aisquared

The AI Squared Team

DOCUMENTATION

1	Installation															3
1.1.1 aisquared package										3						
		1.1.1.1	Module cont	ents .						 	 	 				3
		1.1.1.2	Subpackages							 	 	 				3
2	Changelog															61
Ру	thon Module	Index														65
In	dex															67



This package contains utilities to interact with the AI Squared technology stack, particularly with developing and deploying models to the AI Squared Platform or other applications developed through the AI Squared JavaScript SDK.

Current Production Version: 0.3.6

View this Documentation in PDF Format.

DOCUMENTATION 1

2 DOCUMENTATION

CHAPTER

ONE

INSTALLATION

This package is available through Pypi and can be installed by running the following command:

pip install aisquared

Alternatively, the latest version of the software can be installed directly from GitHub using the following command:

pip install git+https://github.com/AISquaredInc/aisquared

1.1 aisquared

1.1.1 aisquared package

1.1.1.1 Module contents

This package contains utilities to interact with the AI Squared technology stack, particularly with developing and deploying models to the AI Squared Browser Extension or other applications developed through the AI Squared JavaScript SDK.

1.1.1.2 Subpackages

aisquared.base package

Module contents

The aisquared base package contains both some basic objects that are used across the aisquared package backend and some objects which are designed to facilitate simple use cases of the technology.

Submodules

aisquared.base.BaseObject module

class aisquared.base.BaseObject.BaseObject

Bases: object

Base class used for all other classes within the aisquared package. This class is not meant to be used by any end user of this package, but is rather used throughout this package as a parent class.

```
to_dict() → dict
Get the object as a dictionary
to_json() → str
Return the object as a json string
```

aisquared.base.CustomObject module

aisquared.base.rendering module

Some allowed configuration parameters - not meant to be directly called by the end user

aisquared.base.stages module

Some allowed configuration parameters - not designed to be directly called by the user

aisquared.config package

Module contents

The aisquared.config subpackage contains utilities and objects for packaging aisquared configuration steps and models.

For in-depth examples of how to build out .air files using the utilities and classes in this library, please visit our GitHub repository at https://github.com/AISquaredInc/airFiles

Subpackages

aisquared.config.analytic package

Module contents

The aisquared.config.analytic subpackage contains objects for packaging individual analytics.

Submodules

aisquared.config.analytic.DeployedAnalytic module

Bases: BaseObject

Interaction with a remote analytic

Example usage:

```
>>> import aisquared
     >>> analytic = aisquared.config.analytic.DeployedAnalytic(
         'analytic_url',
          'text'
     >>> analytic.to_dict()
     {'className': 'DeployedAnalytic',
     'params': {'url': 'analytic_url',
     'inputType': 'text',
     'secret': 'request',
     'header': None}}
     property header
     property input_type
     property secret
     to\_dict() \rightarrow dict
         Get the object as a dictionary
     property url
aisquared.config.analytic.DeployedModel module
class aisquared.config.analytic.DeployedModel.DeployedModel(url: str, input_type: str, secret: str =
                                                                  'request', header: Optional[dict] =
                                                                  None)
     Bases: BaseObject
     Interaction with a remote model
     Example usage:
     >>> import aisquared
     >>> analytic = aisquared.config.analytic.DeployedModel(
         'model_url',
         'text'
     >>> analytic.to_dict()
     {'className': 'DeployedModel',
     'params': {'url': 'model_url',
     'inputType': 'text',
     'secret': 'request',
     'header': None}}
     property header
     property input_type
     property secret
     to_dict() \rightarrow dict
          Get the config object as a dictionary
```

property url

aisquared.config.analytic.LocalAnalytic module

Bases: BaseObject

Interaction with an analytic (lookup table) saved to the local file system

Example usage:

```
property all
property input_type
property path
to_dict() → dict
    Get the configuration object as a dictionary
```

aisquared.config.analytic.LocalModel module

```
class aisquared.config.analytic.LocalModel.LocalModel(path: str, input_type: str)
```

Bases: BaseObject

Interaction with a model currently saved to the local file system

Example usage:

```
>>> import aisquared
>>> analytic = aisquared.config.analytic.LocalModel(
    'model_path',
    'text'
)
>>> analytic.to_dict()
{'className': 'LocalModel',
    'params': {'path': 'model_path',
    'inputType': 'text'}}
```

property input_type

```
property path \label{eq:configuration} \textbf{to\_dict}() \to \text{dict} Get the configuration object as a dictionary
```

aisquared.config.analytic.ReverseMLWorkflow module

Bases: BaseObject

Interaction with a ReverseML CSV stored in S3

Example usage:

```
>>> import aisquared
>>> analytic = aisquared.config.analytic.ReverseMLWorkflow(
    'bucket_name',
    'file_name',
    'column_name',
    'text'
)
>>> analytic.to_dict()
{'className': 'ReverseMLWorkflow',
    'params': {'bucket': 'bucket_name',
    'fileNames': ['file_name'],
    'inputType': 'text',
    'column': 'column_name',
    'period': None,
    'secret': ''}}
```

```
property bucket

property column

property filenames

property filter_by_columns

property filter_type

property input_type

property period

property secret

to_dict() → dict

Get the configuration object as a dictionary
```

aisquared.config.feedback package

Module contents

The aisquared.config.feedback subpackage contains objects for configuring feedback in aisquared models.

Submodules

aisquared.config.feedback.BinaryFeedback module

Feedback for binary classification

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.feedback.BinaryFeedback(['class1', 'class2'])
>>> my_obj.to_dict()
{'className': 'BinaryFeedback', 'params': {'labelMap': ['class1', 'class2']}}
```

```
property label_map
```

```
to\_dict() \rightarrow dict
```

Return the object as a dictionary

aisquared.config.feedback.ModelFeedback module

${\bf class} \ {\bf aisquared.config.feedback.ModelFeedback.ModelFeedback}$

Bases: BaseObject

Feedback object for questions and answers for an individual model.

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.feedback.ModelFeedback()
>>> my_obj.add_question(
    'How is the model performing?',
    choices = ['very poorly', 'poorly', 'neutral', 'well', 'very well']
)
>>> my_obj.add_question(
    'Any additional feedback?',
    'text'
)
>>> my_obj.to_dict()
{'className': 'ModelFeedback',
'params': {'questions': [{'question': 'How is the model performing?',
'answerType': 'singleChoice',
'choices': ['very poorly', 'poorly', 'neutral', 'well', 'very well']},
{'question': 'Any additional feedback?', 'answerType': 'text'}]}
```

```
add_question(question: str, answer_type: str = 'singleChoice', choices: list = []) Add a question to be asked.
```

Parameters

- **question** (*str*) The question to be asked.
- answer_type (str (default 'singleChoice')) One of either 'singleChoice', 'multiChoice', or 'text'
- choices (list (default [])) The choices to be provided, if answer_type is 'single-Choice' or 'multiChoice'

```
to_dict() \rightarrow dict
```

Return the object as a dictionary

aisquared.config.feedback.MulticlassFeedback module

class aisquared.config.feedback.MulticlassFeedback.MulticlassFeedback(label_map: list)

Bases: BaseObject

Feedback for multiclass classification

Example Usage:

property label_map

```
to\_dict() \rightarrow dict
```

Return the object as a dictionary

aisquared.config.feedback.QualitativeFeedback module

 ${\bf class} \ {\bf aisquared.config.feedback.Qualitative Feedback.Qualitative Feedback}$

Bases: BaseObject

Feedback object for questions and answers for individual predictions.

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.feedback.QualitativeFeedback()
>>> my_obj.add_question('Any additional feedback?', 'text')
>>> my_obj.to_dict()
{'className': 'QualitativeFeedback',
'params': {'questions': [{'question': 'Any additional feedback?',
'answerType': 'text'}]}}
```

```
add\_question(question: str, answer\_type: str = 'singleChoice', choices: list = []) Add a question to be asked.
```

Parameters

- **question** (*str*) The question to be asked.
- answer_type (str (default 'singleChoice')) One of either 'singleChoice', 'multiChoice', or 'text'
- **choices** (*list* (*default* [])) The choices to be provided, if *answer_type* is 'single-Choice' or 'multiChoice'

```
to_dict() \rightarrow dict
```

Return the object as a dictionary

aisquared.config.feedback.RegressionFeedback module

class aisquared.config.feedback.RegressionFeedback.RegressionFeedback

Bases: BaseObject

Feedback for regression

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.feedback.RegressionFeedback()
>>> my_obj.to_dict()
{'className': 'RegressionFeedback', 'params': {}}
```

```
to_dict() \rightarrow dict
```

Return the object as a dictionary

aisquared.config.feedback.SimpleFeedback module

class aisquared.config.feedback.SimpleFeedback

```
Bases: BaseObject
```

Simple thumbs-up/thumbs-down feedback for predictions

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.feedback.SimpleFeedback()
>>> my_obj.to_dict()
{'className': 'SimpleFeedback', 'params': {}}
```

```
\textbf{to\_dict()} \rightarrow dict
```

Return the object as a dictionary

aisquared.config.harvesting package

Module contents

The aisquared.config.harvesting subpackage contains objects for configuring harvesting of data.

Submodules

aisquared.config.harvesting.lmageHarvester module

```
class aisquared.config.harvesting.ImageHarvester.ImageHarvester(how: str = 'all')
     Bases: BaseObject
     Object to harvest images
     Example usage:
     >>> import aisquared
     >>> my_obj = aisquared.config.harvesting.ImageHarvester()
     >>> my_obj.to_dict()
     {'className': 'ImageHarvester', 'params': {'how' : 'all'}}
     property how
```

```
to_dict() \rightarrow dict
```

Get the configuration object as a dictionary

aisquared.config.harvesting.InputHarvester module

```
class aisquared.config.harvesting.InputHarvester.InputHarvester(input_type: str = 'text',
                                                                          max_length: Optional[int] =
                                                                          None, features: Optional[list] =
                                                                          None)
```

Bases: BaseObject

Object to harvest user-input text

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.harvesting.InputHarvester()
>>> my_obj.to_dict()
{'className': 'InputHarvester',
'params': {'inputType': 'text', 'maxLength': None, 'features': None}}
```

```
property features
property input_type
property max_length
to\_dict() \rightarrow dict
     Get the configuration object as a dictionary
```

aisquared.config.harvesting.QueryParameterHarvester module

```
class aisquared.config.harvesting.QueryParameterHarvester.QueryParameterHarvester(query_keys:
```

Union[str, list], url_locations: Union[str, list], attributes: *Union[str,* list])

Bases: BaseObject

Harvester for Query Parameters

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.harvesting.QueryParameterHarvester(
    'test_key',
    'test_url',
    'test_attribute'
>>> my_obj.to_dict()
{'className': 'QueryParameterHarvester',
'params': {'queryKeys': ['test_key'],
'urlLocations': ['test_url'],
'attributes': ['test_attribute']}}
```

```
property attributes
```

```
property query_keys
```

```
to\_dict() \rightarrow dict
```

Get the configuration object as a dictionary

property url_locations

aisquared.config.harvesting.TextHarvester module

```
class aisquared.config.harvesting.TextHarvester.TextHarvester(how: str = 'all', regex:
```

Optional[str] = None, flags: str ='gu', $body_only$: bool = False, keywords: Optional[Union[str, list]] = None, limit: Optional[int] = None)

Bases: BaseObject Object to harvest text

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.harvesting.TextHarvester(
```

(continues on next page)

(continued from previous page)

```
how = 'all',
  body_only = True
)
>>> my_obj.to_dict()
{'className': 'TextHarvester',
  'params': {'how': 'all',
  'regex': None,
  'flags': 'gu',
  'bodyOnly': True,
  'limit': None}}
```

```
property body_only
property flags
property how
property limit
property regex
to_dict() → dict
Get the configuration object as a dictionary
```

aisquared.config.postprocessing package

Module contents

The aisquared.config.postprocessing subpackage contains objects for configuring how predictions are postprocessed.

Submodules

aisquared.config.postprocessing.BinaryClassification module

class aisquared.config.postprocessing.BinaryClassification.BinaryClassification(label_map: list, threshold: float = 0.5)

Bases: BaseObject

Postprocesssing configuration object for binary classification

Example usage

```
>>> import aisquared
>>> my_obj = aisquared.config.postprocessing.BinaryClassification(
    ['class1', 'class2']
)
>>> my_obj.to_dict()
{'className': 'BinaryClassification',
    'params': {'labelMap': ['class1', 'class2'], 'threshold': 0.5}}
```

```
property label_map
property threshold

to_dict() → dict
Get the configuration object as a dictionary
```

aisquared.config.postprocessing.MulticlassClassification module

Bases: BaseObject

Postprocessing configuration object for multiclass classification

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.postprocessing.MulticlassClassification(
    ['class1', 'class2', 'class3']
)
>>> my_obj.to_dict()
{'className': 'MulticlassClassification',
'params': {'labelMap': ['class1', 'class2', 'class3']}}
```

property label_map

```
to\_dict() \rightarrow dict
```

Get the configuration object as a dictionary

aisquared.config.postprocessing.ObjectDetection module

class aisquared.config.postprocessing.ObjectDetection.ObjectDetection($label_map: list, threshold: float = 0.5$)

Bases: BaseObject

Postprocessing configuration object for object detection

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.postprocessing.ObjectDetection(
    ['class1', 'class2', 'class3']
)
>>> my_obj.to_dict()
{'className': 'ObjectDetection',
    'params': {'labelMap': ['class1', 'class2', 'class3'], 'threshold': 0.5}}
```

```
property label_map
```

property threshold

```
to\_dict() \rightarrow dict
```

Get the configuration object as a dictionary

aisquared.config.postprocessing.Regression module

class aisquared.config.postprocessing.Regression.Regression(min: Optional[Union[int, float]] = None, max: Optional[Union[int, float]] = None, round: bool = False)

Bases: BaseObject

Postprocessing configuration object for Regression

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.postprocessing.Regression(
        10,
        100
)
>>> my_obj.to_dict()
{'className': 'Regression', 'params': {'min': 10, 'max': 100, 'round': False}}
```

