
interpret-community

Release 0.24.0

Microsoft

Jan 10, 2022

CONTENTS

1	API Reference	3
2	Indices and tables	19
	Python Module Index	21
	Index	23

The code is available from [GitHub](#).

API REFERENCE

1.1 interpret_community package

1.1.1 Subpackages

`interpret_community.adapter` package

Submodules

`interpret_community.adapter.explanation_adapter` module

`interpret_community.common` package

Common infrastructure, class hierarchy and utilities for model explanations.

class `interpret_community.common.ModelSummary`

Bases: `object`

A structure for gathering and storing the parts of an explanation asset.

add_from_get_model_summary(*name*, *artifact_metadata_tuple*)

Update artifacts and metadata with new information.

Parameters

- **name** (`str`) – The name the new data should be associated with.
- **artifact_metadata_tuple** (`(list[dict], dict)`) – The tuple of artifacts and metadata to add to existing.

get_artifacts()

Get the list of artifacts.

Returns Artifact list.

Return type `list[list[dict]]`

get_metadata_dictionary()

Get the combined dictionary of metadata.

Returns Metadata dictionary.

Return type `dict`

Subpackages

`interpret_community.common.model_wrapper` package

Submodules

`interpret_community.common.aggregate` module

`interpret_community.common.base_explainer` module

Defines the base explainer API to create explanations.

```
class interpret_community.common.base_explainer.BaseExplainer(*args, **kwargs)
    Bases: interpret_community.common.base_explainer.GlobalExplainer, interpret_community.common.base_explainer.LocalExplainer
```

The base class for explainers that create global and local explanations.

```
class interpret_community.common.base_explainer.GlobalExplainer(*args, **kwargs)
    Bases: abc.ABC, interpret_community.common.chained_identity.ChainedIdentity
```

The base class for explainers that create global explanations.

```
abstract explain_global(*args, **kwargs)
```

Abstract method to globally explain the given model.

Note evaluation examples can be optional on derived classes since some explainers don't support it, for example MimicExplainer.

Returns A model explanation object containing the global explanation.

Return type `GlobalExplanation`

```
class interpret_community.common.base_explainer.LocalExplainer(*args, **kwargs)
    Bases: abc.ABC, interpret_community.common.chained_identity.ChainedIdentity
```

The base class for explainers that create local explanations.

```
abstract explain_local(evaluation_examples, **kwargs)
```

Abstract method to explain local instances.

Parameters `evaluation_examples` (*object*) – The evaluation examples.

Returns A model explanation object containing the local explanation.

Return type `LocalExplanation`

`interpret_community.common.blackbox_explainer` module

`interpret_community.common.chained_identity` module

Defines a light-weight chained identity for logging.

```
class interpret_community.common.chained_identity.ChainedIdentity(**kwargs)
    Bases: object
```

The base class for logging information.

interpret_community.common.constants module

Defines constants for interpret community.

```
class interpret_community.common.constants.Attributes
```

Bases: `object`

Provide constants for attributes.

```
EXPECTED_VALUE = 'expected_value'
```

```
class interpret_community.common.constants.DNNFramework
```

Bases: `object`

Provide DNN framework constants.

```
PYTORCH = 'pytorch'
```

```
TENSORFLOW = 'tensorflow'
```

```
class interpret_community.common.constants.Defaults
```

Bases: `object`

Provide constants for default values to explain methods.

```
AUTO = 'auto'
```

```
DEFAULT_BATCH_SIZE = 100
```

```
HDBSCAN = 'hdbscan'
```

```
MAX_DIM = 50
```

```
class interpret_community.common.constants.Dynamic
```

Bases: `object`

Provide constants for dynamically generated classes.

```
GLOBAL_EXPLANATION = 'DynamicGlobalExplanation'
```

```
LOCAL_EXPLANATION = 'DynamicLocalExplanation'
```

```
class interpret_community.common.constants.ExplainParams
```

Bases: `object`

Provide constants for interpret community (init, explain_local and explain_global) parameters.

```
BATCH_SIZE = 'batch_size'
```

```
CLASSES = 'classes'
```

```
CLASSIFICATION = 'classification'
```

```
EVAL_DATA = 'eval_data'
```

```
EVAL_Y_PRED = 'eval_y_predicted'
```

```
EVAL_Y_PRED_PROBA = 'eval_y_predicted_proba'
```

```
EXPECTED_VALUES = 'expected_values'
```

```
EXPLAIN_SUBSET = 'explain_subset'
```

```
EXPLANATION_ID = 'explanation_id'
```

```
FEATURES = 'features'
```

```
GLOBAL_IMPORTANCE_NAMES = 'global_importance_names'
```

```
GLOBAL_IMPORTANCE_RANK = 'global_importance_rank'
GLOBAL_IMPORTANCE_VALUES = 'global_importance_values'
GLOBAL_NAMES = 'global_names'
GLOBAL_RANK = 'global_rank'
GLOBAL_VALUES = 'global_values'
ID = 'id'
INCLUDE_LOCAL = 'include_local'
INIT_DATA = 'init_data'
IS_ENG = 'is_engineered'
IS_LOCAL_SPARSE = 'is_local_sparse'
IS_RAW = 'is_raw'
LOCAL_EXPLANATION = 'local_explanation'
LOCAL_IMPORTANCE_VALUES = 'local_importance_values'
METHOD = 'method'
MODEL_ID = 'model_id'
MODEL_TASK = 'model_task'
MODEL_TYPE = 'model_type'
NUM_CLASSES = 'num_classes'
NUM_EXAMPLES = 'num_examples'
NUM_FEATURES = 'num_features'
PER_CLASS_NAMES = 'per_class_names'
PER_CLASS_RANK = 'per_class_rank'
PER_CLASS_VALUES = 'per_class_values'
PROBABILITIES = 'probabilities'
SAMPLING_POLICY = 'sampling_policy'
SHAP_VALUES_OUTPUT = 'shap_values_output'
```

```
classmethod get_private(explain_param)
```

