

ML Concepts

Home

Crash Course

Filter

- Data
- Working with numerical data (85 min)

Introduction (3 min)

How a model ingests data with feature vectors (5 min)

First steps (5 min)

Programming exercises (10 min)

Normalization (20 min)

Binning (15 min)

Scrubbing (5 min)

Qualities of good numerical features (5 min)

Polynomial transforms (5 min)

Test your knowledge (10 min)

Conclusion (2 min)

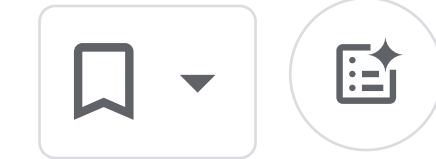
What's next
- Working with categorical data (50 min)
- Datasets, generalization, and overfitting (105 min)
- Advanced ML models
- Neural networks (75 min)



Home > Products > Machine Learning > ML Concepts > Crash Course

Was this helpful?

# Numerical data: Qualities of good numerical features



Send feedback

- On this page
- Clearly named

Checked or tested before training

Sensible

This unit has explored ways to map raw data into suitable **feature vectors**. Good numerical **features** share the qualities described in this section.

## Clearly named

Each feature should have a clear, sensible, and obvious meaning to any human on the project. For example, the meaning of the following feature value is confusing:

**Not recommended**

```
house_age: 851472000
```

In contrast, the following feature name and value are far clearer:

**Recommended**

```
house_age_years: 27
```

**Note:** Although your co-workers will rebel against confusing feature and label names, the model won't care (assuming you normalize values properly).

## Checked or tested before training

Although this module has devoted a lot of time to **outliers**, the topic is important enough to warrant one final mention. In some cases, bad data (rather than bad engineering choices) causes unclear values. For example, the following `user_age_in_years` came from a source that didn't check for appropriate values:

**Not recommended**

```
user_age_in_years: 224
```

But people *can* be 24 years old:

**Recommended**

```
user_age_in_years: 24
```

Check your data!

## Sensible

A "magic value" is a purposeful discontinuity in an otherwise continuous feature. For example, suppose a continuous feature named `watch_time_in_seconds` can hold any floating-point value between 0 and 30 but represents the *absence* of a measurement with the magic value -1:

**Not recommended**

```
watch_time_in_seconds: -1
```

A `watch_time_in_seconds` of -1 would force the model to try to figure out what it means to watch a movie backwards in time. The resulting model would probably not make good predictions.

A better technique is to create a separate Boolean feature that indicates whether or not a `watch_time_in_seconds` value is supplied. For example:

**Recommended**

```
watch_time_in_seconds: 4.82
is_watch_time_in_seconds_defined=True

watch_time_in_seconds: 0
is_watch_time_in_seconds_defined=False
```

This is a way to handle a continuous dataset with missing values. Now consider a **discrete** numerical feature, like `product_category`, whose values must belong to a finite set of values. In this case, when a value is missing, signify that missing value using a new value in the finite set. With a discrete feature, the model will learn different weights for each value, including original weights for missing features.

For example, we can imagine possible values fitting in the set:

```
{0: 'electronics', 1: 'books', 2: 'clothing', 3: 'missing_category'}.
```

- Key terms:**
- Outliers

Feature

Feature vector

Help Center

[Previous](#)  
 **Scrubbing (5 min)**

**Polynomial transforms (5 min)** [Next](#)

Was this helpful?



Send feedback

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see the [Google Developers Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2025-05-12 UTC.

- Connect

Blog

Instagram

LinkedIn

X (Twitter)

YouTube
- Programs

Google Developer Groups

Google Developer Experts

Accelerators

Women Techmakers

Google Cloud & NVIDIA
- Developer consoles

Google API Console

Google Cloud Platform Console

Google Play Console

Firebase Console

Actions on Google Console

Cast SDK Developer Console

Chrome Web Store Dashboard

Google Home Developer Console