```
property max
property min
property round
to_dict() → dict
Get the configuration object as a dictionary
```

aisquared.config.preprocessing package

Module contents

The aisquared.config.preprocessing subpackage contains utilities to configure the preprocessing of data in the data pipeline. It contains

three separate subpackages, aisquared.config.preprocessing.text, aisquared.config.preprocessing.image, and aisquared.config.preprocessing.tabular, which configure the preprocessing of different types of data.

Subpackages

aisquared.config.preprocessing.image package

Module contents

The aisquared.config.preprocessing.image subpackage contains objects for configuring image preprocessing.

Submodules

aisquared.config.preprocessing.image.lmagePreprocessing module

```
class aisquared.config.preprocessing.image.ImagePreprocessing.ImagePreprocesser(steps: Op-
                                                                                           tional[list]
                                                                                           = None)
     Bases: BaseObject
     Preprocesser object for image data
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
          aisquared.config.preprocessing.image.AddValue(255.0)
     )
     add_step(step)
          Add a step to the preprocesser object
     property step_dict
     \textbf{to\_dict()} \rightarrow dict
          Get the configuration object as a dictionary
aisquared.config.preprocessing.image.Steps module
class aisquared.config.preprocessing.image.Steps.AddValue(value: Union[int, float])
     Bases: BaseObject
     Preprocessing step to add a value to all pixels in an image
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
          aisquared.config.preprocessing.image.AddValue(255.0)
     \textbf{to\_dict()} \rightarrow dict
          Get the configuration object as a dictionary
     property value
class aisquared.config.preprocessing.image.Steps.ConvertToColor(color: str)
     Bases: BaseObject
     Preprocessing step to convert images to a color scheme
```

Example usage:

```
>>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
         aisquared.config.preprocessing.image.ConvertToColor('RGB')
     property color
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
class aisquared.config.preprocessing.image.Steps.DivideValue(value: Union[int, float])
     Bases: BaseObject
     Preprocessing step to divide all pixels in an image by a value
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
         aisquared.config.preprocessing.image.DivideValue(255.0)
     )
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
     property value
class aisquared.config.preprocessing.image.Steps.MultiplyValue(value: Union[int, float])
     Bases: BaseObject
     Preprocessing step to multiply all pixels in an image by a value
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
         aisquared.config.preprocessing.image.MultiplyValue(2.0)
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
     property value
class aisquared.config.preprocessing.image.Steps.Resize(size: list, method: str = 'bilinear',
                                                              preserve_aspect_ratio: bool = False)
     Bases: BaseObject
     Preprocessing step to resize an image
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
                                                                                    (continues on next page)
```

```
(continued from previous page)
```

```
aisquared.config.preprocessing.image.Resize([100, 100])
     property method
     property preserve_aspect_ratio
     property size
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
class aisquared.config.preprocessing.image.Steps.SubtractValue(value: Union[int, float])
     Bases: BaseObject
     Preprocessing step to subtract a value from all pixels in an image
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.image.ImagePreprocesser()
     >>> preprocesser.add_step(
          aisquared.config.preprocessing.image.SubtractValue(255.0)
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
     property value
```

aisquared.config.preprocessing.tabular package

Module contents

The aisquared.config.preprocessing.tabular subpackage contains objects for preprocessing tabular data.

Submodules

aisquared.config.preprocessing.tabular.Steps module

```
class aisquared.config.preprocessing.tabular.Steps.DropColumn(column: int)
    Bases: BaseObject
    Drop a column from tabular data
    Example usage:
```

```
>>> import aisquared
>>> preprocesser = aisquared.config.preprocessing.tabular.TabularPreprocesser()
>>> preprocesser.add_step(
    aisquared.config.preprocessing.tabular.DropColumn(
    3
```

(continues on next page)

```
(continued from previous page)
         )
     )
     property column
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
class aisquared.config.preprocessing.tabular.Steps.MinMax(mins: list, maxs: list, columns:
                                                                  Optional[list] = None
     Bases: BaseObject
     Min-Max Scaling preprocessing step
     Min-Max Scaling takes all associated columns and maps values relative to the minimum and maximum values
     of the training data.
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.tabular.TabularPreprocesser()
     >>> preprocesser.add_step(
         aisquared.config.preprocessing.tabular.MinMax(
              [0, 1.1, 2],
              [0.2, 14, 18.3]
         )
     property columns
     property maxs
     property mins
     to\_dict() \rightarrow dict
          Get the configuration object as a dictionary
class aisquared.config.preprocessing.tabular.Steps.OneHot(column: int, values: list)
     Bases: BaseObject
     One Hot encoding preprocessing step
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.tabular.TabularPreprocesser()
     >>> preprocesser.add_step(
         aisquared.config.preprocessing.tabular.OneHot(
              ['one', 'two', 'three']
         )
```

property column

)

```
to_dict() \rightarrow dict
```

Get the configuration object as a dictionary

property values

Bases: BaseObject

Z-Score normalization preprocessing step

Z-Score normalization takes each supplied column value, subtracts that column's provided mean, and divides by the provided standard deviation.

Example usage:

```
>>> import aisquared
>>> preprocesser = aisquared.config.preprocessing.tabular.TabularPreprocesser()
>>> preprocesser.add_step(
    aisquared.config.preprocessing.tabular.ZScore(
        [0, 1, 2],
        [0.2, 0.4, 0.6]
    )
)
```

```
property columns
```

property means

property stds

 $to_dict() \rightarrow dict$

Get the configuration object as a dictionary

aisquared.config.preprocessing.tabular.TabularPreprocessing module

class aisquared.config.preprocessing.tabular.TabularPreprocessing.TabularPreprocesser(steps:

Optional[list]

riorital rist

None)

Bases: BaseObject

Preprocesser object for tabular data

Example usage:

Example usage:

```
>>> import aisquared
>>> preprocesser = aisquared.config.preprocessing.tabular.TabularPreprocesser()
>>> preprocesser.add_step(
    aisquared.config.preprocessing.tabular.ZScore(
        [0, 1, 2],
        [0.2, 0.4, 0.6]
)
)
```

```
add_step(step)
    Add a step to the preprocesser object

to_dict()
    Get the configuration object as a dictionary
```

aisquared.config.preprocessing.text package

Module contents

The aisquared.config.preprocessing.text subpackage contains objects for preprocessing text data.

Submodules

aisquared.config.preprocessing.text.Steps module

```
class aisquared.config.preprocessing.text.Steps.ConvertToCase(lowercase: bool = True)
    Bases: BaseObject
```

Text preprocessing object to convert inputs to all lowercase or all uppercase

Example usage:

```
>>> import aisquared
>>> preprocesser = aisquared.config.preprocessing.text.TextPreprocesser()
>>> preprocesser.add_step(
    aisquared.config.preprocessing.text.ConvertToCase()
)
```

property lowercase

```
\textbf{to\_dict()} \rightarrow dict
```

Get the configuration object as a dictionary

Bases: BaseObject

Text preprocessing object to convert tokens to integer vocabularies

Example usage:

```
>>> import aisquared
>>> preprocesser = aisquared.config.preprocessing.text.TextPreprocesser()
>>> preprocesser.add_step(
    aisquared.config.preprocessing.text.ConvertToVocabulary(
    {
        'test': 3,
        'vocabulary': 4
    }
```

(continues on next page)

(continued from previous page))) property max_vocab property oov_character property start_character $to_dict() \rightarrow dict$ Get the configuration object as a dictionary property vocabulary **class** aisquared.config.preprocessing.text.Steps.**PadSequences**(pad_character: int = 0, length: int = 128, $pad_location$: str = 'post', $truncate_location: str = 'post')$ Bases: BaseObject Text preprocessing object to pad sequences Example usage: >>> import aisquared >>> preprocesser = aisquared.config.preprocessing.text.TextPreprocesser() >>> preprocesser.add_step(aisquared.config.preprocessing.text.PadSequences() property length property pad_character property pad_location $to_dict() \rightarrow dict$ Get the configuration object as a dictionary property truncate_location **class** aisquared.config.preprocessing.text.Steps.**RemoveCharacters**(remove_digits: bool = True, remove_punctuation: bool = True) Bases: BaseObject Preprocessing step to remove characters from text Example usage: >>> import aisquared >>> preprocesser = aisquared.config.preprocessing.text.TextPreprocesser()

```
property remove_digits
```

>>> preprocesser.add_step(

aisquared.config.preprocessing.text.RemoveCharacters()

```
property remove_punctuation
     to\_dict() \rightarrow dict
          Get the configuration object as a dictionary
class aisquared.config.preprocessing.text.Steps.Tokenize(split_sentences: bool = False,
                                                                   split \ words: bool = True, token \ pattern:
                                                                   str = \langle x08 \rangle \langle w \rangle \langle w + \langle x08' \rangle
     Bases: BaseObject
     Preprocessing Step to tokenize text
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.text.TextPreprocesser()
     >>> preprocesser.add_step(
          aisquared.config.preprocessing.text.Tokenize()
     property split_sentences
     property split_words
     to\_dict() \rightarrow dict
          Get the configuration object as a dictionary
     property token_pattern
class aisquared.config.preprocessing.text.Steps.Trim
     Bases: BaseObject
     Text preprocessing class to trim whitespace from text
     Example usage:
     >>> import aisquared
     >>> preprocesser = aisquared.config.preprocessing.text.TextPreprocesser()
     >>> preprocesser.add_step(
          aisquared.config.preprocessing.text.Trim()
     to\_dict() \rightarrow dict
          Get the configuration object as a dictionary
aisquared.config.preprocessing.text.TextPreprocessing module
class aisquared.config.preprocessing.text.TextPreprocessing.TextPreprocesser(steps:
                                                                                           Optional[list] =
                                                                                           None)
     Bases: BaseObject
     Preprocesser object for natural language
     Example usage:
```

aisquared.config.rendering package

Module contents

The aisquared.config.rendering subpackage contains objects for configuring how rendering of predictions is to occur.

Submodules

aisquared.config.rendering.BarChartRendering module

chart_name: str,
container_id: str,
prediction_name_key: str,
prediction_value_key: str,
prediction_name_value:
str, display_legend: bool,
legend_icon: str,
labels_key: Optional[str]
= None, width: str =
'auto', height: str = 'auto',
xOffset: str = '0', yOffset:
str = '0', labels:
Optional[list] = None,
consolidate_rows: bool =
True, css_params:
Optional[dict] = None)

Bases: BaseObject

Rendering class for rendering a Bar Chart

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.BarChartRendering(
   'my_label',
   'my_id',
```

(continues on next page)

(continued from previous page)

```
'my_bar_chart',
    'my_container_id',
    'name',
    'value',
    'name_value',
    True,
    'circle'
>>> my_obj.to_dict()
{'className': 'BarChartRendering',
    'label': 'my_label',
    'params': {'id': 'my_id',
    'chartName': 'my_bar_chart',
    'containerId': 'my_container_id',
    'displayLegend': True,
    'legendIcon': 'circle',
    'width': 'auto',
    'height': 'auto',
    'x0ffset': '0',
    'yOffset': '0',
    'datasource': [{'labels': None,
        'labelsKey': None,
        'consolidateRows': True,
        'predictionNameKey': 'name',
        'predictionValueKey': 'value',
        'predictionNameValue': 'name_value'}]}}
```

$to_dict() \rightarrow dict$

Get the configuration object as a dictionary

aisquared.config.rendering.ContainerRendering module

```
\textbf{class} \  \, \textbf{aisquared.config.rendering.ContainerRendering.ContainerRendering.ContainerRendering.} \\ \textbf{class} \  \, \textbf{aisquared.config.rendering.ContainerRendering.ContainerRendering.} \\ \textbf{class} \  \, \textbf{containerRendering.ContainerRendering.ContainerRendering.} \\ \textbf{class} \  \, \textbf{containerRendering.ContainerRendering.ContainerRendering.} \\ \textbf{class} \  \, \textbf{class} \\ \textbf{class} \  \, \textbf{class} \\ \textbf{class} \  \, \textbf{class} \\ \textbf{class} \  \, \textbf{class} \\ \textbf{class} \  \, \textbf{cla
```

query_selector: str,
position: str =
'absolute',
static_position:
Optional[str] = None,
width: str = 'auto',
height: str = 'flex',
xOffset: str = '0',
yOffset: str = '0',
orientation: str =
'column', css_params:
Optional[dict] = None)

Bases: BaseObject

Rendering for a container

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.ContainerRendering(
   'my container',
    'myContainerID',
    "[data-id='tabpanel-general']"
)
>>> my_obj.to_dict()
{'className': 'ContainerRendering',
'label': 'my container',
'params': {'id': 'myContainerID',
'width': 'auto',
'height': 'auto',
'display': 'flex',
'xOffset': '0',
'yOffset': '0',
'position': 'absolute',
'orientation': 'column',
'querySelector': "[data-id='tabpanel-general']",
'staticPosition': None}}
property display
property height
property id
property label
property orientation
property position
property query_selector
property static_position
to_dict() \rightarrow dict
    Get the configuration object as a dictionary
property width
property xOffset
property yOffset
```

aisquared.config.rendering.DashboardReplacementRendering module

 $\textbf{class} \ a is quared. config. rendering. Dashboard Replacement Rendering. Dashboard Rendering.$

```
str,
where_replace
str
=
",
la-
bel:
str
=
")"
```

Bases: BaseObject

Rendering for dashboard replacement

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.DashboardReplacementRendering(
    'test_anchor_selector'
)
>>> my_obj.to_dict()
{'className': 'DashboardReplacementRendering',
    'label': '',
    'params': {'anchorSelector': 'test_anchor_selector', 'whereReplace': ''}}
```

```
property anchor_selector
property label

to_dict() → dict
    Get the configuration object as a dictionary
property where_replace
```

aisquared.config.rendering.DocumentRendering module

class aisquared.config.rendering.DocumentRendering.DocumentRendering(prediction_key: str =

'className', words: Optional[Union[list, dict, str]] = None, documents: Optional[Union[list, dict, str]] = None,include_probability: bool = False, probability_key: str = 'probability', $underline_color: str =$ 'blue', classes: Optional[list] = None,threshold_key: Optional[str] = None,threshold_value: Optional[Union[int, float]] = None

Bases: BaseObject

Object which dictates how to render predictions on entire documents

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.DocumentRendering()
>>> my_obj.to_dict()
{'className': 'DocumentRendering',
    'params': {'predictionKey': 'className',
    'words': None,
    'documents': None,
    'includeProbability': False,
    'probabilityKey': 'probability',
    'underlineColor': 'blue',
    'classes': None,
    'thresholdKey': None,
    'thresholdValue': None}}
```

