Numerical data: First steps

Before creating feature vectors, we recommend studying numerical data in two ways:

- Visualize your data in plots or graphs.
- Get statistics about your data.

Visualize your data

Graphs can help you find anomalies or patterns hiding in the data. Therefore, before getting too far into analysis, look at your data graphically, either as scatter plots or histograms. View graphs not only at the beginning of the data pipeline, but also throughout data transformations. Visualizations help you continually check your assumptions.

We recommend working with pandas for visualization:

- Working with Missing Data (pandas Documentation)
- Visualizations (pandas Documentation)

Note that certain visualization tools are optimized for certain data formats. A visualization tool that helps you evaluate protocol buffers may or may not be able to help you evaluate CSV data.

Statistically evaluate your data

Beyond visual analysis, we also recommend evaluating potential features and labels mathematically, gathering basic statistics such as:

- mean and median
- standard deviation
- the values at the quartile divisions: the 0th, 25th, 50th, 75th, and 100th percentiles. The 0th percentile is the minimum value of this column; the 100th percentile is the maximum value of this column. (The 50th percentile is the median.)

Find outliers

An **outlier** is a value *distant* from most other values in a feature or label. Outliers often cause problems in model training, so finding outliers is important.

When the delta between the 0th and 25th percentiles differs significantly from the delta between the 75th and 100th percentiles, the dataset probably contains outliers.

Note: Don't over-rely on basic statistics. Anomalies can also hide in seemingly well-balanced data.

Outliers can fall into any of the following categories:

- The outlier is due to a *mistake*. For example, perhaps an experimenter mistakenly entered an extra zero, or perhaps an instrument that gathered data malfunctioned. You'll generally delete examples containing mistake outliers.
- The outlier is a legitimate data point, *not a mistake*. In this case, will your trained model ultimately need to infer good predictions on these outliers?
  - If yes, keep these outliers in your training set. After all, outliers in certain features sometimes mirror outliers in the label, so the outliers could actually *help* your model make better predictions. Be careful, extreme outliers can still hurt your model.
  - If no, delete the outliers or apply more invasive feature engineering techniques, such as **clipping**.

Key terms:

- Clipping
- Outliers

Last updated 2025-02-26 UTC.