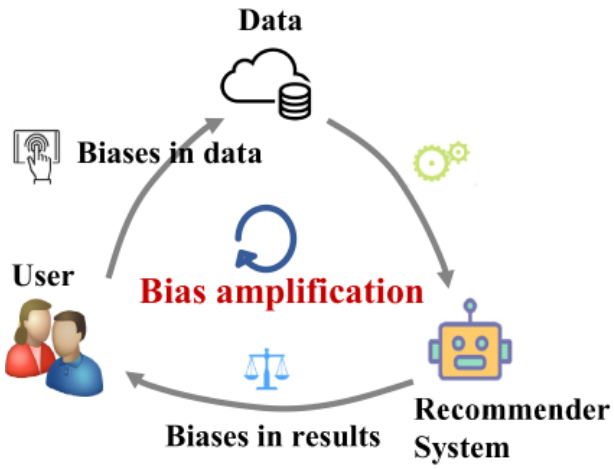



Figure 12: Feedback Loops.

The main problem with Recommender Systems, is that can enhance the patterns of society, notably because of **feedback loops** (Figure 12). These loops can lead to many issues, with different degrees of gravity. A common one is that the popular items get more popular, the so called rich get richer effect. Another common phenomenon is echo chambers, where users only see content that validates their prior opinions, leading to increasingly polarized views.

## 9.3 Models

There are different approaches to recommendation, but the most common and effective one is known as **collaborative filtering**. In collaborative filtering, the paradigm is that similar users will like similar items. The most common approach used in collaborative filtering is **matrix factorization**. Matrix factorization relies upon the assumptions that there are hidden latent features for users and items, from which scores can be reconstructed as a dot product (Figure 13).

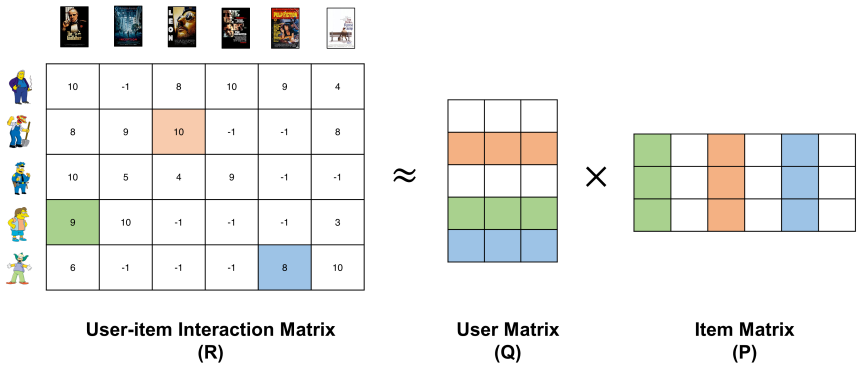



Figure 13: Matrix Factorization.

# 10 Explainability

## 10.1 Explicit Factor Models

As mentioned in the previous section, a common approach to Recommendation makes use of matrix factorization. As with many other AI methods, matrix factorization leads to obscurity into how the results are obtained. That is because the latent features are highly abstract, and not semantically meaningful to humans. The paper Explicit factor models for explainable recommendation based on phrase-level sentiment analysis proposes that we use explicit latent factors instead. This enforces latent features to have semantic meaning (Figure 14).

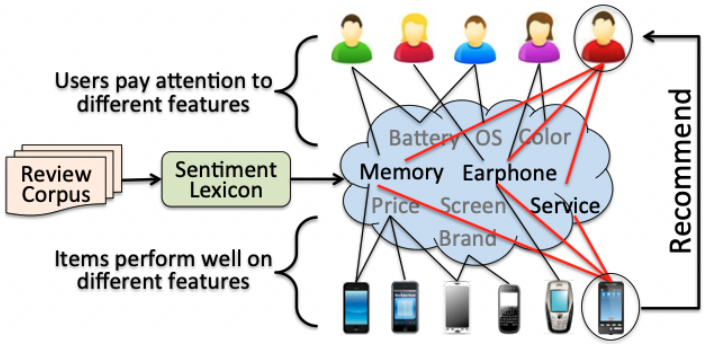



Figure 14: Explicit Factor Models.

