Revision Scoring Documentation

Release 2.5.2

Revision Scoring team

Aug 23, 2019

Contents

1 Key Features **3**
- 1.1 Scoring Models . . . 3
- 1.2 Feature extraction . . . 3
- 1.3 Language support . . . 4

2 Indices and tables **5**

Python Module Index **7**

Index **9**

This library contains a set of facilities for constructing and applying `ScorerModel` s to MediaWiki revisions. This library eases the training and testing of Machine Learning-based scoring strategies.

- See the API reference for detailed information.

CHAPTER 1

Key Features

1.1 Scoring Models

Scoring `Model` are the core of the *revscoring* system. Provide a simple interface with complex internals. Most commonly, a `Learned` (Machine Learned) is `train()`'d and `test()`'d on labeled data to provide a basis for scoring. We currently support `Gradient Boosting`, `Random Forest`, `Linear Regression`, `Support Vector Classifier`, and `Naive Bayes` type models. See `revscoring.scoring`

Example:

```
>>> import mwapi
>>> from revscoring import Model
>>> from revscoring.extractors import api
>>>
>>> with open("models/enwiki.damaging.linear_svc.model") as f:
...     model = Model.load(f)
...
>>> extractor = api.Extractor(mwapi.Session(host="https://en.wikipedia.org",
...                                         user_agent="revscoring demo"))
>>> values = extractor.extract(123456789, model.features)
>>> print(model.score(values))
{'prediction': True,
 'probability': {False: 0.4694409344514984,
                 True: 0.5305590655485017}}
```

1.2 Feature extraction

Revscoring provides a dependency-injection-based feature extraction framework that allows new features to be built on top of old. This allows a powerful means to expressing new features and a simple way to address efficiency concerns. See `revscoring.features`, `revscoring.datasources`, and `revscoring.extractors`

Example:

```
>>>     >>> from mwapi import Session
>>> from revscoring.extractors import api
>>> from revscoring.features import temporal, wikitext
>>>
>>> session = Session("https://en.wikipedia.org/w/api.php", user_agent="test")
>>> api_extractor = api.Extractor(session)
>>>
>>> features = [temporal.revision.day_of_week,
...             temporal.revision.hour_of_day,
...             wikitext.revision.parent.headings_by_level(2)]
>>>
>>> values = api_extractor.extract(624577024, features)
>>> for feature, value in zip(features, values):
...     print("     {0}: {1}".format(feature, repr(value)))
...
    <temporal.revision.day_of_week>: 6
    <temporal.revision.hour_of_day>: 19
    <wikitext.revision.parent.headings_by_level(2)>: 5
```

1.3 Language support

Many features require language specific utilities to be available to support feature extraction. In order to support this, we provide a collection of language feature sets that work like other features except that they are language-specific. Language-specific feature sets are available for the following languages: `arabic`, `czech`, `dutch`, `english`, `estonian`, `french`, `german`, `hebrew`, `hindi`, `hungarian`, `indonesian`, `italian`, `japanese`, `korean`, `norwegian`, `persian`, `polish`, `portuguese`, `romanian`, `russian`, `spanish`, `swedish`, `tamil`, `turkish`, `ukrainian`, and `vietnamese`. See `revscoring.languages`

Example:

```
>>> from revscoring.datasources.revision_oriented import revision
>>> from revscoring.dependencies import solve
>>> from revscoring.languages import english, spanish
>>>
>>> features = [english.informals.revision.matches,
...             spanish.informals.revision.matches]
>>> values = solve(features, cache={revision.text: "I think it is stupid."})
>>>
>>> for feature, value in zip(features, values):
...     print("     {0}: {1}".format(feature, repr(value)))
...
    <len(<english.informals.revision.matches>)>: 2
    <len(<spanish.informals.revision.matches>)>: 0
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

r

revscoring, **??**

Index

R

revscoring (*module*), 1