Datasets, Generalisation, and Overfitting

Data Characteristics

A **dataset** is a collection of **examples**.

Many datasets store data in tables (grids), for example, as comma-separated values (CSV) or directly from spreadsheets or database tables.

Regardless of the format, your ML model is only as good as the data it trains on.

Quantity of Data

As a rough rule of thumb, your model should train on at least an order of magnitude (or two) more examples than trainable parameters. However, good models generally train on *substantially* more examples than that.

Models trained on large datasets with few **features** generally outperform models trained on small datasets with a lot of features

Quality and Reliability of Data

> A high-quality dataset helps your model accomplish its goal. A low quality dataset inhibits your model from accomplishing its goal.

A high-quality dataset is usually also reliable. **Reliability** refers to the degree to which you can *trust* your data. A model trained on a reliable dataset is more likely to yield useful predictions than a model trained on unreliable data.

Complete vs. Incomplete Examples

In a perfect world, each example is **complete**; that is, each example contains a value for each feature.
In a perfect world, each example is
complete; that is, each example contains a value for each feature.

Don't train a model on incomplete examples. Instead, fix or eliminate incomplete examples by doing one of the following:

- Delete incomplete examples.
- **Impute** missing values; that is, convert the incomplete example to a complete example by providing well-reasoned guesses for the missing values.

Labels

Direct labels, which are labels identical to the prediction your model is trying to make. That is, the prediction your model is trying to make is exactly present as a column in your dataset

Proxy labels, which are labels that are similar—but not identical—to the prediction your model is trying to make
Direct labels are generally better than proxy labels. If your dataset provides a possible direct label, you should probably use it. Oftentimes though, direct labels aren't available.

Imbalanced Datasets

Consider a dataset containing a categorical label whose value is either *Positive* or *Negative*. In a **balanced** dataset, the number of *Positive* and *Negative* labels is about equal. However, if one label is more common than the other label, then the dataset is **imbalanced**. The predominant label in an imbalanced dataset is called the **majority class**; the less common label is called the **minority class**

Downsampling and Upweighting

One way to handle an imbalanced dataset is to downsample and upweight the majority class. Here are the definitions of those two new terms:

- **Downsampling** (in this context) means training on a disproportionately low subset of the majority class examples.
- **Upweighting** means adding an example weight to the downsampled class equal to the factor by which you downsampled.

Dividing the Original Dataset

Training, Validation and Test Sets

You should test a model against a *different* set of examples than those used to train the model. As you'll learn a little later, testing on different examples is stronger proof of your model's fitness than testing on the same set of examples. Where do you get those different examples? Traditionally in machine learning, you get those different examples by splitting the original dataset. You might assume, therefore, that you should split the original dataset into two subsets:

- A **training set** that the model trains on.
- A **test set** for evaluation of the trained model.

In addition to the training set and the test set, the third subset is:

- A **validation set** performs the initial testing on the model as it is being trained.

Generalisation

When training a model on data, you are teaching it to perform well on that specific set of data.

If the model makes high-quality predictions on training data but much lower-quality predictions on brand-new data, then the model did not generalise. In this case, we say that the model has overfitted the training data.

Overfitting

Overfitting means creating a model that matches (*memorizes*) the **training set** so closely that the model fails to make correct predictions on new data. An overfit model is analogous to an invention that performs well in the lab but is worthless in the real world.

Fitting, Overfitting and Underfitting

A model must make good predictions on *new* data. That is, you're aiming to create a model that "fits" new data.

As you've seen, an overfit model makes excellent predictions on the training set but poor predictions on new data. An **underfit** model doesn't even make good predictions on the training data. If an overfit model is like a product that performs well in the lab but poorly in the real world, then an underfit model is like a product that doesn't even do well in the lab.

Generalization is the opposite of overfitting. That is, a model that *generalizes well* makes good predictions on new data. Your goal is to create a model that generalizes well to new data.

Completion

You earned the **Machine Learning Crash Course: Datasets, generalization, and overfitting** badge!

The badge has been added to your profile.