```
property classes

property documents

property include_probability

property prediction_key

property probability_key

property threshold_key

property threshold_value

to_dict() → dict

Get the configuration object as a dictionary

property underline_color

property words
```

aisquared.config.rendering.DoughnutChartRendering module

class aisquared.config.rendering.DoughnutChartRendering.DoughnutChartRendering(label: str, id:

```
str,
chart_name:
str,
container_id:
str, predic-
tion_name_key:
str, predic-
tion_value_key:
str, predic-
tion_name_value:
str, dis-
play_legend:
bool,
legend_icon:
str,
labels_key:
Optional[str]
= None.
width: str =
'auto', height:
str = 'auto',
xOffset: str =
'0', yOffset:
str = '0',
labels: Op-
tional[list] =
None,
consoli-
date_rows:
bool = True,
css_params:
Op-
tional[dict]
= None)
```

Bases: BaseObject

Rendering class for rendering a Doughnut Chart

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.DoughnutChartRendering(
    'my_label',
    'my_id',
    'my_doughnut_chart',
    'my_container_id',
    'name',
    'value',
    'name_value',
    True,
    'circle'
)
>>> my_obj.to_dict()
```

(continues on next page)

(continued from previous page)

```
{'className': 'DoughnutChartRendering',
'label': 'my_label',
'params': {'id': 'my_id',
'chartName': 'my_doughnut_chart',
'containerId': 'my_container_id',
'displayLegend': True,
'legendIcon': 'circle',
'width': 'auto',
'height': 'auto',
'x0ffset': '0',
'yOffset': '0',
'datasource': [{'labels': None,
    'labelsKey': None,
    'consolidateRows': True,
    'predictionNameKey': 'name',
    'predictionValueKey': 'value',
    'predictionNameValue': 'name_value'}]}}
```

```
to\_dict() \rightarrow dict
```

Get the configuration object as a dictionary

aisquared.config.rendering.FilterRendering module

Bases: BaseObject

Object which dictates how predictions are to be passed to downstream analytics

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.FilterRendering(
    'inputs',
    'key',
    'gt',
    0.2
)
>>> my_obj.to_dict()
{'className': 'FilterRendering',
    'params': {'source': 'inputs', 'key': 'key', 'qualifier': 'gt', 'value': 0.2}}
```

```
property key
property qualifier
property source
to_dict() → dict
    Get the configuration object as a dictionary
property value
```

aisquared.config.rendering.HTMLTagRendering module

Bases: BaseObject
Rendering for HTML tags

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.HTMLTagRendering(
    'my HTML tag',
    'MyHTMLTagRenderingID',
    'MyContainerID',
    'Example Text',
    'extra_tag'.
    'append',
    'name_key',
    'value_key',
    'name_value'
>>> my_obj.to_dict()
{'className': 'HTMLTagRendering',
'label': 'my HTML tag',
'params': {'id': 'MyHTMLTagRenderingID',
'containerId': 'MyContainerID',
'htmlContent': 'Example Text',
'extraContentTag': 'extra_tag',
'injectionAction': 'append',
'predictionNameKey': 'name_key',
'predictionValueKey': 'value_key'
'predictionNameValue': 'name_value',
'content': ''}}
```

 $to_dict() \rightarrow dict$

Return the configuration object as a dictionary

aisquared.config.rendering.lmageRendering module

Get the configuration object as a dictionary

```
class aisquared.config.rendering.ImageRendering.ImageRendering(color: str = 'blue', thickness: str =
                                                                       '5', placement: str = 'bottomleft',
                                                                       include\_probability: bool = False,
                                                                       badge\_color: str = 'white',
                                                                       font_color: str = 'black', font_size:
                                                                       str = '5', classes: Optional[list] =
                                                                       None, threshold_key:
                                                                       Optional[str] = None,
                                                                       threshold_value:
                                                                       Optional[Union[int, float]] =
                                                                       None)
     Bases: BaseObject
     Object which dictates how to render images
     Example usage:
     >>> import aisquared
     >>> my_obj = aisquared.config.rendering.ImageRendering()
     >>> my_obj.to_dict()
     {'className': 'ImageRendering',
     'params': {'color': 'blue',
     'thickness': '5',
     'placement': 'bottomleft',
     'includeProbability': False,
     'badgeColor': 'white',
     'fontColor': 'black',
     'fontSize': '5',
     'classes': None,
     'thresholdKey': None,
     'thresholdValue': None}}
     property badge_color
     property classes
     property color
     property font_color
     property font_size
     property include_probability
     property placement
     property thickness
     property threshold_key
     property threshold_value
     to\_dict() \rightarrow dict
```

= True, css_params: Optional[dict] = None)

aisquared.config.rendering.LineChartRendering module

class aisquared.config.rendering.LineChartRendering.LineChartRendering(label: str, id: str, chart_name: str, container_id: str, prediction_name_key: str, prediction value key: str, prediction_name_value: str, display_legend: bool, legend_icon: str, labels_key: str, width: str = 'auto', height: str= 'auto', xOffset: str ='0', yOffset: str = '0', labels: Optional[list] = None, consolidate_rows: bool

Bases: BaseObject

Rendering class for rendering a Line Chart

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.LineChartRendering(
    'my_label',
    'my_id',
    'my_line_chart',
    'my_container_id',
    'name',
    'value',
    'name_value',
    True,
    'circle',
    'labels'
>>> my_obj.to_dict()
{'className': 'LineChartRendering',
'label': 'my_label',
'params': {'id': 'my_id',
'chartName': 'my_line_chart',
'containerId': 'my_container_id',
'displayLegend': True,
'legendIcon': 'circle',
'width': 'auto',
'height': 'auto',
'x0ffset': '0',
'yOffset': '0',
'datasource': [{'labels': None,
    'labelsKey': 'labels',
```

(continues on next page)

(continued from previous page)

```
'consolidateRows': True,
'predictionNameKey': 'name',
'predictionValueKey': 'value',
'predictionNameValue': 'name_value'}]}}
```

```
to\_dict() \rightarrow dict
```

Get the configuration object as a dictionary

aisquared.config.rendering.ObjectRendering module

```
class aisquared.config.rendering.ObjectRendering.ObjectRendering(color: str = 'blue', thickness: str = '5', placement: str = 'bottomleft', include_probability: bool = False, badge_color: str = 'white', font_color: str = 'black', font_size: str = '5')
```

Bases: BaseObject

Object which dictates how to render object detection in images

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.ObjectRendering()
>>> my_obj.to_dict()
{'className': 'ObjectRendering',
'params': {'color': 'blue',
'thickness': '5',
'placement': 'bottomleft',
'includeProbability': False,
'badgeColor': 'white',
'fontColor': 'black',
'fontSize': '5'}}
```

```
property badge_color
property color
property font_color
property font_size
property include_probability
property placement
property thickness
to_dict() → dict
Get the configuration object as a dictionary
```

aisquared.config.rendering.PieChartRendering module

container_id: str,
prediction_name_key: str,
prediction_value_key: str,
prediction_name_value:
str, display_legend: bool,
legend_icon: str,
labels_key: Optional[str]
= None, width: str =
'auto', height: str = 'auto',
xOffset: str = '0', yOffset:
str = '0', labels:
Optional[list] = None,
consolidate_rows: bool =
True, css_params:
Optional[dict] = None)

Bases: BaseObject

Rendering class for rendering a Pie Chart

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.PieChartRendering(
        'my_label',
        'my_id',
        'my_doughnut_chart',
        'my_container_id',
        'name',
        'value',
        'name_value',
        True,
        'circle'
    )
>>> my_obj.to_dict()
{'className': 'PieChartRendering',
'label': 'my_label',
'params': {'id': 'my_id',
'chartName': 'my_doughnut_chart',
'containerId': 'my_container_id',
'displayLegend': True,
'legendIcon': 'circle',
'width': 'auto',
'height': 'auto',
'xOffset': '0',
'yOffset': '0',
'datasource': [{'labels': None,
    'labelsKey': None,
    'consolidateRows': True,
    'predictionNameKey': 'name',
    'predictionValueKey': 'value',
```

(continues on next page)

(continued from previous page)

```
'predictionNameValue': 'name_value'}]}}
     to_dict() \rightarrow dict
          Get the configuration object as a dictionary
aisquared.config.rendering.SOSRendering module
class aisquared.config.rendering.SOSRendering.SOSRendering(can_toggle: bool, label: str = ")
     Bases: BaseObject
     Rendering of an SOS dashboard
```

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.SOSRendering(True)
>>> my_obj.to_dict()
{'className': 'SOSRendering', 'label': '', 'params': {'canToggle': True}}
```

```
property can_toggle
property label
to\_dict() \rightarrow dict
     Get the configuration object as a dictionary
```

aisquared.config.rendering.TableRendering module

```
class aisquared.config.rendering.TableRendering(label: str, id: str, container_id:
                                                                       str, prediction_name_key: str,
                                                                       prediction_value_key: str,
                                                                       prediction name values: str,
                                                                       table\_name: str = ", css\_params:
                                                                       Optional[dict] = None)
```

Bases: BaseObject

Class for rendering tables

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.TableRendering(
    'my table',
    'MyTableID',
    'MyContainerID',
    'name_key',
    'value_key',
    'name_values'
>>> my_obj.to_dict()
{'className': 'TableRendering',
'label': 'my table',
```

(continues on next page)

(continued from previous page)

```
'params': {'id': 'MyTableID',
'containerId': 'MyContainerID',
'predictionNameKey': 'name_key',
'predictionValueKey': 'value_key',
'predictionNameValues': 'name_values',
'tableName': ''}}
```

```
to\_dict() \rightarrow dict
```

Get the configuration object as a dictionary

aisquared.config.rendering.WordRendering module

```
class aisquared.config.rendering.WordRendering(word_list: str = 'input', result_key:

Optional[str] = None, content_key:

Optional[str] = None, badge_shape:

str = 'star', badge_color: str = 'blue',

classes: Optional[list] = None,

threshold_key: Optional[str] = None,

threshold_value:

Optional[Union[int, float]] = None,

threshold_value:

threshold_value:
```

Bases: BaseObject

Object for rendering badges on individual words

Example usage:

```
>>> import aisquared
>>> my_obj = aisquared.config.rendering.WordRendering()
>>> my_obj.to_dict()
{'className': 'WordRendering',
'params': {'wordList': 'input',
'resultKey': None,
'contentKey': None,
'badgeShape': 'star',
'badgeColor': 'blue',
'classes': None,
'thresholdKey': None,
'thresholdValue': None
'position': 'after'}}
```

```
property badge_color
property badge_shape
property classes
property content_key
property position
property result_key
```

```
property threshold_key
property threshold_value

to_dict() → dict
    Get the configuration object as a dictionary
property word_list
```

Submodules

aisquared.config.GraphConfiguration module

```
class aisquared.config.GraphConfiguration.GraphConfiguration(name: str, stage: str = 'experimental', version:

Optional[int] = None, description:

str = ", mlflow\_uri: Optional[str] = None, mlflow\_user: Optional[str] = None, mlflow\_token: Optional[str] = None, owner: Optional[str] = None, owner: Optional[str] = owner: owner
```

Bases: BaseObject

Configuration object for deploying a set of processing steps and/or analytics as a dependency graph

add_node(step: BaseObject, dependencies: Optional[Union[int, list]] = None) \rightarrow int Add a node to the configuration graph

Parameters

- **step** (aisquared configuration step) The step to add
- **dependencies** (*int*, *list of int*, *or None*) The ids of nodes which must be run before the added node

Returns

node id – The integer id of the node that is added

Return type

int

property auto_run

```
compile(filename: Optional[str] = None, dtype: Optional[str] = None) \rightarrow None
```

Compile the object into a '.air' file, which can then be dragged and dropped into applications using the AI Squared JavaScript SDK

Parameters

- **filename** (path-like or None (default None)) Filename to compile to. If None, defaults to '{NAME}.air', where {NAME} is the name of the analytic
- **dtype** (*str or None* (*default None*)) The datatype to use for the model weights when using a Keras model. If None, defaults to 'float32'

property description

None, stage: str = 'experimental', version: Optional[int] = None, description: str = ", mlflow_uri: Optional[str] = None, mlflow_user: Optional[str] = None, mlflow_token: Optional[str] = None, owner: Optional[str] = None, url: str = '*',

auto_run: bool = False,
documentation_link: str = ")

```
property documentation_link
     get_filenames() \rightarrow list
          Get filenames for all models in the configuration
     property mlflow_token
     property mlflow_uri
     property mlflow_user
     property name
     property owner
     property stage
     to_dict() \rightarrow dict
          Get the object as a dictionary
     property url
     property version
aisquared.config.ModelConfiguration module
class aisquared.config.ModelConfiguration.ModelConfiguration(name: str, harvesting_steps:
                                                                         Optional[Union[BaseObject, list]] =
                                                                        None, preprocessing_steps:
                                                                         Optional[Union[BaseObject, list]] =
                                                                        None, analytic:
                                                                         Optional[Union[BaseObject, list]] =
                                                                        None, postprocessing_steps:
                                                                         Optional[Union[BaseObject, list]] =
                                                                        None, rendering_steps:
                                                                         Optional[Union[BaseObject, list]] =
                                                                        None, feedback_steps:
                                                                        Optional[Union[BaseObject, list]] =
```

```
Bases: BaseObject
Configuration object for deploying a model or analytic
property analytic
property analytic_dict
property auto_run
```

```
compile(filename: Optional[str] = None, dtype: Optional[str] = None) <math>\rightarrow None
```

Compile the object into a '.air' file, which can then be dragged and dropped into applications using the AI Squared JavaScript SDK

Parameters

- **filename** (path-like or None (default None)) Filename to compile to. If None, defaults to '{NAME}.air', where {NAME} is the name of the analytic
- **dtype** (*str or None* (*default None*)) The datatype to use for the model weights. If None, defaults to 'float32'

