Daimensions™ Frequently Asked Questions

Q: Why do I need to measure anything other than **accuracy**?

A: Without doing the up-front work of measuring and "right-sizing" the machine learning model you're building before you train on your data, you have no way of knowing whether the predictor you build will actually do what you want it to do.

There are lots of things that can go wrong if you only look at the performance accuracy of your machine learning model, including**:**

- You might accidentally build a model that makes predictions that are too closely fixed to what was in the training data, and which won't generalize well when new data is processed. (This is called "overfitting".)
- You have no way of knowing whether you built your model using the right amount of training data – did you gather enough data? Or, perhaps, did you use way more data than you really needed?
- Is there any pre-processing needed to the data? Which is the right model?
- Is my model biased? How resilient is my model to changes?

Q: How good are the measurements?

We are testing our measurements on a wide variety of data sets, including a 176-task subset from OpenML which includes binary and multiclass classification problems from a multitude of sources, including bio/medical, finance, speech, vision, and natural language data. Our measurement-based process is able to automate the creation of predictors on these datasets and completely eliminate the need for hyper-parameter tuning. Our predictors are usually 2 orders of magnitude smaller than the state of the art and the reduction in training time is usually at least on order of magnitude. In 70% of the cases, our measurement-based approach beats the state-of-the-art accuracy reported on OpenML.

Our results are reproducible at: http://github.com/brainome/OpenML

Q: Why is measuring **generalization** important?

A: When you're building a machine learning model for deployment in the real world, you want to be sure that the model is the right one for use on all the data that may show up to be processed. To be successful, you must have the ability to trade off generalization and accuracy to get the most successful performance of your system. In most cases, it's much better to give away a few small points of accuracy to gain the stability and long-term usefulness that a more general model will provide.

Q: Beyond being able to control **generalization** and **accuracy**, why should I use Daimensions™?

A: Without doing the up-front work of measuring and "right-sizing" the machine learning model you're building before you do training on your data, you have no way of knowing whether you're spending the right amount of compute and storage to achieve your goals.

There are multiple things that often go wrong if you don't "measure before you cut":

- You build a model that is much larger than it needs to be.
- Your training time is much longer than it needs to be.
- Your run times are much longer than they need to be.
- Your model is less general than it could be.

Q: How does Daimensions™ handle missing data in features or target variables?

A: Missing targets are handled differently than missing feature cells.

- Missing values in the target column of a row of training data will generate an error.
- Empty cells within a row of training data are handled as if they were empty strings. A unique numerical identifier is generated for each empty cell. This allows Daimensions™ to learn relationships between missing values and the target column.

Q: Can Daimensions™ build other types of models than Neural Network (NN) and Decision Tree (DT)?

A: Currently Daimensions™ supports only NN and DT. Random Forest will become available in Q3 2020. Additional types of models will be supported in future releases.

Q: Can Daimensions™ be used to do regression?

A: No. Currently Daimensions™ only supports classification tasks. However, in some cases a regression target can be quantized into a series of consecutive classes and can thus be trained using Daimensions™ after mapping.

Q: Why should I measure the **learnability** of my training data?

A: If you don't measure the learnability of your training data before you build your model, you have no way of knowing whether you can really build a quality model from what you have. Is the data essentially random? At the other extreme, is it demonstrably reliable? Or, as is often

the case, is it somewhere in between? We use an iterative measurement approach over what we call a "capacity progression" to give a quantifiable answer to this question for your training data so that you can make conscious choices about investing in the gathering and labeling of the right amount of training data.

Q: What is **noise resilience**?

A: Noise resilience is just another way of talking about generalization. Generalization is measured in bits/bit – the higher the generalization, the more a machine learning model can predict using the same sized model. Noise resilience is measured in dB – the more general the model, the greater the noise resilience.

Q: How many data points are required for training?

A: If you're starting from absolute scratch and don't have any data to measure yet, a good rule of thumb is to gather a minimum of 100 data points for each class. However, it is possible to train with fewer samples than that if the data is highly generalizable. On the other hand, it's possible that more data might be required. The best thing to do is to run the learnability measurement process on a regular basis to be sure you're investing in collecting the right amount of training data for your task.

Q: Where can I find the datasets used to create the predictors in https://github.com/brainome/OpenML?

A: In that repository, the file testfiles.csv contains one column ($Id) with IDs and another column with filenames ending in CSV ($Fname). The URLs encode as: https://www.openml.org/d/$ID/$Fname . The script validate.sh composes the URL and does a wget to copy the file locally.