A dataset comprises rows and columns, where each row is an entry into the dataset. Datasets are made up of feature and target values. Let's recap about what makes them different.

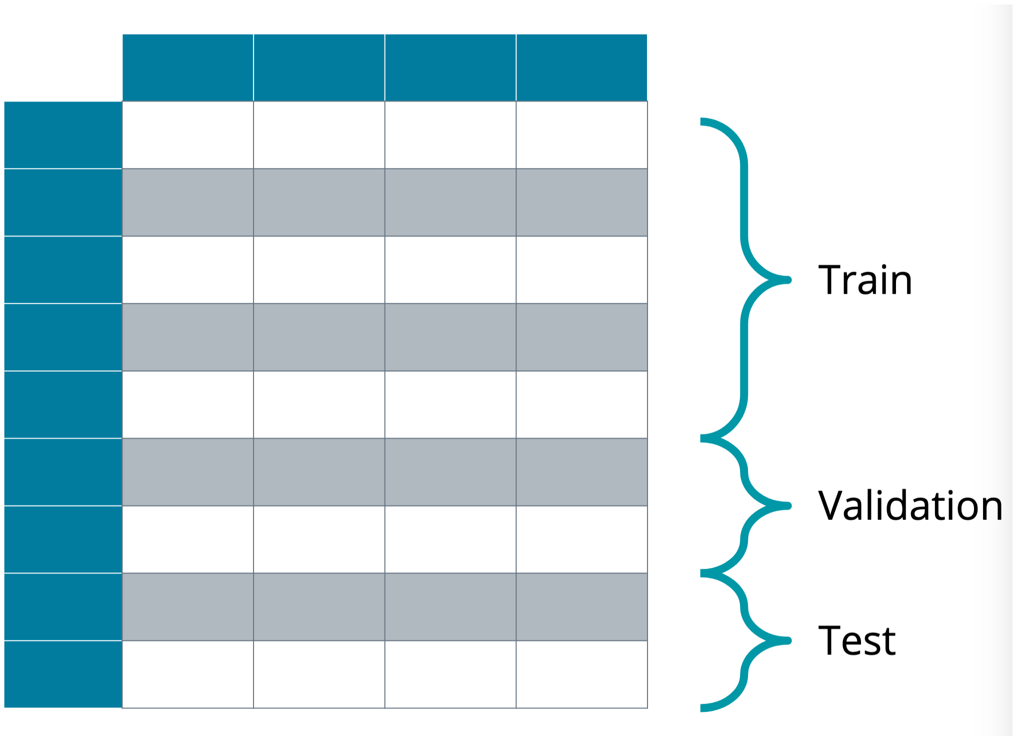
Feature Values

- Features are the columnar data for a row.
- Features are used by models for training to predict the target value.
- Features can be described as the **attributes** of a given data point.

Target Values

- Column value selected to have the model trained to predict.
- In reality, any feature could be a target value. You need to choose it as your target value, and be sure to exclude it from your feature values.

Creating You Dataset



Visualization of a table split into train, validation, and test splits

Training a model on the entire dataset and then testing it on the same dataset, allows your model to memorize what the data looks like. In order to have a true idea of its performance, we need to test it on unseen data. Data sets are then split before training to prevent overfitting the model. However you want to train your model on the most data you possibly can, so the training dataset is usually the largest split when separating the data.

When splitting the data, there are two methods of doing so:

1. Split the data in **train** and **test**. You then train the model on the training dataset and test and tune the model on the **test** dataset.
2. Split the data in **train**, **validation**, and **test**. Then allows you the train the model on the training dataset, tune the model on the **validation** dataset, and finally test your model on the **test** dataset.

The main benefit of including the **validation** dataset is that your **test** dataset is not part of your model tuning, and therefore is a more accurate measure of completely unseen data.

Example split ratios are:

1. Train: 75%, Test: 25%
2. Train: 60%, Test: 40%
3. Train: 50%, Validation: 25%, Test: 25%
4. Train: 70%, Validation: 15%, Test: 15%
5. Train: 33%, Validation: 33%, Test: 33%

Repeatability

In most cases, with creating dataset or training models, it is desired to have repeatability in processes. By default, most data preparation methods and models randomly select and process data. Meaning each time you rerun your data preparation or train a model, there will be a different outcome. To control for this, you can define the **random state,** typically the parameter `random_state`. When defined, data will be selected, although technically "random", in a way that produces the same results upon repeating.

Code Example of Splitting Data

```
import pandas as pd
from sklearn.model_selection import train_test_split

df = pd.DataFrame([[1, 2], [3, 4], [5, 6], [7, 8]])

df_train, df_test = train_test_split(
    df,
    test_size=0.2, # 20% goes to test
    random_state=0 # makes splitting data repeatable
)
print(df_train)
print(df_test)


# output
   0  1
3  7  8
1  3  4
0  1  2

   0  1
2  5  6
```

Additional Resources

- To learn more about the `train_test_split` method from Scikit-learn, you can visit their documentation page: train_test_split. There you will find additional parameters and examples for more insight.
- If you want to learn more about training, validation, and test in general, visit the Wikipedia page entry for the topic.
- For a more technical and in-depth look at creating train/validation/test sets, you can visit the Real Python blog post: Split Your Dataset With scikit-learn's train_test_split()