Return the private version of the ExplainParams property.

Parameters

- **cls** ([ExplainParams](#)) – ExplainParams input class.
- **explain_param** (*str*) – The ExplainParams property to get private version of.

Returns The private version of the property.

Return type *str*

```
classmethod get_serializable()
```

Return only the ExplainParams properties that have meaningful data values for serialization.

Parameters **cls** ([ExplainParams](#)) – ExplainParams input class.

Returns A set of property names, e.g., 'GLOBAL_IMPORTANCE_VALUES', 'MODEL_TYPE', etc.

Return type `set[str]`

class `interpret_community.common.constants.ExplainType`

Bases: `object`

Provide constants for model and explainer type information, useful for visualization.

CLASSIFICATION = 'classification'

DATA = 'data_type'

EXPLAIN = 'explain_type'

EXPLAINER = 'explainer'

FUNCTION = 'function'

GLOBAL = 'global'

HAN = 'han'

IS_ENG = 'is_engineered'

IS_RAW = 'is_raw'

LIME = 'lime'

LOCAL = 'local'

METHOD = 'method'

MIMIC = 'mimic'

MODEL = 'model_type'

MODEL_CLASS = 'model_class'

MODEL_TASK = 'model_task'

PFI = 'pfi'

REGRESSION = 'regression'

SHAP = 'shap'

SHAP_DEEP = 'shap_deep'

SHAP_GPU_KERNEL = 'shap_gpu_kernel'

SHAP_KERNEL = 'shap_kernel'

SHAP_LINEAR = 'shap_linear'

SHAP_TREE = 'shap_tree'

TABULAR = 'tabular'

class `interpret_community.common.constants.ExplainableModelType(value)`

Bases: `str, enum.Enum`

Provide constants for the explainable model type.

LINEAR_EXPLAINABLE_MODEL_TYPE = 'linear_explainable_model_type'

TREE_EXPLAINABLE_MODEL_TYPE = 'tree_explainable_model_type'

```
class interpret_community.common.constants.ExplanationParams
    Bases: object

    Provide constants for explanation parameters.

    CLASSES = 'classes'

    EXPECTED_VALUES = 'expected_values'

class interpret_community.common.constants.Extension
    Bases: object

    Provide constants for extensions to interpret package.

    BLACKBOX = 'blackbox'

    GLASSBOX = 'model'

    GLOBAL = 'global'

    GREYBOX = 'specific'

    LOCAL = 'local'

class interpret_community.common.constants.InterpretData
    Bases: object

    Provide Data and Visualize constants for interpret core.

    BASE_VALUE = 'Base Value'

    EXPLANATION_CLASS_DIMENSION = 'explanation_class_dimension'

    EXPLANATION_TYPE = 'explanation_type'

    EXTRA = 'extra'

    FEATURE_LIST = 'feature_list'

    GLOBAL_FEATURE_IMPORTANCE = 'global_feature_importance'

    INTERCEPT = 'intercept'

    LOCAL_FEATURE_IMPORTANCE = 'local_feature_importance'

    MLI = 'mli'

    MULTICLASS = 'multiclass'

    NAMES = 'names'

    OVERALL = 'overall'

    PERF = 'perf'

    SCORES = 'scores'

    SINGLE = 'single'

    SPECIFIC = 'specific'

    TYPE = 'type'

    UNIVARIATE = 'univariate'

    VALUE = 'value'

    VALUES = 'values'
```

```

class interpret_community.common.constants.LightGBMParams
    Bases: object

    Provide constants for LightGBM.

    CATEGORICAL_FEATURE = 'categorical_feature'

class interpret_community.common.constants.LightGBMSerializationConstants
    Bases: object

    Provide internal class that defines fields used for MimicExplainer serialization.

    IDENTITY = '_identity'
    LOGGER = '_logger'
    MODEL_STR = 'model_str'
    MULTICLASS = 'multiclass'
    OBJECTIVE = 'objective'
    REGRESSION = 'regression'
    TREE_EXPLAINER = '_tree_explainer'
    enum_properties = ['_shap_values_output']
    nonify_properties = ['_logger', '_tree_explainer']
    save_properties = ['_lgbm']

class interpret_community.common.constants.MimicSerializationConstants
    Bases: object

    Provide internal class that defines fields used for MimicExplainer serialization.

    ALLOW_ALL_TRANSFORMATIONS = '_allow_all_transformations'
    FUNCTION = 'function'
    IDENTITY = '_identity'
    INITIALIZATION_EXAMPLES = 'initialization_examples'
    LOGGER = '_logger'
    MODEL = 'model'
    ORIGINAL_EVAL_EXAMPLES = '_original_eval_examples'
    PREDICT_PROBA_FLAG = 'predict_proba_flag'
    RESET_INDEX = 'reset_index'
    TIMESTAMP_FEATURIZER = '_timestamp_featurizer'
    enum_properties = ['_shap_values_output']
    nonify_properties = ['_logger', 'model', 'function', 'initialization_examples',
                        '_original_eval_examples', '_timestamp_featurizer']
    save_properties = ['surrogate_model']

class interpret_community.common.constants.ModelTask(value)
    Bases: str, enum.Enum

    Provide model task constants. Can be 'classification', 'regression', or 'unknown'.

