AntiNex AI Utilities Documentation

Release 1.0.0

Jay Johnson

Nov 02, 2018

Contents

CHAPTER 1

AntiNex Stack Status

AntiNex AI Utilities is part of the AntiNex stack:

Component	Build	Docs Link	Docs Build
REST API		Docs	
Core Worker		Docs	
Network Pipeline		Docs	
AI Utils		Docs	
Client		Docs	

CHAPTER 2

Table of Contents

These are the docs for the AntiNex AI Utilities repository.

2.1 Make Predictions

Large helper method for driving all AI-related tasks.

Handles running:

1. Building Models
2. Compiling Models
3. Creating Datasets for Train, Test, and Predictions
4. Fitting Models
5. Evaluating Models
6. Cross Validating Models
7. Merging Predictions with Original Records

Here is the file on GitHub in case the `automodule` failed to process:

make_predictions.py

`antinex_utils.make_predictions.`**`build_regression_dnn`**(*num_features*, *compile_data*, *label=''*, *model_json=None*, *model_desc=None*)

Parameters

- **`num_features`** – input_dim for the number of features in the data
- **`compile_data`** – dictionary of compile options
- **`label`** – log label for tracking this method
- **`model_json`** – keras model json to build the model

- **model_desc** – optional dictionary for model

antinex_utils.make_predictions.**build_classification_dnn**(*num_features*, *compile_data*, *label=”*, *model_json=None*, *model_desc=None*)

Parameters

- **num_features** – input_dim for the number of features in the data
- **compile_data** – dictionary of compile options
- **label** – log label for tracking this method
- **model_json** – keras model json to build the model
- **model_desc** – optional dictionary for model

antinex_utils.make_predictions.**check_request**(*req*)

Parameters **req** – dictionary to check values

antinex_utils.make_predictions.**save_prediction_image**(*label='not-set'*, *history=None*, *histories=[]*, *image_file=None*)

Parameters

- **history** – model prediction history
- **histories** – histories to generate in the image
- **image_file** – save to file

antinex_utils.make_predictions.**make_predictions**(*req*)

Parameters **req** – dictionary for making predictions

2.2 Convert Records to Scaler Dataset

Helper method for converting records into a scaler dataset. This means all data is bounded between a range like: `[-1, 1]`.

antinex_utils.build_scaler_dataset_from_records.**build_scaler_dataset_from_records**(*record_list*, *label='build scaled-dataset'*, *min_feature= 1*, *max_feature* *cast_to_type*

Parameters

- **record_list** – list of json records to scale between min/max
- **label** – log label for tracking
- **min_feature** – min feature range for scale normalization
- **max_feature** – max feature range for scale normalization
- **cast_to_type** – cast all of the dataframe to this datatype

2.3 Convert Records to Scaler Train and Test Datasets

Helper method for converting records into a scaler datasets that are split using `sklearn.model_selection.train_test_split`. This means all training and tests data is bounded between a range like: `[-1, 1]`.

`antinex_utils.build_scaler_train_and_test_datasets.`**`build_scaler_train_and_test_datasets`**(*labe trai test df, test seed sca min 1, max*

Parameters

- **`label`** – log label
- **`train_features`** – features to train
- **`test_feature`** – target feature name
- **`df`** – dataframe to build scalers and test and train datasets
- **`test_size`** – percent of test to train rows
- **`min_feature_range`** – min scaler range
- **`max_feature_range`** – max scaler range

2.4 Build a Training Request

Helper for building a common training request.

`antinex_utils.build_training_request.`**`build_training_request`**(*csv_file='/tmp/cleaned_attack_scans.csv', meta_file='/tmp/cleaned_metadata.json', predict_feature='label_value', ignore_features=['label_name', 'ip_src', 'ip_dst', 'eth_src', 'eth_dst'], seed=None, test_size=0.2, preproc_rules=None*)

Parameters

- **`csv_file`** – csv file built with prepare_dataset.py
- **`meta_file`** – metadata file built with prepare_dataset.py
- **`predict_feature`** – feature (column) to predict
- **`ignore_features`** – features to remove from the csv before the split of test + train data
- **`seed`** – integer to seed

- **test_size** – percent of records to split into test vs train
- **preproc_rules** – future preprocessing rules hooks

2.5 Constant Values

```
SUCCESS = 0
FAILED = 1
ERR = 2
EX = 3
NOTRUN = 4
INVALID = 5
NOTDONE = 6
```

2.6 Merge Inverse Datasets into Original Records

Helper method for merging predictions with the original rows

antinex_utils.merge_inverse_data_into_original.**merge_inverse_data_into_original**(*req*, *sort_on_index=* *ordered_columns=*

Parameters

- **req** – managed dictionary
- **sort_on_index** – sort the dataframe on this column name
- **ordered_columns** – column list to rename the inverse transform

2.7 Preparing a New Dataset

Helper for preparing a new dataset.

antinex_utils.prepare_dataset_tools.**find_all_headers**(*use_log_id=None*, *pipeline_files=[]*, *label_rules=None*)

Parameters

- **use_log_id** – label for debugging in logs
- **pipeline_files** – list of files to prep
- **label_rules** – dict of rules to apply

antinex_utils.prepare_dataset_tools.**build_csv**(*pipeline_files=[]*, *fulldata_file=None*, *clean_file=None*, *post_proc_rules=None*, *label_rules=None*, *use_log_id=None*, *meta_suffix='metadata.json'*)

Parameters

- **pipeline_files** – list of files to process
- **fulldata_file** – output of non-edited merged data

- **clean_file** – cleaned csv file should be ready for training
- **post_proc_rules** – apply these rules to post processing (clean)
- **label_rules** – apply labeling rules (classification only)
- **use_log_id** – label for tracking the job in the logs
- **meta_suffix** – file suffix

antinex_utils.prepare_dataset_tools.**find_all_pipeline_csvs**(*use_log_id=None*, *csv_glob_path='/opt/antinex/datasets/**/*.csv*

Parameters

- **use_log_id** – label for logs
- **csv_glob_path** – path to files to process

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index