

Daimensions™ CLI Quickstart

Requirements:

- Dimension CLI is currently supported on the following operating systems:
 - Linux Ubuntu 16.04 and later, CentOS 7 and later, Debian 8 and later, RedHat 7 and later.
 - MacOS Mojave and Catalina
- You will need a user login (email) and password to login.
- python3 needs to be installed to run the predictors created.
- Administrator privilege on your machine to install the btc executable

Installing btc

btc is the command line interface to Daimensions™, Brainome's cloud machine learning engine. First download the installation script at: <https://download.brainome.net/btc-cli/btc-setup.sh>

From the download directory run the following command:

```
mymachine: sh btc-setup.sh
```

The script will check that your operating system is supported, download the latest btc client to your machine and install it in /usr/local/bin. You will be prompted to enter the administrator password to install the software.

NOTE: After installation, make sure that "/usr/local/bin" is in your search path.

Login in for the first time:

You will need your user credentials to login to daimensions. The first time you login, your license key will be downloaded automatically. Please use the default password that was provided to you.

```
mymachine: btc WIPE
Please login on daimensions.brainome.ai
Daimensions(tm) user email: user@yourcompany.com
Daimensions(tm) password: *****
All cloud files wiped.
```

Changing you default password

From the shell, do the following:

```
mymachine: btc CHPASSWD
Brainome Daimensions(tm) account change password.
Note: Brainome does not believe in providing password generation rules as
they make cracking easier.
Instead, we rely on measuring the complexity of a new password. A good
password is complex but easy to remember.
For example, good passwords are the beginning letters of the words of an
imaginary sentence as described here:
http://obgyn.ufl.edu/creating-and-using-strong-passwords/
Try to vary upper and lower case and also add special characters or
numbers.
The minimum password length is 10 characters.

Enter old password: *****
New password: *****
Confirm password: *****
Password changed.
```

That's it. You're ready to start running Daimensions.

Running Daimensions™

For this tutorial, we will use a data file from OpenML.org located at:

https://www.openml.org/data/get_csv/53488/spectrometer.arff

Step 1: Download the example data file into your data directory

```
mymachine: curl https://www.openml.org/data/get_csv/53488/spectrometer.arff -o spectrometer.csv
```

NOTE: Version v0.96 of Daimensions supports only data files in CSV format.

Step 2: Get your measurements

The “-measureonly” option performs measurement only on the data set:

```
mymachine: btc -measureonly spectrometer.csv
Brainome Daimensions(tm) 0.96 Copyright (c) 2019, 2020 by Brainome, Inc. All Rights
Reserved.
Licensed to: Your Name
Expiration date: 2020-08-30 (89 days left)
Number of threads: 1
Maximum file size: 1GB
Connected to: daimensions.brainome.ai

Note: Assuming machine learner type QC for compiling efficiency (not accuracy).
Note: Choice can be overridden with -f parameter.
Data:
Number of instances: 531
Number of attributes: 102
Number of classes: 2
Class balance: 89.64% 10.17%

Learnability:
Best guess accuracy: 89.64%
Capacity progression (# of decision points): [3, 5, 6, 6, 7, 7]
Quick Clustering: 100 parameters
Estimated Memory Equivalent Capacity for Neural Networks: 729 parameters

Risk that model needs to overfit for high accuracies...
using Quick clustering: 37.66%
using Neural Networks: 77.80%
```

```
Expected Generalization...
using Quick clustering: 5.31 bits/bit
using a Neural Network: 0.73 bits/bit

Recommendations:
Note: Maybe enough data to generalize. [yellow]
Note: Quick Clustering may outperform Neural Networks. Try with -f QC.

Time estimate for a Neural Network:
Estimated time to architect: 0d 0h 0m 1s
Estimated time to prime (subject to change after model architecting): 0d 0h 3m 43s

Time estimate for Quick Clustering:
Estimated time to prime a quick classifier: a few seconds
```

Congratulations! You have successfully run Daimensions measurements!

Because we did not specify a model type, notice that Daimensions selected a clustering network (QC) as the default model for this dataset to minimize the risk of overfitting the data. You can override this using the -f option. “-f NN” for neural networks or “-f QC” for clustering networks.

Step 3: Build a model

When building models, Daimensions™ offers multiple options:.

Create a python neural network called predict.py:

```
mymachine: btc -v -v -f NN spectrometer.csv -o predict.py
```

Create a python clustering predictor called predict.py

```
mymachine: btc -v -v -f QC spectrometer.csv -o predict.py
```

The “effort” option will increase compute time to lead to better accuracy. (for smaller sets)

```
mymachine: btc -v -v -f NN spectrometer.csv -o predict.py -e 5
```

The “-e 5” effort will take 5 times as long to run and possibly improve the accuracy. It can be used with QC models and NN models.

For a complete list of all options (all options available will print):

```
mymachine: btc -h
```

IMPORTANT! BTC assumes that the last data column is the predictor. If it is not, specify the column name using the `-target` option:

```
mymachine: btc -v -v -f NN spectrometer.csv -o predict.py -target binaryClass
```

For a complete tutorial on using Daimensions, please refer to the Daimensions™ ' User Manual

Step 4: Run the predictor

Once you have created your predictor, you can use it to validate an existing data set or predict with new data. With the predictor created in the example above, you can validate the existing data spectrometer.csv set by running:

```
mymachine: python3 predict.py -validate spectrometer.csv
```

Alternatively, if you have a new dataset (without the prediction target column), you can predict the outcome by running:

```
mymachine: python3 predict.py newdata.csv
```