Elliot

Release Elliot

Vito Walter Anelli, Alejandro Bellogín, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Tommaso Di Noia

Mar 01, 2021

Table of Contents

1	elliot	package	93
	1.1	Subpackages	. 5
	1.2	Submodules	93
		elliot.run module	
	1.4	Module contents	93
	Pytho	on Module Index	95
	Index	· · · · · · · · · · · · · · · · · · ·	101

#ELLIOT

«Every hacker has her fixation. You hack people, I hack time.»

```
\label{eq:control_state} $$ ![PyPI - Python Version](https://img.shields.io/pypi/pyversions/scikit-daisy) $$ [![Version](https://img.shields.io/badge/version-v1.1.2-orange)](https://github.com/sisinflab/elliot) $$ ![GitHub](https://img.shields.io/github/license/sisinflab/elliot) $$ ![GitHub](https://img.shields.io/github/license/sisinflab/elliot) $$ $$ $$ $$ | https://img.shields.io/github/license/sisinflab/elliot) $$ | h
```

Elliot is a comprehensive recommendation framework that analyzes the recommendation problem from the researcher's perspective. It conducts a whole experiment, from dataset loading to results gathering. The core idea is to feed the system with a simple and straightforward configuration file that drives the framework through the experimental setting choices. Elliot untangles the complexity of combining splitting strategies, hyperparameter model optimization, model training, and the generation of reports of the experimental results.

The framework loads, filters, and splits the data considering a vast set of strategies (splitting methods and filtering approaches, from temporal training-test splitting to nested K-folds Cross-Validation). Elliot optimizes hyperparameters for several recommendation algorithms, selects the best models, compares them with the baselines providing intra-model statistics, computes metrics spanning from accuracy to beyond-accuracy, bias, and fairness, and conducts statistical analysis (Wilcoxon and Paired t-test).

Elliot aims to keep the entire experiment reproducible and put the user in control of the framework. ## Installation Elliot works with the following operating systems:

- Linux
- Windows 10
- macOS X

Elliot requires Python version 3.6 or later.

RecBole requires tensorflow version 2.4.1 or later. If you want to use Elliot with GPU, please ensure that CUDA or cudatoolkit version is XXX.XXX or later. This requires NVIDIA driver version >= XXX.XXX (for Linux) or >= XXX.XXX (for Windows10).

Install from source

```
#### CONDA `bash git clone https://github.com//sisinflab/elliot.git && cd
elliot conda create --name elliot_env python=3.8 conda activate pip install
-e . --verbose `
```

```
#### VIRTUALENV `bash git clone https://github.com//sisinflab/elliot.git &&
cd elliot python3 -m venv ./venv source venv/bin/activate pip install -e .
--verbose `
```

Quick Start

Elliot's entry point is the function run_experiment, which accepts a configuration file that drives the

whole experiment. In the following, a sample configuration file is shown to demonstrate how a sample and explicit structure can generate a rigorous experiment.

```python from elliot.run import run\_experiment

```
run_experiment("configuration/file/path") ```
```

The following file is a simple configuration for an experimental setup. It contains all the instructions to get the MovieLens-1M catalog from a specific path and perform a train test split in a random sample way with a ratio of 20%.

This experiment provides a hyperparameter optimization with a grid search strategy for an Item-KNN model. Indeed, it is seen that the possible values of neighbors are closed in squared brackets. It indicates that two different models equipped with two different neighbors' values will be trained and compared to select the best configuration. Moreover, this configuration obliges Elliot to save the recommendation lists with at most 10 items per user as suggest by top\_k property.

In this basic experiment, only a simple metric is considered in the final evaluation study. The candidate metric is nDCG for a cutoff equal to top\_k, unless otherwise noted.

```yaml experiment:

```
dataset: movielens_1m data_config:
    strategy: dataset dataset_path: ../data/movielens_1m/dataset.tsv splitting:
        test_splitting:
        strategy: random_subsampling test_ratio: 0.2

models:
    ItemKNN:
        meta:
            hyper_opt_alg: grid save_recs: True
            neighbors: [50, 100] similarity: cosine

evaluation:
    simple_metrics: [nDCG]

top_k: 10
```

Contributing

There are many ways to contribute to Elliot! You can contribute code, make improvements to the documentation, report or investigate [bugs and issues](https://github.com/sisinflab/elliot/issues)

We welcome all contributions from bug fixes to new features and extensions.

Feel free to share with us your custom configuration files. We are creating a vault of reproducible experiments, and we would be glad of mentioning your contribution.

Reference Elliot in your blogs, papers, and articles.

Talk about Elliot on social media with the hashtag #elliotrs.

The Team Elliot is developed by * Vito Walter Anelli^{[*](#f1)} (vitowalter.anelli@poliba.it) * Alejandro Bellogín (alejandro.bellogin@uam.es) * Antonio Ferrara (antonio.ferrara@poliba.it) * Daniele Malitesta (daniele.malitesta@poliba.it) * Felice Antonio Merra (felice.merra@poliba.it) * Claudio Pomo^{[*](#f1)} (claudio.pomo@poliba.it) * Tommaso Di Noia (tommaso.dinoia@poliba.it)

It is maintained by [SisInfLab Group](http://sisinflab.poliba.it/) and [Information Retrieval Group](http://ir.ii.uam.es/).

d="f1"><sup>* Corresponding authors ## License ELLIOT uses [APACHE2 License](./LICENSE).

Acknowledgements

We refer to the following repositories to improve our code:

2 Chapter .

• SliM and KNN-CF parts with [RecSys2019_DeepLearning_Evaluation](https://github.com/MaurizioFD/RecSys2019_DeepLearning_Evaluation]

. 3

4 Chapter .

elliot package

1.1 Subpackages

1.1.1 elliot.dataset package

```
Subpackages
elliot.dataset.dataloader package
Submodules
elliot.dataset.dataloader.knowledge_aware_chains module
Module description:
class
elliot.dataset.dataloader.knowledge_aware_chains.KnowledgeChainsDataObject (
config, data_tuple, side_information_data, *args, **kwargs )
    Bases: object
    Load train and test dataset
    build_dict ( dataframe, users )
    build_sparse()
    build_sparse_ratings()
    dataframe_to_dict ( data )
    get_test()
    get_validation()
class elliot.dataset.dataloader.knowledge_aware_chains.KnowledgeChainsLoader (
config, *args, **kwargs)
    Bases: object
    Load train and test dataset
    check_timestamp ( d: pandas.core.frame.DataFrame ) → pandas.core.frame.DataFrame
    generate_dataobjects ( ) \rightarrow List[object]
    generate\_dataobjects\_mock() \rightarrow List[object]
```

load_attribute_file (attribute_file, separator='\t')

```
load_dataset_dataframe ( file_ratings, separator='\t', attribute_file=None, feature_file=None,
    properties_file=None, column_names=['userId', 'itemId', 'rating', 'timestamp'], additive=True,
    threshold=10)
    load_dataset_dict ( file_ratings, separator='\t', attribute_file=None, feature_file=None,
    properties_file=None, additive=True, threshold=10 )
    load_feature_names ( infile, separator='\t' )
    load_item_set ( ratings_file, separator='\t', itemPosition=1 )
    load_properties ( properties_file )
    read_splitting ( folder_path )
    reduce_attribute_map_property_selection ( map, items, feature_names, properties,
    additive, threshold=10)
    reduce_dataset_by_item_list ( ratings_file, items, separator='\t')
elliot.dataset.dataloader.visual dataloader module
Module description:
class elliot.dataset.dataloader.visual_dataloader.VisualDataObject (
data_tuple, side_information_data, *args, **kwargs )
    Bases: object
    Load train and test dataset
    build_dict ( dataframe, users )
    build_sparse()
    build_sparse_ratings()
    dataframe_to_dict ( data )
    get_test()
    get_validation()
    read_images ( images_folder, image_set, size_tuple )
    read_images_multiprocessing (images_folder, image_set, size_tuple )
    static read single image (images folder, image set, size tuple, image path)
class elliot.dataset.dataloader.visual_dataloader.VisualLoader ( config, *args,
**kwargs)
    Bases: object
    Load train and test dataset
    check_timestamp ( d: pandas.core.frame.DataFrame ) → pandas.core.frame.DataFrame
    generate_dataobjects ( ) \rightarrow List[object]
    generate_dataobjects_mock() \rightarrow List[object]
                                         file_ratings,
                                                        separator=' \ t',
                                                                         visual_feature_set=None,
    load_dataset_dataframe
    column_names=['userId', 'itemId', 'rating', 'timestamp'] )
    read_splitting ( folder_path )
```

```
reduce_dataset_by_item_list ( ratings_file, items, separator='\t')
Module contents
elliot.dataset.samplers package
Submodules
elliot.dataset.samplers.custom_pointwise_sparse_sampler module
Module description:
         elliot.dataset.samplers.custom_pointwise_sparse_sampler.Sampler
indexed_ratings, sp_i_train )
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.custom_sampler module
Module description:
class elliot.dataset.samplers.custom_sampler.Sampler(indexed_ratings)
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.custom_sparse_sampler module
Module description:
class elliot.dataset.samplers.custom_sparse_sampler.Sampler ( indexed_ratings, sp_i_-
train)
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.pairwise_sampler module
Module description:
class elliot.dataset.samplers.pairwise_sampler.Sampler ( ratings, users, items )
    Bases: object
    step ( events: int )
elliot.dataset.samplers.pipeline_sampler module
Module description:
class elliot.dataset.samplers.pipeline_sampler.Sampler (indexed_ratings, item_indices,
images_path, output_image_size, epochs )
    Bases: object
    pipeline ( num_users, batch_size )
    pipeline_eval ( batch_size )
    read_image ( item )
    read_images_triple ( user, pos, neg )
    step ( events: int, batch_size: int )
elliot.dataset.samplers.pointwise_cfgan_sampler module
Module description:
```

```
class elliot.dataset.samplers.pointwise cfgan sampler.Sampler ( indexed ratings,
sp_i_train, s_zr, s_pm)
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.pointwise_pos_neg_ratings_sampler module
Module description:
                                                                                              (
        elliot.dataset.samplers.pointwise_pos_neg_ratings_sampler.Sampler
indexed_ratings, sparse_i_ratings )
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.pointwise_pos_neg_ratio_ratings_sampler module
Module description:
class elliot.dataset.samplers.pointwise_pos_neg_ratio_ratings_sampler.Sampler
(indexed_ratings, sparse_i_ratings, neg_ratio)
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.pointwise pos neg sampler module
Module description:
class elliot.dataset.samplers.pointwise_pos_neg_sampler.Sampler (indexed_ratings)
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.pointwise_wide_and_deep_sampler module
Module description:
class elliot.dataset.samplers.pointwise_wide_and_deep_sampler.Sampler ( data )
    Bases: object
    step ( events: int, batch_size: int )
elliot.dataset.samplers.sparse sampler module
Module description:
class elliot.dataset.samplers.sparse_sampler.Sampler (sp_i_train)
    Bases: object
    step ( users: int, batch_size: int )
Module contents
Module description:
Submodules
elliot.dataset.abstract_dataset module
class elliot.dataset.abstract_dataset.AbstractDataset(*args, **kwargs)
    Bases: object
    abstract build_dict()
    abstract build_sparse ( *args )
```

```
abstract get_test ( *args )
    required_attributes = ['config', 'args', 'kwargs', 'users', 'items', 'num_users', 'num_items',
    'private_users', 'public_users', 'private_items', 'public_items', 'transactions', 'train_dict',
    'i_train_dict', 'sp_i_train', 'test_dict']
class \verb| elliot.dataset.abstract_dataset.ForceRequiredAttributeDefinitionMeta|\\
    Bases: type
    check_required_attributes ( class_object )
elliot.dataset.dataset module
Module description:
class elliot.dataset.dataset.DataSet (*args, **kwargs)
    Bases: elliot.dataset.abstract_dataset.AbstractDataset
    Load train and test dataset
    build_dict ( dataframe, users )
    build_sparse()
    build_sparse_ratings()
    dataframe_to_dict ( data )
    get_test()
    get_validation()
class elliot.dataset.dataset.DataSetLoader ( config, *args, **kwargs )
    Bases: object
    Load train and test dataset
    check_timestamp ( d: pandas.core.frame.DataFrame ) → pandas.core.frame.DataFrame
    generate_dataobjects () \rightarrow List[object]
    generate\_dataobjects\_mock() \rightarrow List[object]
    read_splitting ( folder_path )
Module contents
```

Module description:

1.1.2 elliot.evaluation package

Subpackages

```
elliot.evaluation.metrics package
Subpackages
elliot.evaluation.metrics.accuracy package
Subpackages
elliot.evaluation.metrics.accuracy.AUC package
Submodules
elliot.evaluation.metrics.accuracy.AUC.auc module
```

This is the implementation of the global AUC metric. It proceeds from a system-wise computation.

class elliot.evaluation.metrics.accuracy.AUC.auc.AUC (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the global AUC recommendation metric. Passing 'AUC' to the metrics list will enable the computation of the metric.

Note:

This metric does not calculate group-based AUC which considers the AUC scores averaged across users. It is also not limited to k. Instead, it calculates the scores on the entire prediction results regardless the users.

```
\label{eq:mathrm aluc} $$\operatorname{AUC} = \frac{(M+1)}{2}  \{\{M\} \times \{N\}\}\}$
```

M is the number of positive samples. N is the number of negative samples. rank_i is the ascending rank of the ith positive sample.

```
eval()
```

Evaluation function :return: the overall value of AUC

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

```
static needs_full_recommendations()
```

elliot.evaluation.metrics.accuracy.AUC.gauc module

This is the implementation of the GroupAUC metric. It proceeds from a user-wise computation, and average the AUC values over the users.

class elliot.evaluation.metrics.accuracy.AUC.gauc.GAUC (recommendations, config,
params, eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the GroupAUC recommendation metric. Passing 'GAUC' to the metrics list will enable the computation of the metric.

Note:

It calculates the AUC score of each user, and finally obtains GAUC by weighting the user AUC. It is also not limited to k. Due to our padding for *scores_tensor* in *RankEvaluator* with *-np.inf*, the padding value will influence the ranks of origin items. Therefore, we use descending sort here and make an identity transformation to the formula of *AUC*, which is shown in *auc_* function. For readability, we didn't do simplification in the code.

 $\label{eq:mathrm} $$ \{GAUC\} = \frac{\{\{\{M\} \setminus \{(M+N+1)\} - \frac{M}{2}\} - \sum_{i=1}^M \frac{\{\{M\} \setminus \{M\}\}}{\{M\} \setminus \{M\}\}} - \frac{M+N+1}{2}} - \frac{M+M+1}} - \frac{M+N+1}{2}} - \frac{M+N+1}{2}} - \frac{M+N+1}{2}} - \frac{M+N+1}{2$

M is the number of positive samples. N is the number of negative samples. rank_i is the descending rank of the ith positive sample.

```
eval()
```

Evaluation function :return: the overall averaged value of AUC

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of AUC per user

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

```
static needs_full_recommendations()
```

elliot.evaluation.metrics.accuracy.AUC.lauc module

This is the implementation of the Limited AUC metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.accuracy.AUC.lauc.LAUC ( recommendations, config,
params, eval_objects )
         Bases: elliot.evaluation.metrics.base_metric.BaseMetric
         This class represents the implementation of the Limited AUC recommendation metric. Passing
         'LAUC' to the metrics list will enable the computation of the metric.
         eval_user_metric()
                 Evaluation function :return: the overall averaged value of LAUC per user
         static name ( )
                 Metric Name Getter :return: returns the public name of the metric
Module contents
elliot.evaluation.metrics.accuracy.DSC package
Submodules
elliot.evaluation.metrics.accuracy.DSC.dsc module
This is the implementation of the Sørensen-Dice coefficient metric. It proceeds from a user-wise
computation, and average the values over the users.
class elliot.evaluation.metrics.accuracy.DSC.dsc.DSC (recommendations, config, params,
eval_objects, additional_data )
         Bases: elliot.evaluation.metrics.base_metric.BaseMetric
         This class represents the implementation of the Sørensen-Dice coefficient recommendation
         metric. Passing 'DSC' to the metrics list will enable the computation of the metric.
                 mathrm \ \{F1@K\} = frac\{1+beta^{2}\}\{frac\{1\}\{text \ \{precision@k \}\}+frac\{beta^{2}\}\{text \ \{recall@k\}\}\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\{text \ \{precision@k \}\}+frac\{beta^{2}\}\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\{text \ \{precision@k \}\}+frac\{beta^{2}\}\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\{text \ \{precision@k \}\}+frac\{beta^{2}\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\{text \ \{precision@k \}\}+frac\{beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\{text \ \{precision@k \}\}+frac\{beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1\}\}\} = frac\{1+beta^{2}\}\{frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\{frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\}\} = frac\{1+beta^{2}\} = frac\{1+beta^{2
                 }}}
         eval_user_metric()
                 Evaluation function :return: the overall averaged value of Sørensen-Dice coefficient per user
         static name ( )
                 Metric Name Getter :return: returns the public name of the metric
Module contents
elliot.evaluation.metrics.accuracy.f1 package
Submodules
elliot.evaluation.metrics.accuracy.f1.extended_f1 module
This is the implementation of the F-score metric. It proceeds from a user-wise computation, and
average the values over the users.
                        elliot.evaluation.metrics.accuracy.fl.extended_fl.ExtendedF1
                                                                                                                                                                                                       (
recommendations, config, params, eval_objects, additional_data)
         Bases: elliot.evaluation.metrics.base_metric.BaseMetric
         This class represents the implementation of the F-score recommendation metric. Passing 'Extend-
         edF1' to the metrics list will enable the computation of the metric.
         eval_user_metric()
         get ( )
         static name ( )
                 Metric Name Getter :return: returns the public name of the metric
         process()
                  Evaluation function :return: the overall value of Bias Disparity
```

elliot.evaluation.metrics.accuracy.f1.f1 module

This is the implementation of the F-score metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.f1.f1.F1 (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the F-score recommendation metric. Passing 'F1' to the metrics list will enable the computation of the metric.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of F-score

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.accuracy.hit_rate package

Submodules

elliot.evaluation.metrics.accuracy.hit_rate.hit_rate module

This is the implementation of the Hit Rate metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.hit_rate.hit_rate.HR (recommendations:
 Dict[int, List[Tuple[int, float]]], config, params, eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Hit Rate recommendation metric. Passing 'HR' to the metrics list will enable the computation of the metric.

 $\label{lem:mathrm} $$\operatorname{HR}@K$ = \frac{\operatorname{Number \ space \ of \ space \ Hits \ @K}{|GT|}$}$

 ${\tt HR}$ is the number of users with a positive sample in the recommendation list. ${\tt GT}$ is the total number of samples in the test set.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Hit Rate per user

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.accuracy.map package

Submodules

elliot.evaluation.metrics.accuracy.map.map module

This is the implementation of the Mean Average Precision metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.map.map.MAP (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Mean Average Precision recommendation metric. Passing 'MAP' to the metrics list will enable the computation of the metric.

Note:

In this case the normalization factor used is $\{ \min (m, N) \}$, which prevents your AP score from being unfairly suppressed when your number of recommendations couldn't possibly capture all the correct ones.

 $\left(AP@N \right) = \frac{1}{mathrm{min}(m,N)}$

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Mean Average Precision per user

static name ()

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.accuracy.mar package

Submodules

elliot.evaluation.metrics.accuracy.mar.mar module

This is the implementation of the Mean Average Recall metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.mar.mar.MAR (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Mean Average Recall recommendation metric. Passing 'MAR' to the metrics list will enable the computation of the metric.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Mean Average Recall per user

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.accuracy.mrr package

Submodules

elliot.evaluation.metrics.accuracy.mrr.mrr module

This is the implementation of the Mean Reciprocal Rank metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.mrr.mrr.MRR (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Mean Reciprocal Rank recommendation metric. Passing 'MRR' to the metrics list will enable the computation of the metric.

```
\mathrm {MRR} = \frac{1}{|\{U\}|} \sum_{i=1}^{|\{U\}|} \frac{1}{rank_i}
```

U is the number of users, $rank_i$ is the rank of the first item in the recommendation list in the test set results for user i.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Mean Reciprocal Rank per user

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.accuracy.ndcg package

Submodules

elliot.evaluation.metrics.accuracy.ndcg.ndcg module

This is the implementation of the normalized Discounted Cumulative Gain metric. It proceeds from a

user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.ndcg.ndcg.NDCG (recommendations, config,
params, eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the nDCG recommendation metric. Passing 'nDCG' to the metrics list will enable the computation of the metric.

 $\label{log_continuous} $$\left\{ \frac{DCG@K}=\sum_{i=1}^{K} \frac{2^{rel_i}-1}{\log_{2}{(i+1)}} \right. $$ \mathrm $$\{IDCG@K\}=\sum_{i=1}^{K}\frac{1}{\log_{2}{(i+1)}} \mathrm $$\{NDCG_u@K\}=\frac{DCG_u@K}{IDCG_u@K} \mathrm $$\{NDCG_u@K\}=\frac{DCG_u@K}{u^{te}} \n u^{te}NDCG_u@K} \$

K stands for recommending K items. And the rel_i is the relevance of the item in position i in the recommendation list. 2^{ℓ} equals to 1 if the item hits otherwise 0. U^{ℓ} is for all users in the test set.

```
compute_idcg ( user, cutoff: int ) \rightarrow float
```

Method to compute Ideal Discounted Cumulative Gain :param gain_map: :param cutoff: :return:

```
compute_user_ndcg ( user\_recommendations: List, user, cutoff: int ) <math>\rightarrow float
```

Method to compute normalized Discounted Cumulative Gain :param sorted_item_predictions: :param gain_map: :param cutoff: :return:

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of normalized Discounted Cumulative Gain per user

```
static name ( )
```

Metric Name Getter: return: returns the public name of the metric

Module contents

This is the nDCG metric module.

This module contains and expose the recommendation metric.

elliot.evaluation.metrics.accuracy.precision package

Submodules

elliot.evaluation.metrics.accuracy.precision.precision module

This is the implementation of the Precision metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.accuracy.precision.precision.Precision
recommendations, config, params, eval_objects)
```

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Precision recommendation metric. Passing 'Precision' to the metrics list will enable the computation of the metric.

```
\mathbf{Precision@K} = \frac{| Rel_u \land Rec_u |}{Rec_u}
```

Rel_u is the set of items relevant to user U, Rec_u is the top K items recommended to users. We obtain the result by calculating the average Precision@K of each user.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Precision

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

This is the Precision metric module.

This module contains and expose the recommendation metric.

elliot.evaluation.metrics.accuracy.recall package

Submodules

elliot.evaluation.metrics.accuracy.recall.recall module

This is the implementation of the Recall metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.accuracy.recall.recall.Recall (recommendations,
config, params, eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Recall recommendation metric. Passing 'Recall' to the metrics list will enable the computation of the metric.

. _Recall: https://en.wikipedia.org/wiki/Precision_and_recall#Recall

 $\mathcal{K} = \frac{|Rel_u \times Rec_u|}{|Rel_u \times Rec_u|}$

Rel_u is the set of items relevant to user U, Rec_u is the top K items recommended to users. We obtain the result by calculating the average Recall@K of each user.

```
eval_user_metric()
```

Evaluation Function :return: the overall averaged value of Recall per user

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

This is the Recall metric implementation.

This module contains and expose the recommendation metric.

Module contents

elliot.evaluation.metrics.bias package

Subpackages

elliot.evaluation.metrics.bias.aclt package

Submodules

elliot.evaluation.metrics.bias.aclt.aclt module

This is the implementation of the Average coverage of long tail items metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.bias.aclt.aclt.ACLT (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Average coverage of long tail items recommendation metric. Passing 'ACLT' to the metrics list will enable the computation of the metric.

Himan Abdollahpouri, Robin Burke, Bamshad Mobasher Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, 2019

 U_{t} is the number of users in the test set. L_{u} is the recommended list of items for user u. 1 (i \in \Gamma) is an indicator function and it equals to 1 when i is in Gamma.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of ACLT

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.bias.aplt package

Submodules

elliot.evaluation.metrics.bias.aplt.aplt module

This is the implementation of the Average percentage of long tail items metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.bias.aplt.aplt.APLT (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Average percentage of long tail items recommendation metric. Passing 'APLT' to the metrics list will enable the computation of the metric.

Abdollahpouri, H.; Burke, R.; and Mobasher Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017

 U_{t} is the number of users in the test set. L_{u} is the recommended list of items for user u. $\gamma \in \mathbb{R}$ medium-tail items.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of APLT

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.bias.arp package

Submodules

elliot.evaluation.metrics.bias.arp.arp module

This is the implementation of the Average Recommendation Popularity metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.bias.arp.arp.ARP (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base metric.BaseMetric
```

This class represents the implementation of the Average Recommendation Popularity recommendation metric. Passing 'ARP' to the metrics list will enable the computation of the metric.

Yin, H.; Cui, B.; Li, J.; Yao, J.; and Chen, C. 2012. Challenging the long tail recommendation. Proceedings of the VLDB Endowment

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of ARP

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.bias.pop_reo package

Submodules

elliot.evaluation.metrics.bias.pop reo.extended pop reo module

This is the implementation of the Popularity-based Ranking-based Equal Opportunity (REO) metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.bias.pop_reo.extended_pop_reo.ExtendedPopREO (
recommendations, config, params, eval_objects, additional_data)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Popularity-based Ranking-based Equal Opportunity (REO) recommendation metric. Passing 'ExtendedPopREO' to the metrics list will enable the computation of the metric.

Zhu, Ziwei, Jianling Wang, and James Caverlee. "Measuring and Mitigating Item Under-Recom-

mendation Bias in Personalized Ranking Systems." Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

```
eval()
```

Evaluation function :return: the overall averaged value of PopREO

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

elliot.evaluation.metrics.bias.pop_reo.pop_reo module

This is the implementation of the Popularity-based Ranking-based Equal Opportunity (REO) metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.bias.pop_reo.pop_reo.PopREO ( recommendations,
config, params, eval_objects )
```

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Popularity-based Ranking-based Equal Opportunity (REO) recommendation metric. Passing 'PopREO' to the metrics list will enable the computation of the metric.

Zhu, Ziwei, Jianling Wang, and James Caverlee. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

 $\label{left} $$\operatorname{REO}=\frac{\left(\operatorname{std} \left(P \right) - \operatorname{g-g_{1}}, \ y=1\right) \cdot P\left(R(a) + g_{A}, \ y=1\right) \cdot \left(\operatorname{g-g_{1}}, \ y=1\right) \cdot P\left(R(a) + g_{A}, \ y$

Y\left (u, R_{u, i}\right) identifies the ground-truth label of a user-item pair $left(u, R_{u, i})$ is liked by user?, returns 1, otherwise 0

```
\sum_{i=1}^{k} G_{g_a}}\left(R_{u,i}\right) Y\left(u, R_{u,i}\right)  counts how many items in test set from group g_a are ranked in top-? for user u
```

 $\sum_{i \in \mathbb{Z}_{a}} (i) \ Y(u, i)$ counts the total number of items from group $\{g_a\}$? in test set for user u

eval()

Evaluation function :return: the overall averaged value of PopREO

static name ()

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.bias.pop_rsp package

Submodules

elliot.evaluation.metrics.bias.pop rsp.extended pop rsp module

This is the implementation of the Popularity-based Ranking-based Statistical Parity (RSP) metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.bias.pop_rsp.extended_pop_rsp.ExtendedPopRSP (
recommendations, config, params, eval_objects, additional_data)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Popularity-based Ranking-based Statistical Parity (RSP) recommendation metric. Passing 'ExtendedPopRSP' to the metrics list will enable the computation of the metric.

Zhu, Ziwei, Jianling Wang, and James Caverlee. "Measuring and Mitigating Item Under-Recommendation Bias in Personalized Ranking Systems." Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

```
eval()
```

Evaluation function :return: the overall averaged value of PopRSP

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

elliot.evaluation.metrics.bias.pop_rsp.pop_rsp module

This is the implementation of the Popularity-based Ranking-based Statistical Parity (RSP) metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.bias.pop_rsp.pop_rsp.PopRSP (recommendations,
config, params, eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Popularity-based Ranking-based Statistical Parity (RSP) recommendation metric. Passing 'PopRSP' to the metrics list will enable the computation of the metric.

Zhu, Ziwei, Jianling Wang, and James Caverlee. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

```
\label{left:continuous} $$\operatorname{RSP}=\frac{std}\left(P\left(R @ k \right), \cdot g_{1}\right), \cdot g_{1}\right),
```

 $\sum_{i=1}^{k} G_{g_a} \left(R_{u, i}\right)$ calculates how many un-interacted items from group g_a are ranked in top-? for user u.

 $\label{lem:condition} $$\sum_{i \in \mathbb{Z}_{a}} (i) \ calculates \ how many un-interacted items belong to group $\{g_a\}$ for u$

eval()

Evaluation function :return: the overall averaged value of PopRSP

static name ()

Metric Name Getter :return: returns the public name of the metric

Module contents

Module contents

elliot.evaluation.metrics.coverage package

Subpackages

elliot.evaluation.metrics.coverage.item coverage package

Submodules

elliot.evaluation.metrics.coverage.item_coverage.item_coverage module

This is the implementation of the Item Coverage metric. It directly proceeds from a system-wise computation, and it considers all the users at the same time.

class

```
elliot.evaluation.metrics.coverage.item_coverage.item_coverage.ItemCoverage (recommendations, config, params, eval_objects)
```

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Item Coverage recommendation metric. Passing 'ItemCoverage' to the metrics list will enable the computation of the metric.

Ricci F, Rokach L, Shapira B, Kantor P. 2015

Note:

The simplest measure of catalog coverage is the percentage of all items that can ever be recommended. This measure can be computed in many cases directly given the algorithm and the input data set.

```
eval()
```

Evaluation function :return: the overall averaged value of Item Coverage

static name ()

Metric Name Getter: return: returns the public name of the metric

Module contents

This is the Item Coverage metric module.

This module contains and expose the recommendation metric.

elliot.evaluation.metrics.coverage.num_retrieved package

Submodules

elliot.evaluation.metrics.coverage.num_retrieved.num_retrieved module

This is the implementation of the NumRetrieved metric. It proceeds from a user-wise computation, and average the values over the users.

class

elliot.evaluation.metrics.coverage.num_retrieved.num_retrieved.NumRetrieved (recommendations, config, params, eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the NumRetrieved recommendation metric. Passing 'NumRetrieved' to the metrics list will enable the computation of the metric.

```
eval_user_metric()
```

Evaluation function: return: the overall averaged value of NumRetrieved

```
static name ( )
```

Metric Name Getter: return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.coverage.user_coverage package

Submodules

elliot.evaluation.metrics.coverage.user_coverage.user_coverage module

This is the implementation of the User Coverage metric. It directly proceeds from a system-wise computation, and it considers all the users at the same time.

class

```
elliot.evaluation.metrics.coverage.user_coverage.user_coverage.UserCoverage (recommendations, config, params, eval_objects)
```

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the User Coverage recommendation metric. Passing 'UserCoverage' to the metrics list will enable the computation of the metric.

Ricci F, Rokach L, Shapira B, Kantor P. 2015

Note:

The proportion of users or user interactions for which the system can recommend items. In many applications the recommender may not provide recommendations for some users due to, e.g. low confidence in the accuracy of predictions for that user.

```
eval()
```

Evaluation function :return: the overall averaged value of User Coverage

static name ()

Metric Name Getter: return: returns the public name of the metric

elliot.evaluation.metrics.coverage.user_coverage.user_coverage_at_n module

This is the implementation of the User Coverage metric. It directly proceeds from a system-wise computation, and it considers all the users at the same time.

```
class
```

elliot.evaluation.metrics.coverage.user_coverage.user_coverage_at_n.**UserCoverageAtN** (recommendations, config, params, eval_objects)

Bases: elliot.evaluation.metrics.base_metric.BaseMetric

This class represents the implementation of the User Coverage recommendation metric. Passing 'UserCoverageAtN' to the metrics list will enable the computation of the metric.

eval()

Evaluation function :return: the overall averaged value of User Coverage

static name ()

Metric Name Getter :return: returns the public name of the metric

Module contents

Module contents

elliot.evaluation.metrics.diversity package

Subpackages

elliot.evaluation.metrics.diversity.SRecall package

Submodules

elliot.evaluation.metrics.diversity.SRecall.srecall module

This is the implementation of the SRecall metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.diversity.SRecall.srecall.SRecall
    (recommendations, config, params, eval_objects, additional_data)
```

Bases: elliot.evaluation.metrics.base_metric.BaseMetric

This class represents the implementation of the SRecall recommendation metric. Passing 'SRecall' to the metrics list will enable the computation of the metric.

3. 24. Zhai, W. W. Cohen, and J. Lafferty, 2003

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of SRecall

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

elliot.evaluation.metrics.diversity.gini_index package

Submodules

elliot.evaluation.metrics.diversity.gini_index.gini_index module

This is the implementation of the Gini Index metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.diversity.gini_index.gini_index.GiniIndex
recommendations, config, params, eval_objects)
```

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Gini Index recommendation metric. Passing 'Gini' to the metrics list will enable the computation of the metric.

Ricci F, Rokach L, Shapira B, Kantor P. 2015

 $\label{lem:linear_series} $$ \mathbf{GiniIndex} = \frac{1}{n-1} \sum_{j=1}^{n}(2 j-n-1) p \left(i_{j}\right)\right) $$$

 i_{j} is the list of items ordered according to increasing p(i)

```
eval()
        Evaluation function :return: the overall averaged value of Gini Index
    static name ( )
        Metric Name Getter: return: returns the public name of the metric
Module contents
elliot.evaluation.metrics.diversity.shannon entropy package
Submodules
elliot.evaluation.metrics.diversity.shannon_entropy.shannon_entropy module
This is the implementation of the Shannon Entropy metric. It proceeds from a user-wise computation,
and average the values over the users.
class
elliot.evaluation.metrics.diversity.shannon_entropy.shannon_entropy.ShannonEntropy
( recommendations, config, params, eval_objects )
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the Shannon Entropy recommendation metric.
    Passing 'SEntropy' to the metrics list will enable the computation of the metric.
    Ricci F, Rokach L, Shapira B, Kantor P. 2015
    \mathrm{ShannonEntropy}=-\mathrm{Sum}_{i=1}^{n} p(i) \log p(i)
    eval()
        Evaluation function :return: the overall value of Shannon Entropy
    static name ( )
        Metric Name Getter :return: returns the public name of the metric
Module contents
Module contents
elliot.evaluation.metrics.fairness package
Subpackages
elliot.evaluation.metrics.fairness.BiasDisparity package
Submodules
elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBD module
This is the implementation of the Bias Disparity metric. It proceeds from a user-wise computation,
and average the values over the users.
class
elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBD.BiasDisparityBD
( recommendations, config, params, eval_objects, additional_data )
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the Bias Disparity recommendation metric. Passing
    'BiasDisparityBD' to the metrics list will enable the computation of the metric.
    Tsintzou, Virginia, Evaggelia Pitoura, and Panayiotis Tsaparas. Proceedings of the Workshop on
    Recommendation in Multi-stakeholder Environments co-located with the 13th {ACM} Conference
    on Recommender Systems (RecSys 2019)
    \mathrm{BD}(G, C) = \mathrm{FR}(G, C) - B_{S}(G, C) + B_{S}(G, C)
    eval()
    get ()
    name()
```

1.1. Subpackages 21

Metric Name Getter :return: returns the public name of the metric

```
process ( )
```

Evaluation function :return: the overall value of Bias Disparity

elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBR module

This is the implementation of the Bias Disparity - Bias Recommendations metric. It proceeds from a user-wise computation, and average the values over the users.

class

elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBR.BiasDisparityBR (recommendations, config, params, eval_objects, additional_data)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Bias Disparity - Bias Recommendations recommendation metric. Passing 'BiasDisparityBR' to the metrics list will enable the computation of the metric.

Tsintzou, Virginia, Evaggelia Pitoura, and Panayiotis Tsaparas. Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th {ACM} Conference on Recommender Systems (RecSys 2019)

```
\mathrm {BD(G, C)}=
rac{B_{R}(G, C)-B_{S}(G, C)}{B_{S}(G, C)}
eval()
get()
get_BR()
name()
    Metric Name Getter :return: returns the public name of the metric
process()
```

Evaluation function :return: the overall value of Bias Disparity - Bias Recommendations

elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBS module

This is the implementation of the Bias Disparity - Bias Source metric. It proceeds from a user-wise computation, and average the values over the users.

class

elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBS.BiasDisparityBS (recommendations, config, params, eval_objects, additional_data)

```
Bases: \verb|elliot.evaluation.metrics.base\_metric.BaseMetric|\\
```

This class represents the implementation of the Bias Disparity - Bias Source recommendation metric. Passing 'BiasDisparityBS' to the metrics list will enable the computation of the metric.

Tsintzou, Virginia, Evaggelia Pitoura, and Panayiotis Tsaparas. Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th {ACM} Conference on Recommender Systems (RecSys 2019)

```
\mathrm {B_{S} (G, C)}=
rac{P R_{S}(G, C)}{P(C)}
eval()
get_BS()
```

```
name ( )
        Metric Name Getter :return: returns the public name of the metric
    process ( )
        Evaluation function :return: the overall value of Bias Disparity - Bias Source
Module contents
elliot.evaluation.metrics.fairness.MAD package
Submodules
elliot.evaluation.metrics.fairness.MAD.ItemMADranking module
This is the implementation of the Item MAD ranking metric. It proceeds from a user-wise computa-
tion, and average the values over the users.
class elliot.evaluation.metrics.fairness.MAD.ItemMADranking.ItemMADranking (
recommendations, config, params, eval_objects, additional_data)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the Item MAD ranking recommendation metric.
    Passing 'ItemMADranking' to the metrics list will enable the computation of the metric.
          Deldjoo, Yashar, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogin, and Tommaso Di
          Noia. User Modeling and User-Adapted Interaction (2020): 1-47.
           \mathcal{L}_{avg}_{i, j}(\{MAD\}(R^{(i)}, R^{(j)}))
    Math \{MAD\}=\{avg\}_{\{i, j\}}(\{MAD\}\} \{t(R^{(i)}, R^{(j)})\})
    eval()
        Evaluation function :return: the overall averaged value of Item MAD ranking
    get ( )
    name()
        Metric Name Getter :return: returns the public name of the metric
elliot.evaluation.metrics.fairness.MAD.ItemMADrating module
This is the implementation of the Item MAD rating metric. It proceeds from a user-wise computation,
and average the values over the users.
       elliot.evaluation.metrics.fairness.MAD.ItemMADrating.ItemMADrating
                                                                                                 (
recommendations, config, params, eval_objects, additional_data)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the Item MAD rating recommendation metric.
    Passing 'ItemMADrating' to the metrics list will enable the computation of the metric.
    Zhu, Ziwei, Xia Hu, and James Caverlee. Proceedings of the 27th ACM International Conference
    on Information and Knowledge Management. 2018.
    \mathcal{MAD}=\{avg\} \{i, j\}(\{MAD\}(R^{(i)}\}, R^{(j)}\})
    Math \{MAD\}=\{avg\}_{i, j}(\{MAD\}\{i(R^{(i)}\}, R^{(j)}\})\}
        Evaluation function :return: the overall averaged value of Item MAD rating
    get ()
    name ( )
        Metric Name Getter :return: returns the public name of the metric
```

This is the implementation of the User MAD ranking metric. It proceeds from a user-wise computation, and average the values over the users.