Since the latent features are set explicitly, the system merely has to learn the sentiment of each user towards each of the features. That can be done using NLP on the user's reviews.

### 10.2 Interaction with fairness

Having explainable recommendations can have a variety of benefits. The first is that systems can be held accountable, and audited for ethical flaws. For instance it is easier to know if some groups of users are being discriminated against. Other benefits of explainable recommendations are listed below:

- Recommendation that are explainable are less likely to discriminate against groups, and it is easier to find out if they do.
- Users can be provided with explanations for the items recommended to them.
- More trustworthy systems can also lead to increased sales and therefore increased revenue for the owner of the system.

## 11 Robustness

### 11.1 Overview

As mentioned in previous sections, robustness has to do with how vulnerable a system is to being attacked, misused or broken by small data changes. This risk is all the more critical in recommendation because of how interlinked they are with humans. According to the Recommender Systems Robustness literature, attacks can be classified into two broad groups:

- Hand engineered Shilling Attacks
- Machine Learning Adversarial Attacks

### 11.2 Shilling

Shilling attacks consist of injecting a large number of fake profiles into a system so as to alter the behaviour of the system. These alterations can take a number of different forms and motives, for instance the attacker could be:

- promoting their own items
- demoting competitor's items
- destroying system performance
- altering system performance towards unethical behaviours

For a broad overview of shilling Attacks and Defenses in Recommendation, see Shilling attacks against recommender systems: a comprehensive survey.

### 11.3 ML Adversarial Attacks

ML Adversarial Attacks take a different approach. In this type of attack, the paradigm is to find minimal perturbations of the user item interaction matrix, that lead to the largest possible drop or alteration in system performance. Some of the different types of attacks we find withing the literature are:

- Adversarial perturbation of model parameters
- Adversarial perturbation of content data
- Data poisoning attacks

For a review of the field, see Adversarial Attacks and Detection on Reinforcement Learning-Based Interactive Recommender Systems.

### 11.4 Interaction with fairness

The strength of modern recommender systems is how powerful they are at learning user preferences. However, their greatest strength is also what leads to their danger. That is because malevolent attackers can use attacks to significantly alter the system behaviour. Some possible consequences are:

- Attackers disadvantaging an item provider, out of competition or revenge
- Attackers altering performance of a system to discriminate against one or more groups
- Attackers altering system performance to display unsafe contents to users

## 12 Privacy

### 12.1 Overview

The main principles of Privacy are the same for all AI tasks. They are:

- Private information: the critical or valuable information that needs restriction of access. For example, user identity and sensitive user attributes such as gender, age, and address.
- Ownership: only the authorized entities can access and control the corresponding private information, where the entity may refer to a user or even the platform itself.
- Threat: malicious entities (inside or outside the system) aiming to get access to or manipulate private information. Note that such entities may utilize auxiliary public information to engage its infiltration or attacks.
- Goal of privacy protection: to maintain the ownership of the private information and find countermeasures for the threats

In Recommendation, Privacy is all the more critical because of the amount of personal information involved.

## 12.2 Privacy Preserving Matrix Factorization

We have seen in previous sections that matrix factorization is a commonly used model in recommendation. In the paper Privacy-preserving matrix factorization, the authors show how such models can be implemented with absolute privacy. The method is such that the owner of the system never has access to user ratings, nor to any information about them. Still, a model can be trained, and items can be recommended to users.

Such methods are rarely used in practice because of the extra implementation burden for the owner of the recommender system, and (slightly) increased training time.

## 12.3 Interaction with fairness

The practices and regulation around data are complex and often obscure. Even if users are protected by the law, it is common for users not to be aware of how their data is used, sold, leaked, etc. This is all the more true in Recommendation, where vast amounts of personal data is collected. Even if such data is not collected, it can often be inferred as shown in Robust Deanonymization of Large Sparse Datasets.