```
property description
property documentation_link
property feedback_dict
property feedback_steps
get_model_filenames() \rightarrow list
    Get filenames for all models in the configuration
property harvester_dict
property harvesting_steps
property mlflow_token
property mlflow_uri
property mlflow_user
property name
property owner
property postprocesser_dict
property postprocessing_steps
property preprocesser_dict
property preprocessing_steps
property render_dict
property rendering_steps
property stage
\textbf{to\_dict()} \rightarrow dict
    Get the object as a dictionary
property url
property version
```

aisquared.logging package

Module contents

The aisquared.logging subpackage contains utilities for performing experiments within aisquared.

This functionality is inhereted from MLFlow. Please see the MFLow documentatation at https://mlflow.org.

aisquared.platform package

Module contents

Utilities for interacting with the AI Squared Platform.

The primary class within this subpackage is the *AISquaredPlatformClient* class, which has the capabilities to interact with much of the functionality in the AI Squared platform. For more information about this class, please see its documentation.

Submodules

aisquared.platform.AISquaredPlatformClient module

 ${\bf class} \ {\bf aisquared.platform.AISquaredPlatformClient. {\bf AISquaredPlatformClient} (use_port:\ bool = False)}$

Bases: object

Client for interacting with the AI Squared platform programmatically

When using the client for the first time, it is important to run the *client.login()* method. When doing so, the client will ask for any required information interactively.

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> # If you have never logged in before, run the following code:
>>> client.login()
>>> # Test connection
>>> client.test_connection()
True
```

 $\label{local_str} \textbf{add_users_to_group}(\textit{group_id: str, user_ids: list, port: int} = 8086, \textit{use_port: Optional[bool]} = \textit{None}) \rightarrow \\ \textbf{bool}$

Add users to a group

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.add_users_to_group('group_id', ['user_id_1', 'user_id_2'])
True
```

Parameters

- **group_id** (str) The group to add the users to
- user_ids (list of str) The IDs of the users to add

- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success – Returns True if operation was successful

Return type

bool

property base_url: str

The base URL associated with the client

create_group($display_name: str, role_id: str, port: int = 8086, use_port: Optional[bool] = None) <math>\rightarrow$ dict Create a group in the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.create_group(
    'group display name',
    'role_id'
)
*dictionary containing group information*
```

Parameters

- **display_name** (*str*) The display name of the group
- role_id (str) The role ID for the group
- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

group_info - Metadata about the created group

Return type

dict

 $create_user(user_name: str, given_name: str, family_name: str, email: str, role_id: str, active: bool = True, middle_name: Optional[str] = None, company_id: Optional[str] = None, password: Optional[str] = None, port: int = 8085, use_port: Optional[bool] = None) <math>\rightarrow$ dict

Create a user within the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.create_user(
   'user name',
   'given_name',
   'family_name',
   'user_email',
   'role_id'
```

(continues on next page)

(continued from previous page)

```
)
*Dictionary with user information*
```

Parameters

- user_name (str) The display name of the user
- **given_name** (str) The user's first name
- family_name (str) The user's last name
- **email** (*str*) The user's email
- role_id (str) The ID of the role to be given to the user
- active (bool (default True)) Whether the user is active
- middle_name (str or None (default None)) The user's middle name
- company_id (str or None (default None)) The user's company ID
- password (str or None (default None)) The user's password
- **port** (*int* (*default* 8085)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

user data - Metadata about the user

Return type

dict

 $delete_group(group_id, port=8086, use_port: Optional[bool] = None) \rightarrow bool$

Delete a group from the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.delete_group('group_id')
True
```

Parameters

- **group_id** (str) The ID of the group to delete
- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port(bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

result - Returns True if successful

Return type

bool

 $delete_model(id: str, port: int = 8080, use_port: Optional[bool] = None) \rightarrow bool$

Delete a model

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.delete_model('model_id')
True
```

Parameters

- id (str) The ID for the model
- **port** (*int* (*default* 8080)) The API port for the model. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success - Whether the action was successful

Return type

bool

 $delete_user(user_id: str, port: int = 8085, use_port: Optional[bool] = None) \rightarrow bool$

Delete a user from the system

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.delete_user('user_id')
True
```

Parameters

- user_id (str) The user's ID
- **port** (*int* (*default* 8085)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

result - Returns True if the call is successful

Return type

bool

 $\texttt{get_group}(\textit{group_id: str}, \textit{port: int} = 8086, \textit{use_port: Optional[bool]} = \textit{None}) \rightarrow \textit{dict}$

Retrieve information about a group

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_group('group_id')
*dictionary containing group data*
```

Parameters

- **group_id** (str) The ID of the group requested
- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

group info – The information about the group

Return type

dict

get_group_id_by_name($group_name: str, port: int = 8083, use_port: Optional[bool] = None) <math>\rightarrow$ str Get the ID of a group by searching for its display name

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_group_id_by_name('Group Name')
*group_id*
```

Parameters

- **group_name** (str) The display name of the group
- **port** (*int* (*default* 8083)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

group_id - The ID of the group

Return type

str

 $get_model(id: str, port: int = 8080, use_port: Optional[bool] = None) \rightarrow dict$ Retrieve a model configuration

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_model('model_id')
*JSON Response including model data and metadata*
```

Parameters

- **id** (*str*) The ID for the model
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

model – Metadata about the model coupled with the model's configuration information

Return type

dictionary

get_model_id_by_name($model_name: str, port: int = 8080, use_port: Optional[bool] = None) <math>\rightarrow$ str Retrieve a model's ID using the name of the model

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_model_id_by_name('my_awesome_model')
*model_id*
```

Parameters

- **model_name** (*str*) The name of the model
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

model_id - The model's ID

Return type

stı

get_role_id_by_role_name($role_name$: str, port: $int = 8086, use_port$: Optional[bool] = None) \rightarrow str Get the ID of a role by searching for its display name

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_role_id_by_role_name('Role Name')
*role_id*
```

Parameters

- $role_name(str)$ The name of the role
- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

role id – The ID of the role

Return type

str

 $\texttt{get_user}(user_id: str, port: int = 8085, use_port: Optional[bool] = None) \rightarrow \text{dict}$

Retrieve a user's information from the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_user('user_id')
*dictionary with results*
```

Parameters

- user_id (str) The ID of the user
- **port** (*int* (*default* 8085)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

user_info – The information about the user

Return type

dict

get_user_id_by_name($name: str, port: int = 8080, use_port: Optional[bool] = None) <math>\rightarrow$ str Get a user's ID from their display name

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_user_id_by_name('User Name')
*user_id*
```

Parameters

- **name** (*str*) The display name of the user
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port(bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

id – The ID of the user

Return type

str

property headers

Headers used for authentication with the AI Squared Platform

```
list_group_users(group\_id: str, as\_df: bool = True, port: int = 8083, use\_port: Optional[bool] = None) 
 <math>\rightarrow Union[DataFrame, dict]
```

List users in a group

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_group_users('group_id')
*DataFrame with results*
```

Parameters

- **group_id** (str) The ID for the group
- **as_df** (bool (default True)) Whether to return the response as a pandas DataFrame
- **port** (*int* (*default* 8083)) The API port for the call. This can be handled automatically by the platform ALB

• use_port (bool or None (default None)) – Whether to use port in URL formatting. If None, defaults to class value

Returns

users – The response from the API

Return type

pandas DataFrame or dictionary

 $\label{list_groups} \begin{subarray}{l} \textbf{list_groups}(max_count: int = 100, as_df: bool = True, port: int = 8083, use_port: Optional[bool] = None) \rightarrow Union[DataFrame, dict] $\end{subarray}$

List all groups

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_groups()
*DataFrame with results*
```

Parameters

- max_count (int (default 100)) The maximum number of groups to return
- as_df (bool (default True)) Whether to return the result as a pandas DataFrame
- **port** (*int* (*default* 8083)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

groups – The response from the API

Return type

pandas DataFrame or dictionary

 $\label{list_model_feedback} \textbf{(} model_id: str, limit: int = 10, as_df: bool = True, port: int = 8080, use_port: \\ Optional[bool] = None) \rightarrow \textbf{Union[dict, DataFrame]}$

List feedback on a model

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_model_feedback('model_id')
*DataFrame with Results*
```

Parameters

- model_id (str) The ID of the model
- limit (int (default 10)) The maximum number of feedback items to return
- **port** (*int* (*default* 8080)) The API port to use. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

feedback - The feedback

Return type

dict or pandas DataFrame

list_model_prediction_feedback($model_id: str, as_df: bool = True, port: int = 8080, use_port: Optional[bool] = None) <math>\rightarrow$ Union[dict, DataFrame]

List all feedback for a model

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_model_prediction_feedback('model_id')
*DataFrame with Results*
```

Parameters

- model_id (str) The ID of the model requested
- as_df (bool (default True)) Whether to return the results as a pandas DataFrame
- **port** (*int* (*default* 8080)) The API port to use. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

results – The results from the platform

Return type

dict or pandas DataFrame

list_model_usage_metrics($model_id$: str, period: str = 'hourly', as_df : bool = True, port: int = 8080, use_port : Optional[bool] = None) \rightarrow Union[dict, DataFrame]

Get usage metrics for a model

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_model_usage_metrics('model_id')
*DataFrame with results*
```

Parameters

- $model_id(str)$ The ID of the model
- **period** (str (default 'hourly')) The period to group metrics into
- as_df (bool (default True)) Whether to return results as a pandas DataFrame
- port (int (default 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

results – The results from the platform

Return type

pandas DataFrame or dict

list_model_users($id: str, as_df: bool = True, port: int = 8080, use_port: Optional[bool] = None) <math>\rightarrow$ Union[DataFrame, dict]

List users for a model

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_model_users('model_id')
*DataFrame with results*
```

Parameters

- **id** (*str*) The ID for the model
- as_df (bool (default True)) Whether to return the response as a Pandas DataFrame
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

model_users - The users for the model

Return type

pandas DataFrame or dictionary

 $\label{list_models} \textbf{list_models}(\textit{as_df: bool} = \textit{True}, \textit{port: int} = 8080, \textit{use_port: Optional[bool]} = \textit{None}) \rightarrow \text{Union[DataFrame, dict]}$

List models within the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_models()
*DataFrame with results*
```

Parameters

- as_df (bool (default True)) Whether to return the response as a pandas DataFrame
- **port** (*default None*) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

models – The models

Return type

pandas DataFrame or dictionary

```
list_prediction_feedback(prediction\_id: str, as\_df: bool = True, port: int = 8080, use\_port: Optional[bool] = None) <math>\rightarrow Union[DataFrame, dict]
```

List prediction feedback given a prediction ID

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_prediction_feedback('prediction_id')
*DataFrame with results*
```

Parameters

- prediction_id (str) The prediction ID
- as_df (bool (default True)) Whether to return the results as a pandas DataFrame
- **port** (*int* (*default* 8080)) The API port to use. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

results – The results from the platform

Return type

pandas DataFrame or dict

 $\label{eq:list_roles} \textbf{list_roles}(\textit{as_df: bool} = \textit{True, port: int} = 8086, \textit{use_port: Optional[bool]} = \textit{None}) \rightarrow \text{Union[DataFrame, dict]}$

List the roles available in the platform

Example usage:

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_roles()
*DataFrame with results*
```

Parameters

- as_df (bool (default True)) Whether to return the results as a pandas DataFrame
- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

roles – The roles

Return type

pandas DataFrame or dict

list_user_usage_metrics($user_id$: str, period: str = 'hourly', as_df : bool = True, port: int = 8080, use_port : Optional[bool] = None) \rightarrow Union[dict, DataFrame]

Get usage metrics for a user

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.get_user_usage_metrics('user_id')
*DataFrame with results*
```

Parameters

- user_id (str) The ID of the user
- **period** (str (default 'hourly')) The period to group metrics into
- as_df (bool (default True)) Whether to return results as a pandas DataFrame
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

results – The results from the platform

Return type

pandas DataFrame or dict

 $\label{list_users} \textbf{(max_count: int = 100, as_df: bool = True, port: int = 8080, use_port: Optional[bool] = None)} \\ \rightarrow \textbf{Union[DataFrame, dict]}$

List all users

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.list_users()
*DataFrame with results*
```

Parameters

- max_count (int (default 100)) The maximum number of users to return
- as_df (bool (default True)) Whether to return the data as a Pandas DataFrame
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

users – The response from the API

Return type

pandas DataFrame or dictionary

```
login(url: Optional[str] = None, port: int = 8080, username: Optional[str] = None, password: Optional[str] = None, use_port: Optional[bool] = None) <math>\rightarrow None
```

Log in to the platform programmatically. If no url, username, or password are provided, logs in interactively

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.login()
Enter URL: https://platform.squared.ai
Enter Username: your.email@your_domain.com
Enter Password: <hidden>
```

Parameters

- url (str or None (default None)) The URL for the platform API
- port (int or None (default 8080)) The API port for the call. This can be handled automatically by the platform ALB
- username (str or None (default None)) The username
- password (str or None (default None)) The password
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

property password: str

The password associated with the client

remove_users_from_group($group_id: str, user_ids: list, port: int = 8086, use_port: Optional[bool] = None) <math>\rightarrow$ bool

Remove users from a group

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.remove_users_from_group('group_id', ['user_id_1', 'user_id_2'])
True
```

Parameters

- **group_id** (str) The ID of the group
- user_ids (list of str) The IDs of the users to remove
- port (int (default = 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port(bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success - Returns True if successful

Return type

bool

 $share_model_with_group(model_id: str, group_id: str, port: int = 8080, use_port: Optional[bool] = None) \rightarrow bool$

Share a model with a group

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.share_model_with_group('model_id', 'group_id')
True
```

Parameters

- model_id (str) The ID for the model to be shared
- **group_id** (str) The ID for the group to be shared with. This can be handled automatically by the platform ALB
- **port** (*int* (*default* 8080)) The API port to use. This can be handled automatically by the platform ALB

• use_port (bool or None (default None)) – Whether to use port in URL formatting. If None, defaults to class value

Returns

success - Returns True if successful

Return type

bool

 $share_model_with_user(model_id: str, user_id: str, port: int = 8080, use_port: Optional[bool] = None) \rightarrow bool$

Share a model with a user

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.share_model_with_user('model_id', 'user_id')
True
```

Parameters

- model_id (str) The ID for the model
- user_id (str) The ID for the user
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success – Whether the action was successful

Return type

bool

 $test_connection(port: int = 8080, use_port: Optional[bool] = None) \rightarrow bool$

Test whether there is a healthy connection to the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.test_connection()
True
```

Parameters

- port (int (default 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success - True if connection was successful

Return type

bool

property token: str

The token associated with the client

```
unshare_model_with_group(model\_id: str, group\_id: str, port: int = 8080, use\_port: Optional[bool] = None) <math>\rightarrow bool
```

Unshare a model with a group