```

By default the model domain is inferred if 'unknown', but this can be overridden if you specify 'classification' or 'regression'.

Classification = 'classification'

Regression = 'regression'

Unknown = 'unknown'

class interpret_community.common.constants.**ResetIndex**(value)

Bases: `str`, `enum.Enum`

Provide index column handling constants. Can be 'ignore', 'reset' or 'reset_teacher'.

By default the index column is ignored, but you can override to reset it and make it a feature column that is then featurized to numeric, or reset it and ignore it during featurization but set it as the index when calling predict on the original model.

Ignore = 'ignore'

Reset = 'reset'

ResetTeacher = 'reset_teacher'

class interpret_community.common.constants.**SHAPDefaults**

Bases: `object`

Provide constants for default values to SHAP.

INDEPENDENT = 'independent'

class interpret_community.common.constants.**SKLearn**

Bases: `object`

Provide scikit-learn related constants.

EXAMPLES = 'examples'

LABELS = 'labels'

PREDICTIONS = 'predictions'

PREDICT_PROBA = 'predict_proba'

class interpret_community.common.constants.**Scipy**

Bases: `object`

Provide scipy related constants.

CSR_FORMAT = 'csr'

class interpret_community.common.constants.**ShapValuesOutput**(value)

Bases: `str`, `enum.Enum`

Provide constants for the SHAP values output from the explainer.

Can be 'default', 'probability' or 'teacher_probability'. If 'teacher_probability' is specified, we use the probabilities from the teacher model.

DEFAULT = 'default'

PROBABILITY = 'probability'

TEACHER_PROBABILITY = 'teacher_probability'

```

class interpret_community.common.constants.Spacy
    Bases: object
    Provide spaCy related constants.
    EN = 'en'
    NER = 'ner'
    TAGGER = 'tagger'

class interpret_community.common.constants.Tensorflow
    Bases: object
    Provide TensorFlow and TensorBoard related constants.
    CPU0 = '/CPU:0'
    TFLOG = 'tflog'

```

interpret_community.common.error_handling module

Defines error handling utilities.

interpret_community.common.exception module

Defines different types of exceptions that this package can raise.

```

exception interpret_community.common.exception.ScenarioNotSupportedException
    Bases: Exception
    An exception indicating that some scenario is not supported.
    Parameters exception_message (str) – A message describing the error.

```

interpret_community.common.explanation_utils module

Defines helpful utilities for summarizing and uploading data.

interpret_community.common.gpu_kmeans module

The code is based on the similar utility function from SHAP: https://github.com/slundberg/shap/blob/9411b68e8057a6c6f3621765b89b24d82bee13d4/shap/utils/_legacy.py This version makes use of cuml kmeans instead of sklearn for speed.

```

class interpret_community.common.gpu_kmeans.Data
    Bases: object

class interpret_community.common.gpu_kmeans.DenseData(data, group_names, *args)
    Bases: interpret_community.common.gpu_kmeans.Data

interpret_community.common.gpu_kmeans.kmeans(X, k, round_values=True)
    Summarize a dataset with k mean samples weighted by the number of data points they each represent. Parameters
    ——— X : numpy.array or pandas.DataFrame or any scipy.sparse matrix
        Matrix of data samples to summarize (# samples x # features)

```

k [int] Number of means to use for approximation.

round_values [bool] For all i, round the ith dimension of each mean sample to match the nearest value from $X[:,i]$. This ensures discrete features always get a valid value.

DenseData object.

interpret_community.common.metrics module

Defines metrics for validating model explanations.

`interpret_community.common.metrics.dcg(validate_order, ground_truth_order_relevance, top_values=10)`
Compute the discounted cumulative gain (DCG).

Compute the DCG as the sum of relevance scores penalized by the logarithmic position of the result. See https://en.wikipedia.org/wiki/Discounted_cumulative_gain for reference.