1.1. Subpackages

elliot.evaluation.metrics.fairness.MAD.UserMADranking module

```
class elliot.evaluation.metrics.fairness.MAD.UserMADranking.UserMADranking (
recommendations, config, params, eval_objects, additional_data)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the User MAD ranking recommendation metric.
    Passing 'UserMADranking' to the metrics list will enable the computation of the metric.
           \mathcal{MAD} = \{avg\}_{\{i, j\}} (\{MAD\} (R^{(i)}\}, R^{(j)}\})
    Math \{MAD\}=\{avg\}_{\{i, j\}}(\{MAD\}\} \{t(R^{(i)}, R^{(j)})\})
    compute_idcg ( user: int, cutoff: int ) \rightarrow float
        Method to compute Ideal Discounted Cumulative Gain :param gain_map: :param cutoff:
        :return:
    compute_user_ndcg ( user\_recommendations: List, user: int, cutoff: int ) <math>\rightarrow float
        Method to compute normalized Discounted Cumulative Gain :param sorted_item_predic-
        tions: :param gain_map: :param cutoff: :return:
    eval()
        Evaluation function :return: the overall averaged value of User MAD ranking
    get ()
    name ( )
        Metric Name Getter :return: returns the public name of the metric
elliot.evaluation.metrics.fairness.MAD.UserMADrating module
This is the implementation of the User MAD rating metric. It proceeds from a user-wise computation,
and average the values over the users.
       elliot.evaluation.metrics.fairness.MAD.UserMADrating.UserMADrating
recommendations, config, params, eval_objects, additional_data)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the User MAD rating recommendation metric.
    Passing 'UserMADrating' to the metrics list will enable the computation of the metric.
    Zhu, Ziwei, Xia Hu, and James Caverlee. Proceedings of the 27th ACM International Conference
    on Information and Knowledge Management. 2018.
    \mathcal{MAD}=\{avg\}_{\{i, j\}}(\{MAD\}(R^{(i)}, R^{(j)}))
    Math \{MAD\}=\{avg\}_{\{i, j\}}(\{MAD\}\{left(R^{(i)}\}, R^{(j)}\}))
    eval()
        Evaluation function :return: the overall averaged value of User MAD rating
    get ()
    name()
        Metric Name Getter :return: returns the public name of the metric
Module contents
This is the Precision metric module.
```

This module contains and expose the recommendation metric.

elliot.evaluation.metrics.fairness.reo package

Submodules

elliot.evaluation.metrics.fairness.reo.reo module

This is the implementation of the Ranking-based Equal Opportunity (REO) metric. It proceeds from a user-wise computation, and average the values over the users.

25

```
class elliot.evaluation.metrics.fairness.reo.reo.REO ( recommendations, config, params,
eval_objects, additional_data )
```

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Ranking-based Equal Opportunity (REO) recommendation metric. Passing 'REO' to the metrics list will enable the computation of the metric.

Zhu, Ziwei, Jianling Wang, and James Caverlee. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

 $\label{left} $$\operatorname{REO}=\frac{\left(\operatorname{std} \left(P \right) - \operatorname{g-g_{1}}, \ y=1\right) \cdot P\left(R(a) + g_{A}, \ y=1\right) \cdot \left(\operatorname{g-g_{1}}, \ y=1\right) \cdot P\left(R(a) + g_{A}, \ y$

Y\left (u, R_{u, i}\right) identifies the ground-truth label of a user-item pair $left(u, R_{u, i})$ is liked by user?, returns 1, otherwise 0

 $\label{lem:counts} $$\sum_{i \in \mathbb{Z}_a} i \in \mathbb{Z}_a . $$(i) Y(u, i) counts the total number of items from group $\{g_a\}$? in test set for user $u$$

```
eval()
get()
name()
```

Metric Name Getter :return: returns the public name of the metric

```
process ( )
```

Evaluation function :return: the overall value of Ranking-based Equal Opportunity (REO)

Module contents

elliot.evaluation.metrics.fairness.rsp package

Submodules

elliot.evaluation.metrics.fairness.rsp.rsp module

This is the implementation of the Ranking-based Statistical Parity (RSP) metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.fairness.rsp.rsp.RSP (recommendations, config, params,
eval_objects, additional_data)

```
Bases: \verb|elliot.evaluation.metrics.base\_metric.BaseMetric|\\
```

This class represents the implementation of the Ranking-based Statistical Parity (RSP) recommendation metric. Passing 'RSP' to the metrics list will enable the computation of the metric.

Zhu, Ziwei, Jianling Wang, and James Caverlee. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020.

```
\mathrm {RSP}=
```

$rac\{\{std\}(P(R @ k mid g=g_{1}\}), ldots, P(R @ k mid g=g_{1}\})\}$

```
 \{\{mean\}(P(R@k\ mid\ g=g_{1}),\ ldots,\ P(R@k\ mid\ g=g_{A}))\}  P(R@k\mid\ g=g_{a}) \} is`
```

$rac\{sum_{u=1}^{N} sum_{i=1}^{k} G_{g_{a}}(R_{u, i})\}$

```
\{sum_{u=1}^{N} sum_{i in I ?ackslash I_{u}^{+}} G_{g_{a}}(i)\}
```

 $\sum_{i=1}^{k} G_{g_a} (R_u, i)$ calculates how many un-interacted items from group g_a are ranked in top-? for user u.

 $\sum_{i \in \{g_{a}\}(i)} I$ ackslash $I_{u}^{+}\} G_{g_{a}}(i)$ calculates how many un-in-

teracted items belong to group $\{g_a\}$ for u

```
eval()
    get ()
    name ( )
        Metric Name Getter :return: returns the public name of the metric
    process()
        Evaluation function :return: the overall value of Ranking-based Statistical Parity (RSP)
Module contents
Module contents
elliot.evaluation.metrics.novelty package
Subpackages
elliot.evaluation.metrics.novelty.EFD package
Submodules
elliot.evaluation.metrics.novelty.EFD.efd module
This is the implementation of the Expected Free Discovery metric. It proceeds from a user-wise
computation, and average the values over the users.
class elliot.evaluation.metrics.novelty.EFD.efd.EFD ( recommendations, config, params,
eval_objects)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
         This class represents the implementation of the Expected Free Discovery recommendation
         metric. Passing 'EFD' to the metrics list will enable the computation of the metric.
         Note:
              EFD can be read as the expected ICF of seen recommended items
           -\log_{2} p(i \neq seen),
                                                  heta))
    eval_user_metric()
        Evaluation function :return: the overall averaged value of Expected Free Discovery per user
    static name ( )
        Metric Name Getter :return: returns the public name of the metric
elliot.evaluation.metrics.novelty.EFD.extended efd module
This is the implementation of the Expected Free Discovery metric. It proceeds from a user-wise
computation, and average the values over the users.
                                                                                             (
          elliot.evaluation.metrics.novelty.EFD.extended_efd.ExtendedEFD
recommendations, config, params, eval_objects, additional_data)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the Expected Free Discovery recommendation metric.
    Passing 'ExtendedEFD' to the metrics list will enable the computation of the metric.
    eval_user_metric()
        Evaluation function :return: the overall averaged value of Expected Free Discovery per user
    static name ( )
        Metric Name Getter :return: returns the public name of the metric
Module contents
elliot.evaluation.metrics.novelty.EPC package
Submodules
```

elliot.evaluation.metrics.novelty.EPC.epc module

This is the implementation of the Expected Popularity Complement metric. It proceeds from a user-wise computation, and average the values over the users.

class elliot.evaluation.metrics.novelty.EPC.epc.EPC (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Expected Popularity Complement recommendation metric. Passing 'EPC' to the metrics list will enable the computation of the metric.

S. Vargas and P. Castells Proceedings of RecSys 2011

Note:

EPC can be read as the expected number of seen relevant recommended items not previously seen

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Expected Popularity Complement per user

```
static name ( )
```

Metric Name Getter: return: returns the public name of the metric

elliot.evaluation.metrics.novelty.EPC.extended_epc module

This is the implementation of the Expected Popularity Complement metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.novelty.EPC.extended_epc.ExtendedEPC
recommendations, config, params, eval_objects, additional_data )
```

```
Bases: elliot.evaluation.metrics.base metric.BaseMetric
```

This class represents the implementation of the Expected Popularity Complement recommendation metric. Passing 'ExtendedEPC' to the metrics list will enable the computation of the metric.

```
eval_user_metric()
```

Evaluation function :return: the overall averaged value of Expected Popularity Complement per user

```
static name ( )
```

Metric Name Getter :return: returns the public name of the metric

Module contents

Module contents

elliot.evaluation.metrics.rating package

Subpackages

elliot.evaluation.metrics.rating.mae package

Submodules

elliot.evaluation.metrics.rating.mae.mae module

This is the implementation of the Mean Absolute Error metric. It proceeds from a system-wise computation.

class elliot.evaluation.metrics.rating.mae.mae.MAE (recommendations, config, params,
eval_objects)

```
Bases: elliot.evaluation.metrics.base_metric.BaseMetric
```

This class represents the implementation of the Mean Absolute Error recommendation metric. Passing 'MAE' to the metrics list will enable the computation of the metric.

```
\mathbf{MAE}=\frac{1}{|T}| \sum_{u, i}r_{u, i} \inf T}\right
```

```
T is the test set, \hat{r}_{u} i } is the score predicted by the model, and r_{u} i } the actual
    score of the test set.
    eval()
        Evaluation function :return: the overall averaged value of Mean Absolute Error
    eval_user_metric()
        Evaluation function :return: the overall averaged value of Mean Absolute Error per user
    static name ( )
        Metric Name Getter :return: returns the public name of the metric
    static needs_full_recommendations()
Module contents
elliot.evaluation.metrics.rating.mse package
Submodules
elliot.evaluation.metrics.rating.mse.mse module
This is the implementation of the Mean Squared Error metric. It proceeds from a system-wise compu-
tation.
class elliot.evaluation.metrics.rating.mse.mse.MSE ( recommendations, config, params,
eval_objects)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
          This class represents the implementation of the Mean Squared Error recommendation
          metric. Passing 'MSE' to the metrics list will enable the computation of the metric.
           \mathbb{MSE} =
    rac{1}{|T}| sum_{(u, i)} in {T}(hat{r}_{u i}-r_{u i})^{2}
          T is the test set, \hat{r}_{u} i is the score predicted by the model, and r_{u} i the
          actual score of the test set.
    eval()
        Evaluation function :return: the overall averaged value of Mean Squared Error
    eval_user_metric()
        Evaluation function :return: the overall averaged value of Mean Squared Error per user
    static name ( )
        Metric Name Getter :return: returns the public name of the metric
    static needs_full_recommendations()
Module contents
elliot.evaluation.metrics.rating.rmse package
Submodules
elliot.evaluation.metrics.rating.rmse.rmse module
This is the implementation of the Root Mean Squared Error metric. It proceeds from a user-wise
computation, and average the values over the users.
class elliot.evaluation.metrics.rating.rmse.rmse.RMSE ( recommendations, config,
params, eval_objects)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    This class represents the implementation of the Root Mean Squared Error recommendation
    metric. Passing 'RMSE' to the metrics list will enable the computation of the metric.
    \mathbf{RMSE} = \sqrt{T}[\{T\}] \sum_{(u, i) \in T}(\hat{T}_{u i}-r_{u i})^{2}
    T is the test set, \hat{r}_{u} i } is the score predicted by the model, and r_{u} i } the actual
```

```
score of the test set.
    eval()
        Evaluation function :return: the overall averaged value of Root Mean Squared Error
    eval_user_metric()
        Evaluation function :return: the overall averaged value of Root Mean Squared Error
    static name ( )
        Metric Name Getter :return: returns the public name of the metric
    static needs_full_recommendations()
Module contents
Module contents
Submodules
elliot.evaluation.metrics.base metric module
This is the implementation of the Precision metric. It proceeds from a user-wise computation, and
average the values over the users.
class elliot.evaluation.metrics.base_metric.BaseMetric ( recommendations, config,
params, evaluation_objects, additional_data=None )
    Bases: abc.ABC
    This class represents the implementation of the Precision recommendation metric. Passing 'Preci-
    sion' to the metrics list will enable the computation of the metric.
    eval()
    get ()
    abstract name ( )
    static needs_full_recommendations()
elliot.evaluation.metrics.metrics utils module
class elliot.evaluation.metrics.metrics_utils.ProxyMetric ( name='ProxyMetric',
val=0, needs_full_recommendations=False )
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    eval()
    name ( )
    needs_full_recommendations()
         elliot.evaluation.metrics.metrics_utils.ProxyStatisticalMetric
name='ProxyMetric', val=0, user_val=0, needs_full_recommendations=False)
    Bases: elliot.evaluation.metrics.base_metric.BaseMetric
    eval()
    eval_user_metric()
    name ( )
    needs_full_recommendations()
elliot.evaluation.metrics.statistical_array_metric module
```

This is the implementation of the Precision metric. It proceeds from a user-wise computation, and average the values over the users.

```
class elliot.evaluation.metrics.statistical_array_metric.StatisticalMetric
Bases: object
```

This class represents the implementation of the Precision recommendation metric. Passing 'Precision' to the metrics list will enable the computation of the metric.

```
abstract eval_user_metric()
```

Module contents

This is the metrics' module.

This module contains and expose the recommendation metrics. Each metric is encapsulated in a specific package.

See the implementation of Precision metric for creating new per-user metrics. See the implementation of Item Coverage for creating new cross-user metrics.

```
elliot.evaluation.metrics.parse_metric ( metric )
elliot.evaluation.metrics.parse_metrics ( metrics )
elliot.evaluation.popularity utils package
Submodules
elliot.evaluation.popularity_utils.popularity module
Module description: This module provides a popularity class based on number of users who have
experienced an item (user-item repetitions in the dataset are counted once)
class elliot.evaluation.popularity_utils.popularity.Popularity (data, pop_ratio=0.8
    Bases: object
    get_custom_pop_obj ( pop_ratio=0.8 )
    get_long_tail()
    get_pop_items()
    get_short_head()
    get_sorted_pop_items()
Module contents
elliot.evaluation.relevance package
Submodules
elliot.evaluation.relevance.relevance module
Module description:
class elliot.evaluation.relevance.relevance.AbstractRelevanceSingleton
    Bases: abc.ABC
    abstract get_rel ( user, item )
    static logarithmic_ranking_discount (k:int) \rightarrow float
        Method to compute logarithmic discount :param k: :return:
class elliot.evaluation.relevance.relevance.BinaryRelevance(test, rel_threshold)
    Bases: elliot.evaluation.relevance.relevance.AbstractRelevanceSingleton
    get_rel ( user, item )
```

get_user_rel (user)

```
get_user_rel_gains ( user )
class elliot.evaluation.relevance.relevance.DiscountedRelevance (test, rel_threshold
    Bases: elliot.evaluation.relevance.relevance.AbstractRelevanceSingleton
    get_rel ( user, item )
    get_user_rel ( user )
    get_user_rel_gains ( user )
class elliot.evaluation.relevance.relevance.Relevance(test, rel_threshold)
    Bases: object
    property binary_relevance
    property discounted_relevance
    get_test()
Module contents
Module description:
Submodules
elliot.evaluation.evaluator module
Module description:
class elliot.evaluation.evaluator.Evaluator ( data: elliot.dataset.dataset.DataSet, params:
types.SimpleNamespace )
    Bases: object
    eval (recommendations )
        Runtime Evaluation of Accuracy Performance (top-k):return:
    eval_at_k ( recommendations, k )
    get_needed_recommendations()
elliot.evaluation.statistical_significance module
Module description:
class elliot.evaluation.statistical_significance.PairedTTest
    Bases: object
    static common_users ( arr_0: Dict[int, float], arr_1: Dict[int, float] )
    static compare ( arr_0: Dict[int, float], arr_1: Dict[int, float], users: List[int] )
class elliot.evaluation.statistical_significance.WilcoxonTest
    Bases: object
    static common_users ( arr_0: Dict[int, float], arr_1: Dict[int, float] )
    static compare ( arr_0: Dict[int, float], arr_1: Dict[int, float], users: List[int] )
Module contents
```

1.1. Subpackages 31

Module description:

1.1.3 elliot.hyperoptimization package

Submodules

elliot.hyperoptimization.model_coordinator module

Module description:

class elliot.hyperoptimization.model_coordinator.ModelCoordinator (data_objs, base:
types.SimpleNamespace, params, model_class: ClassVar)

Bases: object

This class handles the selection of hyperparameters for the hyperparameter tuning realized with HyperOpt.

```
objective ( args )
```

This function respect the signature, and the return format required for HyperOpt optimization :param args: a Dictionary that contains the new hyper-parameter values that will be used in the current run :return: it returns a Dictionary with loss, and status being required by HyperOpt, and params, and results being required by the framework

```
single()
```

This function respect the signature, and the return format required for HyperOpt optimization :param args: a Dictionary that contains the new hyper-parameter values that will be used in the current run :return: it returns a Dictionary with loss, and status being required by HyperOpt, and params, and results being required by the framework

Module contents

Module description:

```
elliot.hyperoptimization.parse_algorithms (opt_alg)
elliot.hyperoptimization.suggest (new_ids, domain, trials, seed, nbMaxSucessiveFailures=1000)
```

1.1.4 elliot.namespace package

Submodules

elliot.namespace_namespace_model module

Module description:

elliot.namespace.namespace model builder module

Module description:

```
class elliot.namespace.namespace_model_builder.Builder
```

The Builder interface specifies methods for creating the different parts of the Product objects.

abstract property base

```
abstract models () \rightarrow None
```

class elliot.namespace.namespace_model_builder.NameSpaceBuilder (config_path,
base_folder_path_elliot, base_folder_path_config)

Bases: elliot.namespace.namespace_model_builder.Builder

property base

 $models() \rightarrow tuple$

Module contents

Module description:

1.1.5 elliot.prefiltering package

Submodules

elliot.prefiltering.standard_prefilters module

```
Bases: object
```

class elliot.prefiltering.standard_prefilters.PreFilter

static filter (d: pandas.core.frame.DataFrame, ns: types.SimpleNamespace) \rightarrow pandas.core.frame.DataFrame

static filter_items_by_popularity (d: pandas.core.frame.DataFrame, threshold) \rightarrow pandas.core.frame.DataFrame

static filter_iterative_k_core (*d: pandas.core.frame.DataFrame, threshold*) → pandas.core.frame.DataFrame

static filter_ratings_by_global_average (d: pandas.core.frame.DataFrame) \rightarrow pandas.core.frame.DataFrame

 $\begin{tabular}{ll} \textbf{static filter_ratings_by_threshold} (\textit{d: pandas.core.frame.DataFrame, threshold} \) \rightarrow \textbf{pandas.core.frame.DataFrame} \\ \end{tabular}$

static filter_ratings_by_user_average (d: pandas.core.frame.DataFrame) \rightarrow pandas.core.frame.DataFrame

static filter_retain_cold_users (d: pandas.core.frame.DataFrame, threshold) \rightarrow pandas.core.frame.DataFrame

static filter_rounds_k_core (*d: pandas.core.frame.DataFrame, threshold, n_rounds*) → pandas.core.frame.DataFrame

Module contents

Module description:

1.1.6 elliot.recommender package

```
Subpackages
elliot.recommender.NN package
Subpackages
elliot.recommender.NN.attribute_item_knn package
Submodules
elliot.recommender.NN.attribute item knn.attribute item knn module
Module description:
elliot.recommender.NN.attribute_item_knn.attribute_item_knn.AttributeItemKNN
( data, config, params, *args, **kwargs )
    Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
   build_feature_sparse()
    get_recommendations (k: int = 100)
    property name
    restore_weights()
    train()
elliot.recommender.NN.attribute item knn.attribute item knn similarity module
class
elliot.recommender.NN.attribute_item_knn.attribute_item_knn_similarity.Similarity
( data, attribute_matrix, num_neighbors, similarity )
    Bases: object
    Simple kNN class
    compute_cosine ( i_index, j_index )
    compute_neighbors()
    get_item_neighbors ( item )
    get_model_state()
    get_transactions()
    get_user_recs(u,k)
    initialize()
        This function initialize the data model
   process_similarity ( similarity )
    static score_item ( neighs, user_items )
    set_model_state ( saving_dict )
Module contents
elliot.recommender.NN.attribute user knn package
Submodules
```

elliot.recommender.NN.attribute_user_knn.attribute_user_knn module

Module description:

```
class
elliot.recommender.NN.attribute_user_knn.attribute_user_knn.AttributeUserKNN
(data, config, params, *args, **kwargs)
                            elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
   build_feature_sparse()
   build_feature_sparse_values()
    compute_binary_profile ( user_items_dict: Dict )
    get_recommendations (k: int = 100)
    property name
    restore_weights()
    train()
elliot.recommender.NN.attribute_user_knn.attribute_user_knn_similarity module
elliot.recommender.NN.attribute_user_knn.attribute_user_knn_similarity.Similarity
( data, attribute_matrix, num_neighbors, similarity )
    Bases: object
    Simple kNN class
    compute_neighbors()
    get_model_state()
    get_transactions()
    get_user_neighbors ( item )
    get_user_recs ( u, k )
    initialize()
       This function initialize the data model
    process_similarity ( similarity )
    static score_item ( neighs, user_neighs_items )
    set_model_state ( saving_dict )
elliot.recommender.NN.attribute user knn.tfidf utils module
class elliot.recommender.NN.attribute_user_knn.tfidf_utils.TFIDF ( map: Dict[int,
List[int]])
    Bases: object
    get_profiles ( ratings: Dict[int, Dict[int, float]] )
    tfidf()
Module contents
```

```
elliot.recommender.NN.item_knn package
Submodules
elliot.recommender.NN.item knn.aiolli ferrari module
Created on 23/10/17 @author: Maurizio Ferrari Dacrema
class elliot.recommender.NN.item_knn.aiolli_ferrari.AiolliSimilarity (
                                                                                       data.
maxk=40, shrink=100, similarity='cosine', normalize=True')
    Bases: object
    get_user_recs ( user, k=100 )
    initialize()
   predict ( u, i )
        elliot.recommender.NN.item_knn.aiolli_ferrari.Compute_Similarity
           topK=100,
                         shrink=0,
                                    normalize=True,
                                                     asymmetric alpha=0.5,
                                                                           tversky alpha=1.0,
tversky_beta=1.0, similarity='cosine', row_weights=None)
    Bases: object
    applyAdjustedCosine()
        Remove from every data point the average for the corresponding row :return:
    applyPearsonCorrelation()
        Remove from every data point the average for the corresponding column :return:
    compute_similarity ( start_col=None, end_col=None, block_size=100 )
        Compute the similarity for the given dataset :param self: :param start_col: column to begin
        with :param end_col: column to stop before, end_col is excluded :return:
    useOnlyBooleanInteractions()
elliot.recommender.NN.item knn.aiolli ferrari.check matrix(X, format='csc', dtype-
=<class 'numpy.float32'>)
    This function takes a matrix as input and transforms it into the specified format. The matrix in
    input can be either sparse or ndarray. If the matrix in input has already the desired format, it is
    returned as-is the dtype parameter is always applied and the default is np.float32 :param X:
    :param format: :param dtype: :return:
elliot.recommender.NN.item knn.item knn module
Module description:
class elliot.recommender.NN.item_knn.item_knn.ItemKNN ( data, config, params, *args,
**kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    restore_weights()
    train()
elliot.recommender.NN.item_knn_similarity module
class elliot.recommender.NN.item_knn.item_knn_similarity.Similarity (
                                                                                        data.
num_neighbors, similarity )
    Bases: object
```

```
Simple kNN class
    compute_cosine ( i_index, j_index )
    compute_neighbors()
    get_item_neighbors ( item )
    get_model_state()
    get_transactions()
    get_user_recs(u,k)
    initialize()
        This function initialize the data model
    process_cosine()
    process_similarity ( similarity )
    static score_item ( neighs, user_items )
    set_model_state ( saving_dict )
Module contents
elliot.recommender.NN.user_knn package
Submodules
elliot.recommender.NN.user_knn.aiolli_ferrari module
Created on 23/10/17 @author: Maurizio Ferrari Dacrema
class elliot.recommender.NN.user_knn.aiolli_ferrari.AiolliSimilarity (
maxk=40, shrink=100, similarity='cosine', normalize=True')
    Bases: object
    get_user_recs ( user, k=100 )
    initialize()
   predict (u, i)
class
        elliot.recommender.NN.user_knn.aiolli_ferrari.Compute_Similarity
                                                     asymmetric_alpha=0.5,
                                                                           tversky_alpha=1.0,
dataMatrix,
             topK=100,
                        shrink=0, normalize=True,
tversky_beta=1.0, similarity='cosine', row_weights=None')
    Bases: object
    applyAdjustedCosine()
        Remove from every data point the average for the corresponding row :return:
    applyPearsonCorrelation()
        Remove from every data point the average for the corresponding column :return:
    compute_similarity ( start_col=None, end_col=None, block_size=100 )
        Compute the similarity for the given dataset :param self: :param start_col: column to begin
        with :param end_col: column to stop before, end_col is excluded :return:
    useOnlyBooleanInteractions()
elliot.recommender.NN.user_knn.aiolli_ferrari.check_matrix(X,format='csc',dtype-
=<class 'numpy.float32'>)
    This function takes a matrix as input and transforms it into the specified format. The matrix in
```

input can be either sparse or ndarray. If the matrix in input has already the desired format, it is returned as-is the dtype parameter is always applied and the default is np.float32 :param X: :param format: :param dtype: :return:

```
elliot.recommender.NN.user knn.user knn module
```

```
Module description:
class elliot.recommender.NN.user_knn.user_knn.UserKNN ( data, config, params, *args,
**kwargs)
   Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
   elliot.recommender.base_recommender_model.BaseRecommenderModel
   get_recommendations (k: int = 100)
   property name
   restore_weights()
   train()
elliot.recommender.NN.user_knn.user_knn_similarity module
class elliot.recommender.NN.user knn.user knn similarity.Similarity (
                                                                                    data.
num_neighbors, similarity )
   Bases: object
   Simple kNN class
   compute_neighbors()
   get_model_state()
   get_transactions()
   get_user_neighbors ( item )
   get_user_recs ( u, k )
   initialize()
       This function initialize the data model
   process_similarity ( similarity )
   static score_item ( neighs, user_neighs_items )
   set_model_state ( saving_dict )
Module contents
Module contents
elliot.recommender.adversarial package
Subpackages
elliot.recommender.adversarial.AMF package
Submodules
elliot.recommender.adversarial.AMF.AMF module
Module description:
class elliot.recommender.adversarial.AMF.AMF ( data, config, params, *args, **kwargs )
                            elliot.recommender.recommender_utils_mixin.RecMixin,
   elliot.recommender.base_recommender_model.BaseRecommenderModel
   get_recommendations (k: int = 100)
```

property name

```
train()
```

elliot.recommender.adversarial.AMF.AMF_model module

Module description:

```
class elliot.recommender.adversarial.AMF.AMF_model.AMF_model (*args, **kwargs)
Bases: tensorflow.python.keras.engine.training.Model
```

build_perturbation (batch)

Evaluate Adversarial Perturbation with FGSM-like Approach

```
call (inputs, training=None)
```

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

get_config()

Returns the config of the layer.

A layer config is a Python dictionary (serializable) containing the configuration of a layer. The same layer can be reinstantiated later (without its trained weights) from this configuration.

The config of a layer does not include connectivity information, nor the layer class name. These are handled by *Network* (one layer of abstraction above).

Returns:

Python dictionary.

```
get_positions ( predictions, train_mask, items, inner_test_user_true_mask )
get_top_k ( predictions, train_mask, k=100 )
predict ( start, stop, **kwargs )
```

Generates output predictions for the input samples.

Computation is done in batches. This method is designed for performance in large scale inputs. For small amount of inputs that fit in one batch, directly using __call__ is recommended for faster execution, e.g., model(x), or model(x, training=False) if you have layers such as tf.keras.layers.BatchNormalization that behaves differently during inference. Also, note the fact that test loss is not affected by regularization layers like noise and dropout.

Arguments:

x: Input samples. It could be:

- A Numpy array (or array-like), or a list of arrays (in case the model has multiple inputs).
- A TensorFlow tensor, or a list of tensors (in case the model has multiple inputs).
- A tf.data dataset.

• A generator or *keras.utils.Sequence* instance.

A more detailed description of unpacking behavior for iterator types (Dataset, generator, Sequence) is given in the *Unpacking behavior for iterator-like inputs* section of *Model.fit*.

batch_size: Integer or None.

Number of samples per batch. If unspecified, *batch_size* will default to 32. Do not specify the *batch_size* if your data is in the form of dataset, generators, or *keras.util-s.Sequence* instances (since they generate batches).

verbose: Verbosity mode, 0 or 1. steps: Total number of steps (batches of samples)

before declaring the prediction round finished. Ignored with the default value of *None*. If x is a *tf.data* dataset and *steps* is None, *predict* will run until the input dataset is exhausted.

callbacks: List of keras.callbacks.Callback instances.

List of callbacks to apply during prediction. See [callbacks](/api_docs/python/t-f/keras/callbacks).

max_queue_size: Integer. Used for generator or keras.utils.Sequence

input only. Maximum size for the generator queue. If unspecified, <code>max_queue_size</code> will default to 10.

workers: Integer. Used for generator or keras.utils.Sequence input

only. Maximum number of processes to spin up when using process-based threading. If unspecified, *workers* will default to 1. If 0, will execute the generator on the main thread.

use_multiprocessing: Boolean. Used for generator or

keras.utils.Sequence input only. If *True*, use process-based threading. If unspecified, *use_multiprocessing* will default to *False*. Note that because this implementation relies on multiprocessing, you should not pass non-picklable arguments to the generator as they can't be passed easily to children processes.

See the discussion of *Unpacking behavior for iterator-like inputs* for *Model.fit*. Note that Model.predict uses the same interpretation rules as *Model.fit* and *Model.evaluate*, so inputs must be unambiguous for all three methods.

Returns:

Numpy array(s) of predictions.

Raises:

Runtime Error: If model.predict is wrapped in tf.function. Value Error: In case of mismatch between the provided

input data and the model's expectations, or in case a stateful model receives a number of samples that is not a multiple of the batch size.

train_step (batch, user_adv_train=False)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's*

```
metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.
```

Module contents

Module description:

elliot.recommender.adversarial.AMR package

Submodules

elliot.recommender.adversarial.AMR.AMR module

Module description:

```
get_recommendations (k: int = 100)
```

property name

```
train()
```

elliot.recommender.adversarial.AMR.AMR_model module

Module description:

```
class elliot.recommender.adversarial.AMR.AMR_model.AMR_model (*args, **kwargs)
Bases: tensorflow.python.keras.engine.training.Model
```

build_perturbation (batch)

Evaluate Adversarial Perturbation with FGSM-like Approach

```
call ( inputs, training=None, mask=None )
```

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the Network in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

get_config()

Returns the config of the layer.

A layer config is a Python dictionary (serializable) containing the configuration of a layer. The same layer can be reinstantiated later (without its trained weights) from this configuration.

The config of a layer does not include connectivity information, nor the layer class name. These are handled by *Network* (one layer of abstraction above).

Returns:

Python dictionary.

```
get_top_k ( preds, train_mask, k=100 )
```

```
predict ( start, stop )
```

Generates output predictions for the input samples.

Computation is done in batches. This method is designed for performance in large scale

inputs. For small amount of inputs that fit in one batch, directly using <u>__call__</u> is recommended for faster execution, e.g., model(x), or model(x), training=False) if you have layers such as tf.keras.layers.BatchNormalization that behaves differently during inference. Also, note the fact that test loss is not affected by regularization layers like noise and dropout.

Arguments:

x: Input samples. It could be:

- A Numpy array (or array-like), or a list of arrays (in case the model has multiple inputs).
- A TensorFlow tensor, or a list of tensors (in case the model has multiple inputs).
- A tf.data dataset.
- A generator or *keras.utils.Sequence* instance.

A more detailed description of unpacking behavior for iterator types (Dataset, generator, Sequence) is given in the *Unpacking behavior for iterator-like inputs* section of *Model.fit*.

batch_size: Integer or None.

Number of samples per batch. If unspecified, *batch_size* will default to 32. Do not specify the *batch_size* if your data is in the form of dataset, generators, or *keras.util-s.Sequence* instances (since they generate batches).

verbose: Verbosity mode, 0 or 1. steps: Total number of steps (batches of samples)

before declaring the prediction round finished. Ignored with the default value of *None*. If x is a *tf.data* dataset and *steps* is None, *predict* will run until the input dataset is exhausted.

callbacks: List of keras.callbacks.Callback instances.

List of callbacks to apply during prediction. See [callbacks](/api_docs/python/t-f/keras/callbacks).

max_queue_size: Integer. Used for generator or keras.utils.Sequence

input only. Maximum size for the generator queue. If unspecified, <code>max_queue_size</code> will default to 10.

workers: Integer. Used for generator or keras.utils.Sequence input

only. Maximum number of processes to spin up when using process-based threading. If unspecified, *workers* will default to 1. If 0, will execute the generator on the main thread.

use_multiprocessing: Boolean. Used for generator or

keras.utils.Sequence input only. If *True*, use process-based threading. If unspecified, use_multiprocessing will default to False. Note that because this implementation relies on multiprocessing, you should not pass non-picklable arguments to the generator as they can't be passed easily to children processes.

See the discussion of *Unpacking behavior for iterator-like inputs* for *Model.fit*. Note that Model.predict uses the same interpretation rules as *Model.fit* and *Model.evaluate*, so inputs must be unambiguous for all three methods.

Returns:

Numpy array(s) of predictions.

Raises:

RuntimeError: If *model.predict* is wrapped in *tf.function*. ValueError: In case of mismatch between the provided

input data and the model's expectations, or in case a stateful model receives a number of samples that is not a multiple of the batch size.

train_step (batch, user_adv_train=False)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's* metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

Module contents

Module description:

Module contents

elliot.recommender.algebric package

Subpackages

elliot.recommender.algebric.slope one package

Submodules

Module contents

elliot.recommender.algebric.slope_one.slope_one module

Module description: Lemire, Daniel, and Anna Maclachlan. "Slope one predictors for online rating-based collaborative filtering." Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics

elliot.recommender.algebric.slope one.slope one model module

Lemire, Daniel, and Anna Maclachlan. "Slope one predictors for online rating-based collaborative filtering." Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics

```
class elliot.recommender.algebric.slope_one.slope_one_model.SlopeOneModel
data)
   Bases: object
   get_model_state()
   get_user_recs(u,k)
   initialize()
   predict(user,item)
   set_model_state(saving_dict)
```

```
Module contents
elliot.recommender.attentive package
Subpackages
elliot.recommender.attentive.afm package
Submodules
elliot.recommender.attentive.afm.afm module
Module description:
class elliot.recommender.attentive.afm.afm(data, config, params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    property name
    train()
elliot.recommender.attentive.afm.afm model module
Module description:
class elliot.recommender.attentive.afm.afm_model.AFMModel (*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch )
Module contents
Module contents
Module description:
elliot.recommender.autoencoders package
Subpackages
elliot.recommender.autoencoders.dae package
Submodules
elliot.recommender.autoencoders.dae.multi dae module
Module description:
class elliot.recommender.autoencoders.dae.multi_dae.MultiDAE ( data, config, params,
*args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    property name
    train()
```

```
elliot.recommender.autoencoders.dae.multi_dae_model module
Module description:
class elliot.recommender.autoencoders.dae.multi_dae_model.Decoder ( *args, **kwargs
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    Converts z, the encoded vector, back into a uaser interaction vector.
    call (inputs, **kwargs)
class
elliot.recommender.autoencoders.dae.multi_dae_model.DenoisingAutoEncoder
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    Combines the encoder and decoder into an end-to-end model for training.
    call (inputs, training=None, **kwargs)
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get\_top\_k (preds, train\_mask, k=100)
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    train_step ( batch )
class elliot.recommender.autoencoders.dae.multi_dae_model.Encoder ( *args, **kwargs
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    Maps user-item interactions to a triplet (z_mean, z_log_var, z).
    call (inputs, training=None)
Module contents
Module description:
elliot.recommender.autoencoders.vae package
Submodules
elliot.recommender.autoencoders.vae.multi_vae module
Module description:
class elliot.recommender.autoencoders.vae.multi_vae.MultiVAE ( data, config, params,
*args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    property name
```

```
train()
elliot.recommender.autoencoders.vae.multi_vae_model module
Module description:
class elliot.recommender.autoencoders.vae.multi_vae_model.Decoder ( *args, **kwargs
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    Converts z, the encoded digit vector, back into a readable digit.
    call (inputs, **kwargs)
class elliot.recommender.autoencoders.vae.multi_vae_model.Encoder ( *args, **kwargs
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    Maps MNIST digits to a triplet (z_mean, z_log_var, z).
    call (inputs, training=None)
class elliot.recommender.autoencoders.vae.multi_vae_model.Sampling
                                                                                         *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    Uses (z_mean, z_log_var) to sample z, the vector encoding a digit.
    call (inputs)
class
elliot.recommender.autoencoders.vae.multi_vae_model.VariationalAutoEncoder (
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    Combines the encoder and decoder into an end-to-end model for training.
    call (inputs, training=None, **kwargs )
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get\_top\_k (preds, train\_mask, k=100)
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch, anneal_ph=0.0, **kwargs )
Module contents
Module description:
Module contents
Module description:
```

```
elliot.recommender.content_based package
Subpackages
elliot.recommender.content based.VSM package
Submodules
elliot.recommender.content_based.VSM.tfidf_utils module
class elliot.recommender.content_based.VSM.tfidf_utils.TFIDF ( map: Dict[int,
List[int]])
    Bases: object
    get_profiles ( ratings: Dict[int, Dict[int, float]] )
    tfidf()
elliot.recommender.content_based.VSM.vector_space_model module
Module description:
class elliot.recommender.content_based.VSM.vector_space_model.VSM ( data, config,
params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base recommender model.BaseRecommenderModel
    build_feature_sparse ( feature_dict, num_entities )
    build_feature_sparse_values ( feature_dict, num_entities )
    compute_binary_profile ( user_items_dict: Dict )
    get_recommendations (k: int = 100)
    property name
    restore_weights()
    train()
elliot.recommender.content based.VSM.vector space model similarity module
class
elliot.recommender.content_based.VSM.vector_space_model_similarity.Similarity
( data, user_profile_matrix, item_attribute_matrix, similarity )
    Bases: object
    Simple kNN class
    get_model_state()
    get_transactions()
    get_user_recs(u,k)
    initialize()
        This function initialize the data model
    process_similarity ( similarity )
    set_model_state ( saving_dict )
Module contents
Module contents
```

```
elliot.recommender.gan package
Subpackages
elliot.recommender.gan.CFGAN package
Submodules
elliot.recommender.gan.CFGAN.cfgan module
Module description:
class elliot.recommender.gan.CFGAN.cfgan.CFGAN ( data, config, params, *args, **kwargs )
   Bases:
                            elliot.recommender.recommender utils mixin.RecMixin,
   elliot.recommender.base_recommender_model.BaseRecommenderModel
   get recommendations ( k: int = 100 )
   property name
   train()
elliot.recommender.gan.CFGAN.cfgan_model module
Module description:
class elliot.recommender.gan.CFGAN.cfgan_model.CFGAN_model(*args, **kwargs)
   Bases: tensorflow.python.keras.engine.training.Model
   get_config()
```

Returns the config of the layer.

A layer config is a Python dictionary (serializable) containing the configuration of a layer. The same layer can be reinstantiated later (without its trained weights) from this configuration.

The config of a layer does not include connectivity information, nor the layer class name. These are handled by *Network* (one layer of abstraction above).

Returns:

Python dictionary.