```
>>> import aisquared
>>> client = aisquared.client.AISquaredPlatformClient()
>>> client.unshare_model_with_group('model_id', 'group_id')
True
```

Parameters

- model_id (str) The ID of the model
- **group_id** (str) The ID of the group
- **port** (*int* (*default* 8080)) The API port to use. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success - Returns True if successful

Return type

bool

```
unshare\_model\_with\_user(model\_id: str, user\_id: str, port: int = 8080, use\_port: Optional[bool] = None) \rightarrow bool
```

Unshare a model with a user

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.unshare_model_with_user('model_id', 'user_id')
True
```

Parameters

- $model_id(str)$ The ID for the model
- user_id (str) The ID for the user
- **port** (*int* (*default* 8080)) The API port for the call. This can be handled automatically by the platform ALB
- use_port(bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success – Whether the action was successful

Return type

bool

 $update_group(group_id: str, display_name: str, role_id: str, port: int = 8086, use_port: Optional[bool] = None) \rightarrow bool$

Update information about a group

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.update_group(
    'group_id',
    'group display name',
    'role_id'
)
True
```

Parameters

- **group_id** (str) The ID of the group to update
- **display_name** (*str*) The display name of the group
- role_id (str) The ID of the role for the group
- **port** (*int* (*default* 8086)) The API port for the call. This can be handled automatically by the platform ALB
- use_port(bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success - Returns True if successful

Return type

bool

update_user($user_id: str, user_name: str, given_name: str, family_name: str, email: str, role_id: str, active: bool = True, middle_name: Optional[str] = None, company_id: Optional[str] = None, password: Optional[str] = None, port: int = 8085, use_port: Optional[bool] = None) <math>\rightarrow$ bool

Update information about a user

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.update_user(
    'user_id',
    'user name',
    'given_name',
    'family_name',
    'user_email',
    'role_id'
)
True
```

Parameters

- **user_id** (str) The ID of the user to update
- **user_name** (*str*) The display name of the user
- **given_name** (str) The first name of the user

- **family_name** (str) The last name of the user
- email (str) The user's email
- role_id (str) The ID of the user's role
- active (bool (default True)) Whether the user is active
- middle_name (str or None (default None)) The user's middle name
- company_id (str or None (default None)) The user's company ID
- password (str or None (default None)) The user's password
- **port** (*int* (*default* 8085)) The API port for the call. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

success - Returns True if update is successful

Return type

bool

upload_model($model_file: str, port: int = 8081, use_port: Optional[bool] = None) <math>\rightarrow$ str Upload a model to the platform

```
>>> import aisquared
>>> client = aisquared.platform.AISquaredPlatformClient()
>>> client.upload_model('my_model_filename.air')
True
```

Parameters

- model_file (path or path-like) The path to the model file
- **port** (*int* (*default* 8081)) The API port to use. This can be handled automatically by the platform ALB
- use_port (bool or None (default None)) Whether to use port in URL formatting. If None, defaults to class value

Returns

successful – Whether the action was successful

Return type

bool

property use_port

property username: str

The username associated with the client

aisquared.serving package

Module contents

The aisquared serving package contains utilities to serve models to a local REST endpoint.

Here is an example of how to serve a simple keras model using these utilities:

```
>>> # Assume model is already trained and stored in memory as model
>>> from aisquared import serving
>>> serving.save_keras_model(model, 'my_model')
>>> serving.deploy_model(
    'my_model',
    'keras',
    additional_functions_file = '<optional file containing `preprocess` and_
    ¬`postprocess` functions, if applicable>'
)
App created successfullly. Serving and awaiting requests
```

And to retrieve predictions from the model:

Submodules

aisquared.serving.deploy_model module

```
aisquared.serving.deploy_model.deploy_model(saved_model: str, model_type: str, host: str = '127.0.0.1', port: int = 2244, custom_objects: Optional[dict] = None, additional_functions_file: Optional[str] = None)
```

Deploy a model to a Flask server on the specified host

Parameters

- **saved_model** (*Path-like*) The path to the saved model directory or model file
- **model_type** (*str*) The type of model
- host (str (default '127.0.0.1')) The host to deploy to
- port (int (default 2244)) The port to deploy to
- **custom_objects** (*dict or None* (*default None*)) Any custom objects to load when using a BeyondML model
- additional_functions_file (file-like or None (default None)) File name containing additional functions (which have to be named preprocess and postprocess, if created) that are used during the prediction process

```
aisquared.serving.deploy_model.load_beyondml_model(model: str, custom_objects: dict)

Load a BeyondML model with custom objects
```

aisquared.serving.get_remote_prediction module

```
aisquared.serving.get_remote_prediction.get_remote_prediction(data: Union[dict, str, ndarray, list], host: str = '127.0.0.1', port: int = 2244) \rightarrow list
```

Send data to use for prediction

Parameters

- data (dict, str, np.ndarray, or list) The data to be predicted on
- host (str (default '127.0.0.1')) The host to use
- port (int (default '2244')) The port to use

Notes

- If data is a dictionary, it is expected to already be correctly formatted
- If data is a string, it is expected to already be correctly formatted

Returns

predictions – The predictions from the deployed model

Return type

list

aisquared.utils package

Module contents

Additional utilities to use with the *aisquared* package. These utilities currently consist of two functions, the *mimic_model* and *get_model* functions. They utilize functionality that exists in our open source package BeyondML to train teacher-student models

To see in-depth examples of how to use these functions, please visit our GitHub repository at https://github.com/AISquaredInc/MimicModelExamples

Submodules

aisquared.utils.utils module

```
aisquared.utils.utils.get_model(model\_type: str, input\_shape: Union[int, tuple], num\_outputs: int, output\_activation: <math>str, size: str = 'small', vocab\_size: Union[None, int] = None)
```

Get a pre-configured model for different use cases

Parameters

- model_type (str) Either 'cv', 'nlp_embedding', or 'fc', defining the model type
- input_shape (int or tuple of int) The input shape to the model
- **num_outputs** (*int*) The output shape of the model

- output_activation (str or keras activation function) The activation of the final layer of the model
- size (str (default 'small')) One of either 'small', 'medium', or 'large'
- **vocab_size** (*str or None* (*default None*)) Size of the vocab, if model_type is 'nlp_embedding'

Returns

model – The model

Return type

TensorFlow Keras model

aisquared.utils.utils.mimic_model (trained_model: BaseEstimator, nnet: Model, training_data: ndarray, test_data: ndarray, test_labels: ndarray, problem_type: str, loss: str, metrics: Union[str, list], optimizer: str, mimic_proba: bool = False, retention: float = 0.9, batch_size: int = 32, epochs: int = 100, starting_sparsification: int = 0, max_sparsification: int = 99, sparsification_rate: int = 5) \rightarrow Model

Train a sparse neural network to mimic a scikit-learn model

Parameters

- trained_model (sklearn model) The model that is already trained
- nnet (TensorFlow keras Model) The neural network to train to mimic the trained model
- training_data (array or array-like) The input data that was used to train the trained model
- **test_data** (*array or array-like*) The input data to be used for testing
- test_labels (array or array-like) The output data used in testing
- **problem_type** (*str*) The type of problem, either 'classification' or 'regression'
- loss (str or keras loss function) The loss to use
- metrics (str, function or list of str, function) Metrics to measure
- **optimizer** (*str or keras optimizer*) The optimizer to use
- mimic_proba (bool (default False)) For classification, mimic the probability outputs
- **retention** (*float* (*default* 0.9)) The retention of performance to allow further pruning
- batch_size (int (default 32)) The batch size to use while training
- **epochs** (*int* (*default* 100)) The number of epochs (if early stopping is not met beforehand)
- starting_sparsification (int (default 0)) The starting model sparsification
- max_sparsification (int (default 99)) The maximum sparsification to allow
- sparsification_rate (int (default 5)) The sparsification rate when invoked

Returns

nnet – The trained model

Return type

TensorFlow keras Model

TWO

CHANGELOG

• Version 0.1.3

- Added *flags* parameter to *TextHarvester* using regular expression harvesting
- Deleted model_feedback parameter in ModelConfiguration object and included functionality in feedback_steps parameter
- Changed format parameter to header for both deployed analytics
- Added feedback and stages to *DocumentPredictor* and *ImagePredictor* objects
- Non-API changes for ALLOWED_STAGES
- Fixed bugs preventing Windows users from importing the package
- Updated ModelConfiguration to include url parameter
- Changed default tokenization string

• Version 0.2.0

- Moved preprocessing steps under subpackages for specific kinds of preprocessing steps
- Cleaned up documentation to render within programmatic access environments
- Added aisquared.logging subpackage

- Created InputHarvester

- * Allows for harvesting of input text, images, and tabular data
- Created the aisquared.serving subpackage, specifically the deploy_model and get_remote_prediction functions
- Created the GraphConfiguration class
- Added auto-run parameter to ModelConfiguration and GraphConfiguration classes

- Created the aisquared CLI with the following commands:

- * aisquared deploy, which deploys a model locally
- * aisquared predict, which predicts using a local JSON file
- * aisquared airfiles, which contains the subcommands list, delete, download, and upload
- Changed all classes within aisquared.config.analytic to accept 'tabular' as an input type
- Removed aisquared.logging and aisquared.remote from top-level imports
- Added round parameter to Regression postprocesser
- Removed *DocumentPredictor* and *ImagePredictor* classes

- Removed *ChainRendering* class
- Created FilterRendering class
- Altered QUALIFIERS
- Added advanced rendering parameters to rendering objects
- Removed *logging* and *remote* subpackages from top-level *aisquared* import

Version 0.2.1

- Added the S3Connector class to the analytics subpackage, which allows download of an analytic directly from S3
- Updated the documentation and added the docs subdirectory for hosting the documentation on GitHub Pages

• Version 0.2.2

- Fixed bug in to_dict method within ObjectRendering class
- Fixed bug in name of MultiplyValue step
- Fixed bug in datatype checking for text harvester
- Added body_only parameter to TextHarvester
- Added 'underline' to possible badges
- Added threshold_key and threshold_values to relevant rendering classes
- Added *Trim* text preprocessing class
- Added CustomObject in the base package to allow for creation of custom classes
- Added keyword harvesting capabilities
- Added utils subpackage with capabilities to mimic a trained sklearn model
- Small documentation changes
- Changed the required imports for the package to streamline installation process, and created two installation options *aisquared* and *aisquared*[full]

• Version 0.2.3

- Added functionality to add custom preprocessing and postprocessing functions to the model deployment pipeline
- Added all parameter to LocalAnalytic class
- Changed under-the-hood functionality of mimic model function in line with updates to BeyondML
- Altered the ReverseMLWorkflow analytic
- Added the BarChartRendering, ContainerRendering, DashboardReplacementRendering, DoughnutChartRendering, HTMLTagRendering, LineChartRendering, PieChartRendering, SOSRendering, and TableRendering rendering classes
- Added the ${\it QueryParameterHarvester}$ harvester class
- Added the *limit* parameter to the TextHarvester class

• Version 0.3.0

- Added type hinting to documentation strings
- Revamped documentation to use Sphinx

• Version 0.3.1

- Changed Python type hints to allow for backwards compatibility with older versions of Python

• Version 0.3.2

- Added functionality to the AISquaredPlatformClient
- Added top level kwargs parameter to the CustomObject class
- Added *DashboardRendering* class
- Removed 'px' from default values in ImageRendering and ObjectRendering classes
- Added functionality for creating, updating, and deleting users to AISquaredPlatformClient
- Added functionality for creating, updating, and delting groups to AISquaredPlatformClient
- Fixed bug related to requiring *auto_run* parameter to be string (fix involves casting as string)
- Altered schemas for different "Chart" Rendering classes to conform to JavaScript standards
- Streamlined the ModelConfiguration class to allow a more functional interface to build .air files
- Updated ContainerRendering class with parameters for position and static_position
- Updated across-the-board functionality of the AISquaredPlatformClient

• Version 0.3.3

- Updated functionality of the AISquaredPlatformClient to interact directly with the platform ALB
- Changed function names in support of change from MANN to BeyondML
- Added documentation surrounding global configuration objects
- Removed redundant additional dependencies

• Version 0.3.4

- Added support for custom CSS strings to appropriate rendering classes
- Refactored AISquaredPlatformClient to import functions from support files
- Fixed documentation errors for the documentation site
- Checked whether responses returned OK status code rather than 200
- Moved CustomObject to aisquared.config from aisquared.base
- Changed endpoint used to list platform users
- Fixed response behaviors where no data was returned from AISquaredPlatformClient

• Version 0.3.5

- Changed file_name parameter in ReverseMLWorkflow to file_names
- Added documentation_link parameter to ModelConfiguration class

• Version 0.3.6

- Fixed issue with type checking for *ModelConfiguration* Rendering classes
- Restricted TensorFlow version to below 2.12.0 to prevent import issues
- Added position parameter to WordRendering class
- Changed default CSS styling for rendering classes
- Changed name of all *processor* classes to *processer*