Parameters

- **validate_order** (*list*) – The order to validate.
- **ground_truth_order_relevance** (*list*) – The ground truth relevancy of the documents to compare to.
- **top_values** (*int*) – Specifies the top values to compute the DCG for. The default is 10.

`interpret_community.common.metrics.ndcg(validate_order, ground_truth_order, top_values=10)`
Compute the normalized discounted cumulative gain (NDCG).

Compute the NDCG as the ratio of the DCG for the validation order compared to the maximum DCG possible for the ground truth order. If the validation order is the same as the ground truth the NDCG will be the maximum of 1.0, and the least possible NDCG is 0.0. See https://en.wikipedia.org/wiki/Discounted_cumulative_gain for reference.

Parameters

- **validate_order** (*list*) – The order to validate for the documents. The values should be unique.
- **ground_truth_order** (*list*) – The true order of the documents. The values should be unique.
- **top_values** (*int*) – Specifies the top values to compute the NDCG for. The default is 10.

interpret_community.common.model_summary module

Defines a structure for gathering and storing the parts of an explanation asset.

class `interpret_community.common.model_summary.ModelSummary`
Bases: `object`

A structure for gathering and storing the parts of an explanation asset.

add_from_get_model_summary(*name, artifact_metadata_tuple*)
Update artifacts and metadata with new information.

Parameters

- **name** (*str*) – The name the new data should be associated with.

- **artifact_metadata_tuple** (*(list[dict], dict)*) – The tuple of artifacts and meta-data to add to existing.

get_artifacts()

Get the list of artifacts.

Returns Artifact list.

Return type *list[list[dict]]*

get_metadata_dictionary()

Get the combined dictionary of metadata.

Returns Metadata dictionary.

Return type *dict*

interpret_community.common.policy module

Defines explanation policies.

```
class interpret_community.common.policy.SamplingPolicy(allow_eval_sampling=False,  
                                                    max_dim_clustering=50,  
                                                    sampling_method='hdbscan', **kwargs)
```

Bases: *interpret_community.common.chained_identity.ChainedIdentity*

Defines the sampling policy for downsampling the evaluation examples.

The policy is a set of parameters that can be tuned to speed up or improve the accuracy of the explain_model function during sampling.

Parameters

- **allow_eval_sampling** (*bool*) – Default to 'False'. Specify whether to allow sampling of evaluation data. If 'True', cluster the evaluation data and determine the optimal number of points for sampling. Set to 'True' to speed up the process when the evaluation data set is large and you only want to generate model summary info.
- **max_dim_clustering** (*int*) – Default to 50 and only take effect when 'allow_eval_sampling' is set to 'True'. Specify the dimensionality to reduce the evaluation data before clustering for sampling. When doing sampling to determine how aggressively to downsample without getting poor explanation results uses a heuristic to find the optimal number of clusters. Since KMeans performs poorly on high dimensional data PCA or Truncated SVD is first run to reduce the dimensionality, which is followed by finding the optimal k by running KMeans until a local minimum is reached as determined by computing the silhouette score, reducing k each time.
- **sampling_method** (*str*) – The sampling method for determining how much to downsample the evaluation data by. If allow_eval_sampling is True, the evaluation data is downsampled to a max_threshold, and then this heuristic is used to determine how much more to downsample the evaluation data without losing accuracy on the calculated feature importance values. By default, this is set to hdbscan, but you can also specify kmeans. With hdbscan the number of clusters is automatically determined and multiplied by a threshold. With kmeans, the optimal number of clusters is found by running KMeans until the maximum silhouette score is calculated, with k halved each time.

Return type *dict*

Returns The arguments for the sampling policy

property allow_eval_sampling

Get whether to allow sampling of evaluation data.

Returns Whether to allow sampling of evaluation data.

Return type `bool`

property max_dim_clustering

Get the dimensionality to reduce the evaluation data before clustering for sampling.

Returns The dimensionality to reduce the evaluation data before clustering for sampling.

Return type `int`

property sampling_method

Get the sampling method for determining how much to downsample the evaluation data by.

Returns The sampling method for determining how much to downsample the evaluation data by.

Return type `str`

interpret_community.common.progress module

Defines utilities for getting progress status for explanation.

`interpret_community.common.progress.get_tqdm(logger, show_progress)`

Get the tqdm progress bar function.

Parameters

- **logger** (*logger*) – The logger for logging info messages.
- **show_progress** (*bool*) – Default to ‘True’. Determines whether to display the explanation status bar when using PFIExplainer.

Returns The tqdm (<https://github.com/tqdm/tqdm>) progress bar.

Return type `function`

interpret_community.common.serialization_utils module

Defines utility functions for serialization of data.

interpret_community.common.structured_model_explainer module

Defines the structured model based APIs for explainers used on specific types of models.

class `interpret_community.common.structured_model_explainer.PureStructuredModelExplainer`(*model*,
***kwargs*)

Bases: `interpret_community.common.base_explainer.BaseExplainer`

The base PureStructuredModelExplainer API for explainers used on specific models.

