SeqMetrics

Release 2.0.0

Ather Abbas

Apr 08, 2024

CONTENTS

CHAPTER

ONE

INSTALLATION

1.1 using pip

The most easy way to install seqmetrics is using `pip`

```
pip install SeqMetrics
```

However, if you are interested in installing all dependencies of seqmetrics, you can choose to install all of them as well.

```
pip install SeqMetrics[all]
```

We can also specify the seqmetrics version that we want to install as below

```
pip install SeqMetrics==1.3.2
```

To updated the installation run

```
pip install --upgrade SeqMetrics
```

1.2 using github link

You can use github link for install SeqMetrics.

```
python -m pip install git+https://github.com/AtrCheema/SeqMetrics.git
```

1.3 using setup.py file

go to folder where repository is downloaded

```
python setup.py install
```

CHAPTER

TWO

QUICK START

2.1 RegressionMetrics

```
>>> import numpy as np
>>> from SeqMetrics import RegressionMetrics

>>> true = np.random.random((20, 1))
>>> pred = np.random.random((20, 1))

>>> er = RegressionMetrics(true, pred)

>>> for m in er.all_methods: print("{:20}".format(m)) # get names of all availabe methods

>>> er.nse()   # calculate Nash Sutcliff efficiency

>>> er.calculate_all(verbose=True)  # or calculate errors using all available methods
```

2.2 ClassificationMetrics

```
>>> import numpy as np
>>> from SeqMetrics import ClassificationMetrics

using boolean array

>>> t = np.array([True, False, False, False])
>>> p = np.array([True, True, True, True])
>>> metrics = ClassificationMetrics(t, p)
>>> accuracy = metrics.accuracy()

binary classification with numerical labels

>>> true = np.array([1, 0, 0, 0])
>>> pred = np.array([1, 1, 1, 1])
>>> metrics = ClassificationMetrics(true, pred)
>>> accuracy = metrics.accuracy()

multiclass classification with numerical labels
```

(continues on next page)

(continued from previous page)

```
>>> true = np.random.randint(1, 4, 100)
>>> pred = np.random.randint(1, 4, 100)
>>> metrics = ClassificationMetrics(true, pred)
>>> accuracy = metrics.accuracy()

You can also provide logits instead of labels.

>>> predictions = np.array([[0.25, 0.25, 0.25, 0.25],
>>>                         [0.01, 0.01, 0.01, 0.96]])
>>> targets = np.array([[0, 0, 0, 1],
>>>                     [0, 0, 0, 1]])
>>> metrics = ClassificationMetrics(targets, predictions, multiclass=True)
>>> metrics.cross_entropy()
...  0.71355817782
```

CHAPTER

THREE

METRICS

The `Metrics` class does some preprocessing of arrays if required.

3.1 Metrics

CHAPTER

FOUR

REGRESSION METRICS

4.1 Functional API

SeqMetrics also provides a functional API for all the performance metrics.

4.2 Class-Based API

CHAPTER

FIVE

CLASSIFICATION METRICS

5.1 Functional API

SeqMetrics also provides a functional API for all the performance metrics.

5.2 Class-Based API

CHAPTER

SIX

UTILITY FUNCTIONS

6.1 Utils

class SeqMetrics.utils.**plot_metrics**(*metrics: dict*, *ranges: tuple = ((0.0, 1.0), (1.0, 10), (10, 1000))*, *exclude: list | None = None*, *plot_type: str = 'bar'*, *max_metrics_per_fig: int = 15*, *show: bool = True*, *save: bool = False*, *save_path: str = ''*, ***kwargs*)

Plots the metrics given as dictionary as radial or bar plot between specified ranges.

Parameters

- **metrics** – dictionary whose keys are names are erros and values are error values.
- **ranges** – tuple of tuples defining range of errors to plot in one plot
- **exclude** – List of metrics to be excluded from plotting.
- **max_metrics_per_fig** – maximum number of metrics to show in one figure.
- **plot_type** – either of `radial` or `bar`.
- **show** – If, then figure will be shown/drawn
- **save** – if True, the figure will be saved.
- **save_path** – if given, the figure will the saved at this location.
- **kwargs** – keyword arguments for plotting

Examples

```
>>> import numpy as np
>>> from SeqMetrics import RegressionMetrics
>>> from SeqMetrics import plot_metrics
>>> t = np.random.random((20, 1))
>>> p = np.random.random((20, 1))
>>> er = RegressionMetrics(t, p)
>>> all_errors = er.calculate_all()
>>> plot_metrics(all_errors, plot_type='bar', max_metrics_per_fig=50)
>>> # or draw the radial plot
>>> plot_metrics(all_errors, plot_type='radial', max_metrics_per_fig=50)
```

```
```

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

INDEX

P