PYTHON MODULE INDEX

```
а
                                                       15
                                               aisquared.config.preprocessing, 15
aisquared, 3
                                               aisquared.config.preprocessing.image, 15
aisquared.base, 3
                                               aisquared.config.preprocessing.image.ImagePreprocessing,
aisquared.base.BaseObject, 3
aisquared.base.rendering, 4
                                               aisquared.config.preprocessing.image.Steps,
aisquared.base.stages, 4
aisquared.config.4
                                               aisquared.config.preprocessing.tabular, 18
aisquared.config.analytic, 4
                                               aisquared.config.preprocessing.tabular.Steps,
aisquared.config.analytic.DeployedAnalytic,4
aisquared.config.analytic.DeployedModel, 5
                                               aisquared.config.preprocessing.tabular.TabularPreprocessing.
aisquared.config.analytic.LocalAnalytic,6
aisquared.config.analytic.LocalModel, 6
                                               aisquared.config.preprocessing.text, 21
aisquared.config.analytic.ReverseMLWorkflow,
                                               aisquared.config.preprocessing.text.Steps, 21
                                               aisquared.config.preprocessing.text.TextPreprocessing,
aisquared.config.feedback, 8
aisquared.config.feedback.BinaryFeedback, 8
                                               aisquared.config.rendering, 24
aisquared.config.feedback.ModelFeedback, 8
                                               aisquared.config.rendering.BarChartRendering,
aisquared.config.feedback.MulticlassFeedback,
aisquared.config.feedback.QualitativeFeedback,aisquared.config.rendering.ContainerRendering,
                                               aisquared.config.rendering.DashboardReplacementRendering,
aisquared.config.feedback.RegressionFeedback,
                                               aisquared.config.rendering.DocumentRendering,
aisquared.config.feedback.SimpleFeedback, 10
aisquared.config.GraphConfiguration, 38
                                               aisquared.config.rendering.DoughnutChartRendering,
aisquared.config.harvesting, 11
aisquared.config.harvesting.ImageHarvester,
                                               aisquared.config.rendering.FilterRendering,
aisquared.config.harvesting.InputHarvester,
                                               aisquared.config.rendering.HTMLTagRendering,
aisquared.config.harvesting.QueryParameterHarvester. 31
                                               aisquared.config.rendering.ImageRendering, 32
                                               aisquared.config.rendering.LineChartRendering,
aisquared.config.harvesting.TextHarvester, 12
aisquared.config.ModelConfiguration, 39
                                               aisquared.config.rendering.ObjectRendering,
aisquared.config.postprocessing, 13
aisquared.config.postprocessing.BinaryClassification, 34
                                               aisquared.config.rendering.PieChartRendering,
aisquared.config.postprocessing.MulticlassClassification,
                                               aisquared.config.rendering.SOSRendering,36
{\tt aisquared.config.postprocessing.ObjectDetection}, squared.config.rendering. Table Rendering, 36
                                               aisquared.config.rendering.WordRendering, 37
                                               aisquared.logging, 41
aisquared.config.postprocessing.Regression,
```

```
aisquared.platform,41
aisquared.platform.AISquaredPlatformClient,
41
aisquared.serving,58
aisquared.serving.deploy_model,58
aisquared.serving.get_remote_prediction,59
aisquared.utils,59
aisquared.utils.utils,59
```

66 Python Module Index

INDEX

A	aisquared.config.feedback
$\verb"add_node"()" (a is quared. config. Graph Configuration. Graph") \\$	Configuration 8
method), 38	alsquared.config.feedback.BinaryFeedback
add_question() (aisquared.config.feedback.ModelFeedback.mo	aisquared.config.feedback.ModelFeedback
add_question() (aisquared.config.feedback.QualitativeFoundation), 9	eedback.@hdfitativeFeedback aisquared.config.feedback.MulticlassFeedback
add_step() (aisquared.config.preprocessing.image.Image method), 16	aisquared.config.feedback.QualitativeFeedback
add_step() (aisquared.config.preprocessing.tabular.Tabu method), 20	larPreprocessing.TabularPreprocesser aisquared.config.feedback.RegressionFeedback
add_step() (aisquared.config.preprocessing.text.TextPrep method), 24	processing.TextPreprocesser aisquared.config.feedback.SimpleFeedback
add users to group()	module, 10
(aisquared.platform.AISquaredPlatformClient.AI	Saisguared configuGraphConfiguration
method) 41	module, 38
AddValue (class in aisquared.config.preprocessing.image.S	_{tals} quared.config.harvesting
16	module, 11
aisquared	aisquared.config.harvesting.ImageHarvester
module, 3	module, 11
aisquared.base	aisquared.config.harvesting.InputHarvester
module, 3	module, 11
aisquared.base.BaseObject	aisquared.config.harvesting.QueryParameterHarvester
module, 3	module, 12
aisquared.base.rendering	aisquared.config.harvesting.TextHarvester
module. 4	module, 12
aisquared.base.stages	aisquared.config.ModelConfiguration
module, 4	module, 39
aisquared.config	aisquared.config.postprocessing
module, 4	module, 13
aisquared.config.analytic	aisquared.config.postprocessing.BinaryClassification
module, 4	module, 13
aisquared.config.analytic.DeployedAnalytic module, 4	${\tt aisquared.config.postprocessing.MulticlassClassification} \\ {\tt module,} \ 14$
aisquared.config.analytic.DeployedModel module, 5	<pre>aisquared.config.postprocessing.ObjectDetection module, 14</pre>
aisquared.config.analytic.LocalAnalytic module, 6	aisquared.config.postprocessing.Regression module, 15
aisquared.config.analytic.LocalModel module, 6	aisquared.config.preprocessing module, 15
aisquared.config.analytic.ReverseMLWorkflow module, 7	<pre>aisquared.config.preprocessing.image module, 15</pre>

aisquared.config.preprocessing.image.ImagePremodule, 16	pariosappsssiend).serving.deploy_model module,58
aisquared.config.preprocessing.image.Steps module, 16	aisquared.serving.get_remote_prediction module, 59
aisquared.config.preprocessing.tabular module, 18	aisquared.utils module, 59
$\begin{array}{c} {\tt aisquared.config.preprocessing.tabular.Steps} \\ {\tt module,} \ 18 \end{array}$	aisquared.utils.utils module, 59
aisquared.config.preprocessing.tabular.Tabulamodule, 20	
aisquared.config.preprocessing.text module, 21	41 all (aisquared.config.analytic.LocalAnalytic.LocalAnalytic
aisquared.config.preprocessing.text.Steps module, 21	property), 6 analytic (aisquared.config.ModelConfiguration.ModelConfiguration
aisquared.config.preprocessing.text.TextPrepromodule, 23	ocessing property), 39
aisquared.config.rendering	analytic_dict (aisquared.config.ModelConfiguration.ModelConfiguration), 39
module, 24 aisquared.config.rendering.BarChartRendering	anchor_selector(aisquared.config.rendering.DashboardReplacementReproperty), 27
module, 24 aisquared.config.rendering.ContainerRendering	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\begin{tabular}{ll} module, 25\\ ais quared.config.rendering.DashboardReplacement \\ \end{tabular}$	* -* * ·
module, 26 aisquared.config.rendering.DocumentRendering module, 27	auto_run (aisquared.config.ModelConfiguration.ModelConfiguration property), 39
aisquared.config.rendering.DoughnutChartRende.module, 28	
aisquared.config.rendering.FilterRendering	badge_color (aisquared.config.rendering.ImageRendering.ImageRenderi property), 32
module, 30 aisquared.config.rendering.HTMLTagRendering module, 31	badge_color (aisquared.config.rendering.ObjectRendering.Object
aisquared.config.rendering.ImageRendering	badge_color (aisquared.config.rendering.WordRendering.WordRendering property), 37
module, 32 aisquared.config.rendering.LineChartRendering	p. op c. 1577, c. 7
<pre>module, 33 aisquared.config.rendering.ObjectRendering module, 34</pre>	BarChartRendering (class in aisquared.config.rendering.BarChartRendering), 24
aisquared.config.rendering.PieChartRendering module, 35	base_url (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformproperty), 42
aisquared.config.rendering.SOSRendering module, 36	BaseObject (class in aisquared.base.BaseObject), 3 BinaryClassification (class in
aisquared.config.rendering.TableRendering module, 36	aisquared.config.postprocessing.BinaryClassification), 13
<pre>aisquared.config.rendering.WordRendering module, 37</pre>	BinaryFeedback (class in aisquared.config.feedback.BinaryFeedback), 8
aisquared.logging module, 41	body_only (aisquared.config.harvesting.TextHarvester.TextHarvester property), 13
aisquared.platform module, 41	bucket (aisquared.config.analytic.ReverseMLWorkflow.ReverseMLWorkflowproperty), 7
aisquared.platform.AISquaredPlatformClient module, 41	C
aisquared.serving module, 58	can_toggle(aisquared.config.rendering.SOSRendering.SOSRendering property), 36

```
classes (aisquared.config.rendering.DocumentRendering. IdedwrtentRende\)i(xij) squared.platform.AISquaredPlatformClient.AISquaredI
                                                                property), 28
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                method), 44
classes (aisquared.config.rendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.ImageRendering.
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         module
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                aisquared.serving.deploy_model), 58
                                                                property), 32
 classes (aisquared.config.rendering.WordRendering.WordRepdoyegAnalytic
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        (class
                                                                property), 37
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                aisquared.config.analytic.DeployedAnalytic), 4
color (aisquared.config.preprocessing.image.Steps.ConvertTepbergedModel
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              in
                                                                property), 17
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 aisquared.config.analytic.DeployedModel),
color (aisquared.config.rendering.ImageRendering.ImageRendering 5
                                                                                                                                                                                                                                                                                                                                                                                                            {\tt description}\ (ais quared. config. Graph Configuration. Graph Configuration
                                                                property), 32
color (aisquared.config.rendering.ObjectRendering.ObjectRenderingproperty), 38
                                                                                                                                                                                                                                                                                                                                                                                                              {\tt description} \ (a is quared. config. Model Configuration. Model Configuration
                                                                property), 34
column (aisquared.config.analytic.ReverseMLWorkflow.ReverseMLWorkflowrty), 40
                                                                                                                                                                                                                                                                                                                                                                                                              display (aisquared.config.rendering.ContainerRendering.ContainerRende
                                                                property), 7
column (aisquared.config.preprocessing.tabular.Steps.DropColumn property), 26
                                                                property), 19
                                                                                                                                                                                                                                                                                                                                                                                                            DivideValue
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              in
column (aisquared.config.preprocessing.tabular.Steps.OneHot
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                aisquared.config.preprocessing.image.Steps),
                                                                property), 19
columns (aisquared.config.preprocessing.tabular.Steps.Mindboumentation_link (aisquared.config.GraphConfiguration.GraphConfig
                                                                property), 19
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 38
columns (aisquared.config.preprocessing.tabular.Steps.ZScobocumentation_link (aisquared.config.ModelConfiguration.ModelConfig.
                                                                property), 20
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 40
compile() (aisquared.config.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.Grap
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 aisquared.config.rendering.DocumentRendering),
                                                                  method), 38
compile() (aisquared.config.ModelConfiguration.ModelConfiguration)
                                                                                                                                                                                                                                                                                                                                                                                                              {\tt documents} \ (a is quared. config. rendering. Document Render
                                                                method), 39
ContainerRendering
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 28
                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                                                                                 in
                                                                aisquared.config.rendering.ContainerRendering), DoughnutChartRendering
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            in
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              aisquared.config.rendering.DoughnutChartRendering),
content_key (aisquared.config.rendering.WordRendering.WordRendering)
                                                                property), 37
                                                                                                                                                                                                                                                                                                                                                                                                            DropColumn (class in aisquared.config.preprocessing.tabular.Steps),
ConvertToCase
                                                                                                                                                                                                                              (class
                                                                                                                                                                                                                                                                                                                                                                                in
                                                                aisquared.config.preprocessing.text.Steps),
ConvertToColor
                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                                                                                in
                                                                                                                                                                                                                                                                                                                                                                                                            features (aisquared.config.harvesting.InputHarvester.InputHarvester
                                                                aisquared.config.preprocessing.image.Steps),
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 11
                                                                                                                                                                                                                                                                                                                                                                                                              {\tt feedback\_dict}\ (a is quared. config. Model Configuration. Model Configuration) \ (a is quared. configuration) \ (a is 
ConvertToVocabulary
                                                                                                                                                                                                                                                        (class
                                                                                                                                                                                                                                                                                                                                                                                in
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 40
                                                                  aisquared.config.preprocessing.text.Steps),
                                                                                                                                                                                                                                                                                                                                                                                                              {\tt feedback\_steps}\ (a is quared. config. Model Configuration. Model Configuration) (a configuration) (a configuration) (a configuration) (b configuration) (b configuration) (configuration) (configuration
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 40
create_group() (aisquared.platform.AISquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCfieneAdisquaredPlatformCf
                                                                  method), 42
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                property), 7
\verb|create_user(|)| (a is quared. platform. A IS quared Platform Client. \textit{AdS} q \textit{log} \textit{rectPlatform} \textit{Client} \textit{log} \textit{log} \textit{rectPlatform} \textit{Client} \textit{log} \textit
                                                                method), 42
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 7
                                                                                                                                                                                                                                                                                                                                                                                                              filter_type (aisquared.config.analytic.ReverseMLWorkflow.ReverseMLW
D
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 7
                                                                                                                                                                                                                                                                                                                                                                                 in FilterRendering
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              in
DashboardReplacementRendering
                                                                                                                                                                                                                                                                                                  (class
                                                                  a is quared. config. rendering. Dashboard Replacement Rendering \ref{eq:squared.config.} quared. config. rendering. Filter Rendering), and the squared config. The sq
{\tt delete\_group()}\ (a is quared. platform. A IS quared Platform {\it Cleans A Sisquered Platform}. Text Harvester. Text Harve
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                property), 13
                                                                   method), 43
delete_model() (aisquared.platform.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredPlatformCfont.AISquaredP
                                                                  method), 43
                                                                                                                                                                                                                                                                                                                                                                                                                                                                              property), 32
```

```
font_color (aisquared.config.rendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectRendering.ObjectR
                                               property), 34
                                                                                                                                                                                                                                                                                                                                             property), 13
font_size (aisquared.config.rendering.ImageRendering.InHTMRFmdRendering
                                                                                                                                                                                                                                                                                                                                              aisquared.config.rendering.HTMLTagRendering),
                                               property), 32
 font_size(aisquared.config.rendering.ObjectRendering.ObjectRendering
                                               property), 34
G
                                                                                                                                                                                                                                                                                             id (aisquared.config.rendering.ContainerRendering.ContainerRendering
get_filenames() (aisquared.config.GraphConfiguration.GraphConfiguration) 26
                                                                                                                                                                                                                                                                                              ImageHarvester
                                                 method), 39
get_group() (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClien
                                               method), 44
                                                                                                                                                                                                                                                                                             ImagePreprocesser
                                                                                                                                                                                                                                                                                                                                                                                                                                                                        (class
get_group_id_by_name()
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
                                                 (aisquared.platform.AISquaredPlatformClient.AISquaredPlatfsquaredPlatfsquared.platform.AISquaredPlatformClient.AISquaredPlatfsquaredPlatfsquared.platform.AISquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquaredPlatfsquare
                                               method), 45
get_model() (aisquared.platform.AISquaredPlatformClienIndsqRandIndingrmClient
                                                                                                                                                                                                                                                                                                                                                                                                                                                                 (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      in
                                                                                                                                                                                                                                                                                                                                              aisquared.config.rendering.ImageRendering),
                                                method), 45
 get_model() (in module aisquared.utils.utils), 59
 get_model_filenames()
                                                                                                                                                                                                                                                                                              include_probability
                                                (a is quared. config. Model Configuration. Model Configuration\ (a is quared. config. rendering. Document Rendering. Documen
                                                                                                                                                                                                                                                                                                                                             property), 28
                                               method), 40
                                                                                                                                                                                                                                                                                             include_probability
get_model_id_by_name()
                                                 (aisquared.platform.AISquaredPlatformClient.AISquaredPlatfistHOMedra onfig.rendering.ImageRendering.ImageRendering
                                                                                                                                                                                                                                                                                                                                             property), 32
                                               method), 46
                                                                                                                                                                                                                                                                                             include_probability
get_remote_prediction()
                                                                                                                                                                                        (in
                                                                                                                                                                                                                                               module
                                                                                                                                                                                                                                                                                                                                              (aisquared.config.rendering.ObjectRendering.ObjectRendering
                                               aisquared.serving.get_remote_prediction),
                                                                                                                                                                                                                                                                                                                                             property), 34
                                                                                                                                                                                                                                                                                              input_type (aisquared.config.analytic.DeployedAnalytic.DeployedAnalyti
get_role_id_by_role_name()
                                                 (aisquared.platform.AISquaredPlatformClient.AISquaredPlatfoPArCY)เครีย
                                                                                                                                                                                                                                                                                              input_type(aisquared.config.analytic.DeployedModel.DeployedModel
                                               method), 46
\verb"get_user"() (a is quared. platform. A IS quared Platform Client. A IS quared \textit{Platform}. \Cite{Client} in \Cite{Client} 
                                                                                                                                                                                                                                                                                             input_type (aisquared.config.analytic.LocalAnalytic.LocalAnalytic
                                               method), 46
                                                                                                                                                                                                                                                                                                                                             property), 6
get_user_id_by_name()
                                                 (aisquared.platform.AISquaredPlatformClient.AISinnwtdPlupo(miQuared.config.analytic.LocalModel.LocalModel
                                                                                                                                                                                                                                                                                                                                             property), 6
                                               method), 47
                                                                                                                                                                                                                                                                                            input_type (aisquared.config.analytic.ReverseMLWorkflow.ReverseMLWo
GraphConfiguration
                                                                                                                                                                               (class
                                                                                                                                                                                                                                                                                                                                              property), 7
                                                 aisquared.config.GraphConfiguration), 38
                                                                                                                                                                                                                                                                                              input_type (aisquared.config.harvesting.InputHarvester.InputHarvester)
Н
                                                                                                                                                                                                                                                                                                                                             property), 11
harvester_dict(aisquared.config.ModelConfiguration.ModelConfiguration
                                                                                                                                                                                                                                                                                                                                              aisquared.config.harvesting.InputHarvester),
                                               property), 40
harvesting_steps(aisquared.config.ModelConfiguration.ModelConfiguration
                                               property), 40
header (aisquared.config.analytic.DeployedAnalytic.DeployedAnalytic
                                                                                                                                                                                                                                                                                              key (aisquared.config.rendering.FilterRendering.FilterRendering
                                               property), 5
{\tt header} (\it ais quared. config. analytic. Deployed Model. Deployed Model \quad property), 30
                                               property), 5
headers (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformClient
                                                                                                                                                                                                                                                                                              label (aisquared.config.rendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRendering.ContainerRenderin
                                               property), 47
height (aisquared.config.rendering.ContainerRendering.ContainerRendering), 26
                                                                                                                                                                                                                                                                                              label (aisquared.config.rendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.DashboardReplacementRendering.Da
                                               property), 26
{\bf how}\,(a is quared. config. harvesting. Image Harvester. Image Harvester)
                                                                                                                                                                                                                                                                                                                                             property), 27
                                                                                                                                                                                                                                                                                             label (aisquared.config.rendering.SOSRendering.SOSRendering
                                               property), 11
```