Parameters **model** (*object*) – The white box model to explain.

```
class interpret_community.common.structured_model_explainer.StructuredInitModelExplainer(model,
                                                                                          ini-
                                                                                          tial-
                                                                                          iza-
                                                                                          tion_examples,
                                                                                          **kwargs)
```

Bases: `interpret_community.common.base_explainer.BaseExplainer`

The base StructuredInitModelExplainer API for explainers.

Used on specific models that require initialization examples.

Parameters

- **model** (*object*) – The white box model to explain.
- **initialization_examples** (*numpy.array or pandas.DataFrame or scipy.sparse.csr_matrix*) – A matrix of feature vector examples (# examples x # features) for initializing the explainer.

interpret_community.common.warnings_suppressor module

Suppresses warnings on imports.

```
class interpret_community.common.warnings_suppressor.shap_warnings_suppressor
```

Bases: `object`

Context manager to suppress warnings from shap.

```
class interpret_community.common.warnings_suppressor.tf_warnings_suppressor
```

Bases: `object`

Context manager to suppress warnings from tensorflow.

interpret_community.dataset package

Defines a common dataset wrapper and common functions for data manipulation.

Subpackages

`interpret_community.dataset.dataset_wrapper` package

Submodules

`interpret_community.dataset.decorator` module

`interpret_community.explanation` package

Submodules

`interpret_community.explanation.explanation` module

`interpret_community.explanation.serialization` module

interpret_community.lime package

Submodules

interpret_community.lime.lime_explainer module

interpret_community.mimic package

Subpackages

interpret_community.mimic.models package

Submodules

interpret_community.mimic.models.explainable_model module

interpret_community.mimic.models.lightgbm_model module

interpret_community.mimic.models.linear_model module

interpret_community.mimic.models.tree_model module

interpret_community.mimic.models.tree_model_utils module

Submodules

interpret_community.mimic.mimic_explainer module

interpret_community.mimic.model_distill module

interpret_community.mlflow package

Submodules

interpret_community.mlflow.mlflow module

interpret_community.permutation package

Submodules

interpret_community.permutation.metric_constants module

interpret_community.permutation.permutation_importance module

interpret_community.shap package

Submodules

interpret_community.shap.deep_explainer module

interpret_community.shap.gpu_kernel_explainer module

interpret_community.shap.kernel_explainer module

interpret_community.shap.kwargs_utils module

interpret_community.shap.linear_explainer module

interpret_community.shap.tree_explainer module

interpret_community.widget package

Submodules

interpret_community.widget.explanation_dashboard module

1.1.2 Submodules

interpret_community.tabular_explainer module

interpret_community.version module

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

i

- `interpret_community.common`, 3
- `interpret_community.common.base_explainer`, 4
- `interpret_community.common.chained_identity`,
4
- `interpret_community.common.constants`, 5
- `interpret_community.common.error_handling`, 11
- `interpret_community.common.exception`, 11
- `interpret_community.common.explanation_utils`,
11
- `interpret_community.common.gpu_kmeans`, 11
- `interpret_community.common.metrics`, 12
- `interpret_community.common.model_summary`, 12
- `interpret_community.common.policy`, 13
- `interpret_community.common.progress`, 14
- `interpret_community.common.serialization_utils`,
14
- `interpret_community.common.structured_model_explainer`,
14
- `interpret_community.common.warnings_suppressor`,
15
- `interpret_community.dataset`, 15

INDEX

A

`add_from_get_model_summary()` (inter-pret_community.common.model_summary.ModelSummary method), 12

`add_from_get_model_summary()` (inter-pret_community.common.ModelSummary method), 3

`ALLOW_ALL_TRANSFORMATIONS` (inter-pret_community.common.constants.MimicSerializationConstants attribute), 9

`allow_eval_sampling` (inter-pret_community.common.policy.SamplingPolicy property), 13

`Attributes` (class in inter-pret_community.common.constants), 5

`AUTO` (interpret_community.common.constants.Defaults attribute), 5

B

`BASE_VALUE` (interpret_community.common.constants.InterpretData attribute), 8

`BaseExplainer` (class in inter-pret_community.common.base_explainer), 4

`BATCH_SIZE` (interpret_community.common.constants.ExplainParams attribute), 5

`BLACKBOX` (interpret_community.common.constants.Extension attribute), 8

C

`CATEGORICAL_FEATURE` (inter-pret_community.common.constants.LightGBMParams attribute), 9

`ChainedIdentity` (class in inter-pret_community.common.chained_identity), 4

`CLASSES` (interpret_community.common.constants.ExplainParams attribute), 5

`CLASSES` (interpret_community.common.constants.ExplanationParams attribute), 8

`CLASSIFICATION` (inter-pret_community.common.constants.ExplainParams attribute), 5

`CLASSIFICATION` (inter-pret_community.common.constants.ExplainType attribute), 7

