

TUTORIAL: IBM ai360

The [AI Fairness 360 toolkit](#) is an extensible open-source library containing techniques developed by the research community to help detect and mitigate bias in machine learning models throughout the AI application lifecycle. The [AI Fairness 360 Python package](#) includes metrics to test for biases and algorithms to mitigate bias.

A bias detection and/or mitigation tool needs to be tailored to the particular bias of interest. More specifically, it needs to know the attribute or attributes, called *protected attributes*, that are of interest: *race* is one example of a protected attribute, another is *age*.

The tutorial shows how to employ the toolkit to compute fairness metrics and mitigate the bias using a feature transformation.

Handling outliers

Outlier detection algorithms:

Isolation forest

- One efficient way of performing outlier detection in high-dimensional datasets is to use random forests. This method *isolates* observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.

Elliptic envelope

- One common way of performing outlier detection is to assume that data are Gaussian distributed. From this assumption, we can try to define the general shape of the data, which leads to a threshold-approach to calling an observation an outlier.

OneClassSVM

- Use a single class version of a support vector machine to identify an optimal hyperplane that can be used to call an observation an outlier.

For high-dimensional data it is sometimes necessary to pipe the data through a dimension reduction algorithm before applying the outlier detection algorithm. In addition to how you scale the data, there is the choice of dimension reduction algorithm and the choice of outlier detection algorithms. Moreover, there are generally parameters that modify these outlier detection algorithms, such as an assumed level of contamination. Given the number of tunable components, grid-searching and iteratively comparing pipelines to decide on a suitable outlier detection pipeline can be time consuming.

Unsupervised learning

Machine learning models that are used to predict the values of response variables generally fall into the domain of supervised learning models. By contrast, unsupervised learning deals with the discovery of patterns derived from the feature matrix itself. We have already seen how useful derived patterns are in the example of visualizing high-dimensional data and outliers. The domain of unsupervised learning has many families of models. Clustering analysis is one of the main subject areas of unsupervised learning and it is used to partition the observations in a feature matrix into groups. Some models use probabilistic or 'soft' assignments.

We discuss some of the most common clustering algorithms and mixture models.

Clustering algorithms

- [*k*-means](#)
- [Hierarchical clustering](#)
- [Affinity Propagation](#)
- [Spectral Clustering](#)

Mixture modeling

- [Gaussian Mixture Models](#)
- [Dirichlet Process](#) Gaussian Mixture Models

There are a number of [other clustering algorithms available](#)

Model selection

[Silhouette Coefficient](#)

- PROS: very flexible and can be visualized through a silhouette plot CONS: generally higher for convex clusters than other clusters such as density based clusters

[Davies-Bouldin Index](#)

- PROS: The computation of Davies-Bouldin is simpler than that of Silhouette scores. CONS: generally higher for convex clusters than other clusters, such as density based cluster. It is also limited to the distance metrics in Euclidean space.

Calinski-Harabasz Index

- PROS: The score is higher when clusters are dense and well separated, which relates to a standard concept of a cluster CONS: generally higher for convex clusters than other clusters such as density based clusters

CASE STUDY: clustering

In this case study we begin to see the utility of iterative comparison across transformation pipelines. The three central questions to keep in mind are:

- Which preprocessing processing pipeline is the best?
- Which transformation pipeline is the best?
- Which model is the best?

The choice of model is the question we will dive deeply into during the next course in this series, but the first two are very relevant to this course. Similar to the first case study one aspect of this exercise is to create new features with unsupervised learning. Here we use all of clusters, but a domain expert or say someone from marketing might want to weight certain profiles differently than others.