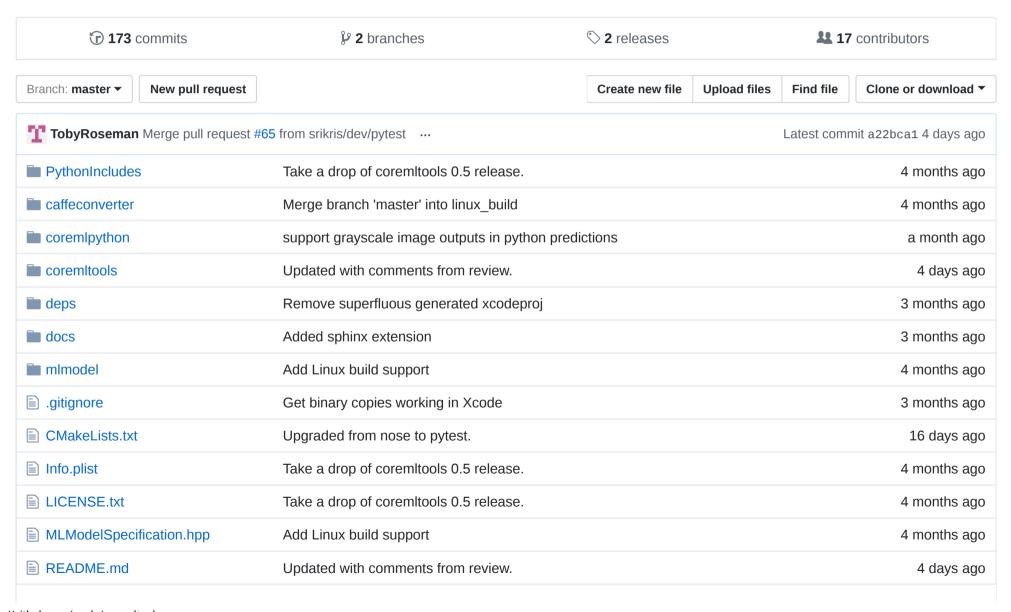
#### apple / coremitools

#### Converter tools for Core ML.



https://github.com/apple/coremltools

README.rst	Fixed typo, updated broken link to main docs and added/updated link t	4 months ago
setup.cfg	Take a drop of coremitools 0.5 release.	4 months ago
setup.py	added setup.py change for individual files inside graph_visualization/	3 months ago

**README.md** 

# **Core ML Community Tools**

Core ML community tools contains all supporting tools for CoreML model conversion and validation. This includes Scikit Learn, LIBSVM, Caffe, Keras and XGBoost.

We recommend using virtualenv to use, install, or build coremltools. Be sure to install virtualenv using your system pip.

pip install virtualenv

#### Installation

The method for installing *coremltools* follows the standard python package installation steps. To create a Python virtual environment called pythonenv follow these steps:

```
# Create a folder for your virtualenv
mkdir mlvirtualenv
cd mlvirtualenv

# Create a Python virtual environment for your CoreML project
virtualenv pythonenv
```

https://github.com/apple/coremltools

To activate your new virtual environment and install coremltools in this environment, follow these steps:

```
# Active your virtual environment
source pythonenv/bin/activate

# Install coremltools in the new virtual environment, pythonenv
(pythonenv) pip install -U coremltools
```

The package documentation contains more details on how to use coremltools.

#### **Dependencies**

coremitools has the following dependencies:

- numpy (1.12.1+)
- protobuf (3.1.0+)

In addition, it has the following soft dependencies that are only needed when you are converting models of these formats:

- Keras (1.2.2, 2.0.4+) with Tensorflow (1.0.x, 1.1.x)
- Xgboost (0.6+)
- scikit-learn (0.15+)
- libSVM

#### **Building from source**

To build the project, you need CMake to configure the project

cmake.

https://github.com/apple/coremItools

after which you can use make to build the project

make

## **Building Installable Wheel**

To make a wheel/egg that you can distribute, you can do the following

make dist

## **Running Unit Tests**

In order to run unit tests, you need pytest, pandas, and h5py.

```
pip install pytest pandas h5py
```

To add a new unit test, add it to the coremltools/test folder. Make sure you name the file with a 'test' as the prefix.

Additionally, running unit-tests would require more packages (like libsvm)

```
pip install numpy scipy scikit-learn
```

To install libsvm

```
git clone https://github.com/cjlin1/libsvm.git
cd libsvm/
```

https://github.com/apple/coremItools

```
make
cd python/
make
```

To make sure you can run libsvm python bindings everywhere, you need the following command, replacing <LIBSVM\_PATH> with the path to the root of your repository.

```
export PYTHONPATH=${PYTHONPATH}:<LIBSVM_PATH>/python
```

To install xgboost

```
git clone --recursive https://github.com/dmlc/xgboost
cd xgboost; cp make/minimum.mk ./config.mk; make
cd python-package; python setup.py develop
```

To install keras (Version >= 2.0)

```
pip install keras tensorflow
```

If you'd like to use the old keras version, you can:

```
pip install keras==1.2.2 tensorflow
```

Finally, to run the most important unit tests, you can use:

```
pytest -rs
```

some tests are marked as slow because they test a lot of combinations. If you want to run, all tests, you can use:

https://github.com/apple/coremItools

5/6

pytest

# **Building Documentation**

First install all external dependencies.

```
pip install Sphinx==1.5.3 sphinx-rtd-theme==0.2.4 numpydoc
pip install -e git+git://github.com/michaeljones/sphinx-to-github.git#egg=sphinx-to-github
```

You also must have the *coremltools* package install, see the *Building* section.

Then from the root of the repository:

```
cd docs
make html
open _build/html/index.html
```

https://github.com/apple/coremitools 6/6