```
get_top_k ( predictions, train_mask, k=100 )
predict ( start, stop, **kwargs )
```

Generates output predictions for the input samples.

Computation is done in batches. This method is designed for performance in large scale inputs. For small amount of inputs that fit in one batch, directly using $_call_$ is recommended for faster execution, e.g., model(x), or model(x, training=False) if you have layers such as tf.keras.layers.BatchNormalization that behaves differently during inference. Also, note the fact that test loss is not affected by regularization layers like noise and dropout.

Arguments:

x: Input samples. It could be:

- A Numpy array (or array-like), or a list of arrays (in case the model has multiple inputs).
- A TensorFlow tensor, or a list of tensors (in case the model has multiple inputs).
- A tf.data dataset.
- A generator or *keras.utils.Sequence* instance.

A more detailed description of unpacking behavior for iterator types (Dataset, generator, Sequence) is given in the *Unpacking behavior for iterator-like inputs* section of *Model.fit*.

batch_size: Integer or None.

Number of samples per batch. If unspecified, *batch_size* will default to 32. Do not specify the *batch_size* if your data is in the form of dataset, generators, or *keras.util-s.Sequence* instances (since they generate batches).

verbose: Verbosity mode, 0 or 1. steps: Total number of steps (batches of samples)

before declaring the prediction round finished. Ignored with the default value of *None*. If x is a *tf.data* dataset and *steps* is None, *predict* will run until the input dataset is exhausted.

callbacks: List of keras.callbacks.Callback instances.

List of callbacks to apply during prediction. See [callbacks](/api_docs/python/t-f/keras/callbacks).

max_queue_size: Integer. Used for generator or keras.utils.Sequence

input only. Maximum size for the generator queue. If unspecified, *max_queue_size* will default to 10.

workers: Integer. Used for generator or keras.utils.Sequence input

only. Maximum number of processes to spin up when using process-based threading. If unspecified, *workers* will default to 1. If 0, will execute the generator on the main thread.

use_multiprocessing: Boolean. Used for generator or

keras.utils.Sequence input only. If *True*, use process-based threading. If unspecified, *use_multiprocessing* will default to *False*. Note that because this implementation relies on multiprocessing, you should not pass non-picklable arguments to the generator as they can't be passed easily to children processes.

See the discussion of *Unpacking behavior for iterator-like inputs* for *Model.fit*. Note that Model.predict uses the same interpretation rules as *Model.fit* and *Model.evaluate*, so inputs must be unambiguous for all three methods.

Returns:

Numpy array(s) of predictions.

Raises:

RuntimeError: If *model.predict* is wrapped in *tf.function*. ValueError: In case of mismatch between the provided

input data and the model's expectations, or in case a stateful model receives a number of samples that is not a multiple of the batch size.

train_step (batch)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's* metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

```
class elliot.recommender.gan.CFGAN.cfgan_model.Discriminator(*args, **kwargs)
Bases: tensorflow.python.keras.engine.training.Model
```

 $discriminate_fake_data(X)$

```
train_step ( batch )
```

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function.*

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for how this logic is run (e.g. tf.function and tf.distribute.Strategy settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

Α dict containing values that will be passed tf.keras.callbacks.CallbackList.on_train_batch_end. Typically, the values of the Model's metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

```
class elliot.recommender.gan.CFGAN.cfgan_model.Generator(*args, **kwargs)
   Bases: tensorflow.python.keras.engine.training.Model
   generate_fake_data ( mask, C_u )
   infer(C_u)
   train_step ( batch )
       The logic for one training step.
```

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function.*

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for how this logic is run (e.g. tf.function and tf.distribute.Strategy settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

Δ values that will dict containing he passed to tf.keras.callbacks.CallbackList.on_train_batch_end. Typically, the values of the Model's metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

Module contents

elliot.recommender.gan.IRGAN package

Submodules

elliot.recommender.gan.IRGAN.irgan module

Module description:

```
class elliot.recommender.gan.IRGAN.irgan.IRGAN ( data, config, params, *args, **kwargs )
   Bases:
                          elliot.recommender.recommender_utils_mixin.RecMixin,
   elliot.recommender.base_recommender_model.BaseRecommenderModel
   get_recommendations (k: int = 100)
   property name
   train()
```

elliot.recommender.gan.IRGAN.irgan model module

Module description:

```
class elliot.recommender.gan.IRGAN.irgan_model.Discriminator(*args, **kwargs)
Bases: tensorflow.python.keras.engine.training.Model
```

call (*inputs*, *training=None*, *mask=None*)

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

train_step (batch)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's* metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

```
class elliot.recommender.gan.IRGAN.irgan_model.Generator(*args, **kwargs)
Bases: tensorflow.python.keras.engine.training.Model
```

call (inputs, training=None, mask=None)

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

train_step(batch)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's* metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

```
train_step_with_reward(batch)
```

```
class elliot.recommender.gan.IRGAN.irgan_model.IRGAN_model (*args, **kwargs)
Bases: tensorflow.python.keras.engine.training.Model
```

call (*inputs*, *training=None*)

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

```
get_config()
```

Returns the config of the layer.

A layer config is a Python dictionary (serializable) containing the configuration of a layer. The same layer can be reinstantiated later (without its trained weights) from this configuration.

The config of a layer does not include connectivity information, nor the layer class name. These are handled by *Network* (one layer of abstraction above).

Returns:

Python dictionary.

```
get_positions ( predictions, train_mask, items, inner_test_user_true_mask )
get_top_k ( predictions, train_mask, k=100 )
pre_train_discriminator ( )
pre_train_generator ( )
predict ( start, stop, **kwargs )
```

Generates output predictions for the input samples.

Computation is done in batches. This method is designed for performance in large scale inputs. For small amount of inputs that fit in one batch, directly using $_call_$ is recommended for faster execution, e.g., model(x), or model(x), training=False) if you have layers such as tf.keras.layers.BatchNormalization that behaves differently during inference. Also, note the fact that test loss is not affected by regularization layers like noise and dropout.

Arguments:

x: Input samples. It could be:

A Numpy array (or array-like), or a list of arrays (in case the model has

multiple inputs).

- A TensorFlow tensor, or a list of tensors (in case the model has multiple inputs).
- A *tf.data* dataset.
- A generator or *keras.utils.Sequence* instance.

A more detailed description of unpacking behavior for iterator types (Dataset, generator, Sequence) is given in the *Unpacking behavior for iterator-like inputs* section of *Model.fit*.

batch_size: Integer or None.

Number of samples per batch. If unspecified, *batch_size* will default to 32. Do not specify the *batch_size* if your data is in the form of dataset, generators, or *keras.util-s.Sequence* instances (since they generate batches).

verbose: Verbosity mode, 0 or 1. steps: Total number of steps (batches of samples)

before declaring the prediction round finished. Ignored with the default value of *None*. If x is a *tf.data* dataset and *steps* is None, *predict* will run until the input dataset is exhausted.

callbacks: List of keras.callbacks.Callback instances.

List of callbacks to apply during prediction. See [callbacks](/api_docs/python/t-f/keras/callbacks).

max_queue_size: Integer. Used for generator or keras.utils.Sequence

input only. Maximum size for the generator queue. If unspecified, *max_queue_size* will default to 10.

workers: Integer. Used for generator or keras.utils.Sequence input

only. Maximum number of processes to spin up when using process-based threading. If unspecified, *workers* will default to 1. If 0, will execute the generator on the main thread.

use_multiprocessing: Boolean. Used for generator or

keras.utils.Sequence input only. If *True*, use process-based threading. If unspecified, *use_multiprocessing* will default to *False*. Note that because this implementation relies on multiprocessing, you should not pass non-picklable arguments to the generator as they can't be passed easily to children processes.

See the discussion of *Unpacking behavior for iterator-like inputs* for *Model.fit*. Note that Model.predict uses the same interpretation rules as *Model.fit* and *Model.evaluate*, so inputs must be unambiguous for all three methods.

Returns:

Numpy array(s) of predictions.

Raises

RuntimeError: If *model.predict* is wrapped in *tf.function*. ValueError: In case of mismatch between the provided

input data and the model's expectations, or in case a stateful model receives a number of samples that is not a multiple of the batch size.

train_step()

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

```
data: A nested structure of `Tensor`s.
        Returns:
            Α
                    dict
                            containing
                                            values
                                                        that
                                                                 will
                                                                          be
                                                                                  passed
                                                                                              to
            tf.keras.callbacks.CallbackList.on_train_batch_end. Typically, the values of the Model's
            metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.
Module contents
Module contents
elliot.recommender.graph_based package
Subpackages
elliot.recommender.graph_based.lightgcn package
Submodules
elliot.recommender.graph based.lightgcn.LightGCN module
Module description:
class elliot.recommender.graph_based.lightgcn.LightGCN.LightGCN ( data, config,
params, *args, **kwargs)
                              elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.graph based.lightgcn.LightGCN model module
Module description:
class elliot.recommender.graph_based.lightgcn.LightGCN_model.LightGCNModel (
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, **kwargs)
        Generates prediction for passed users and items indices
            inputs: user, item (batch) the Network in training mode or inference mode.
        Returns:
            prediction and extracted model parameters
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        tion.
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get_top_k ( preds, train_mask, k=100 )
    predict ( start, stop, **kwargs )
    train_step ( batch )
        Apply a single training step on one batch.
```

```
Args:
            batch: batch used for the current train step
        Returns:
            loss value at the current batch
Module contents
elliot.recommender.graph based.ngcf package
Submodules
elliot.recommender.graph based.ngcf.NGCF module
Module description:
class elliot.recommender.graph_based.ngcf.NGCF.NGCF ( data, config, params, *args, **kwargs
)
                              elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.graph based.ngcf.NGCF model module
Module description:
class elliot.recommender.graph_based.ngcf.NGCF_model.NGCFModel(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, **kwargs)
        Generates prediction for passed users and items indices
            inputs: user, item (batch) the Network in training mode or inference mode.
            prediction and extracted model parameters
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get\_top\_k ( preds, train_mask, k=100 )
    predict (start, stop, **kwargs)
    train_step ( batch )
        Apply a single training step on one batch.
            batch: batch used for the current train step
        Returns:
            loss value at the current batch
```

Module contents

```
Module contents
elliot.recommender.knowledge_aware package
Subpackages
elliot.recommender.knowledge_aware.kaHFM package
Submodules
elliot.recommender.knowledge_aware.kaHFM.ka_hfm module
class elliot.recommender.knowledge_aware.kaHFM.ka_hfm.KaHFM ( data, config, params,
*args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
        Get prediction on the user item pair.
        Returns:
            A single float vaue.
    restore_weights()
    train()
    train_step()
    update_factors ( u: int, i: int, j: int )
class elliot.recommender.knowledge_aware.kaHFM.ka_hfm.MF ( ratings: Dict, map: Dict,
tfidf: Dict, user_profiles: Dict, random: Any, *args)
    Bases: object
    Simple Matrix Factorization class
    get_factors()
    get_item_bias ( item: int )
    get_item_factors ( item: int )
    get_model_state()
    get_transactions()
    get_user_bias ( user: int )
    get_user_factors ( user: int )
    get_user_recs ( user: int, k: int )
    get_user_recs_argpartition ( user: int, k: int )
    initialize ( loc: float = 0, scale: float = 0.1 )
        This function initialize the data model :param loc: :param scale: :return:
    property name
    predict ( user: int, item: int )
    set_item_bias ( item: int, v: float )
```

```
set_item_factors ( item: int, v: float )
    set_model_state ( saving_dict )
    set_user_bias ( user: int, v: float )
    set_user_factors ( user: int, v: float )
elliot.recommender.knowledge_aware.kaHFM.tfidf_utils module
class elliot.recommender.knowledge_aware.kaHFM.tfidf_utils.TFIDF ( map: Dict[int,
List[int]])
    Bases: object
    get_profiles ( ratings: Dict[int, Dict[int, float]] )
    tfidf()
Module contents
elliot.recommender.knowledge_aware.kaHFM_batch package
Submodules
elliot.recommender.knowledge aware.kaHFM batch.kahfm batch module
Module description:
class elliot.recommender.knowledge_aware.kaHFM_batch.kahfm_batch.KaHFMBatch (
data, config, params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base recommender model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.knowledge aware.kaHFM batch.kahfm batch model module
Module description:
class
elliot.recommender.knowledge_aware.kaHFM_batch.kahfm_batch_model.KaHFM_model
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, **kwargs)
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get\_top\_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
    predict_all()
```

```
predict_batch ( start, stop )
         train_step ( batch )
elliot.recommender.knowledge aware.kaHFM batch.tfidf utils module
class elliot.recommender.knowledge_aware.kaHFM_batch.tfidf_utils.TFIDF ( map:
Dict[int, List[int]])
         Bases: object
         get_profiles ( ratings: Dict[int, Dict[int, float]] )
         tfidf()
Module contents
elliot.recommender.knowledge_aware.kahfm_embeddings package
Submodules
elliot.recommender.knowledge aware.kahfm embeddings.kahfm embeddings module
Module description:
\verb|elliot.recommender.knowledge_aware.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.kahfm_embeddings.k
(data, config, params, *args, **kwargs)
         Bases:
                                                                 elliot.recommender.recommender_utils_mixin.RecMixin,
         elliot.recommender.base_recommender_model.BaseRecommenderModel
         \texttt{get\_recommendations} ( k: int = 100 )
         property name
         train()
elliot.recommender.knowledge_aware.kahfm_embeddings.kahfm_embeddings_model module
Module description:
class
elliot.recommender.knowledge_aware.kahfm_embeddings.kahfm_embeddings_model.KaHFMEmbeddi
(*args, **kwargs)
         Bases: tensorflow.python.keras.engine.training.Model
         call (inputs, training=None, **kwargs)
         get_config()
                  Returns the config of the layer.
                  A layer config is a Python dictionary (serializable) containing the configuration of a layer.
                  The same layer can be reinstantiated later (without its trained weights) from this configura-
                  The config of a layer does not include connectivity information, nor the layer class name.
                  These are handled by Network (one layer of abstraction above).
                  Returns:
                           Python dictionary.
         get\_top\_k (preds, train\_mask, k=100)
         predict (inputs, training=False, **kwargs )
        predict_batch ( start, stop )
         train_step ( batch )
```

```
elliot.recommender.knowledge_aware.kahfm_embeddings.tfidf_utils module
class elliot.recommender.knowledge_aware.kahfm_embeddings.tfidf_utils.TFIDF (
map: Dict[int, List[int]] )
    Bases: object
    get_profiles ( ratings: Dict[int, Dict[int, float]] )
    tfidf()
Module contents
Module contents
elliot.recommender.latent_factor_models package
Subpackages
elliot.recommender.latent_factor_models.BPRMF package
Submodules
elliot.recommender.latent_factor_models.BPRMF.BPRMF module
Module description:
class elliot.recommender.latent_factor_models.BPRMF.BPRMF.BPRMF ( data, config,
params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
        Get prediction on the user item pair.
        Returns:
            A single float vaue.
    restore_weights()
    train()
    train_step()
    update_factors ( u: int, i: int, j: int )
class elliot.recommender.latent_factor_models.BPRMF.BPRMF.MF ( F, ratings, random,
*args )
    Bases: object
    Simple Matrix Factorization class
    get_item_bias ( item: int )
    get_item_factors ( item: int )
    get_model_state()
    get_transactions()
    get_user_bias ( user: int )
    get_user_factors ( user: int )
    get_user_recs ( user, k )
```

```
get_user_recs_argpartition ( user: int, k: int )
    initialize ( loc: float = 0, scale: float = 0.1 )
        This function initialize the data model :param loc: :param scale: :return:
    property name
    predict ( user, item )
    set_item_bias ( item: int, v: float )
    set_item_factors ( item: int, v: float )
    set_model_state ( saving_dict )
    set_user_bias ( user: int, v: float )
    set_user_factors ( user: int, v: float )
Module contents
Module description:
elliot.recommender.latent_factor_models.BPRMF_batch package
Submodules
elliot.recommender.latent_factor_models.BPRMF_batch.BPRMF_batch module
Module description:
class
elliot.recommender.latent_factor_models.BPRMF_batch.BPRMF_batch.BPRMF_batch
(data, config, params, *args, **kwargs)
    Bases.
                              elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    restore weights ( )
    train()
elliot.recommender.latent_factor_models.BPRMF_batch.BPRMF_batch_model module
Module description:
elliot.recommender.latent_factor_models.BPRMF_batch_BPRMF_batch_model.BPRMF_batch_model
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None)
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
```

```
get positions (predictions, train mask, items, inner test user true mask)
    get_top_k ( predictions, train_mask, k=100 )
    predict ( start, stop, **kwargs )
    train_step ( batch )
Module contents
Module description:
elliot.recommender.latent factor models.BPRSlim package
Submodules
elliot.recommender.latent factor models.BPRSlim.bprslim module
Module description:
class elliot.recommender.latent_factor_models.BPRSlim.bprslim.BPRSlim ( data,
config, params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
        Get prediction on the user item pair.
        Returns:
            A single float vaue.
    restore_weights()
    train()
elliot.recommender.latent factor models.BPRSlim.bprslim model module
Module description:
class
elliot.recommender.latent_factor_models.BPRSlim.bprslim_model.BPRSlimModel (
data, num_users, num_items, lr, lj_reg, li_reg, sampler, random_seed=42)
    Bases: object
    get_model_state( )
    get_user_recs ( user, k=100 )
    predict ( u, i )
    set_model_state ( saving_dict )
    train_step ( batch )
Module contents
Module description:
elliot.recommender.latent_factor_models.CML package
Submodules
elliot.recommender.latent_factor_models.CML.CML module
Module description:
```

```
class elliot.recommender.latent factor models.CML.CML.CML (data, config, params, *args,
**kwargs)
    Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get recommendations ( k: int = 100 )
    property name
    restore_weights()
    train()
elliot.recommender.latent_factor_models.CML.CML_model module
Module description:
class elliot.recommender.latent_factor_models.CML.CML_model.CML_model ( *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None)
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get_positions ( predictions, train_mask, items, inner_test_user_true_mask )
    get_top_k ( predictions, train_mask, k=100 )
    predict ( start, stop, **kwargs )
    train_step ( batch )
class
       elliot.recommender.latent_factor_models.CML.CML_model.LatentFactor
*args, **kwargs)
    Bases: tensorflow.python.keras.layers.embeddings.Embedding
    censor (censor id)
Module contents
Module description:
elliot.recommender.latent_factor_models.FFM package
Submodules
elliot.recommender.latent_factor_models.FFM.field_aware_factorization_machine module
Module description:
class
elliot.recommender.latent_factor_models.FFM.field_aware_factorization_machine.FFM
( data, config, params, *args, **kwargs )
    Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
```

```
get recommendations (k:int=100)
    property name
   predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.latent_factor_models.FFM.field_aware_factorization_machine_model module
Module description:
class
elliot.recommender.latent_factor_models.FFM.field_aware_factorization_machine_model.Fie
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k (preds, train_mask, k=100)
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.latent factor models.FISM package
Submodules
elliot.recommender.latent factor models.FISM.FISM module
Module description:
class elliot.recommender.latent_factor_models.FISM.FISM.FISM ( data, config, params,
*args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations ( k: int = 100, auc_compute: bool = False )
    property name
    restore_weights()
    train()
elliot.recommender.latent_factor_models.FISM.FISM_model module
Module description:
       elliot.recommender.latent_factor_models.FISM.FISM_model.FISM_model
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
```

```
batch_predict ( user_start, user_stop, **kwargs )
    call (inputs, training=None)
    create_history_item_matrix()
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
            Python dictionary.
    get_positions ( predictions, train_mask, items, inner_test_user_true_mask )
    get_top_k ( predictions, train_mask, k=100 )
    predict ( user, **kwargs )
    train_step ( batch )
class elliot.recommender.latent_factor_models.FISM.FISM_model.LatentFactor (
*args, **kwargs)
    Bases: tensorflow.python.keras.layers.embeddings.Embedding
    censor ( censor_id )
Module contents
Module description:
elliot.recommender.latent factor models.FM package
elliot.recommender.latent factor models.FM.factorization machine module
Module description:
class elliot.recommender.latent_factor_models.FM.factorization_machine.FM (
data, config, params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.latent factor models.FM.factorization machine model module
Module description:
class
elliot.recommender.latent_factor_models.FM.factorization_machine_model.FactorizationMac
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
```

64

```
call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.latent_factor_models.FunkSVD package
Submodules
elliot.recommender.latent factor models.FunkSVD.funk svd module
Module description:
class elliot.recommender.latent_factor_models.FunkSVD.funk_svd.FunkSVD ( data,
config, params, *args, **kwargs)
                              elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get\_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.latent factor models.FunkSVD.funk svd model module
Module description:
class
elliot.recommender.latent_factor_models.FunkSVD.funk_svd_model.FunkSVDModel
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call ( inputs, training=None, mask=None )
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k (preds, train_mask, k=100)
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
```

```
train_step ( batch )
Module contents
elliot.recommender.latent_factor_models.LogisticMF package
Submodules
elliot.recommender.latent_factor_models.LogisticMF.logistic_matrix_factorization module
Module description:
elliot.recommender.latent_factor_models.LogisticMF.logistic_matrix_factorization.Logist
( data, config, params, *args, **kwargs )
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
   predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.latent_factor_models.LogisticMF.logistic_matrix_factorization_model module
Module description:
class
elliot.recommender.latent_factor_models.LogisticMF.logistic_matrix_factorization_model.
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_top_k ( preds, train_mask, k=100 )
   predict_batch ( start, stop, **kwargs )
    set_update_user ( update_user )
    train_step ( batch )
Module contents
elliot.recommender.latent_factor_models.MF package
Submodules
elliot.recommender.latent factor models.MF.matrix factorization module
Module description:
class elliot.recommender.latent_factor_models.MF.matrix_factorization.MF ( data,
config, params, *args, **kwargs)
   Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
   predict ( u: int, i: int )
    restore_weights()
```

```
train()
elliot.recommender.latent_factor_models.MF.matrix_factorization_model module
Module description:
class
elliot.recommender.latent_factor_models.MF.matrix_factorization_model.MatrixFactorization
(*args, **kwargs)
          Bases: tensorflow.python.keras.engine.training.Model
          call (inputs, training=None, mask=None)
          get_recs ( inputs, training=False, **kwargs )
                    Get full predictions on the whole users/items matrix.
                    Returns:
                              The matrix of predicted values.
          get_top_k ( preds, train_mask, k=100 )
          predict (inputs, training=False, **kwargs )
                    Get full predictions on the whole users/items matrix.
                              The matrix of predicted values.
          train_step ( batch )
Module contents
elliot.recommender.latent factor models.NonNegMF package
Submodules
elliot.recommender.latent_factor_models.NonNegMF.non_negative_matrix_factorization module
Module description:
elliot.recommender.latent_factor_models.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegMF.non_negative_matrix_factorization.NonNegative_matrix_factorization.NonNegative_matrix_factorization.NonNe
(data, config, params, *args, **kwargs)
          Bases:
                                                                        elliot.recommender.recommender_utils_mixin.RecMixin,
          elliot.recommender.base_recommender_model.BaseRecommenderModel
          get_recommendations (k: int = 100)
          property name
          predict ( u: int, i: int )
                    Get prediction on the user item pair.
                    Returns:
                              A single float vaue.
          restore_weights()
          train()
elliot.recommender.latent_factor_models.NonNegMF.non_negative_matrix_factorization_model module
Module description:
```

```
class
elliot.recommender.latent_factor_models.NonNegMF.non_negative_matrix_factorization_mode
( data, num_users, num_items, global_mean, embed_mf_size, lambda_weights, learning_rate=0.01,
random_seed=42)
    Bases: object
    get_model_state()
    get_user_recs ( user, k=100 )
   predict ( user, item )
    set_model_state ( saving_dict )
    train_step()
Module contents
Module description:
elliot.recommender.latent_factor_models.PMF package
Submodules
elliot.recommender.latent_factor_models.PMF.probabilistic_matrix_factorization module
Module description:
Mnih, Andriy, and Russ R. Salakhutdinov. "Probabilistic matrix factorization." Advances in neural
information processing systems 20 (2007)
class
elliot.recommender.latent_factor_models.PMF.probabilistic_matrix_factorization.PMF
(data, config, params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
   predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.latent factor models.PMF.probabilistic matrix factorization model module
Module description:
Mnih, Andriy, and Russ R. Salakhutdinov. "Probabilistic matrix factorization." Advances in neural
information processing systems 20 (2007)
class
elliot.recommender.latent_factor_models.PMF.probabilistic_matrix_factorization_model.Pr
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    dot_prod ( layer_0, layer_1 )
    get_recs ( inputs, training=False, **kwargs )
```

Get full predictions on the whole users/items matrix.

```
Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.latent_factor_models.PureSVD package
Submodules
elliot.recommender.latent factor models.PureSVD.pure svd module
Module description:
class elliot.recommender.latent_factor_models.PureSVD.pure_svd.PureSVD ( data,
config, params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base recommender model.BaseRecommenderModel
    get\_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
        Get prediction on the user item pair.
        Returns:
            A single float vaue.
    restore_weights()
    train()
elliot.recommender.latent_factor_models.PureSVD.pure_svd_model module
Module description:
class
elliot.recommender.latent_factor_models.PureSVD.pure_svd_model.PureSVDModel
( factors, data, random_seed )
    Bases: object
    Simple Matrix Factorization class
    get_model_state()
    get_user_recs ( user, k=100 )
    predict ( user, item )
    set_model_state ( saving_dict )
    train_step()
Module contents
elliot.recommender.latent_factor_models.SVDpp package
Submodules
```

```
elliot.recommender.latent_factor_models.SVDpp.svdpp module
Module description:
class elliot.recommender.latent_factor_models.SVDpp.svdpp.SVDpp ( data, config,
params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
    train()
elliot.recommender.latent_factor_models.SVDpp.svdpp_model module
Module description:
class elliot.recommender.latent_factor_models.SVDpp.svdpp_model.SVDppModel (
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.latent_factor_models.Slim package
Submodules
elliot.recommender.latent_factor_models.Slim.slim module
Module description:
class elliot.recommender.latent_factor_models.Slim.slim.Slim ( data, config, params,
*args, **kwargs)
   Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
        Get prediction on the user item pair.
        Returns:
            A single float vaue.
```

```
restore weights()
    train()
elliot.recommender.latent_factor_models.Slim.slim_model module
Module description:
class elliot.recommender.latent_factor_models.Slim.slim_model.SlimModel ( data,
num_users, num_items, l1_ratio, alpha, epochs)
    Bases: object
    get_model_state()
    get_user_recs ( user, k=100 )
    predict ( u, i )
    set_model_state ( saving_dict )
    train ( verbose )
Module contents
Module description:
elliot.recommender.latent_factor_models.WRMF package
elliot.recommender.latent_factor_models.WRMF.wrmf module
Module description:
class elliot.recommender.latent_factor_models.WRMF.wrmf.WRMF ( data, config, params,
*args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base recommender model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
        Get prediction on the user item pair.
        Returns:
            A single float vaue.
    restore_weights()
    train()
elliot.recommender.latent_factor_models.WRMF.wrmf_model module
Module description:
        elliot.recommender.latent_factor_models.WRMF.wrmf_model.WRMFModel
factors, data, random, alpha, reg)
    Bases: object
    Simple Matrix Factorization class
    get_model_state()
    get_user_recs ( user, k=100 )
    predict ( user, item )
```

```
set_model_state ( saving_dict )
    train_step()
Module contents
Module contents
Module description:
elliot.recommender.neural package
Subpackages
elliot.recommender.neural.ConvMF package
Submodules
elliot.recommender.neural.ConvMF.convolutional_matrix_factorization module
Module description:
class
elliot.recommender.neural.ConvMF.convolutional_matrix_factorization.ConvMF (
data, config, params, *args, **kwargs)
    Bases:
                              elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    restore_weights()
    train()
elliot.recommender.neural.ConvMF.convolutional matrix factorization model module
Module description:
class
elliot.recommender.neural.ConvMF.convolutional_matrix_factorization_model.ConvMatrixFac
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=False, **kwargs )
        Calls the model on new inputs.
        In this case call just reapplies all ops in the graph to the new inputs (e.g. build a new compu-
        tational graph from the provided inputs).
        Arguments:
            inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating
            whether to run
                  the Network in training mode or inference mode.
            mask: A mask or list of masks. A mask can be
                 either a tensor or None (no mask).
        Returns:
            A tensor if there is a single output, or a list of tensors if there are more than one outputs.
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
```

```
predict (inputs, training=False, **kwargs )
```

Get full predictions on the whole users/items matrix.

Returns:

The matrix of predicted values.

train_step (batch)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's* metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

class

elliot.recommender.neural.ConvMF.convolutional_matrix_factorization_model.Convolutional(*args, **kwargs)

Bases: tensorflow.python.keras.engine.training.Model

```
call (inputs, **kwargs)
```

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

class

elliot.recommender.neural.ConvMF.convolutional_matrix_factorization_model.MLPComponent (*args, **kwargs)

Bases: tensorflow.python.keras.engine.training.Model

```
call (inputs, training=False, **kwargs)
```

Calls the model on new inputs.

In this case *call* just reapplies all ops in the graph to the new inputs (e.g. build a new computational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

```
Module contents
elliot.recommender.neural.ConvNeuMF package
Submodules
elliot.recommender.neural.ConvNeuMF.convolutional neural matrix factorization module
Module description:
class
elliot.recommender.neural.ConvNeuMF.convolutional_neural_matrix_factorization.ConvNeuMF
(data, config, params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    restore_weights()
    train()
elliot.recommender.neural.ConvNeuMF.convolutional_neural_matrix_factorization_model module
Module description:
elliot.recommender.neural.ConvNeuMF.convolutional_neural_matrix_factorization_model.Con
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=False, **kwargs )
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
   predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch )
class
elliot.recommender.neural.ConvNeuMF.convolutional_neural_matrix_factorization_model.Con
( *args, **kwargs )
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, **kwargs)
```

elliot.recommender.neural.ConvNeuMF.convolutional_neural_matrix_factorization_model.MLP

Bases: tensorflow.python.keras.engine.training.Model

(*args, **kwargs)

```
call (inputs, training=False, **kwargs)
Module contents
elliot.recommender.neural.DMF package
Submodules
elliot.recommender.neural.DMF.deep_matrix_factorization module
Module description:
class elliot.recommender.neural.DMF.deep matrix factorization.DMF ( data, config,
params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base recommender model.BaseRecommenderModel
    get\_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.neural.DMF.deep_matrix_factorization_model module
Module description:
class
elliot.recommender.neural.DMF.deep_matrix_factorization_model.DeepMatrixFactorizationMo
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    cosine ( layer_0, layer_1 )
    dot_prod ( layer_0, layer_1 )
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get\_top\_k ( preds, train_mask, k=100 )
   predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.neural.DeepFM package
Submodules
elliot.recommender.neural.DeepFM.deep_fm module
Module description:
class elliot.recommender.neural.DeepFM.deep_fm.DeepFM ( data, config, params, *args,
**kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
```

```
get_recommendations (k: int = 100)
    property name
   predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.neural.DeepFM.deep_fm_model module
Module description:
class elliot.recommender.neural.DeepFM.deep_fm_model.DeepFMModel (*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.neural.GeneralizedMF package
Submodules
elliot.recommender.neural.GeneralizedMF.generalized_matrix_factorization module
Module description:
class
elliot.recommender.neural.GeneralizedMF.generalized_matrix_factorization.GMF
(data, config, params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
   predict ( u: int, i: int )
    train()
elliot.recommender.neural.GeneralizedMF.generalized_matrix_factorization_model module
Module description:
class
elliot.recommender.neural.GeneralizedMF.generalized_matrix_factorization_model.Generali
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
```

76

```
call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    train_step ( batch )
Module contents
elliot.recommender.neural.ItemAutoRec package
Submodules
elliot.recommender.neural.ltemAutoRec.itemautorec module
Module description:
class elliot.recommender.neural.ItemAutoRec.itemautorec.ItemAutoRec ( data, config,
params, *args, **kwargs)
    Bases:
                            elliot.recommender.recommender utils mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.neural.ltemAutoRec.itemautorec_model module
Module description:
class elliot.recommender.neural.ItemAutoRec.itemautorec_model.Decoder ( *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    call (inputs, **kwargs)
class elliot.recommender.neural.ItemAutoRec.itemautorec_model.Encoder ( *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    call (inputs, training=None)
elliot.recommender.neural.ItemAutoRec.itemautorec_model.ItemAutoRecModel
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, **kwargs )
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
```

These are handled by *Network* (one layer of abstraction above).

```
Returns:
            Python dictionary.
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix. [Is Inverted]
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch )
Module contents
Module description:
elliot.recommender.neural.NAIS package
elliot.recommender.neural.NAIS.nais module
Module description:
class elliot.recommender.neural.NAIS.nais.NAIS ( data, config, params, *args, **kwargs )
                             elliot.recommender.recommender utils mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations ( k: int = 100, auc_compute: bool = False )
    property name
    train()
elliot.recommender.neural.NAIS.nais_model module
Module description:
class elliot.recommender.neural.NAIS.nais_model.LatentFactor(*args, **kwargs)
    Bases: tensorflow.python.keras.layers.embeddings.Embedding
    censor ( censor_id )
class elliot.recommender.neural.NAIS.nais_model.NAIS_model(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    attention ( user_history, target )
    batch_attention ( user_history, target )
    batch_predict (user_start, user_stop)
    batch_softmax ( logits, item_num, similarity, user_bias, item_bias )
    call (inputs, training=None)
    create_history_item_matrix()
    get_config()
        Returns the config of the layer.
```

A layer config is a Python dictionary (serializable) containing the configuration of a layer. The same layer can be reinstantiated later (without its trained weights) from this configuration.

The config of a layer does not include connectivity information, nor the layer class name. These are handled by *Network* (one layer of abstraction above).

```
Returns:
```

```
Python dictionary.
```

```
get_positions ( predictions, train_mask, items, inner_test_user_true_mask )
    get_top_k ( predictions, train_mask, k=100 )
    predict (user, **kwargs)
    softmax (logits, item_num, similarity, user_bias, item_bias, batch_mask_mat=None)
    train_step ( batch )
Module contents
elliot.recommender.neural.NFM package
Submodules
elliot.recommender.neural.NFM.neural fm module
Module description:
class elliot.recommender.neural.NFM.neural_fm.NFM ( data, config, params, *args, **kwargs )
                             elliot.recommender.recommender utils mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    predict ( u: int, i: int )
    restore_weights()
    train()
elliot.recommender.neural.NFM.neural fm model module
Module description:
class
elliot.recommender.neural.NFM.neural_fm_model.NeuralFactorizationMachineModel
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call ( inputs, training=None, mask=None )
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
```

The matrix of predicted values.

Get full predictions on the whole users/items matrix.

predict (inputs, training=False, **kwargs)

Returns:

```
train_step ( batch )
Module contents
elliot.recommender.neural.NPR package
Submodules
elliot.recommender.neural.NPR.neural_personalized_ranking module
Module description:
class elliot.recommender.neural.NPR.neural personalized ranking.NPR (data, config,
params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    Bases:
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get\_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.neural.NPR.neural_personalized_ranking_model module
Module description:
class
elliot.recommender.neural.NPR.neural_personalized_ranking_model.NPRModel
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.neural.NeuMF package
Submodules
elliot.recommender.neural.NeuMF.neural_matrix_factorization module
Module description:
class elliot.recommender.neural.NeuMF.neural_matrix_factorization.NeuMF ( data,
config, params, *args, **kwargs)
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get\_recommendations (k: int = 100)
    property name
```

```
train()
elliot.recommender.neural.NeuMF.neural_matrix_factorization_model module
Module description:
class
elliot.recommender.neural.NeuMF.neural_matrix_factorization_model.NeuralMatrixFactoriza
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
       Get full predictions on the whole users/items matrix.
       Returns:
            The matrix of predicted values.
    get_top_k ( preds, train_mask, k=100 )
    predict (inputs, training=False, **kwargs )
       Get full predictions on the whole users/items matrix.
       Returns:
            The matrix of predicted values.
    train_step ( batch )
Module contents
elliot.recommender.neural.UserAutoRec package
Submodules
elliot.recommender.neural.UserAutoRec.userautorec module
Module description:
class elliot.recommender.neural.UserAutoRec.userautorec.UserAutoRec ( data, config,
params, *args, **kwargs)
                            elliot.recommender.recommender utils mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    property name
    train()
elliot.recommender.neural.UserAutoRec.userautorec model module
Module description:
class elliot.recommender.neural.UserAutoRec.userautorec_model.Decoder ( *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    call (inputs, **kwargs)
class elliot.recommender.neural.UserAutoRec.userautorec_model.Encoder ( *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.base_layer.Layer
    call (inputs, training=None)
```

```
class
elliot.recommender.neural.UserAutoRec.userautorec_model.UserAutoRecModel
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, **kwargs)
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        tion.
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get\_top\_k (preds, train\_mask, k=100)
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
    train_step ( batch )
Module contents
Module description:
elliot.recommender.neural.WideAndDeep package
Submodules
elliot.recommender.neural.WideAndDeep.wide_and_deep module
Module description:
class elliot.recommender.neural.WideAndDeep.wide_and_deep.WideAndDeep ( data,
config, params, *args, **kwargs)
                             elliot.recommender.recommender utils mixin.RecMixin,
    Bases:
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.neural.WideAndDeep.wide_and_deep.build_sparse_features (
data)
elliot.recommender.neural.WideAndDeep.wide and deep model module
Module description:
class
elliot.recommender.neural.WideAndDeep.wide_and_deep_model.WideAndDeepModel (
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=False, **kwargs)
        Calls the model on new inputs.
        In this case call just reapplies all ops in the graph to the new inputs (e.g. build a new compu-
```

tational graph from the provided inputs).

Arguments:

inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating whether to run

the *Network* in training mode or inference mode.

mask: A mask or list of masks. A mask can be

either a tensor or None (no mask).

Returns:

A tensor if there is a single output, or a list of tensors if there are more than one outputs.