`Classification` (inter-pret_community.common.constants.ModelTask attribute), 10

`CPU` (interpret_community.common.constants.Tensorflow attribute), 11

`CSR_FORMAT` (interpret_community.common.constants.Scipy attribute), 10

D

`Data` (class in interpret_community.common.gpu_kmeans), 11

`DATA` (interpret_community.common.constants.ExplainType attribute), 7

`dcg()` (in module interpret_community.common.metrics), 12

`DEFAULT` (interpret_community.common.constants.ShapValuesOutput attribute), 10

`DEFAULT_BATCH_SIZE` (inter-pret_community.common.constants.Defaults attribute), 5

`Defaults` (class in inter-pret_community.common.constants), 5

`DenseData` (class in inter-pret_community.common.gpu_kmeans), 11

`DNNFramework` (class in inter-pret_community.common.constants), 5

`Dynamic` (class in inter-pret_community.common.constants), 5

E

`EN` (interpret_community.common.constants.Spacy attribute), 11

`enum_properties` (inter-pret_community.common.constants.LightGBMSerializationConstants attribute), 9

`enum_properties` (inter-pret_community.common.constants.MimicSerializationConstants attribute), 9

EVAL_DATA (*interpret_community.common.constants.ExplainParams* attribute), 5
EVAL_Y_PRED (*interpret_community.common.constants.ExplainParams* attribute), 5
EVAL_Y_PRED_PROBA (*interpret_community.common.constants.ExplainParams* attribute), 5
EXAMPLES (*interpret_community.common.constants.SKLearn* attribute), 10
EXPECTED_VALUE (*interpret_community.common.constants.Attributes* attribute), 5
EXPECTED_VALUES (*interpret_community.common.constants.ExplainParams* attribute), 5
EXPECTED_VALUES (*interpret_community.common.constants.ExplanationParams* attribute), 8
EXPLAIN (*interpret_community.common.constants.ExplainType* attribute), 7
explain_global() (*interpret_community.common.base_explainer.GlobalExplainer* method), 4
explain_local() (*interpret_community.common.base_explainer.LocalExplainer* method), 4
EXPLAIN_SUBSET (*interpret_community.common.constants.ExplainParams* attribute), 5
ExplainableModelType (class in *interpret_community.common.constants*), 7
EXPLAINER (*interpret_community.common.constants.ExplainType* attribute), 7
ExplainParams (class in *interpret_community.common.constants*), 5
ExplainType (class in *interpret_community.common.constants*), 7
EXPLANATION_CLASS_DIMENSION (*interpret_community.common.constants.InterpretData* attribute), 8
EXPLANATION_ID (*interpret_community.common.constants.ExplainParams* attribute), 5
EXPLANATION_TYPE (*interpret_community.common.constants.InterpretData* attribute), 8
ExplanationParams (class in *interpret_community.common.constants*), 7
Extension (class in *interpret_community.common.constants*), 8
EXTRA (*interpret_community.common.constants.InterpretData* attribute), 8
FEATURE_LIST (*interpret_community.common.constants.InterpretData* attribute), 8
FEATURES (*interpret_community.common.constants.ExplainParams* attribute), 5
FUNCTION (*interpret_community.common.constants.ExplainType* attribute), 7
FUNCTION (*interpret_community.common.constants.MimicSerializationCom* attribute), 9
G
get_artifacts() (*interpret_community.common.model_summary.ModelSummary* method), 13
get_artifacts() (*interpret_community.common.ModelSummary* method), 3
get_metadata_dictionary() (*interpret_community.common.model_summary.ModelSummary* method), 13
get_metadata_dictionary() (*interpret_community.common.ModelSummary* method), 3
get_private() (*interpret_community.common.constants.ExplainParams* class method), 6
get_serializable() (*interpret_community.common.constants.ExplainParams* class method), 6
get_tqdm() (in module *interpret_community.common.progress*), 14
GLASSBOX (*interpret_community.common.constants.Extension* attribute), 8
GLOBAL (*interpret_community.common.constants.ExplainType* attribute), 7
GLOBAL (*interpret_community.common.constants.Extension* attribute), 8
GLOBAL_EXPLANATION (*interpret_community.common.constants.Dynamic* attribute), 5
GLOBAL_FEATURE_IMPORTANCE (*interpret_community.common.constants.InterpretData* attribute), 8
GLOBAL_IMPORTANCE_NAMES (*interpret_community.common.constants.ExplainParams* attribute), 5
GLOBAL_IMPORTANCE_RANK (*interpret_community.common.constants.ExplainParams* attribute), 5
GLOBAL_IMPORTANCE_VALUES (*interpret_community.common.constants.ExplainParams* attribute), 6
GLOBAL_NAMES (*interpret_community.common.constants.ExplainParams* attribute), 6