property), 36

```
label_map (aisquared.config.feedback.BinaryFeedback.BinaryFeedbackBinaryFeedback.BinaryFeedback.BinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBinaryFeedbackBina
                                            property), 8
label_map(aisquared.config.feedback.MulticlassFeedback.MulticlassFeedback)
                                            property), 9
                                                                                                                                                                                                                                                                               \verb"max" (a is quared. config. postprocessing. Regression. Regression
label_map (aisquared.config.postprocessing.BinaryClassification.BingroyClassification
                                            property), 13
                                                                                                                                                                                                                                                                               max_length(aisquared.config.harvesting.InputHarvester.InputHarvester
label_map (aisquared.config.postprocessing.MulticlassClassification, MylticlassClassification
                                            property), 14
                                                                                                                                                                                                                                                                               max_vocab (aisquared.config.preprocessing.text.Steps.ConvertToVocabular
label_map (aisquared.config.postprocessing.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetection.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.ObjectDetectIon.Obj
                                            property), 14
                                                                                                                                                                                                                                                                               maxs (aisquared.config.preprocessing.tabular.Steps.MinMax
length (aisquared.config.preprocessing.text.Steps.PadSequences
                                                                                                                                                                                                                                                                                                                             property), 19
                                             property), 22
                                                                                                                                                                                                                                                                               {\tt means}~(a is quared. config. preprocessing. tabular. Steps. ZS core
limit (aisquared.config.harvesting.TextHarvester.TextHarvester
                                                                                                                                                                                                                                                                                                                             property), 20
                                            property), 13
                                                                                                                                                                                                                                                                               {\tt method} \ (a is quared. config. preprocessing. image. Steps. Resize
LineChartRendering
                                                                                                                                                                      (class
                                                                                                                                                                                                                                                            in
                                                                                                                                                                                                                                                                                                                             property), 18
                                             a is quared. config. rendering. Line Chart Rendering), \\ \texttt{mimic\_model()} \ (in\ module\ a is quared. utils. utils), 60
                                                                                                                                                                                                                                                                               min (aisquared.config.postprocessing.Regression.Regression
list_group_users() (aisquared.platform.AISquaredPlatformClient,AISquaredPlatformClient
                                             method), 47
                                                                                                                                                                                                                                                                               {\tt MinMax}\,(class\,in\,a is quared.config.preprocessing.tabular.Steps),
list\_groups() (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformClient.aisquaredPlatformClient)
                                            method), 48
                                                                                                                                                                                                                                                                               {\tt mins} (aisquared.config.preprocessing.tabular.Steps.MinMax
list_model_feedback()
                                                                                                                                                                                                                                                                                                                             property), 19
                                             (aisquared.platform.AISquaredPlatformClient.AISquaredPlatfeEn/Clientared.config.GraphConfiguration.GraphConfiguration
                                             method), 48
                                                                                                                                                                                                                                                                                                                            property), 39
list_model_prediction_feedback()
                                                                                                                                                                                                                                                                               {\tt mlflow\_token}\ (a is quared. config. Model Configuration. Model Configuration) (a configuration) (a configuration) (b configuration) (b configuration) (configuration) (c
                                              (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformGlieut)
                                             method), 49
                                                                                                                                                                                                                                                                               mlflow_uri(aisquared.config.GraphConfiguration.GraphConfiguration
list_model_usage_metrics()
                                                                                                                                                                                                                                                                                                                              property), 39
                                             (a is quared. platform. AIS quared Platform Client. AIS quared Platform Squared. Config. Model Configuration. Model Configuration and the platform Client. AIS quared Pl
                                             method), 49
                                                                                                                                                                                                                                                                                                                             property), 40
list_model_users() (aisquared.platform.AISquaredPlatformfCligntselfquaredPlatformfglignaphConfiguration.GraphConfiguration
                                              method), 49
                                                                                                                                                                                                                                                                                                                            property), 39
{\tt list\_models()}\ (a is quared. platform. AlS quared Platform Client Als
                                             method), 50
                                                                                                                                                                                                                                                                                                                            property), 40
list_prediction_feedback()
                                                                                                                                                                                                                                                                               ModelConfiguration
                                                                                                                                                                                                                                                                                                                                                                                                                                                      (class
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           in
                                             (aisquared.platform.AISquaredPlatformClient.AISquaredPlatfsquaredPlatfsquared.platform.AISquaredPlatform.AISquaredPlatformClient.AISquaredPlatfsquaredPlatfsquaredPlatform.
                                                                                                                                                                                                                                                                                ModelFeedback
{\tt list\_roles()}\ (a is quared. platform. A IS quared. Platform. Client. A IS quared. Platform. A IS qu
                                              method), 51
                                                                                                                                                                                                                                                                               module
list_user_usage_metrics()
                                                                                                                                                                                                                                                                                                      aisquared, 3
                                              (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformCliente, 3
                                             method), 51
                                                                                                                                                                                                                                                                                                      aisquared.base.BaseObject, 3
list_users() (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClient.AISquaredPlatformClie
                                             method), 52
                                                                                                                                                                                                                                                                                                      aisquared.base.stages, 4
load_beyondml_model()
                                                                                                                                                                                                                                   module
                                                                                                                                                                         (in
                                                                                                                                                                                                                                                                                                      aisquared.config.4
                                              aisquared.serving.deploy_model), 58
                                                                                                                                                                                                                                                                                                      aisquared.config.analytic, 4
LocalAnalytic
                                                                                                                                                       (class
                                                                                                                                                                                                                                                           in
                                                                                                                                                                                                                                                                                                      aisquared.config.analytic.DeployedAnalytic,
                                              aisquared.config.analytic.LocalAnalytic),
                                                                                                                                                                                                                                                                                                      aisquared.config.analytic.DeployedModel,
LocalModel (class in aisquared.config.analytic.LocalModel),
                                                                                                                                                                                                                                                                                                      aisquared.config.analytic.LocalAnalytic,
login() (aisquared.platform.AISquaredPlatformClient.AISquaredPlatformClient
                                             method), 52
                                                                                                                                                                                                                                                                                                       aisquared.config.analytic.LocalModel, 6
```

lowercase (aisquared.config.preprocessing.text.Steps.ConvertToCase

```
aisquared.config.analytic.ReverseMLWorkflow,
                                                                                                                        aisquared.config.rendering.ContainerRendering,
aisquared.config.feedback, 8
aisquared.config.feedback.BinaryFeedback,
                                                                                                                        aisquared.config.rendering.DashboardReplacementRenderi
aisquared.config.feedback.ModelFeedback,
                                                                                                                        aisquared.config.rendering.DocumentRendering,
aisquared.config.feedback.MulticlassFeedback, aisquared.config.rendering.DoughnutChartRendering,
aisquared.config.feedback.QualitativeFeedback,aisquared.config.rendering.FilterRendering,
aisquared.config.feedback.RegressionFeedback, aisquared.config.rendering.HTMLTagRendering,
                                                                                                                         aisquared.config.rendering.ImageRendering,
aisquared.config.feedback.SimpleFeedback,
aisquared.config.GraphConfiguration, 38
                                                                                                                        aisquared.config.rendering.LineChartRendering,
aisquared.config.harvesting, 11
aisquared.config.harvesting.ImageHarvester,
                                                                                                                        aisquared.config.rendering.ObjectRendering,
aisquared.config.harvesting.InputHarvester,
                                                                                                                        aisquared.config.rendering.PieChartRendering,
aisquared.config.harvesting.QueryParameterHarvæistærared.config.rendering.SOSRendering,
aisquared.config.harvesting.TextHarvester,
                                                                                                                        aisquared.config.rendering.TableRendering,
          12
aisquared.config.ModelConfiguration, 39
                                                                                                                        aisquared.config.rendering.WordRendering,
aisquared.config.postprocessing, 13
aisquared.config.postprocessing.BinaryClassifiacistqioamred.logging, 41
                                                                                                                        aisquared.platform, 41
aisquared.config.postprocessing.MulticlassClassisfquantendorplatform.AISquaredPlatformClient,
aisquared.config.postprocessing.ObjectDetectionisquared.serving, 58
                                                                                                                        aisquared.serving.deploy_model, 58
aisquared.config.postprocessing.Regression,
                                                                                                                        aisquared.serving.get_remote_prediction,
aisquared.config.preprocessing, 15
                                                                                                                        aisquared.utils, 59
aisquared.config.preprocessing.image, 15
                                                                                                                        aisquared.utils.utils, 59
aisquared.config.preprocessing.image.ImagePrubprocessingssification
                                                                                                                                                                                                                              in
                                                                                                                                                                                                (class
                                                                                                                                   aisquared.config.postprocessing.MulticlassClassification),
aisquared.config.preprocessing.image.Steps,
                                                                                                              MulticlassFeedback
                                                                                                                                                                                         (class
                                                                                                                                                                                                                              in
aisquared.config.preprocessing.tabular,
                                                                                                                                   aisquared.config.feedback.MulticlassFeedback),
aisquared.config.preprocessing.tabular.StepsltiplyValue
                                                                                                                                                                                  (class
                                                                                                                                                                                                                               in
                                                                                                                                   aisquared.config.preprocessing.image.Steps),
aisquared.config.preprocessing.tabular.TabularPreprocessing,
                                                                                                               Ν
aisquared.config.preprocessing.text, 21
a is quared. config. preprocessing. text. Steps, {\tt name}\ ({\it aisquared.config.GraphConfiguration.GraphConfiguration}) and {\tt name}\ ({\it aisquared.config.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphConfiguration.GraphCon
                                                                                                                                  property), 39
a is quared. config. preprocessing. text. Text Preprocessing. descending. Model Configuration. Model Configuration and the configuration of the configurat
                                                                                                                                  property), 40
aisquared.config.rendering, 24
aisquared.config.rendering.BarChartRendering,
```