```
get_sparse ( u, i )
get_top_k ( preds, train_mask, k=100 )
get_user_recs ( user, k=100 )
predict ( user, **kwargs )
```

Generates output predictions for the input samples.

Computation is done in batches. This method is designed for performance in large scale inputs. For small amount of inputs that fit in one batch, directly using <u>__call__</u> is recommended for faster execution, e.g., model(x), or model(x, training=False) if you have layers such as tf.keras.layers.BatchNormalization that behaves differently during inference. Also, note the fact that test loss is not affected by regularization layers like noise and dropout.

Arguments:

x: Input samples. It could be:

- A Numpy array (or array-like), or a list of arrays (in case the model has multiple inputs).
- A TensorFlow tensor, or a list of tensors (in case the model has multiple inputs).
- A tf.data dataset.
- A generator or *keras.utils.Sequence* instance.

A more detailed description of unpacking behavior for iterator types (Dataset, generator, Sequence) is given in the *Unpacking behavior for iterator-like inputs* section of *Model.fit*.

batch_size: Integer or None.

Number of samples per batch. If unspecified, *batch_size* will default to 32. Do not specify the *batch_size* if your data is in the form of dataset, generators, or *keras.util-s.Sequence* instances (since they generate batches).

verbose: Verbosity mode, 0 or 1. steps: Total number of steps (batches of samples)

before declaring the prediction round finished. Ignored with the default value of *None*. If x is a *tf.data* dataset and *steps* is None, *predict* will run until the input dataset is exhausted.

callbacks: List of keras.callbacks.Callback instances.

List of callbacks to apply during prediction. See [callbacks](/api_docs/python/t-f/keras/callbacks).

max_queue_size: Integer. Used for generator or keras.utils.Sequence

input only. Maximum size for the generator queue. If unspecified, *max_queue_size* will default to 10.

workers: Integer. Used for generator or keras.utils.Sequence input

only. Maximum number of processes to spin up when using process-based threading. If unspecified, *workers* will default to 1. If 0, will execute the generator on the

main thread.

use_multiprocessing: Boolean. Used for generator or

keras.utils.Sequence input only. If *True*, use process-based threading. If unspecified, *use_multiprocessing* will default to *False*. Note that because this implementation relies on multiprocessing, you should not pass non-picklable arguments to the generator as they can't be passed easily to children processes.

See the discussion of *Unpacking behavior for iterator-like inputs* for *Model.fit*. Note that Model.predict uses the same interpretation rules as *Model.fit* and *Model.evaluate*, so inputs must be unambiguous for all three methods.

Returns:

Numpy array(s) of predictions.

Raises:

RuntimeError: If *model.predict* is wrapped in *tf.function*. ValueError: In case of mismatch between the provided

input data and the model's expectations, or in case a stateful model receives a number of samples that is not a multiple of the batch size.

train_step (batch)

The logic for one training step.

This method can be overridden to support custom training logic. This method is called by *Model.make_train_function*.

This method should contain the mathematical logic for one step of training. This typically includes the forward pass, loss calculation, backpropagation, and metric updates.

Configuration details for *how* this logic is run (e.g. *tf.function* and *tf.distribute.Strategy* settings), should be left to *Model.make_train_function*, which can also be overridden.

Arguments:

data: A nested structure of `Tensor`s.

Returns:

A *dict* containing values that will be passed to *tf.keras.callbacks.CallbackList.on_train_batch_end*. Typically, the values of the *Model's* metrics are returned. Example: {'loss': 0.2, 'accuracy': 0.7}.

Module contents

Module contents

elliot.recommender.unpersonalized package

Subpackages

elliot.recommender.unpersonalized.most popular package

Submodules

elliot.recommender.unpersonalized.most_popular.most_popular module

Created on April 4, 2020 Tensorflow 2.1.0 implementation of APR. @author Anonymized

```
get\_recommendations(top\_k)
```

property name

train()

Module contents

 $elliot.recommender.unpersonalized.random_recommender\ package$

Submodules

```
elliot.recommender.unpersonalized.random_recommender.Random module
```

```
Created on April 4, 2020 Tensorflow 2.1.0 implementation of APR. @author Anonymized
class elliot.recommender.unpersonalized.random_recommender.Random.Random ( data,
config, params, *args, **kwargs)
    Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations ( top_k )
    property name
    train()
Module contents
Module contents
elliot.recommender.visual_recommenders package
Subpackages
elliot.recommender.visual_recommenders.ACF package
Submodules
elliot.recommender.visual recommenders.ACF.ACF module
Module description:
class elliot.recommender.visual_recommenders.ACF.ACF ( data, config, params, *args,
**kwargs)
    Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    property name
    train()
elliot.recommender.visual_recommenders.ACF.ACF_model module
Module description:
class elliot.recommender.visual_recommenders.ACF.ACF_model.ACF_model ( *args,
**kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
```

Module contents

Module description:

predict (start, stop)

train_step (batch)

get_top_k (preds, train_mask, k=100)

```
elliot.recommender.visual_recommenders.DVBPR package
Submodules
elliot.recommender.visual recommenders.DVBPR.DVBPR module
Module description:
class elliot.recommender.visual_recommenders.DVBPR.DVBPR.DVBPR (
                                                                                data,
params, *args, **kwargs)
    Bases:
                             elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.visual recommenders.DVBPR.DVBPR model module
Module description:
class elliot.recommender.visual_recommenders.DVBPR.DVBPR_model.DVBPR_model (
*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_config()
    get_top_k ( preds, train_mask, k=100 )
    predict_batch ( start, stop, phi )
    train_step ( batch )
elliot.recommender.visual_recommenders.DVBPR.FeatureExtractor module
elliot.recommender.visual_recommenders.DVBPR.FeatureExtractor.FeatureExtractor
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model,abc.ABC
    call (inputs, training=None, mask=None)
        Calls the model on new inputs.
        In this case call just reapplies all ops in the graph to the new inputs (e.g. build a new compu-
        tational graph from the provided inputs).
        Arguments:
            inputs: A tensor or list of tensors. training: Boolean or boolean scalar tensor, indicating
            whether to run
                  the Network in training mode or inference mode.
            mask: A mask or list of masks. A mask can be
                either a tensor or None (no mask).
            A tensor if there is a single output, or a list of tensors if there are more than one outputs.
Module contents
Module description:
elliot.recommender.visual recommenders.DeepStyle package
```

Submodules

```
elliot.recommender.visual_recommenders.DeepStyle.DeepStyle module
Module description:
class elliot.recommender.visual_recommenders.DeepStyle.DeepStyle.DeepStyle (
data, config, params, *args, **kwargs)
   Bases:
                            elliot.recommender.recommender_utils_mixin.RecMixin,
   elliot.recommender.base_recommender_model.BaseRecommenderModel
   get_recommendations (k: int = 100)
   property name
   train()
elliot.recommender.visual_recommenders.DeepStyle.DeepStyle_model module
Module description:
class
elliot.recommender.visual_recommenders.DeepStyle_DeepStyle_model.DeepStyle_model
(*args, **kwargs)
   Bases: tensorflow.python.keras.engine.training.Model
   call (inputs, training=None)
   get_config()
   get_top_k ( preds, train_mask, k=100 )
   predict ( start, stop, training=False )
   train_step ( batch )
Module contents
Module description:
elliot.recommender.visual_recommenders.VBPR package
Submodules
elliot.recommender.visual recommenders.VBPR.VBPR module
Module description:
class elliot.recommender.visual_recommenders.VBPR.VBPR.VBPR ( data, config, params,
*args, **kwargs)
                            elliot.recommender.recommender_utils_mixin.RecMixin,
   elliot.recommender.base_recommender_model.BaseRecommenderModel
   get\_recommendations (k: int = 100)
   property name
   train()
elliot.recommender.visual_recommenders.VBPR.VBPR_model module
Module description:
class elliot.recommender.visual_recommenders.VBPR.VBPR_model.VBPR_model ( *args,
**kwargs)
```

Bases: tensorflow.python.keras.engine.training.Model

call (inputs, training=None)

```
get_config()
        Returns the config of the layer.
        A layer config is a Python dictionary (serializable) containing the configuration of a layer.
        The same layer can be reinstantiated later (without its trained weights) from this configura-
        The config of a layer does not include connectivity information, nor the layer class name.
        These are handled by Network (one layer of abstraction above).
        Returns:
            Python dictionary.
    get\_top\_k ( preds, train_mask, k=100 )
    predict ( start, stop )
    train_step ( batch )
Module contents
Module description:
elliot.recommender.visual recommenders.VNPR package
Submodules
elliot.recommender.visual recommenders.VNPR.visual neural personalized ranking module
Module description:
class
elliot.recommender.visual_recommenders.VNPR.visual_neural_personalized_ranking.VNPR
(data, config, params, *args, **kwargs)
    Bases:
                              elliot.recommender.recommender_utils_mixin.RecMixin,
    elliot.recommender.base_recommender_model.BaseRecommenderModel
    get_recommendations (k: int = 100)
    property name
    train()
elliot.recommender.visual recommenders.VNPR.visual neural personalized ranking model module
Module description:
class
elliot.recommender.visual_recommenders.VNPR.visual_neural_personalized_ranking_model.VN
(*args, **kwargs)
    Bases: tensorflow.python.keras.engine.training.Model
    call (inputs, training=None, mask=None)
    get_recs ( inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
            The matrix of predicted values.
    get\_top\_k (preds, train\_mask, k=100)
    predict (inputs, training=False, **kwargs )
        Get full predictions on the whole users/items matrix.
        Returns:
            The matrix of predicted values.
```

```
train_step ( batch )
Module contents
Module contents
Module description:
Submodules
elliot.recommender.base recommender model module
Module description:
class elliot.recommender.base_recommender_model.BaseRecommenderModel (
                                                                                       data,
config, params, *args, **kwargs)
    Bases: abc.ABC
    autoset_params()
        Define Parameters as tuples: (variable_name, public_name, shortcut, default, reading_func-
        tion, printing_function) Example:
        self._params_list = [
            ("_similarity", "similarity", "sim", "cosine", None, None), ("_user_profile_type",
            "user_profile", "up", "tfidf", None, None), ("_item_profile_type", "item_profile", "ip",
            "tfidf", None, None), ("_mlpunits", "mlp_units", "mlpunits", "(1,2,3)", lambda x: list(-
            make_tuple(x)), lambda x: str(x).replace(",", "-")),
        1
    abstract get_loss()
    abstract get_params ( )
    get_params_shortcut()
    abstract get_recommendations (*args)
    abstract get_results()
    abstract train ( )
elliot.recommender.base_recommender_model.init_charger(init)
elliot.recommender.recommender_utils_mixin module
class elliot.recommender.recommender_utils_mixin.RecMixin
    Bases: object
    get_loss()
    get_params()
    get_recommendations (k: int = 100)
    get_results()
    get_train_mask ( start, stop )
    restore_weights()
    train()
```

Module contents

Module description:

1.1.7 elliot.result_handler package

```
Submodules
elliot.result handler.result handler module
Module description:
class elliot.result_handler.result_handler.HyperParameterStudy (rel_threshold=1)
    Bases: object
    add_trials(obj)
    save_trials ( output='../results/' )
    save_trials_as_triplets ( output='../results/' )
class elliot.result_handler.result_handler.ResultHandler(rel_threshold=1)
    Bases: object
    add_oneshot_recommender ( **kwargs )
    save_best_models ( output='../results/' )
    save_best_results ( output='../results/' )
    save_best_results_as_triplets ( output='../results/' )
    save_best_statistical_results ( stat_test, output='../results/' )
class elliot.result_handler.result_handler.StatTest (value)
    Bases: enum. Enum
    An enumeration.
    PairedTTest = [<class 'elliot.evaluation.statistical_significance.PairedTTest'>, 'paired_ttest']
    WilcoxonTest = [<class 'elliot.evaluation.statistical_significance.WilcoxonTest'>, 'wilcox-
    on_test']
Module contents
Module description:
1.1.8 elliot.splitter package
Submodules
elliot.splitter.base_splitter module
class elliot.splitter.base_splitter.Splitter ( data: pandas.core.frame.DataFrame,
splitting_ns: types.SimpleNamespace )
    Bases: object
    fold_list_generator ( length, folds=5 )
    generic_split_function
                                  ( data:
                                              pandas.core.frame.DataFrame,
                                                                           **kwargs
```

List[Tuple[pandas.core.frame.DataFrame, pandas.core.frame.DataFrame]]

```
handle hierarchy ( data: pandas.core.frame.DataFrame, valtest splitting ns: types.SimpleNames-
    pace ) \rightarrow List[Tuple[pandas.core.frame.DataFrame, pandas.core.frame.DataFrame]]
    process_splitting()
    read_folder ( folder_path )
                                                           List[Tuple[pandas.core.frame.DataFrame,
    rearrange data
                                         train test:
    pandas.core.frame.DataFrame]],
                                       train val:
                                                       List[List[Tuple[pandas.core.frame.DataFrame,
    pandas.core.frame.DataFrame]]] )
    splitting_best_timestamp ( d: pandas.core.frame.DataFrame, min_below=1, min_over=1 )
    splitting_kfolds ( data: pandas.core.frame.DataFrame, folds=5 )
    splitting_passed_timestamp ( d: pandas.core.frame.DataFrame, timestamp=1 )
    splitting_randomsubsampling_kfolds ( d: pandas.core.frame.DataFrame, folds=5, ratio=0.2 )
    splitting_randomsubsampling_kfolds_leavenout ( d: pandas.core.frame.DataFrame,
   folds=5, n=1)
    splitting_temporal_holdout ( d: pandas.core.frame.DataFrame, ratio=0.2 )
    splitting_temporal_leavenout ( d: pandas.core.frame.DataFrame, n=1 )
    store_splitting(tuple_list)
    subsampling_leavenout_list_generator ( length, n=1 )
    subsampling_list_generator ( length, ratio=0.2 )
Module contents
1.1.9 elliot.utils package
Submodules
elliot.utils.folder module
Module description:
elliot.utils.folder.build_log_folder(path_log_folder)
elliot.utils.folder.build_model_folder(path_output_rec_weight, model)
elliot.utils.folder.create_folder_by_index ( path, index )
elliot.utils.folder.manage_directories ( path_output_rec_result, path_output_rec_weight,
path_output_rec_performance )
elliot.utils.logger_util module
            elliot.utils.logger_util.QueueListenerHandler
                                                                                      handlers,
respect_handler_level=False, auto_run=True, queue=<queue.Queue object>)
    Bases: logging.handlers.QueueHandler
    emit ( record )
        Emit a record.
        Writes the LogRecord to the queue, preparing it for pickling first.
```

1.2. Submodules 91

```
start()
    stop()
elliot.utils.logging module
class elliot.utils.logging.TimeFilter ( name="')
    Bases: logging.Filter
    filter ( record: logging.LogRecord ) \rightarrow bool
        Determine if the specified record is to be logged.
        Is the specified record to be logged? Returns 0 for no, nonzero for yes. If deemed appropri-
        ate, the record may be modified in-place.
elliot.utils.logging.get_logger(name, log_level=10)
elliot.utils.logging.init(path_config, folder_log, log_level=30)
elliot.utils.logging.prepare_logger(name, path, log_level=10)
elliot.utils.read module
Module description:
elliot.utils.read.find_checkpoint(dir, restore_epochs, epochs, rec, best=0)
                  • dir – directory of the model where we start from the reading.
                  • restore_epochs – epoch from which we start from.
                  • epochs – epochs from which we restore (0 means that we have best)
                  • rec – recommender model
                  • best - 0 No Best - 1 Search for the Best
    Returns
elliot.utils.read.load_obj(name)
    Load the pkl object by name :param name: name of file :return:
elliot.utils.read.read_config(sections_fields)
        sections_fields (list): list of fields to retrieve from configuration file
    Return:
        A list of configuration values.
elliot.utils.read.read_csv (filename)
    Args:
        filename (str): csv file path
    Return:
        A pandas dataframe.
elliot.utils.read.read_imagenet_classes_txt (filename)
    Args:
        filename (str): txt file path
    Return:
        A list with 1000 imagenet classes as strings.
```

```
elliot.utils.read.read_multi_config()
    It reads a config file that contains the configuration parameters for the recommendation systems.
    Return:
        A list of configuration settings.
elliot.utils.read.read_np (filename)
    Args:
        filename (str): filename of numpy to load
    Return:
        The loaded numpy.
elliot.utils.write module
Module description:
elliot.utils.write.save_np (npy, filename)
    Store numpy to memory. Args:
          npy: numpy to save filename (str): filename
elliot.utils.write.save_obj(obj, name)
    Store the object in a pkl file :param obj: python object to be stored :param name: file name (Not
    insert .pkl) :return:
elliot.utils.write.store_recommendation(recommendations, path="')
    Store recommendation list (top-k):return:
```

Module contents

Module description:

1.2 Submodules

1.3 elliot.run module

```
Module description:
```

```
elliot.run.config_test (builder, base)
elliot.run.run_experiment (config_path: str = './config/config.yml')
```

1.4 Module contents

Module description:

- genindex
- modindex
- search

1.2. Submodules 93

```
е
                                           elliot.evaluation.metrics.accuracy.AUC.gauc,
elliot,93
                                           elliot.evaluation.metrics.accuracy.AUC.lauc,
   elliot.dataset,9
   elliot.dataset.abstract_dataset,
                                           elliot.evaluation.metrics.accuracy.DSC,
         8
   elliot.dataset.dataloader,7
   elliot.dataset.dataloader.knowledge_aware_chains,
   elliot.dataset.dataloader.visual_dataloader, evaluation.metrics.accuracy.fl,
                                           elliot.evaluation.metrics.accuracy.fl.extend
   elliot.dataset.dataset,9
   elliot.dataset.samplers,8
   elliot.dataset.samplers.custom_pointwise_sparse_samplerion.metrics.accuracy.fl.fl,
                                           elliot.evaluation.metrics.accuracy.hit_rate,
   elliot.dataset.samplers.custom_sampler,
   elliot.dataset.samplers.custom_sparse_sampler,.evaluation.metrics.accuracy.hit_rate.
   elliot.dataset.samplers.pairwise_sampler, elliot.evaluation.metrics.accuracy.map,
   elliot.dataset.samplers.pipeline_sampler, elliot.evaluation.metrics.accuracy.map.map,
   elliot.dataset.samplers.pointwise_cfgan_sampler____elliot.dataset.samplers.accuracy.mar,
   elliot.dataset.samplers.pointwise_pos_neg_ratings_sampier, metrics.accuracy.mar.mar,
   elliot.dataset.samplers.pointwise_pos_neg_liotoeyaluationamptrics.accuracy.mrr,
   elliot.dataset.samplers.pointwise_pos_neg_lintigvaluation.metrics.accuracy.mrr.mrr, 8
   elliot.dataset.samplers.pointwise_wide_afid_detpeyahpation.metrics.accuracy.ndcg,
                                           elliot.evaluation.metrics.accuracy.ndcg.ndcg
   elliot.dataset.samplers.sparse_sampler,
                                           elliot.evaluation.metrics.accuracy.precision
   elliot.evaluation, 31
   elliot.evaluation.evaluator,31
                                           elliot.evaluation.metrics.accuracy.precision
   elliot.evaluation.metrics, 30
   elliot.evaluation.metrics.accuracy,
                                           elliot.evaluation.metrics.accuracy.recall,
   elliot.evaluation.metrics.accuracy.AUC,
                                           elliot.evaluation.metrics.accuracy.recall.re
   elliot.evaluation.metrics.accuracy.AUC.auc,
```

```
elliot.evaluation.metrics.base_metric, elliot.evaluation.metrics.fairness,
elliot.evaluation.metrics.bias,
                                       elliot.evaluation.metrics.fairness.BiasDispa
      18
                                        elliot.evaluation.metrics.fairness.BiasDispa
elliot.evaluation.metrics.bias.aclt,
elliot.evaluation.metrics.bias.aclt.acltelliot.evaluation.metrics.fairness.BiasDispa
elliot.evaluation.metrics.bias.aplt,
                                       elliot.evaluation.metrics.fairness.BiasDispa
elliot.evaluation.metrics.bias.aplt.apltelliot.evaluation.metrics.fairness.MAD,
                                               24
elliot.evaluation.metrics.bias.arp,
                                      elliot.evaluation.metrics.fairness.MAD.ItemM
elliot.evaluation.metrics.bias.arp.arp, elliot.evaluation.metrics.fairness.MAD.ItemM
                                               23
elliot.evaluation.metrics.bias.pop_reo, elliot.evaluation.metrics.fairness.MAD.UserM
      17
elliot.evaluation.metrics.bias.pop_reo.ext&indedepapuateoon.metrics.fairness.MAD.UserM
                                               24
elliot.evaluation.metrics.bias.pop_reo.peplieb, evaluation.metrics.fairness.reo,
elliot.evaluation.metrics.bias.pop_rsp, elliot.evaluation.metrics.fairness.reo.reo,
elliot.evaluation.metrics.bias.pop_rsp.ext&idedepapuation.metrics.fairness.rsp,
      17
                                               26
elliot.evaluation.metrics.bias.pop_rsp.peplisp,evaluation.metrics.fairness.rsp.rsp,
                                               25
elliot.evaluation.metrics.coverage,
                                      elliot.evaluation.metrics.metrics_utils,
                                               29
elliot.evaluation.metrics.coverage.item_edveoagevaluation.metrics.novelty,
elliot.evaluation.metrics.coverage.item_edveoagevatem_tconemegeics.novelty.EFD,
                                               26
elliot.evaluation.metrics.coverage.num_retrievedyaluation.metrics.novelty.EFD.efd,
elliot.evaluation.metrics.coverage.num_retrievedvalua_teominwedics.novelty.EFD.extend
elliot.evaluation.metrics.coverage.user ebweragevaluation.metrics.novelty.EPC,
elliot.evaluation.metrics.coverage.user_ebveragevaseatconemageics.novelty.EPC.epc,
      19
                                               27
elliot.evaluation.metrics.coverage.user_ebveoagevasemtconemetrias.novelty.EPC.extend
elliot.evaluation.metrics.diversity, elliot.evaluation.metrics.rating,
                                               29
elliot.evaluation.metrics.diversity.ginielhdek,evaluation.metrics.rating.mae,
                                               28
elliot.evaluation.metrics.diversity.ginielhdek.gvaluahdex, metrics.rating.mae.mae,
elliot.evaluation.metrics.diversity.shanebhiehtevpypation.metrics.rating.mse,
elliot.evaluation.metrics.diversity.shanadhientevplushannomeentesprating.mse.mse,
elliot.evaluation.metrics.diversity.SRecelliot.evaluation.metrics.rating.rmse,
                                               29
elliot.evaluation.metrics.diversity.SRecellisteewalpation.metrics.rating.rmse.rmse,
      20
```

```
elliot.evaluation.metrics.statistical_array_met45ic,
                                        elliot.recommender.autoencoders.vae,
elliot.evaluation.popularity_utils,
                                        elliot.recommender.autoencoders.vae.multi_va
elliot.evaluation.popularity_utils.popularity, 45
                                        elliot.recommender.autoencoders.vae.multi_va
elliot.evaluation.relevance,31
elliot.evaluation.relevance.relevance, elliot.recommender.base_recommender_model,
elliot.evaluation.statistical significaneelliot.recommender.content_based,
elliot.hyperoptimization, 32
                                        elliot.recommender.content_based.VSM,
elliot.hyperoptimization.model_coordinator,
                                               47
                                        elliot.recommender.content_based.VSM.tfidf_u
elliot.namespace, 33
                                        elliot.recommender.content_based.VSM.vector_
elliot.namespace.namespace_model,
                                               47
elliot.namespace.namespace_model_builderplliot.recommender.content_based.VSM.vector_
                                               47
      32
                                        elliot.recommender.gan, 54
elliot.prefiltering, 33
elliot.prefiltering.standard_prefilters,elliot.recommender.gan.CFGAN,50
                                        elliot.recommender.gan.CFGAN.cfgan,
elliot.recommender, 90
                                        elliot.recommender.gan.CFGAN.cfgan_model,
elliot.recommender.adversarial,
                                        elliot.recommender.gan.IRGAN, 54
elliot.recommender.adversarial.AMF,
                                         elliot.recommender.gan.IRGAN.irgan,
      41
elliot.recommender.adversarial.AMF.AMF,
                                         elliot.recommender.gan.IRGAN.irgan_model,
elliot.recommender.adversarial.AMF.AMF_model, 50
                                        elliot.recommender.graph_based,
elliot.recommender.adversarial.AMR,
                                        elliot.recommender.graph_based.lightgcn,
      43
                                               -55
elliot.recommender.adversarial.AMR.AMR,
                                        elliot.recommender.graph_based.lightgcn.Ligh
elliot.recommender.adversarial.AMR.AMR model, 54
                                        elliot.recommender.graph_based.lightgcn.Ligh
elliot.recommender.algebric,44
elliot.recommender.algebric.slope one, elliot.recommender.graph_based.ngcf,
      43
elliot.recommender.algebric.slope_one.slopeionerecommender.graph_based.ngcf.NGCF,
                                               55
elliot.recommender.algebric.slope_one.slope_ioner_modminender.graph_based.ngcf.NGCF_mod
elliot.recommender.attentive,44
                                        elliot.recommender.knowledge_aware,
elliot.recommender.attentive.afm,
                                         elliot.recommender.knowledge_aware.kaHFM,
elliot.recommender.attentive.afm.afm,
                                        elliot.recommender.knowledge_aware.kaHFM.ka_
                                               56
elliot.recommender.attentive.afm.afm model,
                                        elliot.recommender.knowledge_aware.kaHFM.tfi
elliot.recommender.autoencoders,
                                        elliot.recommender.knowledge_aware.kaHFM_bat
elliot.recommender.autoencoders.dae,
                                        elliot.recommender.knowledge_aware.kaHFM_bat
      45
elliot.recommender.autoencoders.dae.multi_dae, 57
                                        elliot.recommender.knowledge_aware.kaHFM_bat
```

elliot.recommender.autoencoders.dae.multi_dae_model,

```
elliot.recommender.knowledge_aware.kaHFMebatch.recdfimendes,latent_factor_models.Logi
elliot.recommender.knowledge_aware.kahfmebhbeddingsmmender.latent_factor_models.Logi
                                                66
elliot.recommender.knowledge_aware.kahfmethbeddingsmkahdmrehbeddingsctor_models.Logi
elliot.recommender.knowledge_aware.kahfmehhbeddiegsmkehdmeehbeddingschodels.MF,
elliot.recommender.knowledge_aware.kahfmebhbeddingsmbeiderutatent_factor_models.MF.m
elliot.recommender.latent_factor_models, elliot.recommender.latent_factor_models.MF.m
                                                67
elliot.recommender.latent_factor_models.BPRMot.recommender.latent_factor_models.NonN
                                               68
elliot.recommender.latent_factor_models.BPRM5tBPRM5mmender.latent_factor_models.NonN
elliot.recommender.latent_factor_models.BPRMotbaechmmender.latent_factor_models.NonN
elliot.recommender.latent_factor_models.BPRMotbaechmBPRMor_baathnt_factor_models.PMF,
elliot.recommender.latent_factor_models.BPRMotbaechmBPRMor_baathnmodedtor_models.PMF.
\verb|elliot.recommender.latent_factor_models. \verb|BPRSotim| recommender.latent_factor_models. \verb|PMF.| |
elliot.recommender.latent_factor_models.BPRSotimrbpossheimder.latent_factor_models.Pure
elliot.recommender.latent_factor_models.BPRS&timrbposnhemdmode&tent_factor_models.Pure
elliot.recommender.latent_factor_models.@MLiot.recommender.latent_factor_models.Pure
elliot.recommender.latent_factor_models.@MLi@MLrecommender.latent_factor_models.Slim
elliot.recommender.latent_factor_models.@MLi@MLremodenhender.latent_factor_models.Slim
                                                70
elliot.recommender.latent_factor_models.EfMiot.recommender.latent_factor_models.Slim
elliot.recommender.latent_factor_models.EfMi&teldcammenderatatentent_factor_models.SVDp
elliot.recommender.latent_factor_models.EfMi&teldcammendeactatezatfaatmach&delmod&Dp
elliot.recommender.latent_factor_models.&ISMpt.recommender.latent_factor_models.SVDp
                                                70
elliot.recommender.latent_factor_models.&ISMoFISMcommender.latent_factor_models.WRMF
                                                72
elliot.recommender.latent_factor_models.&ISMoFISMommdehder.latent_factor_models.WRMF
elliot.recommender.latent_factor_models.EMliot.recommender.latent_factor_models.WRMF
                                                71
elliot.recommender.latent_factor_models.EMlfactorecommendemarhinel,84
                                        elliot.recommender.neural.ConvMF,
elliot.recommender.latent_factor_models.FM.fact7drization_machine_model,
                                         elliot.recommender.neural.ConvMF.convolution
elliot.recommender.latent_factor_models.FunkSVD7/2
                                         elliot.recommender.neural.ConvMF.convolution
\verb|elliot.recommender.latent_factor_models.FunkSVD|| 72 funk\_svd,
                                         elliot.recommender.neural.ConvNeuMF,
elliot.recommender.latent_factor_models.FunkSVD75funk_svd_model,
      65
                                         elliot.recommender.neural.ConvNeuMF.convolut
```

elliot.recommender.visual_recommenders.DeepS

```
74
                                                                      elliot.recommender.neural.WideAndDeep.wide_a
elliot.recommender.neural.ConvNeuMF.convolution21_neural_matrix_factorization_model,
                                                                      elliot.recommender.NN, 38
elliot.recommender.neural.DeepFM,
                                                                     elliot.recommender.NN.attribute_item_knn,
           76
                                                                                 34
elliot.recommender.neural.DeepFM.deep_fmelliot.recommender.NN.attribute_item_knn.att
\verb|elliot.recommender.neural.DeepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deep_fmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.deepffmetal.peepFM.de
elliot.recommender.neural.DMF,75
                                                                      elliot.recommender.NN.attribute user knn,
elliot.recommender.neural.DMF.deep_matrix_fact@5ization,
           75
                                                                      elliot.recommender.NN.attribute_user_knn.att
elliot.recommender.neural.DMF.deep_matrix_fact@5ization_model,
                                                                      elliot.recommender.NN.attribute_user_knn.att
elliot.recommender.neural.GeneralizedMF,
                                                                                 35
                                                                      elliot.recommender.NN.attribute_user_knn.tfi
elliot.recommender.neural.GeneralizedMF.general35zed_matrix_factorization,
                                                                      elliot.recommender.NN.item knn,
elliot.recommender.neural.GeneralizedMF.general3Vzed_matrix_factorization_model,
                                                                      elliot.recommender.NN.item_knn.aiolli_ferrar
elliot.recommender.neural.ItemAutoRec,
                                                                      elliot.recommender.NN.item_knn.item_knn,
elliot.recommender.neural.ItemAutoRec.itemautor36c,
                                                                      elliot.recommender.NN.item_knn.item_knn_simi
elliot.recommender.neural.ItemAutoRec.itemautor%c_model,
                                                                      elliot.recommender.NN.user_knn,
elliot.recommender.neural.NAIS,
                                                                                 38
           79
                                                                      elliot.recommender.NN.user knn.aiolli ferrar
elliot.recommender.neural.NAIS.nais,
                                                                      elliot.recommender.NN.user_knn.user_knn,
elliot.recommender.neural.NAIS.nais_model,
                                                                                38
                                                                      elliot.recommender.NN.user_knn.user_knn_simi
elliot.recommender.neural.NeuMF,
                                                                      elliot.recommender.recommender_utils_mixin,
elliot.recommender.neural.NeuMF.neural_matrix_factorization,
                                                                      elliot.recommender.unpersonalized,
elliot.recommender.neural.NeuMF.neural_matrix_f%5ctorization_model,
                                                                      elliot.recommender.unpersonalized.most_popul
elliot.recommender.neural.NFM, 80
elliot.recommender.neural.NFM.neural_fm, elliot.recommender.unpersonalized.most_popul
                                                                                 84
elliot.recommender.neural.NFM.neural_fm_modet.recommender.unpersonalized.random_rec
elliot.recommender.neural.NPR,80
                                                                      elliot.recommender.unpersonalized.random_rec
elliot.recommender.neural.NPR.neural_personalized_ranking,
                                                                      elliot.recommender.visual_recommenders,
elliot.recommender.neural.NPR.neural_personaliz89d_ranking_model,
                                                                      elliot.recommender.visual_recommenders.ACF,
elliot.recommender.neural.UserAutoRec,
                                                                      elliot.recommender.visual recommenders.ACF.A
elliot.recommender.neural.UserAutoRec.userautor&c,
                                                                      elliot.recommender.visual_recommenders.ACF.A
elliot.recommender.neural.UserAutoRec.userautor85c_model,
                                                                      elliot.recommender.visual_recommenders.DeepS
elliot.recommender.neural.WideAndDeep,
                                                                      elliot.recommender.visual_recommenders.DeepS
elliot.recommender.neural.WideAndDeep.wide_and_&Teep,
```

Python Module Index 99

82