GLOBAL_RANK (*interpret_community.common.constants.ExplainParams* attribute), 6
 GLOBAL_VALUES (*interpret_community.common.constants.ExplainParams* attribute), 6
 GlobalExplainer (class in *interpret_community.common.base_explainer*), 4
 GREYBOX (*interpret_community.common.constants.ExtensionConstants* attribute), 8
H
 HAN (*interpret_community.common.constants.ExplainType* attribute), 7
 HDBSCAN (*interpret_community.common.constants.Defaults* attribute), 5
I
 ID (*interpret_community.common.constants.ExplainParams* attribute), 6
 IDENTITY (*interpret_community.common.constants.LightGBMSerializationConstants* attribute), 9
 IDENTITY (*interpret_community.common.constants.MimicSerializationConstants* attribute), 9
 Ignore (*interpret_community.common.constants.ResetIndex* attribute), 10
 INCLUDE_LOCAL (*interpret_community.common.constants.ExplainParams* attribute), 6
 INDEPENDENT (*interpret_community.common.constants.SHAPDefaults* attribute), 10
 INIT_DATA (*interpret_community.common.constants.ExplainParams* attribute), 6
 INITIALIZATION_EXAMPLES (*interpret_community.common.constants.MimicSerializationConstants* attribute), 9
 INTERCEPT (*interpret_community.common.constants.InterpretData* attribute), 8
 interpret_community.common module, 3
 interpret_community.common.base_explainer module, 4
 interpret_community.common.chained_identity module, 4
 interpret_community.common.constants module, 5
 interpret_community.common.error_handling module, 11
 interpret_community.common.exception module, 11
 interpret_community.common.explanation_utils module, 11
 interpret_community.common.gpu_kmeans module, 11
 InterpretData (class in *interpret_community.common.constants*), 8
 IS_ENG (*interpret_community.common.constants.ExplainParams* attribute), 6
 IS_ENG (*interpret_community.common.constants.ExplainType* attribute), 7
 IS_LOCAL_SPARSE (*interpret_community.common.constants.ExplainParams* attribute), 6
 IS_RAW (*interpret_community.common.constants.ExplainParams* attribute), 6
 IS_RAW (*interpret_community.common.constants.ExplainType* attribute), 7
 KMeans (in module *interpret_community.common.gpu_kmeans*), 11
L
 LABELS (*interpret_community.common.constants.SKLearn* attribute), 10
 LightGBMParams (class in *interpret_community.common.constants*), 8
 LightGBMSerializationConstants (class in *interpret_community.common.constants*), 9
 LIME (*interpret_community.common.constants.ExplainType* attribute), 7
 LINEAR_EXPLAINABLE_MODEL_TYPE (*interpret_community.common.constants.ExplainableModelType* attribute), 7
 LOCAL (*interpret_community.common.constants.ExplainType* attribute), 7
 LOCAL (*interpret_community.common.constants.ExtensionConstants* attribute), 8
 LOCAL_EXPLANATION (*interpret_community.common.constants.Dynamic* attribute), 5

LOCAL_EXPLANATION (interpret_community.common.constants.ExplainParams attribute), 6
LOCAL_FEATURE_IMPORTANCE (interpret_community.common.constants.InterpretData attribute), 8
LOCAL_IMPORTANCE_VALUES (interpret_community.common.constants.ExplainParams attribute), 6
LocalExplainer (class in interpret_community.common.base_explainer), 4
LOGGER (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9
LOGGER (interpret_community.common.constants.MimicSerializationConstants attribute), 9
M
MAX_DIM (interpret_community.common.constants.Defaults attribute), 5
max_dim_clustering (interpret_community.common.policy.SamplingPolicy property), 14
METHOD (interpret_community.common.constants.ExplainParams attribute), 6
METHOD (interpret_community.common.constants.ExplainType attribute), 7
MIMIC (interpret_community.common.constants.ExplainType attribute), 7
MimicSerializationConstants (class in interpret_community.common.constants), 9
MLI (interpret_community.common.constants.InterpretData attribute), 8
MODEL (interpret_community.common.constants.ExplainType attribute), 7
MODEL (interpret_community.common.constants.MimicSerializationConstants attribute), 9
MODEL_CLASS (interpret_community.common.constants.ExplainType attribute), 7
MODEL_ID (interpret_community.common.constants.ExplainParams attribute), 6
MODEL_STR (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9
MODEL_TASK (interpret_community.common.constants.ExplainParams attribute), 6
MODEL_TASK (interpret_community.common.constants.ExplainType attribute), 7
MODEL_TYPE (interpret_community.common.constants.ExplainParams attribute), 6
ModelSummary (class in interpret_community.common), 3
ModelSummary (class in interpret_community.common.model_summary), 12
ModelTask (class in interpret_community.common), 9
interpret_community.common, 3
interpret_community.common.base_explainer, 4
interpret_community.common.chained_identity, 4
interpret_community.common.constants, 5
interpret_community.common.error_handling, 11
interpret_community.common.exception, 11
interpret_community.common.explanation_utils, 11
interpret_community.common.gpu_kmeans, 11
interpret_community.common.metrics, 12
interpret_community.common.model_summary, 12
interpret_community.common.policy, 13
interpret_community.common.progress, 14
interpret_community.common.serialization_utils, 14
interpret_community.common.structured_model_explainer, 14
interpret_community.common.warnings_suppressor, 15
interpret_community.dataset, 15
MULTICLASS (interpret_community.common.constants.InterpretData attribute), 8
MULTICLASS (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9
N
NAMES (interpret_community.common.constants.InterpretData attribute), 8
SerializationConstants (module in interpret_community.common.metrics), 12
SPACY (interpret_community.common.constants.Spacy attribute), 11
nonify_properties (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9
nonify_properties (interpret_community.common.constants.MimicSerializationConstants attribute), 9
NUM_CLASSES (interpret_community.common.constants.ExplainParams attribute), 6
NUM_EXAMPLES (interpret_community.common.constants.ExplainParams attribute), 6
NUM_FEATURES (interpret_community.common.constants.ExplainParams attribute), 6
O
OBJECTIVE (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9

attribute), 9

ORIGINAL_EVAL_EXAMPLES (interpret_community.common.constants.MimicSerializationConstants attribute), 9

