

---

# **Percept Documentation**

*Release .14*

**Equirio**

August 06, 2013



# CONTENTS

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Description . . . . .	1
1.2	Goals of Percept . . . . .	2
1.3	Contributing . . . . .	2
<b>2</b>	<b>Installation</b>	<b>3</b>
2.1	Installation Overview . . . . .	3
<b>3</b>	<b>Usage</b>	<b>5</b>
3.1	Usage . . . . .	5
<b>4</b>	<b>Indices and tables</b>	<b>7</b>



# OVERVIEW

## 1.1 Description

Percept is a modular, extensible framework for machine learning that emphasizes easy testing and deployability.

There are several important concepts to the percept framework:

1. Settings
2. Workflows
3. Tasks
4. Inputs
5. Formatters

### 1.1.1 Settings

Settings define global attributes, such as what type of caching to use for task values, and where to store results.

### 1.1.2 Workflows

Workflows are high level constructs that tie together multiple tasks. A workflow could take you from data in a database to predicted clusters, for example.

A workflow, when it finishes running, will optionally allow you to store its results. It will also allow you to go into an ipython shell and manipulate and visualize the results when it is done running. Workflows can also be reloaded and used for prediction.

What tasks a workflow ties together, and what data it needs to load are pulled from a configuration file. Workflows can be run from the command line, or programatically.

### 1.1.3 Tasks

Tasks are discrete units of work. One task may load sound files, and another may turn them into feature vectors, for instance. A task can perform any work.

Tasks have the following features: #. Tasks can define which variables they would like to persist. These variables are stored by the cache specified in the settings. #. Tasks can also define a tester class, which will be used to run tests on the tasks. #. Tasks have a namespace, category, and name, which can be used to search for them in the registry later. #. Tasks can define dependencies, which are executed and provided to the task before it runs. #. Tasks can have both

a training method and a prediction method. One can be used to train the classifier or feature extractor, and the other used to perform the same operation on new data. #. Tasks can be executed by themselves, or with a workflow.

### 1.1.4 Inputs

Inputs read data in, and pass it to a formatter class. The data can be from a database, csv file, json file, etc. Each input defines what kind of values it will read in.

### 1.1.5 Formatters

Formatters reformat input data into data that a task can use. Formatters define what they convert from (defined by the input class), and what they convert to (defined by the task class). When a workflow runs, it will inspect its tasks to determine what type of data they accept. It will then pull the data in using an appropriate input and find the right formatter to convert it.

## 1.2 Goals of Percept

Percept is a modular, extensible framework for machine learning that emphasizes easy testing and deployability.

The goals are to create an easy to use tool that can aid in a lot of the pain points that surround creating and deploying machine learning algorithms, such as:

1. Scalability
2. Reusability
3. Deployment

Percept is open source under an Apache license. Alternate licensing is possible; contact [vik@equirio.com](mailto:vik@equirio.com).

## 1.3 Contributing

We welcome contributions.

To contribute:

1. Think of something useful to contribute, or check the github issues tracker.
2. Fork the repository and add in your new feature.
3. Put in a pull request to merge your feature into this repo.

Please feel free to contact [vik@equirio.com](mailto:vik@equirio.com) if you have any questions about contributing.

# INSTALLATION

## 2.1 Installation Overview

### 2.1.1 Manual - Recommended

This assumes that you already have git installed on your computer. The main steps are:

```
$ git clone git://github.com/equirio/percept.git
$ cd percept
$ xargs -a apt-packages.txt apt-get install
$ virtualenv /path/to/percept
$ source /path/to/percept/bin/activate
$ pip install -r pre_requirements.txt
$ pip install -r requirements.txt
```

See usage for usage instructions.

You can skip the virtualenv commands if you like, but they will be a major help in keeping the packages for this repo separate from the rest of your system.

**If you want to easily import the package, do this in the directory you cloned to::** `$ python setup.py install`

**This will install percept as a python package, and will allow you to import it in other projects using::** `$ import percept`

### 2.1.2 PyPi

You can find the percept package [here](#).

Install by doing:

```
$ pip install percept
```

You may need to manually install the requirements afterwards.





# USAGE

## 3.1 Usage

Percept can be used either programatically, or via the command line.

### 3.1.1 Command line

These should all be run from the directory where percept was cloned. If you installed via pip, substitute *percept-admin.py* for *python manage.py* .

Run automated tests:

```
$ python manage.py test --pythonpath='pwd' --settings=tests.test_settings
```

Run a workflow:

```
$ python manage.py run_flow tests/workflow_config/test_save.conf --pythonpath='pwd' --settings=tests
```

List available tasks:

```
$ python manage.py list_tasks --pythonpath='pwd' --settings=tests.test_settings
```

### 3.1.2 Guides

1. Predicting the NFL Season with Percept

### 3.1.3 Examples

1. NFL Season
2. Political Analyzer
3. Evolve your own music
4. Simpsons character analysis



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*