```
elliot.recommender.visual_recommenders.DVBPR,
elliot.recommender.visual_recommenders.DVBPR.DVBPR,
elliot.recommender.visual_recommenders.DVBPR.DVBPR_model,
elliot.recommender.visual recommenders.DVBPR.FeatureExtractor,
elliot.recommender.visual_recommenders.VBPR,
elliot.recommender.visual_recommenders.VBPR.VBPR,
      87
elliot.recommender.visual_recommenders.VBPR.VBPR_model,
elliot.recommender.visual_recommenders.VNPR,
elliot.recommender.visual_recommenders.VNPR.visual_neural_personalized_ranking,
elliot.recommender.visual_recommenders.VNPR.visual_neural_personalized_ranking_model
      88
elliot.result_handler,90
elliot.result_handler.result_handler,
elliot.run,93
elliot.splitter,91
elliot.splitter.base_splitter,90
elliot.utils,93
elliot.utils.folder,91
elliot.utils.logger_util,91
elliot.utils.logging,92
elliot.utils.read,92
elliot.utils.write,93
```

| ^ | applyPearsonCorrelation() (elliot.recom- |
|--|--|
| A | mender.NN.item_knn.aiolli_ferrari.Compute_Similarit |
| AbstractDataset (class in elliot.dataset.abstrac- | method), 36 |
| t_dataset), 8 | applyPearsonCorrelation() (elliot.recom- |
| AbstractRelevanceSingleton (class in elliot.eval-
uation.relevance.relevance), 30 | mender.NN.user_knn.aiolli_ferrari.Compute_Similarity method), 37 |
| ACF (class in elliot.recommender.visual_recom- | ARP (class in elliot.evaluation.metric- |
| menders.ACF.ACF), 85 | s.bias.arp.arp), 16 |
| ACF_model (class in elliot.recommender.visu- | attention() (elliot.recommender.neural.NAIS |
| al_recommenders.ACF_Model), | nais_model.NAIS_model method), 78 |
| 85 | AttributeItemKNN (class in elliot.recom- |
| ACLT (class in elliot.evaluation.metrics.bias.a- | mender.NN.attribute_item_knn.attribute_item_knn), |
| clt.aclt), 15 | 34 |
| add_oneshot_recommender() (elliot.result_han- | AttributeUserKNN (class in elliot.recom- |
| dler.result_handler.ResultHandler | mender.NN.attribute_user_knn.attribute_user_knn), |
| method), 90 | 35 |
| add_trials() (elliot.result_handler.result_han- | AUC (class in elliot.evaluation.metrics.accura- |
| dler.HyperParameterStudy method), | cy.AUC.auc), 10 |
| 90 | autoset_params() (elliot.recom- |
| AFM (class in elliot.recommender.attentive.afm.afm), 44 | mender.base_recommender_model.BaseRecommender method), 89 |
| AFMModel (class in elliot.recommender.atten- | |
| tive.afm.afm_model), 44 | В |
| AiolliSimilarity (class in elliot.recommender.N- | base() (elliot.namespace_mod- |
| N.item_knn.aiolli_ferrari), 36 | el_builder.Builder property), 32 |
| AiolliSimilarity (class in elliot.recommender.N- | base() (elliot.namespace.namespace_mod- |
| N.user_knn.aiolli_ferrari), 37 | el_builder.NameSpaceBuilder proper- |
| AMF (class in elliot.recommender.adversari- | ty), 33 |
| al.AMF.AMF), 38 | BaseMetric (class in elliot.evaluation.metric- |
| AMF_model (class in elliot.recommender.ad- | s.base_metric), 29 |
| versarial.AMF.AMF_model), 39 | BaseRecommenderModel (class in elliot.recom- |
| AMR (class in elliot.recommender.adversari- | mender.base_recommender_model), |
| al.AMR.AMR), 41 | 89 |
| AMR_model (class in elliot.recommender.ad- | batch_attention() (elliot.recommender.neural |
| versarial.AMR.AMR_model), 41 | NAIS.nais_model.NAIS_model |
| APLT (class in elliot.evaluation.metric- | method), 78 |
| s.bias.aplt.aplt), 16 | batch_predict() (elliot.recommender.latent_fac- |
| applyAdjustedCosine() (elliot.recommender.N- | tor_models.FISM.FISM_model.FISM_model |
| N.item_knn.aiolli_ferrari.Compute_Similarit | y method), 64 |
| method), 36 | batch_predict() (elliot.recommender.neural |
| applyAdjustedCosine() (elliot.recommender.N- | NAIS.nais_model.NAIS_model |
| N.user_knn.aiolli_ferrari.Compute_Similarit
method), 37 | |

| batch_softmax() (elliot.recommender.neural
NAIS.nais_model.NAIS_model
method), 78
BiasDisparityBD (class in elliot.evaluation.met- | folder), 91 build_perturbation() (elliot.recommender.adversarial.AMF.AMF_model.AMF_model method), 39 |
|---|--|
| | build_perturbation() (elliot.recommender.ad-
versarial.AMR.AMR_model.AMR_model |
| BiasDisparityBR (class in elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBR), | method), 41 |
| BiasDisparityBS (class in elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparityBS), 22 | build_sparse() (elliot.dataset.dataloader.knowl-
edge_aware_chains.KnowledgeChainsDataObject
method), 5 |
| binary_relevance() (elliot.evaluation.relevance.Relevance property), 31 | build_sparse() (elliot.dataset.dataloader.visu-
al_dataloader.VisualDataObject
method), 6 |
| BinaryRelevance (class in elliot.evaluation.relevance.relevance), 30 | build_sparse() (elliot.dataset.dataset.DataSet method), 9 |
| BPRMF (class in elliot.recommender.latent_fac-
tor_models.BPRMF.BPRMF), 59
BPRMF_batch (class in elliot.recommender.la- | build_sparse_features() (in module elliot.rec-
ommender.neural.WideAndDeep.wide_and_deep),
82 |
| tent_factor_models.BPRMF_batch.BPRMF_b
60 | a dtoh)d_sparse_ratings() (elliot.dataset.dataload-
er.knowledge_aware_chains.KnowledgeChainsDataOb |
| BPRMF_batch_model (class in elliot.recom-
mender.latent_factor_models.BPRMF_batch. | method), 5 BRRM Fspharksh_ratingsl() (elliot.dataset.dataload-
er.visual_dataloader.VisualDataObject |
| BPRSlim (class in elliot.recommender.latent
factor_models.BPRSlim.bprslim), 61 | method), 6 build_sparse_ratings() (elliot.dataset.dataset |
| | DataSet method), 9)Builder (class in elliot.namespace.namespace |
| 61 build_dict() (elliot.dataset.abstract_dataset.AbstractDataset method), 8 | model_builder), 32 |
| build_dict() (elliot.dataset.dataloader.knowl-
edge_aware_chains.KnowledgeChainsDataC
method), 5 | .call() (elliot.recommender.adversarial.AM-
)bject F.AMF_model.AMF_model method), |
| build_dict() (elliot.dataset.dataloader.visual
dataloader.VisualDataObject method), | call() (elliot.recommender.adversarial.AM-
R.AMR_model.AMR_model method), |
| build_dict() (elliot.dataset.dataset.DataSet method), 9 | 41 call() (elliot.recommender.attentive.afm.afm model.AFMModel method), 44 |
| build_feature_sparse() (elliot.recommender
content_based.VSM.vector_space_model.VS
method), 47 | call() (elliot recommender autoencoders dae - |
| build_feature_sparse() (elliot.recommender.N-
N.attribute_item_knn.attribute_item_knn.At
method), 34 | call() (elliot.recommender.autoencoders.dae
tributeltemKNN
multi_dae_model.DenoisingAutoEncoder |
| build_feature_sparse() (elliot.recommender.N-N.attribute_user_knn.attribute_user_knn.Atmethod), 35 | method), 45 call() (elliot.recommender.autoencoders.dae tributeUserKNN_dae_model.Encoder method), |
| | call() (elliot.recommender.autoencoders.vae
odel.VSM
multi_vae_model.Decoder method), |
| havild feature are well-self (alliet reserve | 46 call() (elliot.recommender.autoencoders.vae er_knn.AttributeUserKNN) multi_vae_model.Encoder method), |
| build_log_folder() (in module elliot.utils.fold-
er), 91 | call() (elliot.recommender.autoencoders.vae |
| build_model_folder() (in module elliot.utils | multi_vae_model.Sampling method), |

102 Index

| 46 | call() (elliot.recommender.neural.ConvMF.con- |
|---|---|
| call() (elliot.recommender.autoencoders.vae | volutional_matrix_factorization_model.MLPComponer |
| multi_vae_model.VariationalAutoEncoder | method), 73 |
| method), 46 | call() (elliot.recommender.neural.ConvNeuMF |
| call() (elliot.recommender.gan.IRGAN.irgan | $convolution al_neural_matrix_factorization_model. Con$ |
| model.Discriminator method), 51 | method), 74 |
| call() (elliot.recommender.gan.IRGAN.irgan | call() (elliot.recommender.neural.ConvNeuMF |
| model.Generator method), 51 | convolutional_neural_matrix_factorization_model.Con |
| call() (elliot.recommender.gan.IRGAN.irgan | method), 74 |
| model.IRGAN_model method), 52 | call() (elliot.recommender.neural.ConvNeuMF |
| call() (elliot.recommender.graph_based.lightgc- | convolutional_neural_matrix_factorization_model.MLI |
| n.LightGCN_model.LightGCNModel | method), 75 |
| method), 54 | call() (elliot.recommender.neu- |
| call() (elliot.recommender | ral.DeepFM.deep_fm_model.DeepFMModel |
| graph_based.ngcf.NGCF_model.NGCFMod | |
| method), 55 call() (elliot.recommender.knowl- | <pre>call() (elliot.recommender.neural.DMF.deep</pre> |
| edge_aware.kaHFM_batch.kahfm_batch_mc | |
| method), 57 | call() (elliot.recommender.neural.General- |
| call() (elliot.recommender.knowl- | izedMF.generalized_matrix_factorization_model.General |
| edge_aware.kahfm_embeddings.kahfm_eml | |
| method), 58 | call() (elliot.recommender.neural.ItemAu- |
| call() (elliot.recommender.latent_factor_model- | toRec.itemautorec_model.Decoder |
| s.BPRMF_batch.BPRMF_batch_model.BPRN | |
| method), 60 | call() (elliot.recommender.neural.ItemAu- |
| call() (elliot.recommender.latent_factor_model- | toRec.itemautorec_model.Encoder |
| s.CML.CML_model.CML_model | method), 77 |
| method), 62 | call() (elliot.recommender.neural.ItemAu- |
| call() (elliot.recommender.latent_factor_model- | toRec.itemautorec_model.ItemAutoRecModel |
| | nodel.Field hethreF potorizationMachineModel |
| method), 63 | call() (elliot.recommender.neural.NAIS.nais |
| call() (elliot.recommender.latent_factor_model- | model.NAIS_model method), 78 |
| s.FISM.FISM_model.FISM_model | call() (elliot.recommender.neural.NeuMF.neu- |
| method), 64 | $ral_matrix_factorization_model. Neural Matrix Factorization and the property of the property$ |
| call() (elliot.recommender.latent_factor_model- | method), 81 |
| | atiohly/(adhinteleloubemender.neural.NFM.neu- |
| method), 65 | ral_fm_model.NeuralFactorizationMachineModel |
| call() (elliot.recommender.latent_factor_model- | method), 79 |
| | ekall() (elliot.recommender.neural.NPR.neu- |
| method), 65 | ral_personalized_ranking_model.NPRModel |
| call() (elliot.recommender.latent_factor_model- | method), 80 |
| s.LogisticMF.logistic_matrix_factorization_n | |
| method), 66 | AutoRec.userautorec_model.Decoder |
| call() (elliot.recommender.latent_factor_model-
s.MF.matrix_factorization_model.MatrixFactorization_model.MatrixFactorization_model. | method), 81 |
| method), 67 | AutoRec.userautorec_model.Encoder |
| call() (elliot.recommender.latent_factor_model- | method), 81 |
| s.PMF.probabilistic_matrix_factorization_mo | |
| method), 68 | AutoRec.userautorec_model.UserAutoRecModel |
| call() (elliot.recommender.latent_factor_model- | method), 82 |
| s.SVDpp.svdpp_model.SVDppModel | call() (elliot.recommender.neural.WideAnd- |
| method), 70 | Deep.wide_and_deep_model.WideAndDeepModel |
| call() (elliot.recommender.neural.ConvMF.con- | method), 82 |
| volutional_matrix_factorization_model.Com | |
| method), 72 | menders.ACF.ACF_model.ACF_model |
| call() (elliot.recommender.neural.ConvMF.con- | method), 85 |
| volutional_matrix_factorization_model.Com | |
| method), 73 | menders.DeepStyle.DeepStyle_model.DeepStyle_mode |

Index 103

| method), 87 | content_based.VSM.vector_space_model.VSM |
|---|--|
| call() (elliot.recommender.visual_recom-
menders.DVBPR.DVBPR_model.DVBPR_mo | method), 47 |
| method), 86 | mender.NN.attribute_user_knn.attribute_user_knn.Att |
| call() (elliot.recommender.visual_recom- | method), 35 |
| | kt cortsp rute_cosine() (elliot.recommender.NN.at- |
| method), 86 | tribute_item_knn.attribute_item_knn_similarity.Similar |
| call() (elliot.recommender.visual_recom- | method), 34 |
| menders.VBPR.VBPR_model.VBPR_model | |
| method), 87 | N.item_knn.item_knn_similarity.Similarity |
| call() (elliot.recommender.visual_recom- | method), 37 |
| | _ramkjngt <u>em</u> iddg()VeNRRMdaduhtion.metrics.accu- |
| method), 88 | racy.ndcg.ndcg.NDCG method), 14 |
| censor() (elliot.recommender.latent_factor | compute_idcg() (elliot.evaluation.metrics.fair- |
| models.CML.CML_model.LatentFactor | ness.MAD.UserMADranking.UserMADranking |
| method), 62 | method), 24 |
| censor() (elliot.recommender.latent_factor
models.FISM.FISM_model.LatentFactor | compute_neighbors() (elliot.recommender.N-N.attribute_item_knn.attribute_item_knn_similarity.Sir |
| method), 64 | method), 34 |
| censor() (elliot.recommender.neural.NAIS | compute_neighbors() (elliot.recommender.N- |
| nais_model.LatentFactor method), 78 | N.attribute_user_knn.attribute_user_knn_similarity.Sin |
| CFGAN (class in elliot.recommender.gan.CF- | method), 35 |
| GAN.cfgan), 48 | compute_neighbors() (elliot.recommender.N- |
| CFGAN_model (class in elliot.recommender | N.item_knn.item_knn_similarity.Similarity |
| gan.CFGAN.cfgan_model), 48 | method), 37 |
| <pre>check_matrix() (in module elliot.recom-</pre> | compute_neighbors() (elliot.recommender.N- |
| mender.NN.item_knn.aiolli_ferrari), | N.user_knn.user_knn_similarity.Similarity |
| 36 | method), 38 |
| check_matrix() (in module elliot.recom- | Compute_Similarity (class in elliot.recom- |
| mender.NN.user_knn.aiolli_ferrari),
37 | mender.NN.item_knn.aiolli_ferrari), |
| check_required_attributes() (elliot.dataset.ab- | Compute_Similarity (class in elliot.recom- |
| | itionMeta mender.NN.user_knn.aiolli_ferrari), |
| method), 9 | 37 |
| check_timestamp() (elliot.dataset.dataloader | compute_similarity() (elliot.recommender.N- |
| | sLoader N.item_knn.aiolli_ferrari.Compute_Similarity |
| method), 5 | method), 36 |
| <pre>check_timestamp() (elliot.dataset.dataloader.vi-</pre> | compute_similarity() (elliot.recommender.N- |
| sual_dataloader.VisualLoader | N.user_knn.aiolli_ferrari.Compute_Similarity |
| method), 6 | method), 37 |
| check_timestamp() (elliot.dataset.dataset | compute_user_ndcg() (elliot.evaluation.metric- |
| DataSetLoader method), 9 | s.accuracy.ndcg.ndcg.NDCG method), |
| CML (class in elliot.recommender.latent_fac- | 14 |
| tor_models.CML.CML), 62 | compute_user_ndcg() (elliot.evaluation.metric- |
| CML_model (class in elliot.recommender.laten-
t_factor_models.CML.CML_model), | s.fairness.MAD.UserMADranking.UserMADranking method), 24 |
| 62 | config_test() (in module elliot.run), 93 |
| common_users() (elliot.evaluation.statisti- | ConvMatrixFactorizationModel (class in |
| cal_significance.PairedTTest static | elliot.recommender.neural.ConvMF.convolutional_mat |
| method), 31 | 72 |
| common_users() (elliot.evaluation.statisti- | ConvMF (class in elliot.recommender.neural |
| cal_significance.WilcoxonTest static | ConvMF.convolutional_matrix_factorization), |
| method), 31 | 72 |
| $compare () \ (elliot.evaluation.statistical_signifi-$ | ConvNeuMF (class in elliot.recommender.neu- |
| cance.PairedTTest static method), 31 | ral.ConvNeuMF.convolutional_neural_matrix_factoriza |
| compare() (elliot.evaluation.statistical_signifi- | 74 |

104 Index

 $ConvNeural Matrix Factorization Model\ (class\ in$

 $elliot.recommender.neural. Conv NeuMF. convolutional_$

cance.WilcoxonTest static method), 31

compute_binary_profile() (elliot.recommender.-

| 74 | discounted_relevance() (elliot.evaluation.rele- |
|--|--|
| ConvolutionalComponent (class in elliot.recom- | vance.relevance.Relevance property), |
| mender.neural.ConvMF.convolutional_matri | |
| 73 | DiscountedRelevance (class in elliot.evalua- |
| ConvolutionalComponent (class in elliot.recom- | tion.relevance.relevance), 31 |
| mender.neurai.Convineulvir.convolutionai_r | n edisat<u>i</u>matate_fakto_ritatti()r(ethiotlet); ommender
gan.CFGAN.cfgan_model.Discriminator |
| cosine() (elliot.recommender.neural.DMF.deep- | method), 49 |
| | t Diszatioin Mod (class in elliot.recommender |
| method), 75 | gan.CFGAN.cfgan_model), 49 |
| create_folder_by_index() (in module elliot.util- | Discriminator (class in elliot.recommender |
| s.folder), 91 | gan.IRGAN.irgan_model), 51 |
| create_history_item_matrix() (elliot.recom- | DMF (class in elliot.recommender.neural.DM- |
| mender.latent_factor_models.FISM.FISM_me | |
| method), 64 | dot_prod() (elliot.recommender.latent_factor |
| create_history_item_matrix() (elliot.recom- | models.PMF.probabilistic_matrix_factorization_model. |
| mender.neural.NAIS.nais_model.NAIS_mod | |
| method), 78 | dot_prod() (elliot.recommender.neural.DM- |
| D | F.deep_matrix_factorization_model.DeepMatrixFactori |
| D | method), 75 DSC (class in elliot.evaluation.metrics.accura- |
| dataframe_to_dict() (elliot.dataset.dataloader | - cv DSC dec) 11 |
| knowledge_aware_chains.KnowledgeChains | DataObje Cy.DSC.dsc), 11
DVBPR (class in elliot.recommender.visual_rec- |
| | ommenders.DVBPR.DVBPR), 86 |
| dataframe_to_dict() (elliot.dataset.dataload- | DVBPR_model (class in elliot.recommender.vi- |
| er.visual_dataloader.VisualDataObject
method), 6 | sual_recommenders.DVBPR.DVBPR_model), |
| dataframe_to_dict() (elliot.dataset.dataset | 86 |
| DataSet method), 9 | |
| DataSet (class in elliot.dataset.dataset), 9 | E |
| DataSetLoader (class in elliot.dataset.dataset), 9 | EFD (class in elliot.evaluation.metrics.novel- |
| Decoder (class in elliot.recommender.autoen- | ty.EFD.efd), 26 |
| coders.dae.multi_dae_model), 45 | elliot |
| Decoder (class in elliot.recommender.autoen- | module, 93 |
| coders.vae.multi_vae_model), 46 | elliot.dataset |
| Decoder (class in elliot.recommender.neu- | module, 9 |
| ral.ItemAutoRec.itemautorec_model), | elliot.dataset.abstract_dataset |
| 77 Deceder (class in alliet recommender nous | module, 8 |
| Decoder (class in elliot.recommender.neu-
ral.UserAutoRec.userautorec_model), | elliot.dataset.dataloader
module, 7 |
| 81 | elliot.dataset.dataloader.knowledge_aware_chains |
| DeepFM (class in elliot.recommender.neu- | module, 5 |
| ral.DeepFM.deep_fm), 75 | elliot.dataset.dataloader.visual_dataloader |
| DeepFMModel (class in elliot.recom- | module, 6 |
| mender.neural.DeepFM.deep_fm_model), | elliot.dataset.dataset |
| 76 | module, 9 |
| DeepMatrixFactorizationModel (class in | elliot.dataset.samplers |
| elliot.recommender.neural.DMF.deep_matrix | |
| 75 | elliot.dataset.samplers.custom_pointwise_sparse_sampler |
| DeepStyle (class in elliot.recommender.visu- | module, 7 |
| al_recommenders.DeepStyle.DeepStyle), | elliot.dataset.samplers.custom_sampler |
| 87 | module, 7 |
| DeepStyle_model (class in elliot.recom- | elliot.dataset.samplers.custom_sparse_sampler |
| mender.visual_recommenders.DeepStyle.De
87 | epstyn <u>eonoore</u> 1),
elliot.dataset.samplers.pairwise_sampler |
| Denoising Auto Encoder (class in elliot.recom- | module, 7 |
| mender.autoencoders.dae.multi_dae_model) | |
| 45 | module 7 |

| elliot.dataset.samplers.pointwise_cfgan_sampler module, 7 | elliot.evaluation.metrics.accuracy.precision module, 14 |
|---|--|
| elliot.dataset.samplers.pointwise_pos_neg_ratings_s module, 8 | ampler.evaluation.metrics.accuracy.precision.precision
module, 14 |
| · | |
| elliot.dataset.samplers.pointwise_pos_neg_ratio_rati | |
| module, 8 | module, 15 |
| elliot.dataset.samplers.pointwise_pos_neg_sampler | elliot.evaluation.metrics.accuracy.recall.recall |
| module, 8 | module, 15 |
| elliot.dataset.samplers.pointwise_wide_and_deep_sa | aneplient.evaluation.metrics.base_metric |
| module, 8 | module, 29 |
| elliot.dataset.samplers.sparse_sampler | elliot.evaluation.metrics.bias |
| module, 8 | module, 18 |
| elliot.evaluation | elliot.evaluation.metrics.bias.aclt |
| module, 31 | module, 15 |
| elliot.evaluation.evaluator | elliot.evaluation.metrics.bias.aclt.aclt |
| module, 31 | module, 15 |
| elliot.evaluation.metrics | elliot.evaluation.metrics.bias.aplt |
| | <u>*</u> |
| module, 30 | module, 16 |
| elliot.evaluation.metrics.accuracy | elliot.evaluation.metrics.bias.aplt.aplt |
| module, 15 | module, 16 |
| elliot.evaluation.metrics.accuracy.AUC | elliot.evaluation.metrics.bias.arp |
| module, 11 | module, 16 |
| elliot.evaluation.metrics.accuracy.AUC.auc | elliot.evaluation.metrics.bias.arp.arp |
| module, 9 | module, 16 |
| elliot.evaluation.metrics.accuracy.AUC.gauc | elliot.evaluation.metrics.bias.pop_reo |
| module, 10 | module, 17 |
| elliot.evaluation.metrics.accuracy.AUC.lauc | elliot.evaluation.metrics.bias.pop_reo.extended_pop_reo |
| module, 10 | module, 16 |
| elliot.evaluation.metrics.accuracy.DSC | elliot.evaluation.metrics.bias.pop_reo.pop_reo |
| module, 11 | module, 17 |
| | • |
| elliot.evaluation.metrics.accuracy.DSC.dsc | elliot.evaluation.metrics.bias.pop_rsp |
| module, 11 | module, 18 |
| elliot.evaluation.metrics.accuracy.f1 | elliot.evaluation.metrics.bias.pop_rsp.extended_pop_rsp |
| module, 12 | module, 17 |
| elliot.evaluation.metrics.accuracy.f1.extended_f1 | elliot.evaluation.metrics.bias.pop_rsp.pop_rsp |
| module, 11 | module, 18 |
| elliot.evaluation.metrics.accuracy.f1.f1 | elliot.evaluation.metrics.coverage |
| module, 12 | module, 20 |
| elliot.evaluation.metrics.accuracy.hit_rate | elliot.evaluation.metrics.coverage.item_coverage |
| module, 12 | module, 19 |
| elliot.evaluation.metrics.accuracy.hit_rate.hit_rate | elliot.evaluation.metrics.coverage.item_coverage.item_coverage |
| module, 12 | module, 18 |
| elliot.evaluation.metrics.accuracy.map | elliot.evaluation.metrics.coverage.num_retrieved |
| module, 13 | module, 19 |
| | |
| elliot.evaluation.metrics.accuracy.map.map | elliot.evaluation.metrics.coverage.num_retrieved.num_retrieved |
| module, 12 | module, 19 |
| elliot.evaluation.metrics.accuracy.mar | elliot.evaluation.metrics.coverage.user_coverage |
| module, 13 | module, 20 |
| elliot.evaluation.metrics.accuracy.mar.mar | elliot.evaluation.metrics.coverage.user_coverage.user_coverage |
| module, 13 | module, 19 |
| elliot.evaluation.metrics.accuracy.mrr | elliot.evaluation.metrics.coverage.user_coverage.user_coverage_ |
| module, 13 | module, 19 |
| elliot.evaluation.metrics.accuracy.mrr.mrr | elliot.evaluation.metrics.diversity |
| module, 13 | module, 21 |
| elliot.evaluation.metrics.accuracy.ndcg | elliot.evaluation.metrics.diversity.gini_index |
| module, 14 | module, 21 |
| elliot.evaluation.metrics.accuracy.ndcg.ndcg | elliot.evaluation.metrics.diversity.gini_index.gini_index |
| | |
| module, 13 | module, 20 |

| elliot.evaluation.metrics.diversity.shannon_entropy | elliot.evaluation.metrics.rating.mse |
|--|---|
| module, 21 | module, 28 |
| elliot.evaluation.metrics.diversity.shannon_entropy.sh | · |
| module, 21 | module, 28 |
| elliot.evaluation.metrics.diversity.SRecall | elliot.evaluation.metrics.rating.rmse |
| • | |
| module, 20 | module, 29 |
| elliot.evaluation.metrics.diversity.SRecall.srecall | elliot.evaluation.metrics.rating.rmse.rmse |
| module, 20 | module, 28 |
| elliot.evaluation.metrics.fairness | elliot.evaluation.metrics.statistical_array_metric |
| module, 26 | module, 29 |
| elliot.evaluation.metrics.fairness.BiasDisparity | elliot.evaluation.popularity_utils |
| module, 23 | module, 30 |
| elliot.evaluation.metrics.fairness.BiasDisparity.BiasDis | sphiotyBD luation.popularity_utils.popularity |
| module, 21 | module, 30 |
| elliot. evaluation. metrics. fairness. Bias Disparity. Bias Disparity. Bias Disparity. | s phiotyB Rluation.relevance |
| module, 22 | module, 31 |
| elliot.evaluation.metrics.fairness.BiasDisparity.BiasDis | |
| module, 22 | module, 30 |
| elliot.evaluation.metrics.fairness.MAD | elliot.evaluation.statistical_significance |
| module, 24 | module, 31 |
| | |
| elliot.evaluation.metrics.fairness.MAD.ItemMADrank | |
| module, 23 | module, 32 |
| elliot.evaluation.metrics.fairness.MAD.ItemMADratir | |
| module, 23 | module, 32 |
| elliot.evaluation.metrics.fairness.MAD.UserMADrank | |
| module, 23 | module, 33 |
| elliot. evaluation. metrics. fairness. MAD. User MAD rational fairness and the state of the st | |
| module, 24 | module, 32 |
| elliot.evaluation.metrics.fairness.reo | elliot.namespace.namespace_model_builder |
| module, 25 | module, 32 |
| elliot.evaluation.metrics.fairness.reo.reo | elliot.prefiltering |
| module, 24 | module, 33 |
| elliot.evaluation.metrics.fairness.rsp | elliot.prefiltering.standard_prefilters |
| module, 26 | module, 33 |
| elliot.evaluation.metrics.fairness.rsp.rsp | elliot.recommender |
| module, 25 | module, 90 |
| elliot.evaluation.metrics_utils | elliot.recommender.adversarial |
| module, 29 | module, 43 |
| elliot.evaluation.metrics.novelty | elliot.recommender.adversarial.AMF |
| module, 27 | module, 41 |
| | elliot.recommender.adversarial.AMF.AMF |
| elliot.evaluation.metrics.novelty.EFD | |
| module, 26 | module, 38 |
| elliot.evaluation.metrics.novelty.EFD.efd | elliot.recommender.adversarial.AMF.AMF_model |
| module, 26 | module, 39 |
| elliot.evaluation.metrics.novelty.EFD.extended_efd | elliot.recommender.adversarial.AMR |
| module, 26 | module, 43 |
| elliot.evaluation.metrics.novelty.EPC | elliot.recommender.adversarial.AMR.AMR |
| module, 27 | module, 41 |
| elliot.evaluation.metrics.novelty.EPC.epc | elliot.recommender.adversarial.AMR.AMR_model |
| module, 27 | module, 41 |
| elliot.evaluation.metrics.novelty.EPC.extended_epc | elliot.recommender.algebric |
| module, 27 | module, 44 |
| elliot.evaluation.metrics.rating | elliot.recommender.algebric.slope_one |
| module, 29 | module, 43 |
| | |
| elliot.evaluation.metrics.rating.mae | elliot.recommender.algebric.slope_one.slope_one |
| module, 28 | module, 43 |
| elliot.evaluation.metrics.rating.mae.mae | elliot.recommender.algebric.slope_one.slope_one_model |
| module, 27 | module, 43 |

| elliot.recommender.attentive
module, 44 | elliot.recommender.graph_based.ngcf.NGCF
module, 55 |
|---|--|
| elliot.recommender.attentive.afm
module, 44 | elliot.recommender.graph_based.ngcf.NGCF_model module, 55 |
| elliot.recommender.attentive.afm.afm
module, 44 | elliot.recommender.knowledge_aware module, 59 |
| elliot.recommender.attentive.afm.afm_model module, 44 | elliot.recommender.knowledge_aware.kaHFM module, 57 |
| elliot.recommender.autoencoders
module, 46 | elliot.recommender.knowledge_aware.kaHFM.ka_hfm module, 56 |
| elliot.recommender.autoencoders.dae
module, 45 | elliot.recommender.knowledge_aware.kaHFM.tfidf_utils module, 57 |
| elliot.recommender.autoencoders.dae.multi_dae module, 44 | elliot.recommender.knowledge_aware.kaHFM_batch module, 58 |
| | d el liot.recommender.knowledge_aware.kaHFM_batch.kahfm_bat
module, 57 |
| elliot.recommender.autoencoders.vae
module, 46 | elliot.recommender.knowledge_aware.kaHFM_batch.kahfm_bat
module, 57 |
| elliot.recommender.autoencoders.vae.multi_vae module, 45 | elliot.recommender.knowledge_aware.kaHFM_batch.tfidf_utils module, 58 |
| elliot.recommender.autoencoders.vae.multi_vae_mod
module, 46 | lællliot.recommender.knowledge_aware.kahfm_embeddings
module, 59 |
| elliot.recommender.base_recommender_model module, 89 | elliot.recommender.knowledge_aware.kahfm_embeddings.kahfm
module, 58 |
| elliot.recommender.content_based module, 47 | elliot.recommender.knowledge_aware.kahfm_embeddings.kahfm
module, 58 |
| elliot.recommender.content_based.VSM module, 47 | elliot.recommender.knowledge_aware.kahfm_embeddings.tfidf_module, 59 |
| elliot.recommender.content_based.VSM.tfidf_utils module, 47 | elliot.recommender.latent_factor_models
module, 72 |
| elliot.recommender.content_based.VSM.vector_space
module, 47 | _ehlbotter ecommender.latent_factor_models.BPRMF
module, 60 |
| module, 47 | _ehlboterlesiom:haveinty ler.latent_factor_models.BPRMF.BPRMF
module, 59 |
| elliot.recommender.gan
module, 54 | elliot.recommender.latent_factor_models.BPRMF_batch module, 61 |
| elliot.recommender.gan.CFGAN module, 50 | elliot.recommender.latent_factor_models.BPRMF_batch.BPRMF module, 60 |
| elliot.recommender.gan.CFGAN.cfgan
module, 48 | elliot.recommender.latent_factor_models.BPRMF_batch.BPRMF module, 60 |
| elliot.recommender.gan.CFGAN.cfgan_model module, 48 | elliot.recommender.latent_factor_models.BPRSlim
module, 61 |
| elliot.recommender.gan.IRGAN
module, 54 | elliot.recommender.latent_factor_models.BPRSlim.bprslim module, 61 |
| elliot.recommender.gan.IRGAN.irgan
module, 50 | elliot.recommender.latent_factor_models.BPRSlim.bprslim_mod
module, 61 |
| elliot.recommender.gan.IRGAN.irgan_model
module, 50 | elliot.recommender.latent_factor_models.CML module, 62 |
| elliot.recommender.graph_based
module, 56 | elliot.recommender.latent_factor_models.CML.CML
module, 61 |
| elliot.recommender.graph_based.lightgcn
module, 55 | elliot.recommender.latent_factor_models.CML.CML_model module, 62 |
| elliot.recommender.graph_based.lightgcn.LightGCN module, 54 | module, 63 |
| module, 54 | _reliidel recommender.latent_factor_models.FFM.field_aware_factor
module, 62 |
| elliot.recommender.graph_based.ngcf
module, 55 | elliot.recommender.latent_factor_models.FFM.field_aware_factor
module, 63 |

| elliot.recommender.latent_factor_models.FISM module, 64 | elliot.recommender.latent_factor_models.SVDpp.svdpp_mode module, 70 |
|---|---|
| elliot.recommender.latent_factor_models.FISM.FISM module, 63 | · |
| $elliot.recommender.latent_factor_models.FISM.FISM_$ | _module, 72
_module, 72
_module, 72
_module, 71 |
| module, 63 elliot.recommender.latent_factor_models.FM | $elliot.recommender.latent_factor_models.WRMF.wrmf_model$ |
| module, 65 elliot.recommender.latent_factor_models.FM.factoriz | |
| module, 64
elliot.recommender.latent_factor_models.FM.factoriz | module, 84 |
| module, 64 | module, 74 |
| elliot.recommender.latent_factor_models.FunkSVD module, 66 | elliot.recommender.neural.ConvMF.convolutional_matrix_factor
module, 72 |
| | u rkliot.d ecommender.neural.ConvMF.convolutional_matrix_factors module, 72 |
| elliot.recommender.latent_factor_models.FunkSVD.fu
module, 65 | · |
| • | elliot.recommender.neural.ConvNeuMF.convolutional_neural_
module, 74 |
| | lædiktaematrinefadeorieutialrConvNeuMF.convolutional_neural_
module, 74 |
| elliot.recommender.latent_factor_models.LogisticMF. | |
| module, 66 | module, 76 |
| elliot.recommender.latent_factor_models.MF
module, 67 | elliot.recommender.neural.DeepFM.deep_fm
module, 75 |
| elliot.recommender.latent_factor_models.MF.matrix_module, 66 | fæd livizætiom mender.neural.DeepFM.deep_fm_model
module, 76 |
| elliot.recommender.latent_factor_models.MF.matrix_module, 67 | fæd tiví zætiom_mendel r.neural.DMF
module, 75 |
| | Felliot.recommender.neural.DMF.deep_matrix_factorization module, 75 |
| | F. nedlinotnægatinvmendlerixnefactabilitælltfadeep_matrix_factorization_module, 75 |
| elliot.recommender.latent_factor_models.NonNegMF module, 67 | |
| elliot.recommender.latent_factor_models.PMF module, 69 | elliot.recommender.neural.GeneralizedMF.generalized_matrix_module, 76 |
| | pi elskid<u>.</u>neatnix<u>n</u>fauderizetion l.GeneralizedMF.generalized_matrix
module, 76 |
| elliot.recommender.latent_factor_models.PMF.probal | |
| module, 68 | module, 78 |
| elliot.recommender.latent_factor_models.PureSVD module, 69 | elliot.recommender.neural.ItemAutoRec.itemautorec
module, 77 |
| elliot.recommender.latent_factor_models.PureSVD.pt
module, 69 | urdlist decommender.neural.ItemAutoRec.itemautorec_model module, 77 |
| elliot.recommender.latent_factor_models.PureSVD.pt
module, 69 | u rd<u>l</u>ist d<u>e</u>moderl ender.neural.NAIS
module, 79 |
| elliot.recommender.latent_factor_models.Slim
module, 71 | elliot.recommender.neural.NAIS.nais
module, 78 |
| elliot.recommender.latent_factor_models.Slim.slim module, 70 | elliot.recommender.neural.NAIS.nais_model module, 78 |
| elliot.recommender.latent_factor_models.Slim.slim_n | |
| module, 71 | module, 81 |
| elliot.recommender.latent_factor_models.SVDpp
module, 70 | elliot.recommender.neural.NeuMF.neural_matrix_factorization module, 80 |
| elliot.recommender.latent_factor_models.SVDpp.svd
module, 70 | pplliot.recommender.neural.NeuMF.neural_matrix_factorization
module, 81 |

| elliot.recommender.neural.NFM module, 80 | elliot.recommender.unpersonalized module, 85 |
|--|--|
| elliot.recommender.neural.NFM.neural_fm
module, 79 | elliot.recommender.unpersonalized.most_popular
module, 84 |
| elliot.recommender.neural.NFM.neural_fm_model module, 79 | elliot.recommender.unpersonalized.most_popular.most_popul |
| elliot.recommender.neural.NPR
module, 80 | elliot.recommender.unpersonalized.random_recommender
module, 85 |
| | radhistry commender.unpersonalized.random_recommender.Random_dule, 85 |
| elliot.recommender.neural.NPR.neural_personalized_module, 80 | |
| elliot.recommender.neural.UserAutoRec
module, 82 | elliot.recommender.visual_recommenders.ACF module, 85 |
| elliot.recommender.neural.UserAutoRec.userautorec module, 81 | |
| | rabidelrecommender.visual_recommenders.ACF.ACF_model module, 85 |
| elliot.recommender.neural.WideAndDeep
module, 84 | elliot.recommender.visual_recommenders.DeepStyle module, 87 |
| | delipt.recommender.visual_recommenders.DeepStyle.DeepStyle module, 87 |
| | _delipt_recommenders.DeepStyle.DeepStyle_module, 87 |
| elliot.recommender.NN
module, 38 | elliot.recommender.visual_recommenders.DVBPR module, 86 |
| elliot.recommender.NN.attribute_item_knn
module, 34 | elliot.recommender.visual_recommenders.DVBPR.DVBPR module, 86 |
| | _ieeliotkracommender.visual_recommenders.DVBPR.DVBPR_mod module, 86 |
| | _i eeliotknro_similariey .visual_recommenders.DVBPR.FeatureExtracemodule, 86 |
| elliot.recommender.NN.attribute_user_knn module, 35 | elliot.recommender.visual_recommenders.VBPR module, 88 |
| | usleinotkurraommender.visual_recommenders.VBPR.VBPR module, 87 |
| · | usleinotkurenosiumidandteyr.visual_recommenders.VBPR.VBPR_model module, 87 |
| elliot.recommender.NN.attribute_user_knn.tfidf_utils
module, 35 | |
| elliot.recommender.NN.item_knn
module, 37 | elliot.recommender.visual_recommenders.VNPR.visual_neural_module, 88 |
| elliot.recommender.NN.item_knn.aiolli_ferrari module, 36 | elliot.recommender.visual_recommenders.VNPR.visual_neural_module, 88 |
| elliot.recommender.NN.item_knn.item_knn
module, 36 | elliot.result_handler
module, 90 |
| elliot.recommender.NN.item_knn.item_knn_similarit module, 36 | yelliot.result_handler.result_handler
module, 90 |
| elliot.recommender.NN.user_knn
module, 38 | elliot.run
module, 93 |
| elliot.recommender.NN.user_knn.aiolli_ferrari
module, 37 | elliot.splitter
module, 91 |
| elliot.recommender.NN.user_knn.user_knn
module, 38 | elliot.splitter.base_splitter
module, 90 |
| elliot.recommender.NN.user_knn.user_knn_similarity
module, 38 | yelliot.utils
module, 93 |
| elliot.recommender.recommender_utils_mixin | elliot.utils.folder |
| module, 89 | module, 91 |

| elliot.utils.logger_util | eval() (elliot.evaluation.metrics.fairness.Bias- |
|---|--|
| module, 91 | Disparity.BiasDisparityBR.BiasDisparityBR |
| elliot.utils.logging | method), 22 |
| module, 92 | eval() (elliot.evaluation.metrics.fairness.Bias- |
| elliot.utils.read | Disparity.BiasDisparityBS.BiasDisparityBS |
| module, 92 | method), 22 |
| elliot.utils.write | eval() (elliot.evaluation.metrics.fairness |
| module, 93 | |
| | MAD.ItemMADranking.ItemMADranking |
| emit() (elliot.utils.logger_util.QueueListener- | method), 23 |
| Handler method), 91 | eval() (elliot.evaluation.metrics.fairness |
| Encoder (class in elliot.recommender.autoen- | MAD.ItemMADrating.ItemMADrating |
| coders.dae.multi_dae_model), 45 | method), 23 |
| Encoder (class in elliot.recommender.autoen- | eval() (elliot.evaluation.metrics.fairness |
| coders.vae.multi_vae_model), 46 | MAD.UserMADranking.UserMADranking |
| Encoder (class in elliot.recommender.neu- | method), 24 |
| ral.ItemAutoRec.itemautorec_model), | eval() (elliot.evaluation.metrics.fairness |
| 77 | MAD.UserMADrating.UserMADrating |
| Encoder (class in elliot.recommender.neu- | method), 24 |
| ral.UserAutoRec.userautorec_model), | eval() (elliot.evaluation.metrics.fairness.reo.re- |
| 81 | |
| V = | o.REO method), 25 |
| EPC (class in elliot.evaluation.metrics.novel- | eval() (elliot.evaluation.metrics.fairness.rsp.r- |
| ty.EPC.epc), 27 | sp.RSP method), 26 |
| eval() (elliot.evaluation.evaluator.Evaluator | eval() (elliot.evaluation.metrics.metrics_util- |
| method), 31 | s.ProxyMetric method), 29 |
| eval() (elliot.evaluation.metrics.accura- | eval() (elliot.evaluation.metrics.metrics_util- |
| cy.AUC.auc.AUC method), 10 | s.ProxyStatisticalMetric method), 29 |
| eval() (elliot.evaluation.metrics.accuracy.AUC | eval() (elliot.evaluation.metrics.rating.mae |
| gauc.GAUC method), 10 | mae.MAE method), 28 |
| | |
| eval() (elliot.evaluation.metrics.base_met- | eval() (elliot.evaluation.metrics.rat- |
| ric.BaseMetric method), 29 | ing.mse.mse.MSE method), 28 |
| eval() (elliot.evaluation.metrics.bias.pop_re- | eval() (elliot.evaluation.metrics.rat- |
| o.extended_pop_reo.ExtendedPopREO | ing.rmse.rmse.RMSE method), 29 |
| method), 17 | eval_at_k() (elliot.evaluation.evaluator.Evalu- |
| eval() (elliot.evaluation.metrics.bias.pop_re- | ator method), 31 |
| o.pop_reo.PopREO method), 17 | eval_user_metric() (elliot.evaluation.metrics.ac- |
| eval() (elliot.evaluation.metrics.bias.pop_r- | curacy.AUC.gauc.GAUC method), 10 |
| sp.extended_pop_rsp.ExtendedPopRSP | eval_user_metric() (elliot.evaluation.metrics.ac- |
| method), 18 | curacy.AUC.lauc.LAUC method), 11 |
| eval() (elliot.evaluation.metrics.bias.pop_r- | eval_user_metric() (elliot.evaluation.metrics.ac- |
| | |
| sp.pop_rsp.PopRSP method), 18 | curacy.DSC.dsc.DSC method), 11 |
| eval() (elliot.evaluation.metrics.coverage.item | eval_user_metric() (elliot.evaluation.metrics.ac- |
| coverage.item_coverage.ItemCoverage | curacy.f1.extended_f1.ExtendedF1 |
| method), 19 | method), 11 |
| eval() (elliot.evaluation.metrics.coverage.user | eval_user_metric() (elliot.evaluation.metrics.ac- |
| coverage.user_coverage.UserCoverage | curacy.f1.f1.F1 method), 12 |
| method), 19 | eval_user_metric() (elliot.evaluation.metrics.ac- |
| eval() (elliot.evaluation.metrics.coverage.user | curacy.hit_rate.hit_rate.HR method), |
| coverage_user_coverage_at_n.UserCoverage_A | |
| method), 20 | eval_user_metric() (elliot.evaluation.metrics.ac- |
| | |
| eval() (elliot.evaluation.metrics.diversi- | curacy.map.map.MAP method), 13 |
| ty.gini_index.gini_index.GiniIndex | eval_user_metric() (elliot.evaluation.metrics.ac- |
| method), 21 | curacy.mar.MAR method), 13 |
| eval() (elliot.evaluation.metrics.diversity.shan- | eval_user_metric() (elliot.evaluation.metrics.ac- |
| non_entropy.shannon_entropy.ShannonEntr | |
| method), 21 | eval_user_metric() (elliot.evaluation.metrics.ac- |
| eval() (elliot.evaluation.metrics.fairness.Bias- | curacy.ndcg.ndcg.NDCG method), 14 |
| Disparity.BiasDisparityBD.BiasDisparityBD | |
| method), 21 | curacy.precision.precision.Precision |
| 111001), 21 | curacy.precision.precision.recision |

| method), 14 | 64 |
|--|--|
| eval_user_metric() (elliot.evaluation.metrics.ac- | FeatureExtractor (class in elliot.recom- |
| curacy.recall.recall.Recall method), 15 | mender.visual_recommenders.DVBPR.FeatureExtractor |
| eval_user_metric() (elliot.evaluation.metric- | 86 |
| s.bias.aclt.aclt.ACLT method), 15 | FFM (class in elliot.recommender.latent_fac- |
| eval_user_metric() (elliot.evaluation.metric- | tor_models.FFM.field_aware_factorization_machine), |
| s.bias.aplt.aplt.APLT method), 16 | 62 |
| eval_user_metric() (elliot.evaluation.metric- | FieldAwareFactorizationMachineModel (class |
| s.bias.arp.arp.ARP method), 16 | in elliot.recommender.latent_factor |
| eval_user_metric() (elliot.evaluation.metrics | models.FFM.field_aware_factorization_machine_mode |
| coverage.num_retrieved.num_retrieved.Nur | |
| method), 19 | fill_base() (elliot.namespace.namespace_mod- |
| eval_user_metric() (elliot.evaluation.metrics.di- | el.NameSpaceModel method), 32 |
| versity.SRecall.srecall.SRecall method), | fill_model() (elliot.namespace.namespace_mod- |
| 20 | el.NameSpaceModel method), 32 |
| eval_user_metric() (elliot.evaluation.metrics | filter() (elliot.prefiltering.standard_prefilter- |
| metrics_utils.ProxyStatisticalMetric | s.PreFilter static method), 33 |
| method), 29 | filter() (elliot.utils.logging.TimeFilter method), |
| eval_user_metric() (elliot.evaluation.metric- | 92 |
| s.novelty.EFD.efd.EFD method), 26 | filter_items_by_popularity() (elliot.prefiltering |
| eval_user_metric() (elliot.evaluation.metric- | standard_prefilters.PreFilter static |
| s.novelty.EFD.extended_efd.ExtendedEFD | method), 33 |
| method), 26 | filter_iterative_k_core() (elliot.prefiltering.stan- |
| eval_user_metric() (elliot.evaluation.metric- | dard_prefilters.PreFilter static |
| s.novelty.EPC.epc.EPC method), 27 | method), 33 |
| eval_user_metric() (elliot.evaluation.metric- | filter_ratings_by_global_average() (elliot.pre- |
| s.novelty.EPC.extended_epc.ExtendedEPC | filtering.standard_prefilters.PreFilter |
| | |
| method), 27 | static method), 33 |
| eval_user_metric() (elliot.evaluation.metrics.rat- | filter_ratings_by_threshold() (elliot.prefilter- |
| ing.mae.mae.MAE method), 28 | ing.standard_prefilters.PreFilter static |
| eval_user_metric() (elliot.evaluation.metrics.rat- | method), 33 |
| ing.mse.mse.MSE method), 28 | filter_ratings_by_user_average() (elliot.prefilter- |
| eval_user_metric() (elliot.evaluation.metrics.rat- | ing.standard_prefilters.PreFilter static |
| ing.rmse.rmse.RMSE method), 29 | method), 33 |