OVERALL (interpret_community.common.constants.InterpretData attribute), 8

P

PER_CLASS_NAMES (interpret_community.common.constants.ExplainParams attribute), 6

PER_CLASS_RANK (interpret_community.common.constants.ExplainParams attribute), 6

PER_CLASS_VALUES (interpret_community.common.constants.ExplainParams attribute), 6

PERF (interpret_community.common.constants.InterpretData attribute), 8

PFI (interpret_community.common.constants.ExplainType attribute), 7

PREDICT_PROBA (interpret_community.common.constants.SKLearn attribute), 10

PREDICT_PROBA_FLAG (interpret_community.common.constants.MimicSerializationConstants attribute), 9

PREDICTIONS (interpret_community.common.constants.SKLearn attribute), 10

PROBABILITIES (interpret_community.common.constants.ExplainParams attribute), 6

PROBABILITY (interpret_community.common.constants.ShapValuesOutput attribute), 10

PureStructuredModelExplainer (class in interpret_community.common.structured_model_explainer), 14

PYTORCH (interpret_community.common.constants.DNNFramework attribute), 5

R

REGRESSION (interpret_community.common.constants.ExplainType attribute), 7

REGRESSION (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9

Regression (interpret_community.common.constants.ModelTask attribute), 10

Reset (interpret_community.common.constants.ResetIndex attribute), 10

RESET_INDEX (interpret_community.common.constants.MimicSerializationConstants attribute), 9

ResetIndex (class in interpret_community.common.constants), 10

ResetTeacher (interpret_community.common.constants.ResetIndex attribute), 10

S

SAMPLING_METHOD (interpret_community.common.policy.SamplingPolicy property), 14

SAMPLING_POLICY (interpret_community.common.constants.ExplainParams attribute), 6

SamplingPolicy (class in interpret_community.common.policy), 13

save_properties (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9

Scipy (class in interpret_community.common.constants), 10

SCORES (interpret_community.common.constants.InterpretData attribute), 8

SHAP (interpret_community.common.constants.ExplainType attribute), 7

SHAP_DEEP (interpret_community.common.constants.ExplainType attribute), 7

SHAP_GPU_KERNEL (interpret_community.common.constants.ExplainType attribute), 7

SHAP_KERNEL (interpret_community.common.constants.ExplainType attribute), 7

SHAP_LINEAR (interpret_community.common.constants.ExplainType attribute), 7

SHAP_TREE (interpret_community.common.constants.ExplainType attribute), 7

SHAP_VALUES_OUTPUT (interpret_community.common.constants.ExplainParams attribute), 6

shap_warnings_suppressor (class in interpret_community.common.warnings_suppressor), 15

SHAPDefaults (class in interpret_community.common.constants), 10

ShapValuesOutput (class in interpret_community.common.constants), 10

SINGLE (interpret_community.common.constants.InterpretData attribute), 8

SKLearn (class in interpret_community.common.constants), 10

Spacy (class in interpret_community.common.constants), 10

SPECIFIC (interpret_community.common.constants.InterpretData attribute), 8

StructuredInitModelExplainer (class in interpret_community.common.structured_model_explainer),
14

T

TABULAR (interpret_community.common.constants.ExplainType attribute), 7

TAGGER (interpret_community.common.constants.Spacy attribute), 11

TEACHER_PROBABILITY (interpret_community.common.constants.ShapValuesOutput attribute), 10

Tensorflow (class in interpret_community.common.constants), 11

TENSORFLOW (interpret_community.common.constants.DNNFramework attribute), 5

tf_warnings_suppressor (class in interpret_community.common.warnings_suppressor),
15

TFLOG (interpret_community.common.constants.Tensorflow attribute), 11

TIMESTAMP_FEATURIZER (interpret_community.common.constants.MimicSerializationConstants attribute), 9

TREE_EXPLAINABLE_MODEL_TYPE (interpret_community.common.constants.ExplainableModelType attribute), 7

TREE_EXPLAINER (interpret_community.common.constants.LightGBMSerializationConstants attribute), 9

TYPE (interpret_community.common.constants.InterpretData attribute), 8

U

UNIVARIATE (interpret_community.common.constants.InterpretData attribute), 8

Unknown (interpret_community.common.constants.ModelTask attribute), 10

V

VALUE (interpret_community.common.constants.InterpretData attribute), 8

VALUES (interpret_community.common.constants.InterpretData attribute), 8