0			preproce	ssing_steps		
ObjectDetection	(class	in		(aisquared.config.Mode	elConfiguration.Mod	lelConfiguration
aisquared.config.pos	stprocessing.ObjectDete	ection	- / -	property), 40		
14				_aspect_ratio (aisquared.config.prepr	magagging imaga Star	na Pagiza
ObjectRendering	(class	in		aisquarea.conjig.prepi property), 18	rocessing.image.siep	os.kestze
aisquared.config.rend 34	dering.ObjectRendering	g),	probabil	ity_key(aisquared.co	onfig.rendering.Doc	umentRendering.Doo
OneHot (class in aisquared.com			(ps),	property), 28		
oov_character(aisquared.co	onfig.preprocessing.text	.Step	s.ConvertTo	Vocabulary		
property), 22				r (aisquared.config.ren	ıdering.FilterRendei	ring.FilterRendering
orientation(aisquared.confi			Qualitat	iveFeedback	(class	in
owner (aisquared.config.Graph property), 39	hConfiguration.GraphC	Config	guration o	aisquared.config.feedba Э	ack.QualitativeFeedi	back),
owner (aisquared.config.Mode property), 40	clConfiguration.ModelC	Config		ys (aisquared.config.h property), 12	arvesting.QueryPar	ameterHarvester.Que
				lector(aisquared.cor	nfig.rendering.Conta	iinerRendering.Cont
P				property), 26	. 1	
pad_character(aisquared.co	onfig.preprocessing.text	.Step	s. પ્રયાસકપ્રદૃત્વર ો	क्र ापुर ुterHarvester aisquared.config.harve.	(class sting QueryParamet	in erHarvester)
<pre>property), 22 pad_location(aisquared.com</pre>	nfia proprocessina text	Stans		*	sung.Quer yi aramei	erriar vesier),
property), 22	ijig.preprocessing.iexi	sieps.	_	ees		
PadSequences	(class	in	R			
=	processing.text.Steps),	.,,		equared.config.harvesti property), 13	ing.TextHarvester.Te	xtHarvester
password (aisquared.platform	ı.AISquaredPlatformCli	ient.A		ž .	config.postprocessin	g.Regression),
property), 53				15		
$\verb"path" (a is quared.config.an alytical and statements) and statements and statements are also as a second statement of the statements are also as a second statement of the statements are also as a second statement of the statements are also as a second statement of the statements are also as a second statement of the stateme$	c.LocalAnalytic.LocalA	Analy			(class	in
property), 6				aisquared.config.feedbo	ack.RegressionFeedl	pack),
path (aisquared.config.analyti	ıc.LocalModel.LocalMo	odel		10 igits (gisquared cont	Ga nranrocassina tar	t Stans PamovaChara
property), 6 period(aisquared.config.anal	lutic PavarsaMI Workfle	D a		igits(aisquared.conf Statements) 22	ig.preprocessing.iex	i.sieps.RemoveChart
property), 7	yuc.ReversemEworkjio	w.Ke		unctuation(aisquare	ed.config.preprocess	ing.text.Steps.Remov
PieChartRendering	(class	in		property), 22		
	dering.PieChartRender					
35	O .	0,,		aisquared.platform.AI	SquaredPlatformCli	ent.AISquaredPlatfo
placement (aisquared.config.	rendering.ImageRende1	ring.I				
property), 32			RemoveCh		(class	in
placement (aisquared.config.)	rendering.ObjectRende	ring.	ObjectRend	eisqu ared.config.prepro	ocessing.text.Steps),	
property), 34	I i C i i D	, .	المداد ما محمد	22 30+ (1=22====1 = == 1 = == 6 =	M - 1-1C	. M - 1 - 1C C
position (aisquared.config.reproperty), 26	endering.ContainerRend	derin		жепиенцу агеа.conпg property), 40	моаевСоппдиганоп	.MoaetConfiguranoi
position (aisquared.config.re	endering.WordRenderin	g.Wo			onfig.ModelConfigu	ration.ModelConfigu
property), 37			1	property), 40		
postprocesser_dict(aisque	ared.config.ModelConfi	igura			g.preprocessing.imag	ge.Steps),
property), 40				17 ou (giaguaned config n	an danin a WandDan d	onina WondDondonina
postprocessing_steps	odelConfiguration.Mod	alCo.		ey (aisquared.config.re	enaering.wora k enae	ering.wora k enaering
property), 40	vaeiConjiguraiion.moa	eiCoi	<i>nıguranon p</i> ReverseM	LWorkflow	(class	in
prediction_key(aisquared.o	config.rendering.Docun	nentR			`	
property), 28	J.G. 1 1 1 1 1 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		,	7		**
preprocesser_dict(aisquare	red.config.ModelConfig	urati	o n.Mndeli Cs	aufgnecheiomfig.postproc	cessing.Regression.I	Regression
property), 40			1	property), 15		

S	TextHar		(class	in
secret (aisquared.config.analytic.DeployedAnalytic.Deployerty), 5	oyedAnaly TextPre	_{tte} isquared.config.harve processer	esting.TextHarvester), (class	12 in
secret (aisquared.config.analytic.DeployedModel.Deploy		aisquared.config.prepr 23		-
secret (aisquared.config.analytic.ReverseMLWorkflow.Reproperty), 7	ev erseMEW	Sskfqisquared.config.re property), 32	ndering.ImageRenderi	ing.ImageRendering
share_model_with_group()	thickne	ss (aisquared.config.re	ndering.ObjectRender	ing.ObjectRendering
(aisquared.platform.AISquaredPlatformClient.A method), 53	<i>ISquaredP</i> thresho	1d (aisquared.config.po	estprocessing.BinaryC	lassification.BinaryC
<pre>share_model_with_user()</pre>	+hnocho	property), 14	estava aggina OhiaatD	ataatian OhiaatDataa
(aisquared.platform.AISquaredPlatformClient.A method), 54		property), 14		
SimpleFeedback (class in	thresho	ld_key (aisquared.con property), 28	ng.renaering.Docume	nikenaering.Docume
aisquared.config.feedback.SimpleFeedback), 10	thresho	ld_key (aisquared.con property), 32	fig.rendering.ImageRe	endering.ImageRende
size (aisquared.config.preprocessing.image.Steps.Resize property), 18	thresho	ld_key (aisquared.con property), 37	fig.rendering.WordRen	ıdering.WordRenderi
SOSRendering (class in aisquared.config.rendering.SOSRendering), 36	thresho	ld_value (aisquared.c	onfig.rendering.Docur	mentRendering.Docu
$\verb source (a is quared. config. rendering. Filter Rendering. Filter) $	rRendering	, property), 28 ld_value (aisquared.c	onfia randarina Imaga	Pondorina ImagoPor
property), 30 split_sentences (aisquared.config.preprocessing.text.S		_	onjig.renaering.image	Kenuering.ImageKen
property), 23	thresho	ia_vaiue (<i>aisquarea.c</i>	onfig.rendering.WordI	Rendering.WordRend
<pre>split_words (aisquared.config.preprocessing.text.Steps.7 property), 23</pre>	<i>Tokenize</i> to_dict		e.BaseObject.BaseObj	ect
stage (aisquared.config.GraphConfiguration.GraphConfiguration), 39	<i>guration</i> to_dict	method), 3 () (aisquared.config.an	alytic.DeployedAnaly	tic.DeployedAnalytic
stage (aisquared.config.ModelConfiguration.ModelConfiguration), 40	g <i>uration</i> to_dict	method), 5 () (aisquared.config.an	alytic.DeployedMode	l.DeployedModel
$\verb start_character (a is quared. config. preprocessing. text. Since the configuration of th$	teps.Conve	r M61 V8Edbūlary () (aisquared.config.an	alvtic.LocalAnalvtic.I	LocalAnalytic
property), 22 static_position(aisquared.config.rendering.Container	rRendering			
property), 26 stds (aisquared.config.preprocessing.tabular.Steps.ZScore	e	method), 7		
property), 20	to_dict	() (aisquared.config.an	alytic.ReverseMLWor	kflow.ReverseMLWor
step_dict(aisquared.config.preprocessing.image.ImageIproperty), 16	το_αιсτ	() (aisquarea.conjig.jee	edback.BinaryFeedbac	ck.BinaryFeedback
<pre>step_dict(aisquared.config.preprocessing.text.TextPrep</pre>	rocessing.T to_dict	EXPTEProcesser () (aisquared.config.fee	edback.ModelFeedbac	k.ModelFeedback
SubtractValue (class in		method), 9		
aisquared.config.preprocessing.image.Steps), 18	to_dict	() (aisquared.config.fee method), 9	edback.MulticlassFeed	lback.MulticlassFeed
T	to_dict	() (aisquared.config.fee method), 10	edback.QualitativeFee	dback.QualitativeFee
TableRendering (class in aisquared.config.rendering.TableRendering),	to_dict	() (aisquared.config.fee method), 10	edback.RegressionFeed	dback.RegressionFee
36	to_dict	() (aisquared.config.fee	edback.SimpleFeedbac	ck.SimpleFeedback
TabularPreprocesser (class in aisquared.config.preprocessing.tabular.Tabular.	Preprocessi	method), 10 (aisquared.config.G	raphConfiguration.Gr	aphConfiguration
20		memou j, s		
test_connection() (aisquared.platform.AISquaredPlatform), 54		method), 11 () (aisquared.config.ha		
	to_uict	C) (aisquarea.conjig.na	a vesiing.mpuimui vesi	er.mpuman vester

method), 11	method), 25
to_dict() (aisquared.config.harvesting.QueryParameterHttovedicatt	Qù (x) Pquancel & chlifig vestal ering. ContainerRendering. ContainerRen
method), 12	method), 26
to_dict() (aisquared.config.harvesting.TextHarvester.TextHarvester.TextHarvester.Text	(t) (aisquared.config.rendering.DashboardReplacementRendering
method), 13	method), 27
to_dict() (aisquared.config.ModelConfiguration.ModelConfiguration	6n (aisquared.config.rendering.DocumentRendering.DocumentRen
method), 40	method), 28
to_dict() (aisquared.config.postprocessing.BinaryClassificationids	
method), 14	method), 30
to_dict() (aisquared.config.postprocessing.MulticlassClassofication	
method), 14	method), 30
to_dict() (aisquared.config.postprocessing.ObjectDetection ObjectDetection Obj	
method), 14	method), 31
to_dict() (aisquared.config.postprocessing.Regression.Regressiont	
method), 15	method), 32
to_dict() (aisquared.config.preprocessing.image.ImagePreprodiest	
method), 16	method), 34
to_dict() (aisquared.config.preprocessing.image.Steps.Adtl\addict	
method), 16	method), 34
to_dict() (aisquared.config.preprocessing.image.Steps.ComoedTot	
method), 17	method), 36
to_dict() (aisquared.config.preprocessing.image.Steps.DivioleWinhte	
method), 17	method), 36
to_dict() (aisquared.config.preprocessing.image.Steps.Mtdapdived	
method), 17	method), 37
to_dict() (aisquared.config.preprocessing.image.Steps.Retigedict	
method), 18	method), 38
to_dict() (aisquared.config.preprocessing.image.Steps.Subtrajtson	
method), 18	method), 4
to_dict() (aisquared.config.preprocessing.tabular.Steps.Drokenlum	
method), 19	property), 54
to_dict() (aisquared.config.preprocessing.tabular.Steps.MiolkAnx_p	
method), 19	property), 23
to_dict() (aisquared.config.preprocessing.tabular.Steps.OFmekkntiz	
method), 19	23
to_dict() (aisquared.config.preprocessing.tabular.Steps. ZSrine (cla	ass in aisquared config preprocessing text Steps).
method), 20	23
to_dict() (aisquared.config.preprocessing.tabular.Tabular Preproc	
method), 21	property), 22
to_dict() (aisquared.config.preprocessing.text.Steps.ConvertToCas	
method), 21	
to_dict() (aisquared.config.preprocessing.text.Steps.ConvertdeVoc	abulary or (sign and south a nord spin a Doormant Dondoning Door
method), 22	
to_dict() (aisquared.config.preprocessing.text.Steps.PadSequences	property), 28
method), 22	
to_dict() (aisquared.config.preprocessing.text.Steps.RemoveChara	(aisquared.platform.AISquaredPlatformClient.AISquaredPlatform
1 1 00	· · · · · · · · · · · · · · · · · · ·
to_dict() (aisquared.config.preprocessing.text.Steps.Tokenize	_model_with_user()
method), 23	(aisquared.platform.AISquaredPlatformClient.AISquaredPlatform
to_dict() (aisquared.config.preprocessing.text.Steps.Trimupdate_	method), 55
method), 23	yr oup() (aisquarea.piaiJorm.AiSquareaPiaiJormClient.AiSquarea
to_dict() (aisquared.config.preprocessing.text.TextPreprograme_1	method), 55
d. A) 24	aser (9 (asquarea.piayorm.A1Squarear tatjormCtient.A1Squareat

method), 24 method), 56 to_dict() (aisquared.config.rendering.BarChartRendering.BarChartRendering

method), 24

```
upload_model() (aisquared.platform.AISquaredPlatformClient
                           method), 57
                                                                                                                                                                     yOffset (aisquared.config.rendering.ContainerRendering.ContainerRende
url (aisquared.config.analytic.DeployedAnalytic.DeployedAnalytic
                                                                                                                                                                                              property), 26
                           property), 5
url (aisquared.config.analytic.DeployedModel.DeployedModel
                           property), 5
ZScore (class in aisquared.config.preprocessing.tabular.Steps), url (aisquared.config.GraphConfiguration.GraphConfiguration
                           property), 39
url (aisquared.config.ModelConfiguration.ModelConfiguration
                           property), 40
{\tt url\_locations}\ (a is quared. config. harvesting. Query Parameter Harvester. Query Parameter Harvester)
                           property), 12
{\tt use\_port}\ (ais quared.plat form. AIS quared Plat form Client. AIS qu
                           property), 57
{\tt username} \ (a is quared. platform. A IS quared Platform Client. A IS
                           property), 57
 V
value (aisquared.config.preprocessing.image.Steps.AddValue
                           property), 16
value (aisquared.config.preprocessing.image.Steps.DivideValue
                           property), 17
value (aisquared.config.preprocessing.image.Steps.MultiplyValue
                           property), 17
value (aisquared.config.preprocessing.image.Steps.SubtractValue
                           property), 18
value (aisquared.config.rendering.FilterRendering.FilterRendering
                           property), 30
values (aisquared.config.preprocessing.tabular.Steps.OneHot
                           property), 20
version (aisquared.config.GraphConfiguration.GraphConfiguration
                           property), 39
version (aisquared.config.ModelConfiguration.ModelConfiguration
                           property), 40
vocabulary (aisquared.config.preprocessing.text.Steps.ConvertToVocabulary
                           property), 22
 W
 where_replace (aisquared.config.rendering.DashboardReplacementRendering.DashboardReplacementRendering
                           property), 27
width(aisquared.config.rendering.ContainerRendering.ContainerRendering
                           property), 26
word_list(aisquared.config.rendering.WordRendering.WordRendering
                           property), 38
WordRendering
                                                                                            (class
                                                                                                                                                        in
                           aisquared.config.rendering.WordRendering),
words (aisquared.config.rendering.DocumentRendering.DocumentRendering
                           property), 28
X
xOffset (aisquared.config.rendering.ContainerRendering.ContainerRendering
                           property), 26
```