| eval_user_metric() (elliot.evaluation.metrics.s- | filter_retain_cold_users() (elliot.prefiltering.s- |
| tatistical_array_metric.StatisticalMetric | tandard_prefilters.PreFilter static |
| method), 30 | method), 33 |
| Evaluator (class in elliot.evaluation.evaluator), | filter_rounds_k_core() (elliot.prefiltering.stan- |
| 31 | dard_prefilters.PreFilter static |
| ExtendedEFD (class in elliot.evaluation.metric- | method), 33 |
| s.novelty.EFD.extended_efd), 26 | filter_users_by_profile_size() (elliot.prefilter- |
| ExtendedEPC (class in elliot.evaluation.metric- | ing.standard_prefilters.PreFilter static |
| s.novelty.EPC.extended_epc), 27 | method), 33 |
| ExtendedF1 (class in elliot.evaluation.metric- | find_checkpoint() (in module elliot.utils.read), |
| s.accuracy.f1.extended_f1), 11 | 92 |
| ExtendedPopREO (class in elliot.evaluation | FISM (class in elliot.recommender.latent_fac- |
| metrics.bias.pop_reo.extended_pop_reo), | tor_models.FISM.FISM), 63 |
| 16 | FISM_model (class in elliot.recommender.laten- |
| ExtendedPopRSP (class in elliot.evaluation.met- | t_factor_models.FISM.FISM_model), |
| rics.bias.pop_rsp.extended_pop_rsp), | 63 |
| 17 | FM (class in elliot.recommender.latent_factor |
| | models.FM.factorization_machine), 64 |
| F | fold_list_generator() (elliot.splitter.base_split- |
| F1 (class in elliot.evaluation.metrics.accura- | ter.Splitter method), 90 |
| cy.f1.f1), 12 | ForceRequiredAttributeDefinitionMeta (class in |
| Factorization Machine Model (class in alliet res | elliot.dataset.abstract_dataset), 9 |
| ommender.latent_factor_models.FM.factoriz | FunkSVD (class in elliot.recommender.latent
ration_machine_model),
factor_models.FunkSVD.funk_svd), 65 |

| FunkSVDModel (class in elliot.recommender.latent_factor_models.FunkSVD.funk_svd_models.FunkSVD.funkSVD | get() (elliot.evaluation.metrics.fairness
del), MAD.UserMADrating.UserMADrating
method), 24 |
|--|--|
| | get() (elliot.evaluation.metrics.fairness.reo.re- |
| G | o.REO method), 25 |
| GAUC (class in elliot.evaluation.metrics.accura- | get() (elliot.evaluation.metrics.fairness.rsp.r- |
| cy.AUC.gauc), 10 | sp.RSP method), 26 |
| GeneralizedMatrixFactorizationModel (class in | get_BR() (elliot.evaluation.metrics.fairness.Bias- |
| elliot.recommender.neural.GeneralizedMF.gr | eneralized_matrixyBiasDisparityBR_BiasDisparityBR
method), 22 |
| gararata datachiacts() (alliat datacat dataland | get_BS() (elliot.evaluation.metrics.fairness.Bias- |
| er.knowledge_aware_chains.KnowledgeChaimethod), 5 | insLoader Disparity.BiasDisparityBS.BiasDisparityBS method), 22 |
| generate_dataobjects() (elliot.dataset.dataload- | get_config() (elliot.recommender.adversari- |
| er.visual_dataloader.VisualLoader | al.AMF.AMF_model.AMF_model |
| method), 6 | method), 39 |
| generate_dataobjects() (elliot.dataset.dataset | get_config() (elliot.recommender.adversari- |
| DataSetLoader method), 9 | al.AMR.AMR_model.AMR_model |
| compared detections and alliest detect det | method), 41 |
| aloader.knowledge_aware_chains.Knowledg | get config() (elliot.recommender.autoencoder-
ee hainsLoader
s.dae.multi dae model.DenoisingAutoEncoder |
| method), 5 | overacement grade in the constraint of the const |
| generate_dataobjects_mock() (elliot.dataset.dat- | method), 45 |
| aloader.visual_dataloader.VisualLoader | get_config() (elliot.recommender.autoencoder-
s.vae.multi_vae_model.VariationalAutoEncoder |
| method), 6 | method), 46 |
| generate_dataobjects_mock() (elliot.dataset | get_config() (elliot.recommender.gan.CF- |
| dataset.DataSetLoader method), 9 | GAN.cfgan_model.CFGAN_model |
| generate_fake_data() (elliot.recommender | method), 48 |
| gan.CFGAN.cfgan_model.Generator | get_config() (elliot.recommender.gan.IRGAN.ir- |
| method), 50
Generator (class in elliot.recommender.gan.CF- | gan_model.IRGAN_model method), |
| GAN.cfgan_model), 50 | 52 |
| Generator (class in elliot.recommender.gan.IR- | <pre>get_config() (elliot.recommender.graph_based</pre> |
| GAN.irgan_model), 51 | lightgcn.LightGCN_model.LightGCNModel |
| generic_split_function() (elliot.splitter.base_s- | method), 54 |
| plitter.Splitter method), 90 | get_config() (elliot.recommender |
| get() (elliot.evaluation.metrics.accuracy.f1.ex- | graph_based.ngcf.NGCF_model.NGCFModel |
| tended_f1.ExtendedF1 method), 11 | method), 55 |
| get() (elliot.evaluation.metrics.base_met- | get_config() (elliot.recommender.knowl- |
| ric.BaseMetric method), 29 | edge_aware.kaHFM_batch.kahfm_batch_model.KaH
method), 57 |
| get() (elliot.evaluation.metrics.fairness.BiasDis- | get_config() (elliot.recommender.knowl- |
| parity.BiasDisparityBD.BiasDisparityBD | edge_aware.kahfm_embeddings.kahfm_embeddings |
| method), 21 | method), 58 |
| get() (elliot.evaluation.metrics.fairness.BiasDis- | get_config() (elliot.recommender.latent_factor |
| parity.BiasDisparityBR.BiasDisparityBR | models.BPRMF_batch.BPRMF_batch_model.BPRMF_ |
| method), 22
get() (elliot.evaluation.metrics.fairness.BiasDis- | method), 60 |
| parity.BiasDisparityBS.BiasDisparityBS | <pre>get_config() (elliot.recommender.latent_factor</pre> |
| method), 22 | models.CML.CML_model |
| get() (elliot.evaluation.metrics.fairness | method), 62 |
| MAD.ItemMADranking.ItemMADranking | get_config() (elliot.recommender.latent_factor |
| method), 23 | models.FISM_FISM_model |
| get() (elliot.evaluation.metrics.fairness | method), 64 |
| MAD.ItemMADrating.ItemMADrating | get_config() (elliot.recommender.neu- |
| method), 23 | ral.ItemAutoRec.itemautorec_model.ItemAutoRecMo |

get() (elliot.evaluation.metrics.fairness.-

method), 24

MAD.UserMADranking.UserMADranking

method), 77

 $get_config()\ (elliot.recommender.neural.NAIS.-$

get_config() (elliot.recommender.neural.User-

nais_model.NAIS_model method), 78

| $AutoRec.userautorec_model.UserAutoRecM$ | odel method), 59 |
|--|---|
| method), 82 | get_model_state() (elliot.recommender.latent |
| get_config() (elliot.recommender.visual_recom- | factor_models.BPRSlim.bprslim_model.BPRSlimMode |
| menders.ACF.ACF_model.ACF_model | method), 61 |
| method), 85 | get_model_state() (elliot.recommender.latent |
| get_config() (elliot.recommender.visual_recom- | factor_models.NonNegMF.non_negative_matrix_facto |
| menders.DeepStyle.DeepStyle_model.DeepS | |
| method), 87 | get_model_state() (elliot.recommender.latent |
| get_config() (elliot.recommender.visual_recom- | factor_models.PureSVD.pure_svd_model.PureSVDMo |
| menders.DVBPR.DVBPR_model.DVBPR_mo | |
| method), 86 | get_model_state() (elliot.recommender.latent
factor_models.Slim.slim_model.SlimModel |
| get_config() (elliot.recommender.visual_recommenders.VBPR.VBPR_model.VBPR_model | method), 71 |
| method), 88 | get_model_state() (elliot.recommender.latent |
| get_custom_pop_obj() (elliot.evaluation.popu- | factor_models.WRMF.wrmf_model.WRMFModel |
| larity_utils.popularity.Popularity | method), 71 |
| method), 30 | get_model_state() (elliot.recommender.NN.at- |
| get_factors() (elliot.recommender.knowl- | tribute_item_knn.attribute_item_knn_similarity.Simila |
| edge_aware.kaHFM.ka_hfm.MF | method), 34 |
| method), 56 | get_model_state() (elliot.recommender.NN.at- |
| get_item_bias() (elliot.recommender.knowl- | tribute_user_knn.attribute_user_knn_similarity.Simila |
| edge_aware.kaHFM.ka_hfm.MF | method), 35 |
| method), 56 | get_model_state() (elliot.recommender.N- |
| <pre>get_item_bias() (elliot.recommender.latent_fac-</pre> | N.item_knn.item_knn_similarity.Similarity |
| tor_models.BPRMF.BPRMF.MF | method), 37 |
| method), 59 | get_model_state() (elliot.recommender.N- |
| get_item_factors() (elliot.recommender.knowl- | N.user_knn.user_knn_similarity.Similarity |
| edge_aware.kaHFM.ka_hfm.MF | method), 38 |
| method), 56 | get_needed_recommendations() (elliot.evalua- |
| get_item_factors() (elliot.recommender.latent | tion.evaluator.Evaluator method), 31 |
| factor_models.BPRMF.BPRMF.MF | get_params() (elliot.recommender.base_recom- |
| method), 59 | mender_model.BaseRecommenderModel |
| get_item_neighbors() (elliot.recommender.N- | method), 89 |
| N.attribute_item_knn.attribute_item_knn_si | |
| method), 34 | mender_utils_mixin.RecMixin |
| get_item_neighbors() (elliot.recommender.N- | method), 89 |
| N.item_knn.item_knn_similarity.Similarity | get_params_shortcut() (elliot.recom- |
| method), 37 get_logger() (in module elliot.utils.logging), 92 | mender.base_recommender_model.BaseRecommender |
| | method), 89 |
| get_long_tail() (elliot.evaluation.popularity_u-tils.popularity.Popularity method), 30 | get_pop_items() (elliot.evaluation.populari-
ty_utils.popularity.Popularity |
| get_loss() (elliot.recommender.base_recom- | method), 30 |
| mender_model.BaseRecommenderModel | get_positions() (elliot.recommender.adversari- |
| method), 89 | al.AMF.AMF_model.AMF_model |
| get_loss() (elliot.recommender.recom- | method), 39 |
| mender_utils_mixin.RecMixin | get_positions() (elliot.recommender.gan.IR- |
| method), 89 | GAN.irgan_model.IRGAN_model |
| get_model_state() (elliot.recommender.alge- | method), 52 |
| | Igdelpositions() (elliot.recommender.latent_fac- |
| method), 43 | tor_models.BPRMF_batch.BPRMF_batch_model.BPRN |
| get_model_state() (elliot.recommender.con- | method), 61 |
| | ritotSimilations() (elliot.recommender.latent_fac- |
| method), 47 | tor_models.CML.CML_model.CML_model |
| get_model_state() (elliot.recommender.knowl- | method), 62 |
| edge_aware.kaHFM.ka_hfm.MF | get_positions() (elliot.recommender.latent_fac- |
| method), 56 | tor_models.FISM.FISM_model.FISM_model |
| get_model_state() (elliot.recommender.latent | method), 64 |
| factor models RPRMERPRMEME | get_nositions() (elliot recommender neural - |

| NAIS.nais_model.NAIS_model | get_recommendations() (elliot.recommender.la- |
|---|---|
| method), 79 | tent_factor_models.BPRSlim.bprslim.BPRSlim |
| get_profiles() (elliot.recommender.con- | method), 61 |
| tent_based.VSM.tfidf_utils.TFIDF | get_recommendations() (elliot.recommender.la- |
| method), 47 | tent_factor_models.CML.CML.CML |
| get_profiles() (elliot.recommender.knowl- | method), 62 |
| edge_aware.kaHFM.tfidf_utils.TFIDF | get_recommendations() (elliot.recommender.la- |
| method), 57 | tent_factor_models.FFM.field_aware_factorization_ma |
| get_profiles() (elliot.recommender.knowl- | method), 63 |
| _ | get_recommendations() (elliot.recommender.la- |
| method), 58 | tent_factor_models.FISM.FISM.FISM |
| get_profiles() (elliot.recommender.knowl- | method), 63 |
| | Frecommendations() (elliot.recommender.la- |
| method), 59 | tent_factor_models.FM.factorization_machine.FM |
| get_profiles() (elliot.recommender.NN.at- | method), 64 |
| tribute_user_knn.tfidf_utils.TFIDF | get_recommendations() (elliot.recommender.la- |
| method), 35 | tent_factor_models.FunkSVD.funk_svd.FunkSVD |
| get_recommendations() (elliot.recom- | method), 65 |
| mender.adversarial.AMF.AMF.AMF | get_recommendations() (elliot.recommender.la- |
| method), 38 | tent_factor_models.LogisticMF.logistic_matrix_factoriz |
| get_recommendations() (elliot.recom- | method), 66 |
| mender.adversarial.AMR.AMR.AMR | get_recommendations() (elliot.recommender.la- |
| method), 41 | tent_factor_models.MF.matrix_factorization.MF |
| get_recommendations() (elliot.recommender.al- | method), 66 |
| gebric.slope_one.slope_one.SlopeOne | get_recommendations() (elliot.recommender.la- |
| method), 43 | tent_factor_models.NonNegMF.non_negative_matrix_ |
| get_recommendations() (elliot.recom- | method), 67 |
| | oget <u>n</u> ender/Medications() (elliot.recommender.la- |
| method), 89 | tent_factor_models.PMF.probabilistic_matrix_factoriza |
| get_recommendations() (elliot.recommender | method), 68 |
| method), 47 | Mget_recommendations() (elliot.recommender.la-
tent_factor_models.PureSVD.pure_svd.PureSVD |
| get_recommendations() (elliot.recommender | method), 69 |
| gan.CFGAN.cfgan.CFGAN method), | get_recommendations() (elliot.recommender.la- |
| 48 | tent_factor_models.Slim.slim |
| get_recommendations() (elliot.recommender | method), 70 |
| gan.IRGAN.irgan.IRGAN method), 50 | get_recommendations() (elliot.recommender.la- |
| get_recommendations() (elliot.recommender | tent_factor_models.SVDpp.svdpp.SVDpp |
| graph_based.lightgcn.LightGCN.LightGCN | method), 70 |
| method), 54 | get_recommendations() (elliot.recommender.la- |
| get_recommendations() (elliot.recommender | tent_factor_models.WRMF.wrmf.WRMF |
| graph_based.ngcf.NGCF.NGCF | method), 71 |
| method), 55 | get_recommendations() (elliot.recom- |
| get_recommendations() (elliot.recommender | mender.neural.ConvMF.convolutional_matrix_factoriz |
| knowledge_aware.kaHFM.ka_hfm.KaHFM | method), 72 |
| method), 56 | get_recommendations() (elliot.recom- |
| get_recommendations() (elliot.recommender | mender.neural.ConvNeuMF.convolutional_neural_mat |
| knowledge_aware.kaHFM_batch.kahfm_batch | |
| method), 57 | get_recommendations() (elliot.recom- |
| get_recommendations() (elliot.recommender | mender.neural.DeepFM.deep_fm.DeepFM |
| knowledge_aware.kahfm_embeddings.kahfm | |
| method), 58 | get_recommendations() (elliot.recom- |
| get_recommendations() (elliot.recommender.la- | mender.neural.DMF.deep_matrix_factorization.DMF |
| tent_factor_models.BPRMF.BPRMF.BPRMF | method), 75 |
| method), 59 | get_recommendations() (elliot.recom- |
| get_recommendations() (elliot.recommender.la- | mender.neural.GeneralizedMF.generalized_matrix_fac |
| tent_factor_models.BPRMF_batch.BPRMF_b | |
| method), 60 | get_recommendations() (elliot.recom- |
| ,, ~~ | 0 · - · · · · · · · · · · · · · · · · · |

| mender.neural.ItemAutoRec.itemautorec.Iter | |
|---|--|
| method), 77 | get_recs() (elliot.recommender.latent_factor |
| get_recommendations() (elliot.recom- | models.FunkSVD.funk_svd_model.FunkSVDModel |
| mender.neural.NAIS.nais.NAIS | method), 65 |
| method), 78 | get_recs() (elliot.recommender.latent_factor |
| get_recommendations() (elliot.recom- | $models. MF. matrix_factorization_model. MatrixFactorization_model. Matrix$ |
| mender.neural.NeuMF.neural_matrix_factori | zation.NemMetFiod), 67 |
| method), 80 | get_recs() (elliot.recommender.latent_factor |
| get_recommendations() (elliot.recom- | models.PMF.probabilistic_matrix_factorization_model. |
| mender.neural.NFM.neural_fm.NFM | method), 68 |
| method), 79 | get_recs() (elliot.recommender.latent_factor |
| get_recommendations() (elliot.recom- | models.SVDpp.svdpp_model.SVDppModel |
| mender.neural.NPR.neural_personalized_rar | |
| method), 80 | get_recs() (elliot.recommender.neural.ConvMF |
| get_recommendations() (elliot.recom- | convolutional_matrix_factorization_model.ConvMatrix |
| mender.neural.WideAndDeep.wide_and_de | |
| method), 82 | get_recs() (elliot.recommender.neural.Con- |
| get_recommendations() (elliot.recommender.N- | vNeuMF.convolutional_neural_matrix_factorization_m |
| N.attribute_item_knn.attribute_item_knn.At | |
| method), 34 | get_recs() (elliot.recommender.neu- |
| get_recommendations() (elliot.recommender.N- | ral.DeepFM.deep_fm_model.DeepFMModel |
| N.attribute_user_knn.attribute_user_knn.Att | |
| method), 35 | get_recs() (elliot.recommender.neural.DM- |
| get_recommendations() (elliot.recommender.N- | F.deep_matrix_factorization_model.DeepMatrixFactori |
| N.item_knn.item_knn.ItemKNN | method), 75 |
| | |
| method), 36 | get_recs() (elliot.recommender.neural.General- |
| get_recommendations() (elliot.recommender.N- | izedMF.generalized_matrix_factorization_model.General |
| N.user_knn.user_knn.UserKNN | method), 77 |
| method), 38 | get_recs() (elliot.recommender.neural.ItemAu- |
| get_recommendations() (elliot.recom- | toRec.itemautorec_model.ItemAutoRecModel |
| mender.recommender_utils_mixin.RecMixin | |
| method), 89 | get_recs() (elliot.recommender.neural.NeuM- |
| get_recommendations() (elliot.recom- | F.neural_matrix_factorization_model.NeuralMatrixFac |
| mender.unpersonalized.most_popular.most_ | |
| method), 84 | get_recs() (elliot.recommender.neu- |
| get_recommendations() (elliot.recom- | ral.NFM.neural_fm_model.NeuralFactorizationMachin |
| mender.unpersonalized.random_recommend | |
| method), 85 | get_recs() (elliot.recommender.neural.NPR.neu- |
| get_recommendations() (elliot.recommender.vi- | ral_personalized_ranking_model.NPRModel |
| $sual_recommenders. Deep Style. Deep Styl$ | |
| method), 87 | get_recs() (elliot.recommender.visual_recom- |
| get_recommendations() (elliot.recommender.vi- | menders.VNPR.visual_neural_personalized_ranking_r |
| sual_recommenders.DVBPR.DVBPR.DVBPR | method), 88 |
| method), 86 | get_rel() (elliot.evaluation.relevance.rele- |
| get_recommendations() (elliot.recommender.vi- | vance.AbstractRelevanceSingleton |
| sual_recommenders.VBPR.VBPR | method), 30 |
| method), 87 | get_rel() (elliot.evaluation.relevance.rele- |
| get_recommendations() (elliot.recommender.vi- | vance.BinaryRelevance method), 30 |
| sual_recommenders.VNPR.visual_neural_pe | |
| method), 88 | vance.DiscountedRelevance method), |
| get_recs() (elliot.recommender.atten- | 31 |
| tive.afm.afm_model.AFMModel | get_results() (elliot.recommender.base_recom- |
| method), 44 | mender_model.BaseRecommenderModel |
| get_recs() (elliot.recommender.latent_factor | method), 89 |
| | iget_neodlthfi(eldiAntareonatorinaleicueManhineModel |
| method), 63 | mender_utils_mixin.RecMixin |
| get_recs() (elliot.recommender.latent_factor | method), 89 |
| models.FM.factorization_machine_model.Fac | |
| inoacion macionization macinic illoaci.Ta | -gazarammonarammonumigrourammona uruurun uruurulli.p∪puluulli |

| | ty_utils.popularity.Popularity method), 30 | get_top_ | _k() (elliot.recommender.latent_factor
models.FFM.field_aware_factorization_machine_mode |
|-----------------|---|----------------|---|
| get_sort | ed_pop_items() (elliot.evaluation.popu- | | method), 63 |
| | larity_utils.popularity.Popularity | get_top_ | _k() (elliot.recommender.latent_factor |
| | method), 30 | | models.FISM.FISM_model |
| get_spai | rse() (elliot.recommender.neural.Wide- | | method), 64 |
| | AndDeep.wide_and_deep_model.WideAndl | Doctor | |
| | method), 83 | | models.FM.factorization_machine_model.Factorization |
| get_test(| () (elliot.dataset.abstract_dataset.Ab- | | method), 65 |
| | stractDataset method), 9 | get_top_ | _k() (elliot.recommender.latent_factor |
| get_test(| () (elliot.dataset.dataloader.knowl- | _ | models.FunkSVD.funk_svd_model.FunkSVDModel |
| | edge_aware_chains.KnowledgeChainsDataO | | method), 65 |
| | method), 5 | get_top_ | _k() (elliot.recommender.latent_factor |
| get_test(| () (elliot.dataset.dataloader.visual_dat- | | models.LogisticMF.logistic_matrix_factorization_mode |
| | aloader.VisualDataObject method), 6 | | method), 66 |
| get_test(| () (elliot.dataset.dataset.DataSet | get_top_ | _k() (elliot.recommender.latent_factor |
| 1 (1) | method), 9 | | models.MF.matrix_factorization_model.MatrixFactoriz |
| get_test(| () (elliot.evaluation.relevance.rele- | (. (| method), 67 |
| (| vance.Relevance method), 31 | get_top_ | _k() (elliot.recommender.latent_factor |
| get_top_ | _k() (elliot.recommender.adversari- | | models.PMF.probabilistic_matrix_factorization_model. |
| | al.AMF.AMF_model | | method), 69 |
| | method), 39 | get_top_ | _k() (elliot.recommender.latent_factor |
| get_top_ | _k() (elliot.recommender.adversari- | | models.SVDpp.svdpp_model.SVDppModel |
| | al.AMR.AMR_model.AMR_model | (. (| method), 70 |
| (| method), 41 | get_top_ | _k() (elliot.recommender.neural.Con- |
| get_top_ | _k() (elliot.recommender.atten- | | vMF.convolutional_matrix_factorization_model.ConvN |
| | tive.afm.afm_model.AFMModel | (. (| method), 72 |
| ant tous | method), 44 | get_top_ | _k() (elliot.recommender.neural.Con- |
| get_top_ | _k() (elliot.recommender.autoencoders | | vNeuMF.convolutional_neural_matrix_factorization_m |
| | dae.multi_dae_model.DenoisingAutoEncode | | method), 74 |
| ~~! !~ ~ | method), 45 | get_top_ | _k() (elliot.recommender.neu- |
| get_top_ | _k() (elliot.recommender.autoencoders | O#4 | ral.DeepFM.deep_fm_model.DeepFMModel |
| | vae.multi_vae_model.VariationalAutoEncode | | method), 76 |
| got top | method), 46 | get_top_ | _k() (elliot.recommender.neural.DM-
F.deep_matrix_factorization_model.DeepMatrixFactori |
| get_top_ | _k() (elliot.recommender.gan.CF-
GAN.cfgan_model.CFGAN_model | | method), 75 |
| | method), 48 | got top | |
| got top | _k() (elliot.recommender.gan.IRGAN.ir- | get_top_ | _k() (elliot.recommender.neural.Gener-
alizedMF.generalized_matrix_factorization_model.Gen |
| get_top_ | gan_model.IRGAN_model method), | | method), 77 |
| | 52 | get ton | _k() (elliot.recommender.neu- |
| get ton | _k() (elliot.recommender.graph_based | gci_top_ | ral.ItemAutoRec.itemautorec_model.ItemAutoRecMode |
| get_top_ | lightgcn.LightGCN_model.LightGCNModel | | method), 78 |
| | method), 54 | | _k() (elliot.recommender.neural.NAIS |
| get top | _k() (elliot.recommender | Sci_top_ | nais_model.NAIS_model method), 79 |
| Sci_top_ | graph_based.ngcf.NGCF_model.NGCFMode | bet ton | |
| | method), 55 | Sec_top- | F.neural_matrix_factorization_model.NeuralMatrixFact |
| get top | _k() (elliot.recommender.knowl- | | method), 81 |
| 9r-F- | edge_aware.kaHFM_batch.kahfm_batch_mo | obeelt Khan bl | |
| | method), 57 | O1- | ral.NFM.neural_fm_model.NeuralFactorizationMachin |
| get top | _k() (elliot.recommender.knowl- | | method), 79 |
| 0 - 1- | edge_aware.kahfm_embeddings.kahfm_emb | edditors | |
| | method), 58 | 0 – 0 | PR.neural_personalized_ranking_model.NPRModel |
| get_top | _k() (elliot.recommender.latent_factor | | method), 80 |
| 5 - 1- | models.BPRMF_batch.BPRMF_batch_model | BPRME | |
| | method), 61 | J - 1- | AutoRec.userautorec_model.UserAutoRecModel |
| get_top | _k() (elliot.recommender.latent_factor | | method), 82 |
| J - 1- | models.CML.CML_model.CML_model | get_top_ | _k() (elliot.recommender.neural.Wide- |
| | method), 62 | J = 1- | AndDeep.wide_and_deep_model.WideAndDeepMode |
| | · · · · · · · · · · · · · · · · · · · | | |

| method), 83 | get_user_recs() (elliot.recommender.algebric.s- |
|---|--|
| get_top_k() (elliot.recommender.visual_recom- | lope_one.slope_one_model.SlopeOneModel |
| menders.ACF.ACF_model.ACF_model | method), 43 |
| method), 85 | get_user_recs() (elliot.recommender.con- |
| get_top_k() (elliot.recommender.visual_recom- | tent_based.VSM.vector_space_model_similarity.Similar |
| menders.DeepStyle.DeepStyle_model.DeepS | |
| method), 87 | get_user_recs() (elliot.recommender.knowl- |
| get_top_k() (elliot.recommender.visual_recom- | edge_aware.kaHFM.ka_hfm.MF |
| menders.DVBPR.DVBPR_model.DVBPR_mo | |
| method), 86 | get_user_recs() (elliot.recommender.latent_fac- |
| get_top_k() (elliot.recommender.visual_recom- | tor_models.BPRMF.BPRMF.MF |
| menders.VBPR.VBPR_model.VBPR_model | method), 59 |
| method), 88 | get_user_recs() (elliot.recommender.latent_fac- |
| get_top_k() (elliot.recommender.visual_recom- | tor_models.BPRSlim.bprslim_model.BPRSlimModel |
| menders.VNPR.visual_neural_personalized_ | |
| method), 88 | get_user_recs() (elliot.recommender.latent_fac- |
| get_train_mask() (elliot.recommender.recom- | tor_models.NonNegMF.non_negative_matrix_factoriza |
| mender_utils_mixin.RecMixin | method), 68 |
| method), 89 | get_user_recs() (elliot.recommender.latent_fac- |
| get_transactions() (elliot.recommender.con- | tor_models.PureSVD.pure_svd_model.PureSVDModel |
| tent_based.VSM.vector_space_model_simila | |
| method), 47 | get_user_recs() (elliot.recommender.latent_fac- |
| get_transactions() (elliot.recommender.knowl- | tor_models.Slim.slim_model.SlimModel |
| edge_aware.kaHFM.ka_hfm.MF | method), 71 |
| method), 56 | get_user_recs() (elliot.recommender.latent_fac- |
| get_transactions() (elliot.recommender.latent | tor_models.WRMF.wrmf_model.WRMFModel |
| factor_models.BPRMF.BPRMF.MF | method), 71 |
| method), 59 | get_user_recs() (elliot.recommender.neu- |
| get_transactions() (elliot.recommender.NN.at- | ral.WideAndDeep.wide_and_deep_model.WideAndDe |
| tribute_item_knn.attribute_item_knn_simila | |
| method), 34 | get_user_recs() (elliot.recommender.NN.at- |
| get_transactions() (elliot.recommender.NN.at- | tribute_item_knn.attribute_item_knn_similarity.Similar |
| tribute_user_knn.attribute_user_knn_simila | |
| method), 35 | get_user_recs() (elliot.recommender.NN.at- |
| get_transactions() (elliot.recommender.N- | tribute_user_knn.attribute_user_knn_similarity.Similar |
| N.item_knn.item_knn_similarity.Similarity | method), 35 |
| method), 37 | get_user_recs() (elliot.recommender.N- |
| get_transactions() (elliot.recommender.N- | N.item_knn.aiolli_ferrari.AiolliSimilarity |
| N.user_knn.user_knn_similarity.Similarity | method), 36 |
| method), 38 | get_user_recs() (elliot.recommender.N- |
| get_user_bias() (elliot.recommender.knowl- | N.item_knn.item_knn_similarity.Similarity |
| edge_aware.kaHFM.ka_hfm.MF | method), 37 |
| method), 56 | get_user_recs() (elliot.recommender.N- |
| get_user_bias() (elliot.recommender.latent_fac- | N.user_knn.aiolli_ferrari.AiolliSimilarity |
| tor_models.BPRMF.BPRMF.MF | method), 37 |
| method), 59 | get_user_recs() (elliot.recommender.N- |
| get_user_factors() (elliot.recommender.knowl- | N.user_knn.user_knn_similarity.Similarity |
| edge_aware.kaHFM.ka_hfm.MF | method), 38 |
| method), 56 | get_user_recs_argpartition() (elliot.recom- |
| get_user_factors() (elliot.recommender.latent | mender.knowledge_aware.kaHFM.ka_hfm.MF |
| factor_models.BPRMF.BPRMF.MF | method), 56 |
| method), 59 | get_user_recs_argpartition() (elliot.recom- |
| get_user_neighbors() (elliot.recommender.N- | mender.latent_factor_models.BPRMF.BPRMF.MF |
| N.attribute_user_knn.attribute_user_knn_sii | |
| method), 35 | get_user_rel() (elliot.evaluation.relevance.rele- |
| get_user_neighbors() (elliot.recommender.N- | vance.BinaryRelevance method), 30 |
| N.user_knn.user_knn_similarity.Similarity | get_user_rel() (elliot.evaluation.relevance.rele- |
| method), 38 | vance.DiscountedRelevance method), |

| 31 | initialize() (elliot.recommender.NN.item_kn- |
|--|--|
| get_user_rel_gains() (elliot.evaluation.rele- | n.item_knn_similarity.Similarity |
| vance.relevance.BinaryRelevance | method), 37 |
| method), 31 | initialize() (elliot.recommender.NN.user_kn- |
| get_user_rel_gains() (elliot.evaluation.rele- | n.aiolli_ferrari.AiolliSimilarity |
| vance.relevance.DiscountedRelevance | method), 37 |
| method), 31 | initialize() (elliot.recommender.NN.user_kn- |
| get_validation() (elliot.dataset.dataloader | n.user_knn_similarity.Similarity |
| knowledge_aware_chains.KnowledgeChains | |
| method), 5 | IRGAN (class in elliot.recommender.gan.IR- |
| get_validation() (elliot.dataset.dataloader.visu- | GAN.irgan), 50 |
| al_dataloader.VisualDataObject | IRGAN_model (class in elliot.recommender |
| method), 6 | gan.IRGAN.irgan_model), 52 |
| get_validation() (elliot.dataset.dataset.DataSet | ItemAutoRec (class in elliot.recommender.neu- |
| method), 9 | ral.ItemAutoRec.itemautorec), 77 |
| GiniIndex (class in elliot.evaluation.metrics.di- | ItemAutoRecModel (class in elliot.recom- |
| versity.gini_index.gini_index), 20 | mender.neural.ItemAutoRec.itemautorec_model), |
| GMF (class in elliot.recommender.neural.Gen- | 77 |
| · · | ItemCoverage (class in elliot.evaluation.metric- |
| 76 | s.coverage.item_coverage.item_coverage), |
| | 18 |
| H | ItemKNN (class in elliot.recommender.N- |
| handle_hierarchy() (elliot.splitter.base_split- | N.item_knn.item_knn), 36 |
| ter.Splitter method), 91 | ItemMADranking (class in elliot.evaluation |
| HR (class in elliot.evaluation.metrics.accura- | metrics.fairness.MAD.ItemMADranking), |
| cy.hit_rate.hit_rate), 12 | 23 |
| HyperParameterStudy (class in elliot.re- | ItemMADrating (class in elliot.evaluation.met- |
| sult_handler.result_handler), 90 | rics.fairness.MAD.ItemMADrating), |
| | 23 |
| l | K |
| infer() (elliot.recommender.gan.CFGAN.cfgan | |
| model.Generator method), 50 | KaHFM (class in elliot.recommender.knowl- |
| init() (in module elliot.utils.logging), 92 | edge_aware.kaHFM.ka_hfm), 56 |
| init_charger() (in module elliot.recom- | KaHFM_model (class in elliot.recommender
knowledge_aware.kaHFM_batch.kahfm_batch_model) |
| mender.base_recommender_model), | 57 |
| 89 | KaHFMBatch (class in elliot.recommender |
| initialize() (elliot.recommender.algebric.s- | knowledge_aware.kaHFM_batch.kahfm_batch), |
| lope_one.slope_one_model.SlopeOneModel | 57 |
| method), 43 | KaHFMEmbeddings (class in elliot.recom- |
| initialize() (elliot.recommender.con- | rity.Similanty
58 |
| tent_based.VSM.vector_space_model_simila | rity.5imilarity |
| method), 47 | KaHFMEmbeddingsModel (class in elliot.rec- |
| initialize() (elliot.recommender.knowl- | ommender.knowledge_aware.kahfm_embeddings.kahf |
| edge_aware.kaHFM.ka_hfm.MF | 58 |
| method), 56 | KnowledgeChainsDataObject (class in elliot |
| initialize() (elliot.recommender.latent_factor
models.BPRMF.BPRMF.MF method), | dataset.dataloader.knowledge_aware_chains), |
| 60 | 5 |
| initialization (alliat secondary NINI at | KnowledgeChainsLoader (class in elliot |
| tribute item kan attribute item kan simila | rity.Similarity |
| method), 34 | 5 |
| initialize() (elliot.recommender.NN.at- | |
| tribute_user_knn.attribute_user_knn_similar | ily Similarity |
| method), 35 | LatentFactor (class in elliot.recommender.laten- |
| initialize() (elliot.recommender.NN.item_kn- | t_factor_models.CML.CML_model), |
| n.aiolli_ferrari.AiolliSimilarity | 62 |
| method), 36 | LatentFactor (class in elliot.recommender.laten- |
| | |

| t_factor_models.FISM.FISM_model), | MF (class in elliot.recommender.latent_factor
models.BPRMF.BPRMF), 59 |
|---|--|
| LatentFactor (class in elliot.recommender.neural.NAIS.nais_model), 78 | MF (class in elliot.recommender.latent_factor
models.MF.matrix_factorization), 66 |
| LAUC (class in elliot.evaluation.metrics.accura- | MLPComponent (class in elliot.recom- |
| cy.AUC.lauc), 11 | mender.neural.ConvMF.convolutional_matrix_factoriza |
| LightGCN (class in elliot.recommender | 73 |
| graph_based.lightgcn.LightGCN), 54 | MLPComponent (class in elliot.recom- |
| LightGCNModel (class in elliot.recommender | mender.neural.ConvNeuMF.convolutional_neural_mat |
| graph_based.lightgcn.LightGCN_model), | 74 |
| 54 | ModelCoordinator (class in elliot.hyperopti- |
| load_attribute_file() (elliot.dataset.dataloader | mization.model_coordinator), 32 |
| knowledge_aware_chains.KnowledgeChains | sImadets() (elliot.namespace.namespace_mod- |
| method), 5 | el_builder.Builder method), 33 |
| load_dataset_dataframe() (elliot.dataset.dat- | models() (elliot.namespace.namespace_mod- |
| aloader.knowledge_aware_chains.Knowledg | |
| method), 6 | method), 33 |
| load_dataset_dataframe() (elliot.dataset.dat- | module |
| aloader.visual_dataloader.VisualLoader | elliot, 93 |
| method), 6 | elliot.dataset, 9 |
| load_dataset_dict() (elliot.dataset.dataloader | elliot.dataset.abstract_dataset, 8 |
| knowledge_aware_chains.KnowledgeChains | |
| method), 6 load_feature_names() (elliot.dataset.dataload- | elliot.dataset.dataloader.knowledge_aware_chains, |
| | insLo ædiw t.dataset.dataloader.visual_dataloader, |
| method), 6 | 6 |
| load_item_set() (elliot.dataset.dataloader | elliot.dataset.dataset, 9 |
| knowledge_aware_chains.KnowledgeChains | |
| method), 6 | elliot.dataset.samplers.custom_pointwise_sparse_sampler, |
| load_obj() (in module elliot.utils.read), 92 | 7 |
| load_properties() (elliot.dataset.dataloader | elliot.dataset.samplers.custom_sampler, 7 |
| | Loadeliot.dataset.samplers.custom_sparse_sampler, |
| method), 6 | 7 |
| logarithmic_ranking_discount() (elliot.evalua- | elliot.dataset.samplers.pairwise_sampler, 7 |
| | inglet eh iot.dataset.samplers.pipeline_sampler, 7 |
| static method), 30 | elliot.dataset.samplers.pointwise_cfgan_sampler, |
| LogisticMatrixFactorization (class in elliot.rec- | 7 |
| · · · · · · · · · · · · · · · · · · · | .logis tid<u>i</u>ntataba<u>s</u>fatstamipdeis rpointwise_pos_neg_ratings_sampler |
| 66 | 8 |
| LogisticMatrixFactorizationModel (class in | elliot.dataset.samplers.pointwise_pos_neg_ratio_ratings_sa |
| 66 | gisticMF.logistfc_matrix_factorization_model),
elliot.dataset.samplers.pointwise_pos_neg_sampler, |
| 00 | emot.dataset.sampiers.pointwise_pos_neg_sampier, |
| M | elliot.dataset.samplers.pointwise_wide_and_deep_sampler, |
| | 8 |
| MAE (class in elliot.evaluation.metrics.rating | elliot.dataset.samplers.sparse_sampler, 8 |
| mae.mae), 27 | elliot.evaluation, 31 |
| manage_directories() (in module elliot.utils | elliot.evaluation.evaluator, 31 |
| folder), 91 MAR (class in alliest avaluation matrice acquire | elliot.evaluation.metrics, 30 |
| MAP (class in elliot.evaluation.metrics.accura-
cy.map.map), 12 | elliot.evaluation.metrics.accuracy, 15 |
| MAR (class in elliot.evaluation.metrics.accura- | elliot.evaluation.metrics.accuracy.AUC, 11 |
| cy.mar.mar), 13 | elliot.evaluation.metrics.accuracy.AUC.auc, |
| Matrix Easterization Model (class in alliet res | 9 |
| ommender.latent factor models.MF.matrix | factorization_model), |
| 67 | 10 |
| MF (class in elliot.recommender.knowl- | elliot.evaluation.metrics.accuracy.AUC.lauc, |
| edge_aware.kaHFM.ka_hfm), 56 | 10 |
| - <i>'</i> · | elliot.evaluation.metrics.accuracy.DSC, 11 |

| elliot.evaluation.metrics.accuracy.DSC.dsc, | elliot.evaluation.metrics.coverage.user_coverage.user_cover |
|--|--|
| elliot.evaluation.metrics.accuracy.f1, 12 elliot.evaluation.metrics.accuracy.f1.extended_f1, | elliot.evaluation.metrics.coverage.user_coverage.user_cover 19 |
| 11 | elliot.evaluation.metrics.diversity, 21 |
| elliot.evaluation.metrics.accuracy.f1.f1, 12 | elliot.evaluation.metrics.diversity.gini_index, |
| elliot.evaluation.metrics.accuracy.hit_rate, | 21 |
| 12 | elliot.evaluation.metrics.diversity.gini_index.gini_index, |
| elliot.evaluation.metrics.accuracy.hit_rate.hit_rate, | 20 |
| 12 | elliot.evaluation.metrics.diversity.shannon_entropy, |
| elliot.evaluation.metrics.accuracy.map, 13 | 21 |
| elliot.evaluation.metrics.accuracy.map, 19 | elliot.evaluation.metrics.diversity.shannon_entropy.shannon |
| 12 | 21 |
| | |
| elliot.evaluation.metrics.accuracy.mar, 13 | elliot.evaluation.metrics.diversity.SRecall, |
| elliot.evaluation.metrics.accuracy.mar.mar, | 20 |
| 13 | elliot.evaluation.metrics.diversity.SRecall.srecall, |
| elliot.evaluation.metrics.accuracy.mrr, 13 | 20 |
| elliot.evaluation.metrics.accuracy.mrr.mrr, | elliot.evaluation.metrics.fairness, 26 |
| 13 | elliot.evaluation.metrics.fairness.BiasDisparity, |
| elliot.evaluation.metrics.accuracy.ndcg, 14 | 23 |
| elliot.evaluation.metrics.accuracy.ndcg.ndcg, | elliot. evaluation. metrics. fairness. Bias Disparity. Bias Disparity and the property of th |
| 13 | 21 |
| elliot.evaluation.metrics.accuracy.precision, | elliot. evaluation. metrics. fairness. Bias Disparity. Bias Disparity and the property of th |
| 14 | 22 |
| | elliot.evaluation.metrics.fairness.BiasDisparity.BiasDisparity |
| elliot.evaluation.metrics.accuracy.recall, 15 | elliot.evaluation.metrics.fairness.MAD, 24 |
| elliot.evaluation.metrics.accuracy.recall.recall, | elliot.evaluation.metrics.fairness.MAD.ItemMADranking, |
| 15 | 23 |
| elliot.evaluation.metrics.base_metric, 29 elliot.evaluation.metrics.bias, 18 | elliot.evaluation.metrics.fairness.MAD.ItemMADrating, |
| elliot.evaluation.metrics.bias.aclt, 15 | elliot.evaluation.metrics.fairness.MAD.UserMADranking, |
| | 23 |
| elliot.evaluation.metrics.bias.aclt.aclt, 15 | |
| elliot.evaluation.metrics.bias.aplt, 16 | elliot.evaluation.metrics.fairness.MAD.UserMADrating, |
| elliot.evaluation.metrics.bias.aplt.aplt, 16 | 24 |
| elliot.evaluation.metrics.bias.arp, 16 | elliot.evaluation.metrics.fairness.reo, 25 |
| elliot.evaluation.metrics.bias.arp.arp, 16 | elliot.evaluation.metrics.fairness.reo.reo, 24 |
| elliot.evaluation.metrics.bias.pop_reo, 17 | elliot.evaluation.metrics.fairness.rsp, 26 |
| elliot.evaluation.metrics.bias.pop_reo.extended_pop_ | relijot.evaluation.metrics.fairness.rsp.rsp, 25 |
| 16 | elliot.evaluation.metrics.metrics_utils, 29 |
| elliot.evaluation.metrics.bias.pop_reo.pop_reo, | elliot.evaluation.metrics.novelty, 27 |
| 17 | elliot.evaluation.metrics.novelty.EFD, 26 |
| elliot.evaluation.metrics.bias.pop_rsp, 18 | elliot.evaluation.metrics.novelty.EFD.efd, |
| elliot.evaluation.metrics.bias.pop_rsp.extended_pop_ | |
| 17 | elliot.evaluation.metrics.novelty.EFD.extended_efd, |
| | 26 |
| elliot.evaluation.metrics.bias.pop_rsp.pop_rsp, 18 | |
| | elliot.evaluation.metrics.novelty.EPC, 27 |
| elliot.evaluation.metrics.coverage, 20 | elliot.evaluation.metrics.novelty.EPC.epc, |
| elliot.evaluation.metrics.coverage.item_coverage, | 27 |
| 19 | elliot.evaluation.metrics.novelty.EPC.extended_epc, |
| elliot.evaluation.metrics.coverage.item_coverage.item_ | · · |
| 18 | elliot.evaluation.metrics.rating, 29 |
| elliot.evaluation.metrics.coverage.num_retrieved, | elliot.evaluation.metrics.rating.mae, 28 |
| 19 | elliot.evaluation.metrics.rating.mae.mae, |
| elliot.evaluation.metrics.coverage.num_retrieved.num | |
| 19 | elliot.evaluation.metrics.rating.mse, 28 |
| elliot.evaluation.metrics.coverage.user_coverage, | elliot.evaluation.metrics.rating.mse.mse, 28 |
| 20 | elliot.evaluation.metrics.rating.rmse. 29 |

| alliet evaluation metrics rating rmse rmse | 47 |
|---|--|
| elliot.evaluation.metrics.rating.rmse.rmse, | elliot.recommender.content_based.VSM.vector_space_mod |
| elliot.evaluation.metrics.statistical_array_metric, | 47 |
| 29 | elliot.recommender.content_based.VSM.vector_space_mod |
| <u>-</u> | 47 |
| elliot.evaluation.popularity_utils, 30 | |
| elliot.evaluation.popularity_utils.popularity, | elliot.recommender.gan, 54 |
| 30 | elliot.recommender.gan.CFGAN, 50 |
| elliot.evaluation.relevance, 31 | elliot.recommender.gan.CFGAN.cfgan, 48 |
| elliot.evaluation.relevance.relevance, 30 | elliot.recommender.gan.CFGAN.cfgan_model, |
| elliot.evaluation.statistical_significance, 31 | 48 |
| elliot.hyperoptimization, 32 | elliot.recommender.gan.IRGAN, 54 |
| elliot.hyperoptimization.model_coordinator, | elliot.recommender.gan.IRGAN.irgan, 50 |
| 32 | elliot.recommender.gan.IRGAN.irgan_model, |
| elliot.namespace, 33 | 50 |
| elliot.namespace.namespace_model, 32 | elliot.recommender.graph_based, 56 |
| elliot.namespace.namespace_model_builder, | elliot.recommender.graph_based.lightgcn, |
| 32 | 55 |
| elliot.prefiltering, 33 | elliot.recommender.graph_based.lightgcn.LightGCN, |
| elliot.prefiltering.standard_prefilters, 33 | 54 |
| elliot.recommender, 90 | elliot.recommender.graph_based.lightgcn.LightGCN_mod |
| elliot.recommender.adversarial, 43 | 54 |
| elliot.recommender.adversarial.AMF, 41 | elliot.recommender.graph_based.ngcf, 55 |
| elliot.recommender.adversarial.AMF.AMF, | elliot.recommender.graph_based.ngcf.NGCF, |
| 38 | 55 |
| elliot.recommender.adversarial.AMF.AMF_model, | elliot.recommender.graph_based.ngcf.NGCF_model, 55 |
| elliot.recommender.adversarial.AMR, 43 | elliot.recommender.knowledge_aware, 59 |
| elliot.recommender.adversarial.AMR.AMR, | elliot.recommender.knowledge_aware.kaHFM, |
| 41 | 57 |
| elliot.recommender.adversarial.AMR.AMR_model, | elliot.recommender.knowledge_aware.kaHFM.ka_hfm, |
| | |
| elliot.recommender.algebric, 44 | elliot.recommender.knowledge_aware.kaHFM.tfidf_utils, 57 |
| elliot.recommender.algebric.slope_one, 43 | |
| elliot.recommender.algebric.slope_one.slope_one, | elliot.recommender.knowledge_aware.kaHFM_batch, |
| 43 | 58 |
| ÿ 1 1 | odelljot.recommender.knowledge_aware.kaHFM_batch.kahfr |
| 43 | 57 |
| elliot.recommender.attentive, 44 | elliot.recommender.knowledge_aware.kaHFM_batch.kahfr |
| elliot.recommender.attentive.afm, 44 | 57 |
| elliot.recommender.attentive.afm.afm, 44 | elliot.recommender.knowledge_aware.kaHFM_batch.tfidf_ |
| elliot.recommender.attentive.afm.afm_model, | 58 |
| 44 | elliot.recommender.knowledge_aware.kahfm_embeddings |
| elliot.recommender.autoencoders, 46 | 59 |
| elliot.recommender.autoencoders.dae, 45 | elliot.recommender.knowledge_aware.kahfm_embeddings |
| elliot.recommender.autoencoders.dae.multi_dae, | 58 |
| 44 | elliot.recommender.knowledge_aware.kahfm_embeddings |
| elliot.recommender.autoencoders.dae.multi_dae_mo | |
| 45 | elliot.recommender.knowledge_aware.kahfm_embeddings |
| elliot.recommender.autoencoders.vae, 46 | 59 |
| elliot.recommender.autoencoders.vae.multi_vae, | elliot.recommender.latent_factor_models, |
| 45 | 72 |
| elliot.recommender.autoencoders.vae.multi_vae_mod | dælliot.recommender.latent_factor_models.BPRMF, |
| 46 | 60 |
| elliot.recommender.base_recommender_model, | elliot.recommender.latent_factor_models.BPRMF.BPRMF, |
| 89 | 59 |
| elliot.recommender.content_based, 47 | elliot.recommender.latent_factor_models.BPRMF_batch, |
| elliot.recommender.content_based.VSM, 47 | 61 |
| elliot.recommender.content_based.VSM.tfidf_utils. | elliot.recommender.latent_factor_models.BPRMF_batch.BF |

