

Jubatus

第4回 データ変換と特徴抽出

概要

- データ変換
 - フィルター
 - 特徴抽出

生データ



加工データ



特徴ベクトル

フィルタ

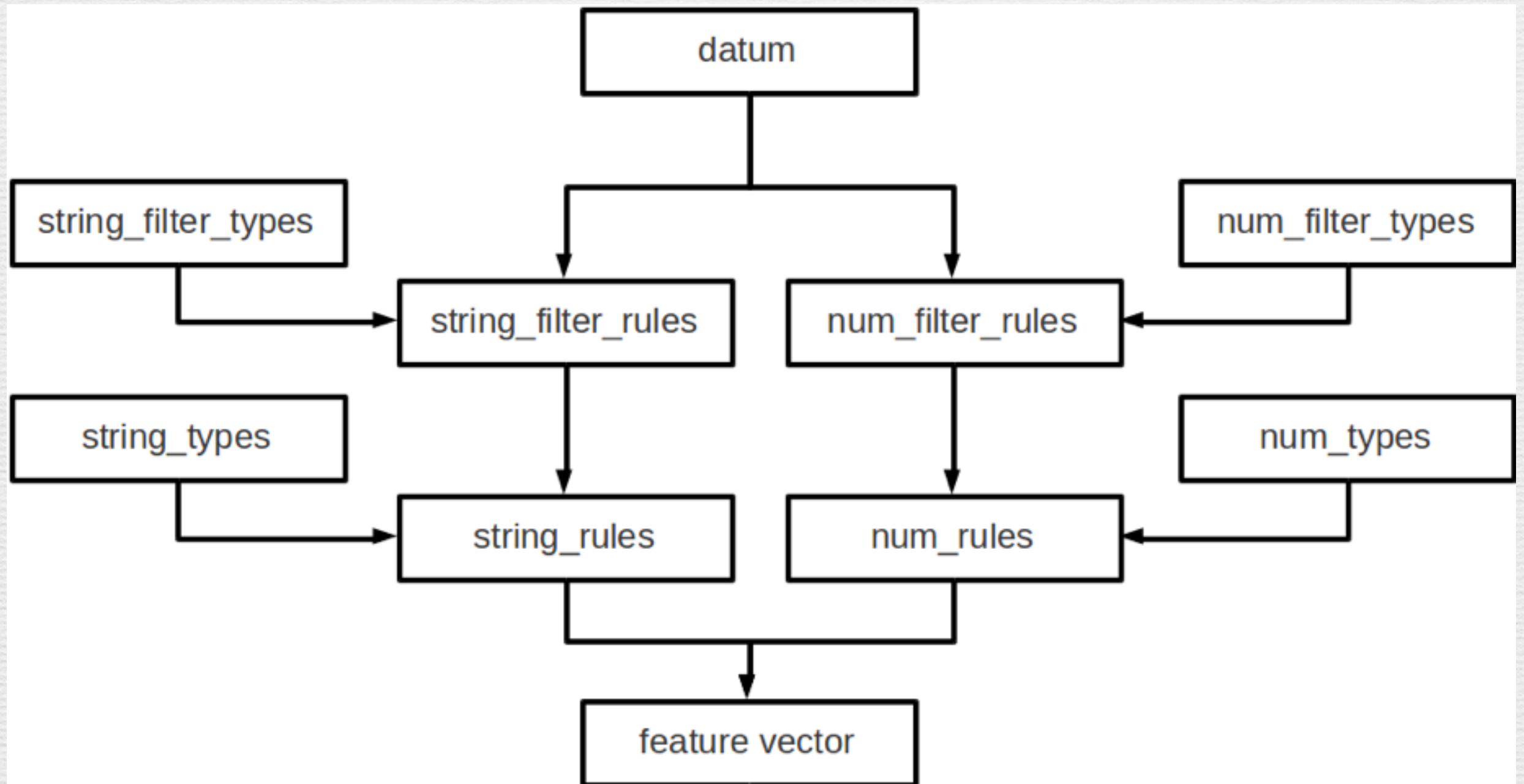
特徴抽出

```
1 <!DOCTYPE HTML PUBLIC "-//  
2 "http://www.w3.org/TR/ht  
3 <html>  
4 <head>  
5   <title>Example</title>  
6   <link rel="stylesheet" href="s  
7 </head>  
8 <body>  
9   <div id="header">  
10    <h1><a href="." title="Back  
11  </div>  
12   <div id="toolbar">  
13    <span class="left">Today <sp  
14    <span class="right">  
15      <span id="time">&nbsp;</sp  
16      <select id="timezone">  
17        <option value="-12">(GMT  
18        <option value="-11">(GMT
```

局は今現在に収まっております、なん
となく生きています。五月最後の更新に
分路りでもしようかと思う。
とは言っても、そこら辺はタラシと書いて
い、こんな感じで自分の行く先を歩いてきた
によって言う一つの参考にもしてみたく
毎度のことながら結構シビアなこと書いてる
に
人ですっていきま

used terms			
od	apple	web	
0.00867	0.04016	0.01075	
0.00454	0.00480	0.01003	
0.00454	0.04016	0.00970	
0.00495	0.00892	0.01890	
0.00117	0.00469	0.01075	
0.00419	0.00201	0.01003	
0.00450	0.00468	0.00947	
0.00398	0.04016	0.01816	
0.00752	0.00010	0.00182	
0.00398	0.00464	0.01890	

