MALT

Release 1.0

Kimeel

Sep 19, 2019

CONTENTS:

CHAPTER

ONE

HOW TO INSTALL MALT

First clone the git repo and install virtualenv:

```
git clone https://github.com/kimeels/MALT.git

python3 -m pip install --user virtualenv
```

Change directories into MALT and create a virtual environment:

```
cd MALT

python3 -m venv malt_env
```

Start the virtual env and install the necessary packages using the requirements file:

```
source malt_env/bin/activate

pip3 install -r requirements.txt
```

CHAPTER

TWO

THE MALT API REFERENCE

2.1 The Lightcurve class

class malt.**Lightcurve**(*filepath*, *interpolate=False*, *interp_func=<function get_gp>*, *ini_t='rand'*, *obs_time=0.3333333333333333*, *sample_size=100*, *obj_type=None*)

extract_features(*feat_ex_method=<function get_wavelet_feature>*)

Extracts features from the given lightcurve with assigned feature extraction method.

self [Lightcurve object] An instance of the Lightcurve class.

feat_ex_method: python function Function to use for the feature extraction.

interpolate(*interp_func=<function get_gp>*, *ini_t='rand'*, *obs_time=0.3333333333333333*, *sample_size=100*, *aug_num=1*)

Interpolates the given lightcurve with assigned interpolation function

self: Lightcurve object An instance of the Lightcurve class.

interp_func: python function A python function that takes in a lightcurve and interpolates it.

ini_t: str or float Initial time to start sampling.

obs_time: float The total length of the interpolated lightcurve.

sample_size: int Number of data points in interpolated lightcurve.

aug_num: int Number of lightcurves to augment to.

loadfile(*filename*)

Loads file to extract time, flux, flux_err ra_dec and class

filename: path to dataset

2.2 The Dataset class

class malt.**Dataset**(*configFile=''*, *feat_ex_method=<function get_wavelet_feature>*, *interpolate=True*, *interp_func=<function get_gp>*, *ini_t='rand'*, *obs_time=0.3333333333333333*, *sample_size=100*, *aug_num=1*, *ml_method=<class 'malt.machine_learning.RFclassifier'>*, *hyperparams={'criterion': ['gini', 'entropy'], 'n_estimators': array([70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89])}*, *n_jobs=-1*, *pca=True*, *n_components=20*)

add(*new_lightcurve*)

Adds new lightcurve to the Dataset then retrains Dataset.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

lightcurve [Lightcurve object] Lightcurve object to add to dataset.

extract_features()
Extracts features from all the lightcurves in the given dataset with assigned feature extraction method.

self [Dataset object] An instance of the Dataset class containing instances of the Lightcurve class.

get_pca()
Performs PCA decomposition of a feature array X.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

interpolate()
Interpolates all the lightcurves in the given dataset with assigned interpolation function.

self [Dataset object] An instance of the Dataset class containing instances of the Lightcurve class.

classmethod load_from_save(*filename*)

Returns a saved Dataset instance using pickle

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

filename: str filename under which the Dataset instance was saved.

populate(*filepaths*)
Initialises an instance of the Dataset class.

self [Database object] An instance of the Database class.

filepaths: list List containing the paths to the data files.

predict(*lightcurve*, *show_prob=False*)

Predicts the type of given lightcurve object using classifier trained on Dataset.

self [Dataset object] An instance of the Dataset class containing instances of the Lightcurve class.

lightcurve [Lightcurve object] Lightcurve object for which to predict

show_prob [boolean.] If True will print full output from predict_proba()

project_pca(*lightcurve=None*)
Projects self.features onto calculated PCA axis from self.pca

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

run_diagnostic()
Runs the Diagnostic test which trains n classifiers on different subsets of the Dataset to test how well it can classify objects.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

save(*filename='saved_dataset'*)

Saves a Dataset instance using a pickle dump

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

filename: str filename under which to store the Dataset instance

train(*verbose=1*)

Trains a ML algorithm on the Dataset with the parameters specified on initialisation.

self [Dataset object] An instance of the Dataset class containing instances of the Lightcurve class.

verbose : How much information to print out.

types (*show_aug_num=False*)
Prints out the counts of each object type stored in the dataset.

self: Dataset object An instance of the Dataset class containing instances of the Lightcurve class.

show_aug_num: boolean Use augmented lightcurve when counting type numbers.

2.3 MALT interpolator

malt.interpolator.**get_gp** (*lightcurve*, *t0*, *obs_time*, *sample_size*, *aug_num*)
Returns a Gaussian Process (george) object marginalised on the data in file.

lightcurve: Lightcurve object An instance of the Lightcurve class.

t0: float Initial time to start sampling.

obs_time: float The total length of the interpolated lightcurve.

sample_size: int Number of data points in interpolated lightcurve.

2.4 MALT feature extraction

malt.feature_extraction.**get_wavelet_feature** (*lightcurve*)
Returns wavelet coefficients for a given lightcurve object.

lightcurve [Lightcurve object] An instance of the Lightcurve class

2.5 MALT machine learning

class malt.machine_learning.**RFclassifier** (*n_estimators='warn'*, *criterion='gini'*)

PYTHON MODULE INDEX

m

INDEX