```
elliot.recommender.latent_factor_models.BPRMF_batchllBPRMf_batchderdatehnt_factor_models.PMF.probabilistic_
elliot.recommender.latent_factor_models.BPRSlim,
                                                                                                                       elliot.recommender.latent_factor_models.PMF.probabilistic_
elliot.recommender.latent\_factor\_models.BPRS lim.bpr \textbf{\textit{elliot}}.recommender.latent\_factor\_models.PureSVD, and the limit of the latent\_factor\_models.PureSVD, and the latent\_factor\_m
elliot.recommender.latent_factor_models.BPRSlim.bprellint_neoderlnender.latent_factor_models.PureSVD.pure_sv
elliot.recommender.latent_factor_models.CML,
                                                                                                                       elliot.recommender.latent_factor_models.PureSVD.pure_sv
elliot.recommender.latent_factor_models.CML.CML, elliot.recommender.latent_factor_models.Slim,
elliot.recommender.latent_factor_models.CML_mblittlrecommender.latent_factor_models.Slim.slim,
elliot.recommender.latent_factor_models.FFM,
                                                                                                                       elliot.recommender.latent\_factor\_models.Slim.slim\_model,
elliot.recommender.latent_factor_models.FFM.field_avelliot_factorizationder.latent_factor_models.SVDpp,
elliot.recommender.latent_factor_models.FFM.field_avelliot_factorizationder.hatchin_factor_idehodels.SVDpp.svdpp,
elliot.recommender.latent_factor_models.FISM,
                                                                                                                       elliot.recommender.latent_factor_models.SVDpp.svdpp_mo
elliot.recommender.latent_factor_models.FISM.FISM, elliot.recommender.latent_factor_models.WRMF,
elliot.recommender.latent_factor_models.FISM.FISM_wrldidelrecommender.latent_factor_models.WRMF.wrmf,
elliot.recommender.latent_factor_models.FM,
                                                                                                                       elliot.recommender.latent_factor_models.WRMF.wrmf_models.
                                                                                                                                           71
elliot.recommender.latent_factor_models.FM.factorizatibiotmachine.ender.neural, 84
                                                                                                                       elliot.recommender.neural.ConvMF, 74
elliot.recommender.latent_factor_models.FM.factorizatibliotmachime.emdeteheural.ConvMF.convolutional_matrix_f
elliot.recommender.latent_factor_models.FunkSVD, elliot.recommender.neural.ConvMF.convolutional_matrix_f
elliot.recommender.latent_factor_models.FunkSVD.funkliot.decommender.neural.ConvNeuMF, 75
                                                                                                                       elliot.recommender.neural.ConvNeuMF.convolutional_neur
elliot.recommender.latent_factor_models.FunkSVD.funk_svd_m76del,
                                                                                                                       elliot.recommender.neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.ConvNeuMF.convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convolutional\_neural.Convo
elliot.recommender.latent_factor_models.LogisticMF,
                                                                                                                       elliot.recommender.neural.DeepFM, 76
elliot.recommender.latent_factor_models.LogisticMF.logistic_constrine_factorizationDeepFM.deep_fm,
elliot.recommender.latent_factor_models.LogisticMF.loglisticgcmatrine_factornieatialrDeapdfell.deep_fm_model,
elliot.recommender.latent_factor_models.MF,
                                                                                                                       elliot.recommender.neural.DMF, 75
                                                                                                                       elliot.recommender.neural.DMF.deep_matrix_factorization,
elliot.recommender.latent_factor_models.MF.matrix_factorization,
                                                                                                                       elliot.recommender.neural.DMF.deep_matrix_factorization_
elliot.recommender.latent_factor_models.MF.matrix_factorization_model,
                                                                                                                       elliot.recommender.neural.GeneralizedMF,
elliot.recommender.latent_factor_models.NonNegMF,
                                                                                                                       elliot.recommender.neural.GeneralizedMF.generalized_mat
elliot.recommender.latent_factor_models.NonNegMF.non_negative_matrix_factorization,
                                                                                                                       elliot.recommender.neural.GeneralizedMF.generalized_mat
elliot.recommender.latent_factor_models.NonNegMF.non_negaTive_matrix_factorization_model,
                                                                                                                       elliot.recommender.neural.ItemAutoRec,
elliot.recommender.latent_factor_models.PMF,
                                                                                                                                           78
```

```
elliot.recommender.neural.ItemAutoRec.itemautorec, elliot.recommender.NN.user_knn.aiolli_ferrari,
elliot.recommender.neural.ItemAutoRec.itemautorec_mbbidelrecommender.NN.user_knn.user_knn,
                                                                                   elliot.recommender.NN.user\_knn.user\_knn\_similarity,
elliot.recommender.neural.NAIS, 79
elliot.recommender.neural.NAIS.nais, 78
elliot.recommender.neural.NAIS.nais_model,
                                                                                   elliot.recommender.recommender_utils_mixin,
elliot.recommender.neural.NeuMF, 81
                                                                                   elliot.recommender.unpersonalized, 85
elliot.recommender.neural.NeuMF.neural_matrix_fact@rlivattionommender.unpersonalized.most_popular,
elliot.recommender.neural.NeuMF.neural_matrix_fact@flizattiono_montelder.unpersonalized.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_popular.most_pop
elliot.recommender.neural.NFM, 80
                                                                                   elliot.recommender.unpersonalized.random_recommender,
elliot.recommender.neural.NFM.neural fm,
                                                                                   elliot.recommender.unpersonalized.random_recommender.
elliot.recommender.neural.NFM.neural_fm_model,
                                                                                   elliot.recommender.visual recommenders,
elliot.recommender.neural.NPR, 80
elliot.recommender.neural.NPR.neural_personalized_relitionrecommender.visual_recommenders.ACF,
elliot.recommender.neural.NPR.neural_personalized_rellibtrrecommender.visual_recommenders.ACF.ACF,
elliot.recommender.neural.UserAutoRec,
                                                                                   elliot.recommender.visual_recommenders.ACF.ACF_model
elliot.recommender.neural.UserAutoRec.userautorec, elliot.recommender.visual_recommenders.DeepStyle,
elliot.recommender.neural.UserAutoRec.userautorec_mbithtlrecommender.visual_recommenders.DeepStyle.DeepS
elliot.recommender.neural.WideAndDeep,
                                                                                   elliot.recommender.visual_recommenders.DeepStyle.DeepS
elliot.recommender.neural.WideAndDeep.wide_and_delipt.recommender.visual_recommenders.DVBPR,
elliot.recommender.neural.WideAndDeep.wide_and_dddipt_neoderlmender.visual_recommenders.DVBPR.DVBPR,
elliot.recommender.NN, 38
                                                                                   elliot.recommender.visual_recommenders.DVBPR.DVBPR_
elliot.recommender.NN.attribute_item_knn,
                                                                                   elliot.recommender.visual_recommenders.DVBPR.FeatureE
elliot.recommender.NN.attribute\_item\_knn.attribute\_item\_knn.86
                                                                                   elliot.recommender.visual recommenders.VBPR,
elliot.recommender.NN.attribute_item_knn.attribute_item_knn8similarity,
                                                                                   elliot.recommender.visual_recommenders.VBPR.VBPR,
elliot.recommender.NN.attribute_user_knn,
                                                                                   elliot.recommender.visual_recommenders.VBPR.VBPR_mo
elliot.recommender.NN.attribute_user_knn.attribute_user_knn,87
                                                                                   elliot.recommender.visual_recommenders.VNPR,
elliot.recommender.NN.attribute_user_knn.attribute_user_knn_89milarity,
                                                                                   elliot.recommender.visual_recommenders.VNPR.visual_ne
elliot.recommender.NN.attribute_user_knn.tfidf_utils,
                                                                                   elliot.recommender.visual_recommenders.VNPR.visual_ne
elliot.recommender.NN.item knn, 37
elliot.recommender.NN.item_knn.aiolli_ferrari,
                                                                                   elliot.result_handler, 90
                                                                                   elliot.result_handler.result_handler, 90
elliot.recommender.NN.item_knn.item_knn,
                                                                                   elliot.run, 93
                                                                                   elliot.splitter, 91
elliot.recommender.NN.item_knn.item_knn_similarityelliot.splitter.base_splitter, 90
                                                                                   elliot.utils, 93
                                                                                   elliot.utils.folder, 91
elliot.recommender.NN.user_knn, 38
```

| elliot.utils.logger_util, 91 | name() (elliot.evaluation.metric- |
|--|--|
| elliot.utils.logging, 92 | s.bias.arp.arp.ARP static method), 16 |
| elliot.utils.read, 92 | name() (elliot.evaluation.metrics.bias.pop_re- |
| elliot.utils.write, 93 | o.extended_pop_reo.ExtendedPopREO |
| MostPop (class in elliot.recommender.unper- | static method), 17 |
| sonalized.most_popular.most_popular), | name() (elliot.evaluation.metrics.bias.pop_re- |
| MDD (aloos in alliet assalssation matrice account | o.pop_reo.PopREO static method), 17 |
| MRR (class in elliot.evaluation.metrics.accura- | name() (elliot.evaluation.metrics.bias.pop_r- |
| cy.mrr.mrr), 13 | sp.extended_pop_rsp.ExtendedPopRSP |
| MSE (class in elliot.evaluation.metrics.rat-
ing.mse.mse), 28 | static method), 18 name() (elliot.evaluation.metrics.bias.pop_r- |
| MultiDAE (class in elliot.recommender.autoen- | sp.pop_rsp.PopRSP static method), 18 |
| coders.dae.multi_dae), 44 | name() (elliot.evaluation.metrics.cover- |
| MultiVAE (class in elliot.recommender.autoen- | age.item_coverage.item_coverage.ItemCoverage |
| coders.vae.multi_vae), 45 | static method), 19 |
| coderon delinaria. (de) | name() (elliot.evaluation.metrics.cover- |
| N | age.num_retrieved.num_retrieved.NumRetrieved |
| | static method), 19 |
| NAIS (class in elliot.recommender.neural
NAIS.nais), 78 | name() (elliot.evaluation.metrics.cover- |
| NAIS_model (class in elliot.recommender.neu- | age.user_coverage.user_coverage.UserCoverage |
| ral.NAIS.nais_model), 78 | static method), 19 |
| name() (elliot.evaluation.metrics.accura- | name() (elliot.evaluation.metrics.cover- |
| cy.AUC.auc.AUC static method), 10 | $age.user_coverage.user_coverage_at_n.UserCoverageA$ |
| name() (elliot.evaluation.metrics.accura- | static method), 20 |
| cy.AUC.gauc.GAUC static method), 10 | name() (elliot.evaluation.metrics.diversi- |
| name() (elliot.evaluation.metrics.accura- | ty.gini_index.gini_index.GiniIndex |
| cy.AUC.lauc.LAUC static method), 11 | static method), 21 |
| name() (elliot.evaluation.metrics.accuracy.D- | name() (elliot.evaluation.metrics.diversity.shan- |
| SC.dsc.DSC static method), 11 | non_entropy.shannon_entropy.ShannonEntropy |
| name() (elliot.evaluation.metrics.accuracy.f1.ex- | static method), 21 |
| tended_f1.ExtendedF1 static method), | name() (elliot.evaluation.metrics.diversity.SRe- |
| 11 | call.srecall.SRecall static method), 20 |
| name() (elliot.evaluation.metrics.accura- | name() (elliot.evaluation.metrics.fairness.Bias- |
| cy.f1.f1.F1 static method), 12 | Disparity.BiasDisparityBD.BiasDisparityBD |
| name() (elliot.evaluation.metrics.accura- | method), 21
name() (elliot.evaluation.metrics.fairness.Bias- |
| cy.hit_rate.hit_rate.HR static method), | Disparity.BiasDisparityBR.BiasDisparityBR |
| 12 | method), 22 |
| name() (elliot.evaluation.metrics.accura- | name() (elliot.evaluation.metrics.fairness.Bias- |
| cy.map.map.MAP static method), 13 | Disparity.BiasDisparityBS.BiasDisparityBS |
| name() (elliot.evaluation.metrics.accuracy.mar | method), 23 |
| mar.MAR static method), 13 | name() (elliot.evaluation.metrics.fairness |
| name() (elliot.evaluation.metrics.accuracy.mr-
r.mrr.MRR static method), 13 | MAD.ItemMADranking.ItemMADranking |
| name() (elliot.evaluation.metrics.accuracy.nd- | method), 23 |
| cg.ndcg.NDCG static method), 14 | name() (elliot.evaluation.metrics.fairness |
| name() (elliot.evaluation.metrics.accuracy.preci- | MAD.ItemMADrating.ItemMADrating |
| sion.precision.Precision static | method), 23 |
| method), 14 | name() (elliot.evaluation.metrics.fairness |
| name() (elliot.evaluation.metrics.accuracy.recal- | MAD.UserMADranking.UserMADranking |
| l.recall.Recall static method), 15 | method), 24 |
| name() (elliot.evaluation.metrics.base_met- | name() (elliot.evaluation.metrics.fairness |
| ric.BaseMetric method), 29 | MAD.UserMADrating.UserMADrating |
| name() (elliot.evaluation.metrics.bias.aclt.a- | method), 24 |
| clt.ACLT static method), 15 | name() (elliot.evaluation.metrics.fairness.reo.re- |
| name() (elliot.evaluation.metric- | o.REO method), 25 |
| s.bias.aplt.aplt.APLT static method), | name() (elliot.evaluation.metrics.fairness.rsp.r- |
| 16 | sp.RSP method), 26 |
| | name() (elliot.evaluation.metrics.metrics_util- |

| s.ProxyMetric method), 29
name() (elliot.evaluation.metrics.metrics_util- | name() (elliot.recommender.latent_factor_mod-
els.BPRMF.BPRMF.BPRMF property), |
|--|---|
| s.ProxyStatisticalMetric method), 29 | 59 |
| name() (elliot.evaluation.metrics.novel- | name() (elliot.recommender.latent_factor_mod- |
| ty.EFD.efd.EFD static method), 26 | els.BPRMF.BPRMF.MF property), 60 |
| name() (elliot.evaluation.metrics.novel- | name() (elliot.recommender.latent_factor_mod- |
| ty.EFD.extended_efd.ExtendedEFD | els.BPRMF_batch.BPRMF_batch.BPRMF_batch |
| static method), 26 | property), 60 |
| name() (elliot.evaluation.metrics.novel- | name() (elliot.recommender.latent_factor_mod- |
| | |
| ty.EPC.epc.EPC static method), 27 | els.BPRSlim.bprslim.BPRSlim proper- |
| name() (elliot.evaluation.metrics.novel- | ty), 61 |
| ty.EPC.extended_epc.ExtendedEPC | name() (elliot.recommender.latent_factor_mod- |
| static method), 27 | els.CML.CML property), 62 |
| name() (elliot.evaluation.metrics.rating.mae | name() (elliot.recommender.latent_factor_mod- |
| mae.MAE static method), 28 | els.FFM.field_aware_factorization_machine.FFM |
| name() (elliot.evaluation.metrics.rat- | property), 63 |
| ing.mse.mse.MSE static method), 28 | name() (elliot.recommender.latent_factor_mod- |
| | |
| name() (elliot.evaluation.metrics.rat- | els.FISM.FISM property), 63 |
| ing.rmse.rmse.RMSE static method), | name() (elliot.recommender.latent_factor_mod- |
| 29 | els.FM.factorization_machine.FM |
| name() (elliot.recommender.adversarial.AM- | property), 64 |
| F.AMF.AMF property), 39 | name() (elliot.recommender.latent_factor_mod- |
| name() (elliot.recommender.adversarial.AM- | els.FunkSVD.funk_svd.FunkSVD |
| R.AMR.AMR property), 41 | property), 65 |
| name() (elliot.recommender.algebric.s- | name() (elliot.recommender.latent_factor_mod- |
| | |
| lope_one.slope_one.SlopeOne proper- | els.LogisticMF.logistic_matrix_factorization.LogisticMa |
| ty), 43 | property), 66 |
| name() (elliot.recommender.atten- | name() (elliot.recommender.latent_factor_mod- |
| tive.afm.afm.AFM property), 44 | els.MF.matrix_factorization.MF prop- |
| name() (elliot.recommender.autoencoders.dae | erty), 66 |
| multi_dae.MultiDAE property), 44 | name() (elliot.recommender.latent_factor_mod- |
| name() (elliot.recommender.autoencoders.vae | els.NonNegMF.non_negative_matrix_factorization.Nor |
| multi_vae.MultiVAE property), 45 | property), 67 |
| name() (elliot.recommender.content_based.VS- | name() (elliot.recommender.latent_factor_mod- |
| | |
| M.vector_space_model.VSM proper- | els.PMF.probabilistic_matrix_factorization.PMF |
| ty), 47 | property), 68 |
| name() (elliot.recommender.gan.CFGAN.cf- | name() (elliot.recommender.latent_factor_mod- |
| gan.CFGAN property), 48 | els.PureSVD.pure_svd.PureSVD prop- |
| name() (elliot.recommender.gan.IRGAN.ir- | erty), 69 |
| gan.IRGAN property), 50 | name() (elliot.recommender.latent_factor_mod- |
| name() (elliot.recommender.graph_based.light- | els.Slim.slim property), 70 |
| gcn.LightGCN.LightGCN property), | name() (elliot.recommender.latent_factor_mod- |
| 54 | els.SVDpp.svdpp.SVDpp property), 70 |
| name() (elliot.recommender | name() (elliot.recommender.latent_factor_mod- |
| | |
| graph_based.ngcf.NGCF.NGCF prop- | els.WRMF.wrmf.WRMF property), 71 |
| erty), 55 | name() (elliot.recommender.neural.ConvMF |
| name() (elliot.recommender.knowl- | $convolutional_matrix_factorization. ConvMF$ |
| edge_aware.kaHFM.ka_hfm.KaHFM | property), 72 |
| property), 56 | name() (elliot.recommender.neural.ConvNeuM- |
| name() (elliot.recommender.knowl- | F.convolutional_neural_matrix_factorization.ConvNeul |
| edge_aware.kaHFM.ka_hfm.MF prop- | property), 74 |
| erty), 56 | name() (elliot.recommender.neu- |
| name() (elliot.recommender.knowl- | |
| | ral.DeepFM.deep_fm.DeepFM proper- |
| edge_aware.kaHFM_batch.kahfm_batch.Ka | • |
| property), 57 | name() (elliot.recommender.neural.DMF.deep |
| name() (elliot.recommender.knowl- | matrix_factorization.DMF property), |
| edge_aware.kahfm_embeddings.kahfm_eml | č č |
| property), 58 | name() (elliot.recommender.neural.General- |

| <pre>izedMF.generalized_matrix_factorization.GN property), 76</pre> | MF tion.metrics.accuracy.AUC.auc.AUC static method), 10 |
|---|--|
| name() (elliot.recommender.neural.ItemAu- | needs_full_recommendations() (elliot.evalua- |
| toRec.itemautorec.ItemAutoRec prop- | tion.metrics.accuracy.AUC.gauc.GAUC |
| erty), 77 | static method), 10 |
| name() (elliot.recommender.neural.NAIS.nais | |
| | needs_full_recommendations() (elliot.evalua- |
| NAIS property), 78 | tion.metrics.base_metric.BaseMetric |
| name() (elliot.recommender.neural.NeuM- | static method), 29 |
| F.neural_matrix_factorization.NeuMF | needs_full_recommendations() (elliot.evalua- |
| property), 80 | tion.metrics.metrics_utils.ProxyMetric |
| name() (elliot.recommender.neural.NFM.neu- | method), 29 |
| ral_fm.NFM property), 79 | needs_full_recommendations() (elliot.evalua- |
| name() (elliot.recommender.neural.NPR.neu- | tion.metrics.metrics_utils.ProxyStatisticalMetric |
| ral_personalized_ranking.NPR prop- | method), 29 |
| erty), 80 | needs_full_recommendations() (elliot.evalua- |
| name() (elliot.recommender.neural.User- | tion.metrics.rating.mae.mae.MAE |
| AutoRec.userautorec.UserAutoRec | static method), 28 |
| property), 81 | needs_full_recommendations() (elliot.evalua- |
| name() (elliot.recommender.neural.WideAnd- | tion.metrics.rating.mse.mse.MSE static |
| Deep.wide_and_deep.WideAndDeep | method), 28 |
| property), 82 | needs_full_recommendations() (elliot.evalua- |
| name() (elliot.recommender.NN.at- | tion.metrics.rating.rmse.rmse.RMSE |
| tribute_item_knn.attribute_item_knn.Attribu | |
| property), 34 | NeuMF (class in elliot.recommender.neu- |
| name() (elliot.recommender.NN.at- | ral.NeuMF.neural_matrix_factorization), |
| tribute_user_knn.attribute_user_knn.Attribu | |
| property), 35 | NeuralFactorizationMachineModel (class in |
| name() (elliot.recommender.NN.item_kn- | elliot.recommender.neural.NFM.neural_fm_model), |
| n.item_knn.ItemKNN property), 36 | 79 |
| name() (elliot.recommender.NN.user_kn- | NeuralMatrixFactorizationModel (class in |
| n.user_knn.UserKNN property), 38 | elliot.recommender.neural.NeuMF.neural_matrix_facto |
| name() (elliot.recommender.unpersonalized | 81 |
| most_popular.most_popular.MostPop | NFM (class in elliot.recommender.neu- |
| property), 84 | ral.NFM.neural_fm), 79 |
| name() (elliot.recommender.unpersonal- | NGCF (class in elliot.recommender |
| ized.random_recommender.Random.Rando | 9 1 |
| property), 85 | NGCFModel (class in elliot.recommender |
| name() (elliot.recommender.visual_recom- | graph_based.ngcf.NGCF_model), 55 |
| menders.ACF.ACF.ACF property), 85 | NonNegMF (class in elliot.recommender.laten- |
| name() (elliot.recommender.visual_recom- | $t_factor_models. NonNegMF. non_negative_matrix_fact$ |
| menders.DeepStyle.DeepStyle.DeepStyle | 67 |
| property), 87 | NonNegMFModel (class in elliot.recom- |
| name() (elliot.recommender.visual_recom- | mender.latent_factor_models.NonNegMF.non_negative |
| menders.DVBPR.DVBPR.DVBPR | 68 |
| property), 86 | NPR (class in elliot.recommender.neural.N- |
| name() (elliot.recommender.visual_recom- | PR.neural_personalized_ranking), 80 |
| menders.VBPR.VBPR.VBPR proper- | NPRModel (class in elliot.recommender.neu- |
| ty), 87 | ral.NPR.neural_personalized_ranking_model), |
| name() (elliot.recommender.visual_recom- | 80 |
| menders.VNPR.visual_neural_personalized | _MukiRetViAVed (class in elliot.evaluation.metric- |
| property), 88 | s.coverage.num_retrieved.num_retrieved), |
| NameSpaceBuilder (class in elliot.namespace | 19 |
| namespace_model_builder), 33 | |
| NameSpaceModel (class in elliot.namespace | 0 |
| namespace_model), 32 | _ |
| NDCG (class in elliot.evaluation.metrics.accu- | objective() (elliot.hyperoptimization.model_co- |
| racy.ndcg.ndcg), 14 | ordinator.ModelCoordinator method), |
| needs_full_recommendations() (elliot.evalua- | 32 |

| | | 52 |
|---|-----------------|---|
| P | predict(| (elliot.recommender.graph_based |
| PairedTTest (class in elliot.evaluation.statisti- | | lightgcn.LightGCN_model.LightGCNModel |
| cal_significance), 31 | | method), 54 |
| PairedTTest (elliot.result_handler.result_han- | predict() | (elliot.recommender |
| dler.StatTest attribute), 90 | | graph_based.ngcf.NGCF_model.NGCFModel |
| parse_algorithms() (in module elliot.hyperopti- | 1 | method), 55 |
| mization), 32 | predict(| (elliot.recommender.knowl- |
| parse_metric() (in module elliot.evaluation.met- | | edge_aware.kaHFM.ka_hfm.KaHFM |
| rics), 30 | nradiat(| method), 56 |
| parse_metrics() (in module elliot.evaluation | predict |) (elliot.recommender.knowl-
edge_aware.kaHFM.ka_hfm.MF |
| metrics), 30 | | method), 56 |
| pipeline() (elliot.dataset.sampler- | predict(|) (elliot.recommender.knowl- |
| s.pipeline_sampler.Sampler method), | predict | edge_aware.kaHFM_batch.kahfm_batch_model.KaHFN |
| ninalina aval() (alliat dataset samplar | | method), 57 |
| pipeline_eval() (elliot.dataset.sampler-
s.pipeline_sampler.Sampler method), | predict(|) (elliot.recommender.knowl- |
| 5.pipeinie_sampier.sampier metriod), 7 | | edge_aware.kahfm_embeddings.kahfm_embeddings_r |
| PMF (class in elliot.recommender.latent_fac- | | method), 58 |
| tor_models.PMF.probabilistic_matrix_factori | predict(| (elliot.recommender.latent_factor |
| 68 | 20101011)) | models.BPRMF.BPRMF.BPRMF |
| PopREO (class in elliot.evaluation.metric- | | method), 59 |
| s.bias.pop_reo.pop_reo), 17 | predict() | (elliot.recommender.latent_factor |
| PopRSP (class in elliot.evaluation.metric- | | models.BPRMF.BPRMF.MF method), |
| s.bias.pop_rsp.pop_rsp), 18 | 1 | 60 |
| Popularity (class in elliot.evaluation.populari- | predict(| (elliot.recommender.latent_factor |
| ty_utils.popularity), 30 | | models.BPRMF_batch.BPRMF_batch_model.BPRMF_b |
| pre_train_discriminator() (elliot.recommender | prodict(| method), 61 |
| gan.IRGAN.irgan_model.IRGAN_model | predict | (elliot.recommender.latent_factor |
| method), 52 | | models.BPRSlim.bprslim.BPRSlim method), 61 |
| pre_train_generator() (elliot.recommender | predict(|) (elliot.recommender.latent_factor |
| gan.IRGAN.irgan_model.IRGAN_model | predict | models.BPRSlim.bprslim_model.BPRSlimModel |
| method), 52 | | method), 61 |
| Precision (class in elliot.evaluation.metrics.ac- | predict(|) (elliot.recommender.latent_factor |
| curacy.precision.precision), 14 predict() (elliot.recommender.adversarial.AM- | 1 | models.CML.CML_model |
| F.AMF_model.AMF_model method), | | method), 62 |
| 39 | predict(| (elliot.recommender.latent_factor |
| predict() (elliot.recommender.adversarial.AM- | | $models.FFM.field_aware_factorization_machine.FFM$ |
| R.AMR_model.AMR_model method), | | method), 63 |
| 41 | predict() |) (elliot.recommender.latent_factor |
| predict() (elliot.recommender.algebric.s- | | models.FFM.field_aware_factorization_machine_mode |
| lope_one.slope_one_model.SlopeOneModel | 1 | method), 63 |
| method), 43 | predict(| (elliot.recommender.latent_factor |
| predict() (elliot.recommender.atten- | | models.FISM_FISM_model |
| tive.afm.afm_model.AFMModel | mundiat(| method), 64 |
| method), 44 | predict |) (elliot.recommender.latent_factor
models.FM.factorization_machine.FM |
| predict() (elliot.recommender.autoencoders | | method), 64 |
| dae.multi_dae_model.DenoisingAutoEncode | er
predict() |) (elliot.recommender.latent_factor |
| method), 45 | predict | models.FM.factorization_machine_model.Factorization |
| predict() (elliot.recommender.autoencoders | | method), 65 |
| vae.multi_vae_model.VariationalAutoEncod
method), 46 | predict(|) (elliot.recommender.latent_factor |
| predict() (elliot.recommender.gan.CFGAN.cf- | | models.FunkSVD.funk_svd.FunkSVD |
| gan_model.CFGAN_model method), | | method), 65 |
| 48 | predict(| (elliot.recommender.latent_factor |
| predict() (elliot.recommender.gan.IRGAN.ir- | | $models. Funk SVD. funk_svd_model. Funk SVD Model$ |
| gan_model.IRGAN_model method), | | method), 65 |

| <pre>predict() (elliot.recommender.latent_factor</pre> | | method), 75 |
|--|----------------|---|
| models.LogisticMF.logistic_matrix_factorizat | iporedict() | ticllitattriecontonizationeural.General- |
| method), 66 | | izedMF.generalized_matrix_factorization.GMF |
| <pre>predict() (elliot.recommender.latent_factor</pre> | | method), 76 |
| models.MF.matrix_factorization.MF | predict() | (elliot.recommender.neural.ItemAu- |
| method), 66 | 1 | toRec.itemautorec_model.ItemAutoRecModel |
| <pre>predict() (elliot.recommender.latent_factor</pre> | | method), 78 |
| models.MF.matrix_factorization_model.Matr | ixaFeachtctr() | |
| method), 67 | Ι | nais_model.NAIS_model method), 79 |
| predict() (elliot.recommender.latent_factor | predict() | (elliot.recommender.neural.NeuM- |
| | | rF.NewNegNaFrix_factorization_model.NeuralMatrixFac |
| method), 67 | CTCTTEMTTO | method), 81 |
| predict() (elliot.recommender.latent_factor | predict() | (elliot.recommender.neural.NFM.neu- |
| models.NonNegMF.non_negative_matrix_fac | | |
| method), 68 | | (elliot.recommender.neural.NFM.neu- |
| predict() (elliot.recommender.latent_factor | F() | ral_fm_model.NeuralFactorizationMachineModel |
| models.PMF.probabilistic_matrix_factorization | on.PMF | |
| method), 68 | | (elliot.recommender.neural.NPR.neu- |
| predict() (elliot.recommender.latent_factor | predict() | ral_personalized_ranking_model.NPRModel |
| models.PMF.probabilistic_matrix_factorization | on mode | |
| method), 69 | | (elliot.recommender.neural.User- |
| predict() (elliot.recommender.latent_factor | predict() | AutoRec.userautorec_model.UserAutoRecModel |
| models.PureSVD.pure_svd.PureSVD | | method), 82 |
| method), 69 | nredict() | (elliot.recommender.neural.WideAnd- |
| predict() (elliot.recommender.latent_factor | predict() | Deep.wide_and_deep_model.WideAndDeepModel |
| models.PureSVD.pure_svd_model.PureSVD | Model | method), 83 |
| method), 69 | | (elliot.recommender.NN.item_kn- |
| predict() (elliot.recommender.latent_factor | predict() | n.aiolli_ferrari.AiolliSimilarity |
| models.Slim.slim.slim method), 70 | | method), 36 |
| predict() (elliot.recommender.latent_factor | prodict() | (elliot.recommender.NN.user_kn- |
| models.Slim.slim_model.SlimModel | predict() | n.aiolli_ferrari.AiolliSimilarity |
| method), 71 | | method), 37 |
| | prodict() | |
| predict() (elliot.recommender.latent_factor | predict() | (elliot.recommender.visual_recom- |
| models.SVDpp.svdpp.SVDpp | | menders.ACF.ACF_model.ACF_model |
| method), 70 | nuodiat() | method), 85 |
| predict() (elliot.recommender.latent_factor | predict() | (elliot.recommender.visual_recom- |
| models.SVDpp.svdpp_model.SVDppModel | | menders.DeepStyle_DeepStyle_model.DeepStyle_model |
| method), 70 | d: -t() | method), 87 |
| predict() (elliot.recommender.latent_factor | predict() | (elliot.recommender.visual_recom- |
| models.WRMF.wrmf.WRMF method), | | menders.VBPR.VBPR_model.VBPR_model |
| 71 | | method), 88 |
| predict() (elliot.recommender.latent_factor | predict() | (elliot.recommender.visual_recom- |
| models.WRMF.wrmf_model.WRMFModel | | menders.VNPR.visual_neural_personalized_ranking_r |
| method), 71 | 1 | method), 88 |
| predict() (elliot.recommender.neural.ConvMF | | all() (elliot.recommender.knowl- |
| | onvMatr | ix Haget_orivzatiohaNfffdM _batch.kahfm_batch_model.KaHF |
| method), 73 | 1 | method), 57 |
| predict() (elliot.recommender.neural.Con- | | batch() (elliot.recommender.knowl- |
| | rization_1 | neolge <u>l. Gware Neal HHM a</u> baitdfaktahrfza <u>t loatMo</u> deoldel. KaHF |
| method), 74 | | method), 58 |
| predict() (elliot.recommender.neu- | predict_ | batch() (elliot.recommender.knowl- |
| ral.DeepFM.deep_fm.DeepFM | | edge_aware.kahfm_embeddings.kahfm_embeddings_r |
| method), 76 | _ | method), 58 |
| predict() (elliot.recommender.neu- | predict_ | batch() (elliot.recommender.latent_fac- |
| ral.DeepFM.deep_fm_model.DeepFMModel | | tor_models.LogisticMF.logistic_matrix_factorization_m |
| method), 76 | _ | method), 66 |
| predict() (elliot.recommender.neural.DM- | | batch() (elliot.recommender.visual_rec- |
| F.deep_matrix_factorization_model.DeepMa | trixFacto | rizzantinoeriMerte:DVBPR.DVBPR_model.DVBPR_model |

| method), 86 | |
|--|---|
| PreFilter (class in elliot.prefiltering.standard- | R |
| _prefilters), 33 | Random (class in elliot.recommender.unper- |
| prepare_logger() (in module elliot.utils.log- | sonalized.random_recommender.Random), |
| ging), 92 | 85 |
| ProbabilisticMatrixFactorizationModel (class in | read_config() (in module elliot.utils.read), 92 |
| elliot.recommender.latent_tactor_models.PN | Tearo Dabilistic matrix finterizationally ofel), |
| 68 | read_folder() (elliot.splitter.base_splitter.S- |
| process() (elliot.evaluation.metrics.accura-
cy.f1.extended_f1.ExtendedF1 | plitter method), 91 |
| method), 11 | read_image() (elliot.dataset.sampler- |
| process() (elliot.evaluation.metrics.fairness.Bi- | s.pipeline_sampler.Sampler method), |
| asDisparity.BiasDisparityBD.BiasDisparityB | Pread_imagenet_classes_txt() (in module elliot.u- |
| method), 22 | tils.read), 92 |
| process() (elliot.evaluation.metrics.fairness.Bi- | read images() (elliot.dataset.dataloader.visu- |
| asDisparity.BiasDisparityBR.BiasDisparityBI | R al_dataloader.VisualDataObject |
| method), 22 | method), 6 |
| process() (elliot.evaluation.metrics.fairness.Bi-
asDisparity.BiasDisparityBS.BiasDisparityBS | read_images_multiprocessing() (elliot.dataset |
| method), 23 | |
| process() (elliot.evaluation.metrics.fairness.re- | method), 6 |
| o.reo.REO method), 25 | read_images_triple() (elliot.dataset.sampler-
s.pipeline_sampler.Sampler method), |
| process() (elliot.evaluation.metrics.fairness.r- | 7 |
| sp.rsp.RSP method), 26 | read_multi_config() (in module elliot.util- |
| process_cosine() (elliot.recommender.N- | s.read), 93 |
| N.item_knn.item_knn_similarity.Similarity | read_np() (in module elliot.utils.read), 93 |
| method), 37 process_similarity() (elliot.recommender.con- | read_single_image() (elliot.dataset.dataload- |
| tent_based.VSM.vector_space_model_simila | er.visual_dataloader.VisualDataObject |
| method), 47 | read_splitting() (elliot.dataset.dataloader |
| | knowledge aware chains Knowledge Chains Loader |
| N.attribute_item_knn.attribute_item_knn_si | knowledge_aware_chains.KnowledgeChainsLoader
milarity.Similarity, 6 |
| memoa), 54 | read splitting() (elliot.dataset.dataloader.visu- |
| process_similarity() (elliot.recommender.N- | al_dataloader.VisualLoader method),
milarity.Similarity |
| N.attribute_user_knn.attribute_user_knn_sii | |
| method), 35 process_similarity() (elliot.recommender.N- | read_splitting() (elliot.dataset.dataset.DataSet- |
| N.item_knn_similarity.Similarity | Loader method), 9 |
| method), 37 | rearrange_data() (elliot.splitter.base_splitter.S-
plitter method), 91 |
| process_similarity() (elliot.recommender.N- | Recall (class in elliot.evaluation.metrics.accura- |
| N.user_knn.user_knn_similarity.Similarity | cy.recall.recall), 15 |
| method), 38 | RecMixin (class in elliot.recommender.recom- |
| process_splitting() (elliot.splitter.base_split- | mender_utils_mixin), 89 |
| ter.Splitter method), 91
ProxyMetric (class in elliot.evaluation.metrics | reduce_attribute_map_property_selection() |
| metrics_utils), 29 | (elliot.dataset.dataloader.knowledge_aware_chains.Kno |
| ProxyStatisticalMetric (class in elliot.evalua- | method), 6 |
| tion.metrics_utils), 29 | reduce_dataset_by_item_list() (elliot.dataset
dataloader.knowledge_aware_chains.KnowledgeChain |
| PureSVD (class in elliot.recommender.latent | method), 6 |
| factor_models.PureSVD.pure_svd), 69 | reduce_dataset_by_item_list() (elliot.dataset |
| PureSVDModel (class in elliot.recommender.la- | dataloader.visual dataloader.VisualLoader |
| tent_factor_models.PureSVD.pure_svd_mod | del), method), 7 |
| 69 | Relevance (class in elliot.evaluation.rele- |
| Q | vance.relevance), 31 |
| | REO (class in elliot.evaluation.metrics.fair- |
| QueueListenerHandler (class in elliot.utils.log-
ger_util), 91 | ness.reo.reo), 25 required_attributes (elliot.dataset.abstract |
| gei_uiii), 🚈 | dataset Abstract Dataset attribute) 0 |

restore_weights() (elliot.recommender.alge-ConvNeuMF.convolutional_neural_matrix_factorization bric.slope_one.slope_one.SlopeOne method), 74 method), 43 restore_weights() (elliot.recommender.neu $ral.DeepFM.deep_fm.DeepFM$ restore_weights() (elliot.recommender.content_based.VSM.vector_space_model.VSM method), 76 method), 47 restore_weights() (elliot.recommender.neurestore_weights() (elliot.recommender.knowlral.NFM.neural_fm.NFM method), 79 edge aware.kaHFM.ka hfm.KaHFM restore weights() (elliot.recommender.NN.atmethod), 56 tribute_item_knn.attribute_item_knn.AttributeItemKN method), 34 restore_weights() (elliot.recommender.latent_factor models.BPRMF.BPRMF.BPRMF restore weights() (elliot.recommender.NN.attribute_user_knn.attribute_user_knn.AttributeUserKN method), 59 restore_weights() (elliot.recommender.latent_method), 35 factor_models.BPRMF_batch.BPRM method), 60 N.item knn.item knn.ItemKNN restore_weights() (elliot.recommender.latent_method), 36 factor_models.BPRSlim.bprslim.BPRSlim restore_weights() (elliot.recommender.N-N.user knn.user knn.UserKNN method), 61 restore_weights() (elliot.recommender.latent_method), 38 factor models.CML.CML.CML restore_weights() (elliot.recommender.recommender_utils_mixin.RecMixin method), 62 restore weights() (elliot.recommender.latent method), 89 factor_models.FFM.field_aware_factorizationResadblitendfed(class in elliot.result_handler.remethod), 63 sult_handler), 90 restore_weights() (elliot.recommender.latent_-RMSE (class in elliot.evaluation.metrics.ratfactor models.FISM.FISM.FISM ing.rmse.rmse), 28 method), 63 RSP (class in elliot.evaluation.metrics.fairness.rrestore_weights() (elliot.recommender.latent_sp.rsp), 25 factor_models.FM.factorization_machine.FM run_experiment() (in module elliot.run), 93 method), 64 restore_weights() (elliot.recommender.latent_factor_models.FunkSVD.funk_svd.FunkSVD Sampler (class in elliot.dataset.samplers.cusmethod), 65 tom_pointwise_sparse_sampler), 7 restore_weights() (elliot.recommender.latent_factor_models.LogisticMF.logistic_matrix_factorization_LogisticMatrix_Factorization method) 66 method), 66 Sampler (class in elliot.dataset.samplers.cusrestore_weights() (elliot.recommender.latent_tom_sparse_sampler), 7 $factor_models.MF.matrix_factorization.MF$ Sampler (class in elliot.dataset.samplers.pairmethod), 66 wise_sampler), 7 restore weights() (elliot.recommender.latent weights() (elliot.recommender.latent_- Sampler (class in elliot.dataset.sampler-factor_models.NonNegMF.non_negative_matrix_factorization_NonNegMF.7 method), 67 Sampler (class in elliot.dataset.samplers.pointrestore_weights() (elliot.recommender.latent_wise_cfgan_sampler), 8 method), 68 wise_pos_neg_ratings_sampler), 8 restore_weights() (elliot.recommender.latent_-Sampler (class in elliot.dataset.samplers.pointfactor_models.PureSVD.pure_svd.PureSVD wise_pos_neg_ratio_ratings_sampler), method), 69 restore_weights() (elliot.recommender.latent_-Sampler (class in elliot.dataset.samplers.pointfactor models.Slim.slim.Slim wise_pos_neg_sampler), 8 method), 71 Sampler (class in elliot.dataset.samplers.pointrestore_weights() (elliot.recommender.latent_wise_wide_and_deep_sampler), 8 $factor_models.WRMF.wrmf.WRMF$ Sampler (class in elliot.dataset.samplers.smethod), 71 parse_sampler), 8 ConvMF.convolutional_matrix_factorization.ConvMF restore_weights() (elliot.recommender.neural.method), 72 save_best_models() (elliot.result_handler.re-

Index 131

restore_weights() (elliot.recommender.neural.-

| sult_handler.ResultHandler method), | set_model_state() (elliot.recommender.latent
factor_models.NonNegMF.non_negative_matrix_factor |
|--|--|
| save_best_results() (elliot.result_handler.re- | method), 68 |
| sult_handler.ResultHandler method), | set_model_state() (elliot.recommender.latent |
| 90 | factor_models.PureSVD.pure_svd_model.PureSVDModels.PureSVD |
| | method), 69 |
| save_best_results_as_triplets() (elliot.re-
sult_handler.result_handler.ResultHandler | set_model_state() (elliot.recommender.latent |
| method), 90 | factor_models.Slim.slim_model.SlimModel |
| save_best_statistical_results() (elliot.result_han- | method), 71 |
| dler.result_handler.ResultHandler | set_model_state() (elliot.recommender.latent |
| method), 90 | factor_models.WRMF.wrmf_model.WRMFModel |
| save_np() (in module elliot.utils.write), 93 | method), 72 |
| save_np() (in module elliot.utils.write), 93 | set_model_state() (elliot.recommender.NN.at- |
| | tribute_item_knn.attribute_item_knn_similarity.Similar |
| save_trials() (elliot.result_handler.result_handler.HyperParameterStudy method) | method), 34 |
| dler.HyperParameterStudy method), | set_model_state() (elliot.recommender.NN.at- |
| save_trials_as_triplets() (elliot.result_han- | tribute_user_knn.attribute_user_knn_similarity.Similar |
| dler.result_handler.HyperParameterStudy | method), 35 |
| 7.1 | |
| method), 90 | set_model_state() (elliot.recommender.N- |
| score_item() (elliot.recommender.NN.at-
tribute_item_knn.attribute_item_knn_simila | N.item_knn.item_knn_similarity.Similarity |
| | |
| static method), 34 | set_model_state() (elliot.recommender.N- |
| score_item() (elliot.recommender.NN.at- | N.user_knn.user_knn_similarity.Similarity |
| tribute_user_knn.attribute_user_knn_simila | |
| static method), 35 | set_update_user() (elliot.recommender.latent |
| score_item() (elliot.recommender.NN.item_kn- | factor_models.LogisticMF.logistic_matrix_factorization |
| n.item_knn_similarity.Similarity static | method), 66 |
| method), 37 | set_user_bias() (elliot.recommender.knowl- |
| score_item() (elliot.recommender.NN.user_kn- | edge_aware.kaHFM.ka_hfm.MF |
| n.user_knn_similarity.Similarity static | method), 57 |
| method), 38 | set_user_bias() (elliot.recommender.latent_fac- |
| set_item_bias() (elliot.recommender.knowl- | tor_models.BPRMF.BPRMF.MF |
| edge_aware.kaHFM.ka_hfm.MF | method), 60 |
| method), 56 | set_user_factors() (elliot.recommender.knowl- |
| set_item_bias() (elliot.recommender.latent_fac- | edge_aware.kaHFM.ka_hfm.MF |
| tor_models.BPRMF.BPRMF.MF | method), 57 |
| method), 60 | set_user_factors() (elliot.recommender.latent |
| set_item_factors() (elliot.recommender.knowl- | factor_models.BPRMF.BPRMF.MF |
| edge_aware.kaHFM.ka_hfm.MF | method), 60 |
| method), 57 | ShannonEntropy (class in elliot.evaluation.met- |
| set_item_factors() (elliot.recommender.latent | rics.diversity.shannon_entropy.shannon_entropy), |
| factor_models.BPRMF.BPRMF.MF | 21 |
| method), 60 | Similarity (class in elliot.recommender.con- |
| set_model_state() (elliot.recommender.alge- | tent_based.VSM.vector_space_model_similarity), |
| bric.slope_one.slope_one_model.SlopeOneN | |
| method), 43 | Similarity (class in elliot.recommender.NN.at- |
| set_model_state() (elliot.recommender.con- | tribute_item_knn.attribute_item_knn_similarity), |
| tent_based.VSM.vector_space_model_simila | |
| method), 47 | Similarity (class in elliot.recommender.NN.at- |
| set_model_state() (elliot.recommender.knowl- | tribute_user_knn.attribute_user_knn_similarity), |
| edge_aware.kaHFM.ka_hfm.MF | 35 |
| method), 57 | Similarity (class in elliot.recommender.N- |
| set_model_state() (elliot.recommender.latent | N.item_knn.item_knn_similarity), 36 |
| factor_models.BPRMF.BPRMF.MF | Similarity (class in elliot.recommender.N- |
| method), 60 | N.user_knn.user_knn_similarity), 38 |
| set_model_state() (elliot.recommender.latent | single() (elliot.hyperoptimization.model_coor- |
| | SlimModedinator.ModelCoordinator method), |
| method), 61 | 32 |

Slim (class in elliot.recommender.latent_facwise_wide_and_deep_sampler.Sampler tor_models.Slim.slim), 70 method), 8 step() (elliot.dataset.samplers.sparse_sam-SlimModel (class in elliot.recommender.latent_factor_models.Slim.slim_model), 71 pler.Sampler method), 8 stop() (elliot.utils.logger_util.QueueListener-SlopeOne (class in elliot.recommender.algebric.slope_one.slope_one), 43 Handler method), 92 SlopeOneModel (class in elliot.recomstore_recommendation() (in module elliot.utilmender.algebric.slope_one.slope_one_model), s.write), 93 store_splitting() (elliot.splitter.base_splitter.Ssoftmax() (elliot.recommender.neural.NAIS.plitter method), 91 subsampling_leavenout_list_generator() nais_model.NAIS_model method), 79 Splitter (class in elliot.splitter.base_splitter), 90 (elliot.splitter.base_splitter.Splitter splitting_best_timestamp() (elliot.splitmethod), 91 ter.base_splitter.Splitter method), 91 subsampling_list_generator() (elliot.splitsplitting_kfolds() (elliot.splitter.base_splitter.Ster.base_splitter.Splitter method), 91 plitter method), 91 suggest() (in module elliot.hyperoptimization), splitting_passed_timestamp() (elliot.splitter.base_splitter.Splitter method), 91 SVDpp (class in elliot.recommender.latent_facsplitting_randomsubsampling_kfolds() (elliot.stor_models.SVDpp.svdpp), 70 plitter.base_splitter.Splitter method), SVDppModel (class in elliot.recommender.latent_factor_models.SVDpp.svdpp_model), splitting_randomsubsampling_kfolds_leavenout() (elliot.splitter.base_splitter.Splitter method), 91 splitting_temporal_holdout() (elliot.split-TFIDF (class in elliot.recommender.conter.base_splitter.Splitter method), 91 tent_based.VSM.tfidf_utils), 47 splitting_temporal_leavenout() (elliot.split-TFIDF (class in elliot.recommender.knowlter.base_splitter.Splitter method), 91 edge_aware.kaHFM.tfidf_utils), 57 SRecall (class in elliot.evaluation.metrics.diver-TFIDF (class in elliot.recommender.knowlsity.SRecall.srecall), 20 edge_aware.kaHFM_batch.tfidf_utils), start() (elliot.utils.logger_util.QueueListener-Handler method), 92 TFIDF (class in elliot.recommender.knowl-StatisticalMetric (class in elliot.evaluation.metedge_aware.kahfm_embeddings.tfidf_utils), rics.statistical_array_metric), 30 StatTest (class in elliot.result_handler.re-TFIDF (class in elliot.recommender.NN.atsult_handler), 90 tribute_user_knn.tfidf_utils), 35 step() (elliot.dataset.samplers.custom_pointtfidf() (elliot.recommender.content_based.VSwise_sparse_sampler.Sampler M.tfidf_utils.TFIDF method), 47 method), 7 tfidf() (elliot.recommender.knowlstep() (elliot.dataset.samplers.custom_sam $edge_aware.kaHFM.tfidf_utils.TFIDF$ pler.Sampler method), 7 method), 57 step() (elliot.dataset.samplers.custom_stfidf() (elliot.recommender.knowlparse_sampler.Sampler method), 7 $edge_aware.kaHFM_batch.tfidf_utils.TFIDF$ step() (elliot.dataset.samplers.pairwise_sammethod), 58 pler.Sampler method), 7 tfidf() (elliot.recommender.knowlstep() (elliot.dataset.samplers.pipeline_samedge_aware.kahfm_embeddings.tfidf_utils.TFIDF pler.Sampler method), 7 method), 59 step() (elliot.dataset.samplers.pointwise_cftfidf() (elliot.recommender.NN.atgan_sampler.Sampler method), 8 tribute_user_knn.tfidf_utils.TFIDF step() (elliot.dataset.samplers.pointwise_method), 35 pos_neg_ratings_sampler.Sampler TimeFilter (class in elliot.utils.logging), 92 method), 8 train() (elliot.recommender.adversarial.AMstep() (elliot.dataset.samplers.pointwise_-F.AMF.AMF method), 39 pos_neg_ratio_ratings_sampler.Sampler train() (elliot.recommender.adversarial.AMmethod), 8 R.AMR.AMR method), 41 step() (elliot.dataset.samplers.pointwise_train() (elliot.recommender.algebric.spos_neg_sampler.Sampler method), 8 lope_one.slope_one.SlopeOne step() (elliot.dataset.samplers.point-

| method), 43 | method), 67 |
|--|--|
| train() (elliot.recommender.atten- | train() (elliot.recommender.latent_factor_mod- |
| tive.afm.afm.AFM method), 44 | els.NonNegMF.non_negative_matrix_factorization.Nor |
| train() (elliot.recommender.autoencoders.dae | method), 67 |
| multi_dae.MultiDAE method), 44 | train() (elliot.recommender.latent_factor_mod- |
| train() (elliot.recommender.autoencoders.vae | els.PMF.probabilistic_matrix_factorization.PMF |
| multi_vae.MultiVAE method), 46 | method), 68 |
| train() (elliot.recommender.base_recom- | train() (elliot.recommender.latent_factor_mod- |
| mender_model.BaseRecommenderModel | els.PureSVD.pure_svd.PureSVD |
| method), 89 | method), 69 |
| train() (elliot.recommender.content_based.VS- | train() (elliot.recommender.latent_factor_mod- |
| | els.Slim.slim.slim method), 71 |
| M.vector_space_model.VSM method), | |
| train() (alliat recommendar can CECAN of | train() (elliot.recommender.latent_factor_mod- |
| train() (elliot.recommender.gan.CFGAN.cf- | els.Slim.slim_model.SlimModel |
| gan.CFGAN method), 48 | method), 71 |
| train() (elliot.recommender.gan.IRGAN.ir- | train() (elliot.recommender.latent_factor_mod- |
| gan.IRGAN method), 50 | els.SVDpp.svdpp.SVDpp method), 70 |
| train() (elliot.recommender.graph_based.light- | train() (elliot.recommender.latent_factor_mod- |
| gcn.LightGCN.LightGCN method), 54 | els.WRMF.wrmf.WRMF method), 71 |
| train() (elliot.recommender | train() (elliot.recommender.neural.ConvMF |
| graph_based.ngcf.NGCF.NGCF | convolutional_matrix_factorization.ConvMF |
| method), 55 | method), 72 |
| train() (elliot.recommender.knowl- | train() (elliot.recommender.neural.ConvNeuM- |
| edge_aware.kaHFM.ka_hfm.KaHFM | F.convolutional_neural_matrix_factorization.ConvNeul |
| method), 56 | method), 74 |
| train() (elliot.recommender.knowl- | train() (elliot.recommender.neu- |
| edge_aware.kaHFM_batch.kahfm_batch.Ka | HFMBatchral.DeepFM.deep_fm.DeepFM |
| method), 57 | method), 76 |
| train() (elliot.recommender.knowl- | train() (elliot.recommender.neural.DMF.deep |
| edge_aware.kahfm_embeddings.kahfm_eml | beddings.KnaHFM_Fandwedztitign.DMF method), 75 |
| method), 58 | train() (elliot.recommender.neural.General- |
| train() (elliot.recommender.latent_factor_mod- | izedMF.generalized_matrix_factorization.GMF |
| els.BPRMF.BPRMF.BPRMF method), | method), 76 |
| 59 | train() (elliot.recommender.neural.ItemAu- |
| train() (elliot.recommender.latent_factor_mod- | toRec.itemautorec.ItemAutoRec |
| els.BPRMF_batch.BPRMF_batch.BPRMF_ba | tch method), 77 |
| method), 60 | train() (elliot.recommender.neural.NAIS.nais |
| train() (elliot.recommender.latent_factor_mod- | NAIS method), 78 |
| els.BPRSlim.bprslim.BPRSlim | train() (elliot.recommender.neural.NeuMF.neu- |
| method), 61 | ral_matrix_factorization.NeuMF |
| train() (elliot.recommender.latent_factor_mod- | method), 81 |
| els.CML.CML method), 62 | train() (elliot.recommender.neural.NFM.neu- |
| train() (elliot.recommender.latent_factor_mod- | ral_fm.NFM method), 79 |
| | FfMn() (elliot.recommender.neural.NPR.neu- |
| method), 63 | ral_personalized_ranking.NPR |
| train() (elliot.recommender.latent_factor_mod- | method), 80 |
| els.FISM.FISM method), 63 | train() (elliot.recommender.neural.User- |
| train() (elliot.recommender.latent_factor_mod- | AutoRec.userautorec.UserAutoRec |
| els.FM.factorization_machine.FM | method), 81 |
| method), 64 | train() (elliot.recommender.neural.WideAnd- |
| train() (elliot.recommender.latent_factor_mod- | Deep.wide_and_deep.WideAndDeep |
| els.FunkSVD.funk_svd.FunkSVD | method), 82 |
| method), 65 | train() (elliot.recommender.NN.at- |
| | tribute_item_knn.attribute_item_knn.AttributeItemKN |
| train() (elliot.recommender.latent_factor_mod- | |
| els.LogisticMF.logistic_matrix_factorization. | |
| method), 66 | train() (elliot.recommender.NN.at- |
| train() (elliot.recommender.latent_factor_mod- | tribute_user_knn.attribute_user_knn.AttributeUserKN |

method), 35

 $els. MF. matrix_factorization. MF$

train() (elliot.recommender.NN.item_kn $lightgcn. LightGCN_model. LightGCNModel\\$ n.item_knn.ItemKNN method), 36 method), 54 train() (elliot.recommender.NN.user_kntrain_step() (elliot.recommender.n.user_knn.UserKNN method), 38 graph_based.ngcf.NGCF_model.NGCFModel train() (elliot.recommender.recommender_utilmethod), 55 s_mixin.RecMixin method), 89 train_step() (elliot.recommender.knowltrain() (elliot.recommender.unpersonalized.edge_aware.kaHFM.ka_hfm.KaHFM method), 56 most_popular.most_popular.MostPop method), 84 train_step() (elliot.recommender.knowledge_aware.kaHFM_batch.kahfm_batch_model.KaHFM train() (elliot.recommender.unpersonalized.random_recommender.Random.Random method), 58 method), 85 train_step() (elliot.recommender.knowl $train() \ (elliot.recommender.visual_recom$ edge_aware.kahfm_embeddings.kahfm_embeddings_r menders.ACF.ACF.ACF method), 85 train() (elliot.recommender.visual_recomtrain_step() (elliot.recommender.latent_factor_menders. Deep Style. Deep Style. Deep Stylemodels. BPRMF. BPRMF. BPRMFmethod), 59 method), 87 train() (elliot.recommender.visual recomtrain_step() (elliot.recommender.latent_factor_menders.DVBPR.DVBPR.DVBPR $models.BPRMF_batch_BPRMF_batch_model.BPRMF_b$ method), 86 method), 61 train() (elliot.recommender.visual_recomtrain_step() (elliot.recommender.latent_factor_ $models. BPRS lim.bprs lim_model. BPRS limModel$ menders.VBPR.VBPR.VBPR method), method), 61 train_step() (elliot.recommender.latent_factor_train() (elliot.recommender.visual_recommenders.VNPR.visual_neural_personalized_ranking.VfNRels.CML_CML_model.CML_model method), 88 method), 62 train_step() (elliot.recommender.adversaritrain_step() (elliot.recommender.latent_factor_ $al. AMF. AMF_model. AMF_model$ models.FFM.field_aware_factorization_machine_mode method), 40 train_step() (elliot.recommender.latent_factor_train_step() (elliot.recommender.adversari $al. AMR. AMR_model. AMR_model$ $models.FISM.FISM_model.FISM_model$ method), 42 method), 64 train_step() (elliot.recommender.attentrain_step() (elliot.recommender.latent_factor_ $models. FM. factorization_machine_model. Factorization$ tive.afm.afm_model.AFMModel method), 44 method), 65 train_step() (elliot.recommender.latent_factor_train_step() (elliot.recommender.autoencoders. $dae.multi_dae_model.DenoisingAutoEncoder$ $models. Funk SVD. funk_svd_model. Funk SVD Model$ method), 45 method), 66 train_step() (elliot.recommender.autoencoders.train_step() (elliot.recommender.latent_factor_vae.multi_vae_model.VariationalAutoEncoder models.LogisticMF.logistic_matrix_factorization_mode method), 46 train_step() (elliot.recommender.gan.CFtrain_step() (elliot.recommender.latent_factor_-GAN.cfgan_model.CFGAN_model $models. MF. matrix_factorization_model. MatrixFactorization_model. MatrixFactorization_models. MatrixFactorizati$ method), 49 method), 67 train_step() (elliot.recommender.gan.CFtrain_step() (elliot.recommender.latent_factor_-GAN.cfgan_model.Discriminator models.NonNegMF.non_negative_matrix_factorization method), 50 method), 68 train_step() (elliot.recommender.gan.CFtrain_step() (elliot.recommender.latent_factor_-GAN.cfgan_model.Generator $models. PMF. probabilistic_matrix_factorization_model.$ method), 50 train_step() (elliot.recommender.latent_factor_train_step() (elliot.recommender.gan.IRGAN.irgan_model.Discriminator method), 51 $models. Pure SVD. pure_svd_model. Pure SVD Model$ train_step() (elliot.recommender.gan.IRGAN.irmethod), 69 gan_model.Generator method), 51 train_step() (elliot.recommender.latent_factor_train_step() (elliot.recommender.gan.IRGAN.ir $models. SVDpp. svdpp_model. SVDppModel$ gan_model.IRGAN_model method), method), 70

Index 135

train_step() (elliot.recommender.graph_based.-

train_step() (elliot.recommender.latent_factor_-

 $models.WRMF.wrmf_model.WRMFModel$

| method), 72 | edge_aware.kaHFM.ka_hfm.KaHFM |
|--|--|
| train_step() (elliot.recommender.neural.Con- | method), 56 |
| | od ep.Catn<u>v</u>fslatnis (f)a ctbiozaticonModed ler.latent |
| method), 73 | factor_models.BPRMF.BPRMF.BPRMF |
| train_step() (elliot.recommender.neural.Con- | method), 59 |
| | orizat@mlynBookelaGInteNeticalMatelkiEtateoizationModel |
| method), 74 | mender.NN.item_knn.aiolli_ferrari.Compute_Similarit |
| train_step() (elliot.recommender.neu- | method), 36 |
| | el useOnlyBooleanInteractions() (elliot.recom- |
| method), 76
train_step() (elliot.recommender.neural.DM- | mender.NN.user_knn.aiolli_ferrari.Compute_Similarity |
| | method), 37
at tixFnAtotriRationNasodn lelliot.recommender.neu- |
| method), 75 | ral.UserAutoRec.userautorec), 81 |
| train_step() (elliot.recommender.neural.Gener- | UserAutoRecModel (class in elliot.recom- |
| | model.Gemenlikech/wataix/Bet/Ariza/Ren/Msadelutorec_model), |
| method), 77 | 82 |
| train_step() (elliot.recommender.neu- | UserCoverage (class in elliot.evaluation.metric- |
| | utoRecModecoverage.user_coverage.user_coverage), |
| method), 78 | 19 |
| train_step() (elliot.recommender.neural.NAIS | UserCoverageAtN (class in elliot.evaluation |
| nais_model.NAIS_model method), 79 | metrics.coverage.user_coverage.user_coverage_at_n), |
| train_step() (elliot.recommender.neural.NeuM- | 20 |
| F.neural_matrix_factorization_model.Neura | |
| method), 81 | N.user_knn.user_knn), 38 |
| train_step() (elliot.recommender.neu- | UserMADranking (class in elliot.evaluation |
| | tionMachi mMrids. Ifairness.MAD.UserMADranking),
24 |
| method), 80
train_step() (elliot.recommender.neural.N- | UserMADrating (class in elliot.evaluation.met- |
| | PRModel rics.fairness.MAD.UserMADrating), |
| method), 80 | 24 |
| train_step() (elliot.recommender.neural.User- | |
| AutoRec.userautorec_model.UserAutoRecM | 1o v jel |
| method), 82 | VariationalAutoEncoder (class in elliot.recom- |
| train_step() (elliot.recommender.neural.Wide- | mender autoencoders vae multi vae model) |
| AndDeep.wide_and_deep_model.WideAnd | mender.autoencoders.vae.multi_vae_model), |
| method), 01 | VBPR (class in elliot.recommender.visual_rec- |
| train_step() (elliot.recommender.visual_recom- | ommenders.VBPR.VBPR), 87 |
| menders.ACF.ACF_model.ACF_model | VBPR_model (class in elliot.recommender.visu- |
| method), 85 | al_recommenders.VBPR.VBPR_model), |
| train_step() (elliot.recommender.visual_recom- | State madel |
| method), 87 | Style_model
VisualDataObject (class in elliot.dataset.dat- |
| metriod), 67 | aloader visual dataloader) 6 |
| menders DVBPR DVBPR model DVBPR m | VisualLoader (class in elliot.dataset.dataload-
odel er.visual_dataloader), 6 |
| method), 86 | |
| train_step() (elliot.recommender.visual_recom- | VNPR (class in elliot.recommender.visual_rec- |
| menders.VBPR.VBPR_model.VBPR_model | ommenders.VNPR.visual_neural_personalized_rankin
88 |
| method), 88 | VNPRModel (class in elliot.recommender.visu- |
| train_step() (elliot.recommender.visual_recom- | al recommenders VNPR visual neural personalized r |
| $menders. VNPR. visual_neural_personalized$ | al_recommenders.VNPR.visual_neural_personalized_r
l_ranking_model.VNPRModel |
| method), 89 | VSM (class in elliot.recommender.con- |
| train_step_with_reward() (elliot.recommender | tent_based.VSM.vector_space_model), |
| gan.IRGAN.irgan_model.Generator | 47 |
| method), 52 | |
| 11 | W |
| U | WideAndDeep (class in elliot.recom- |
| update_factors() (elliot.recommender.knowl- | mender.neural.WideAndDeep.wide and deep). |

WideAndDeepModel (class in elliot.recommender.neural.WideAndDeep.wide_and_deep_model), 82
WilcoxonTest (class in elliot.evaluation.statistical_significance), 31
WilcoxonTest (elliot.result_handler.result_handler.StatTest attribute), 90
WRMF (class in elliot.recommender.latent_factor_models.WRMF.wrmf), 71
WRMFModel (class in elliot.recommender.latent_factent_factor_models.WRMF.wrmf_model), 71