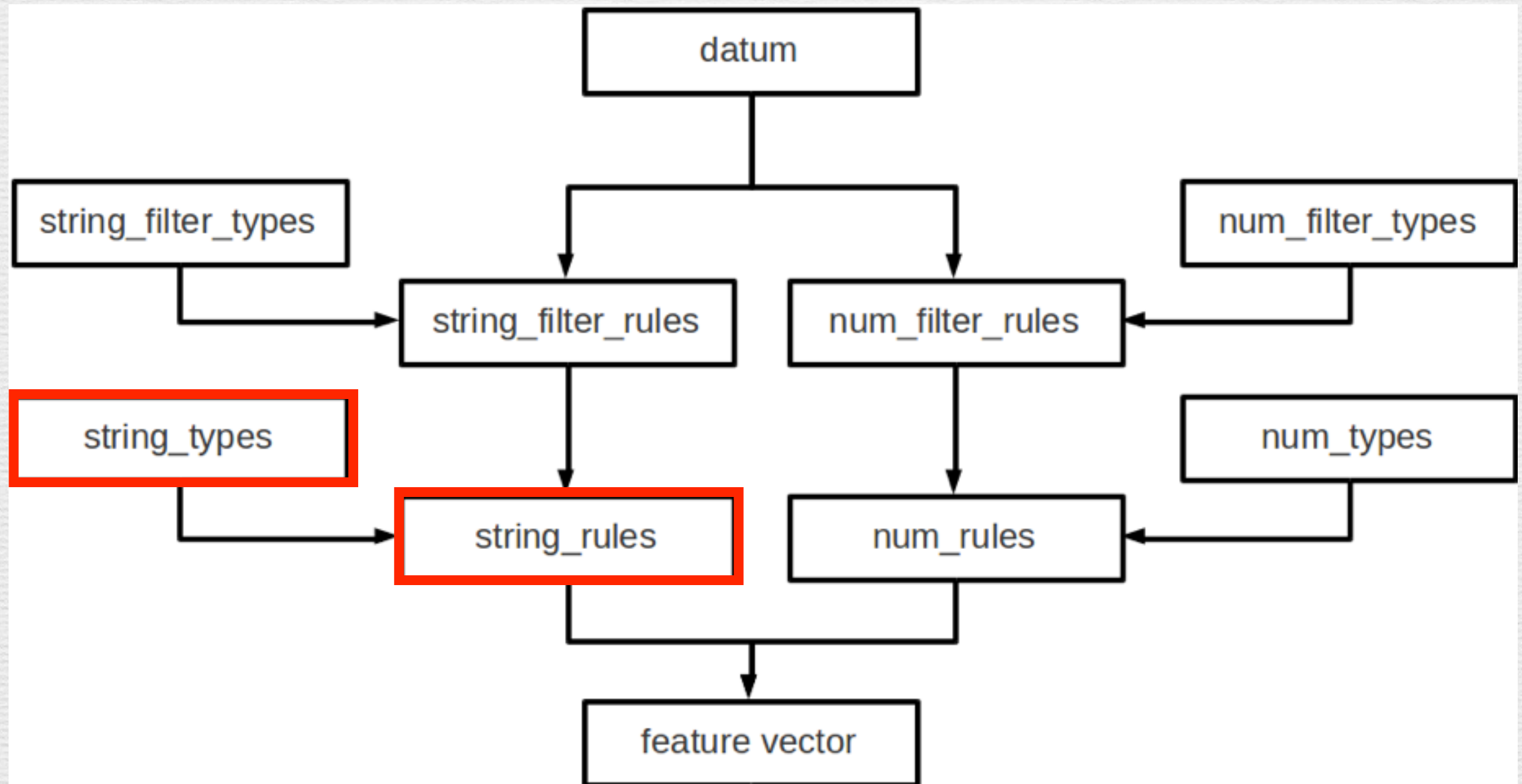
データ変換の流れ



Jubatus

```
{
  "string_filter_types": {},
  "string_filter_rules": [],
  "num_filter_types": {},
  "num_filter_rules": [],
  "string_types": {},
  "string_rules": [
    { "key": "*", "type": "space", "sample_weight": "bin", "global_weight": "bin" }
  ],
  "num_types": {},
  "num_rules": [
    { "key": "*", "type": "num" }
  ],
  "binary_types": {},
  "binary_rules": []
}
```

データ変換の流れ



文字列からの特徴抽出

- 設定の例

```
"string_types": {  
  "bigram": { "method": "ngram", "char_num": "2" }  
},  
"string_rules": [  
  { "key": "user/name",      "type": "str",  "sample_weight": "bin", "global_weight": "bin" },  
  { "key": "message",        "type": "bigram", "sample_weight": "tf", "global_weight": "bin" },  
  { "key": "message-detagged", "type": "space", "sample_weight": "bin", "global_weight": "bin" }  
]
```

- string_types
- string_rules

string_types

- <抽出器名

```
"string_types": {  
  "bigram": { "method": "ngram", "char_num": "2" }  
},
```

- method
 - ngram
 - regexp
 - dynamic

ngram

- 隣接する

```
"string_types": {  
  "bigram": { "method": "ngram", "char_num": "2" }  
},
```

- char_num

“message”	“<H>Hello World</H>”	“<H”	1
		“H>”	2
		“>H”	1
		“He”	1
		“el”	1

regexp

- 正規表現を利用して特徴量として利用
 - pattern
 - group

```
"string_types": {  
  "age": {  
    "method": "regexp",  
    "pattern": "(age|Age)([ :=])([0-9]+)",  
    "group": "3"  
  }  
}
```

string_rules

- 抽出規則適用の指定

```
"string_rules": [  
  { "key": "user/name",      "type": "str",  "sample_weight": "bin", "global_weight": "bin" },  
  { "key": "message",        "type": "bigram", "sample_weight": "tf", "global_weight": "bin" },  
  { "key": "message-detagged", "type": "space", "sample_weight": "bin", "global_weight": "bin" } ]
```

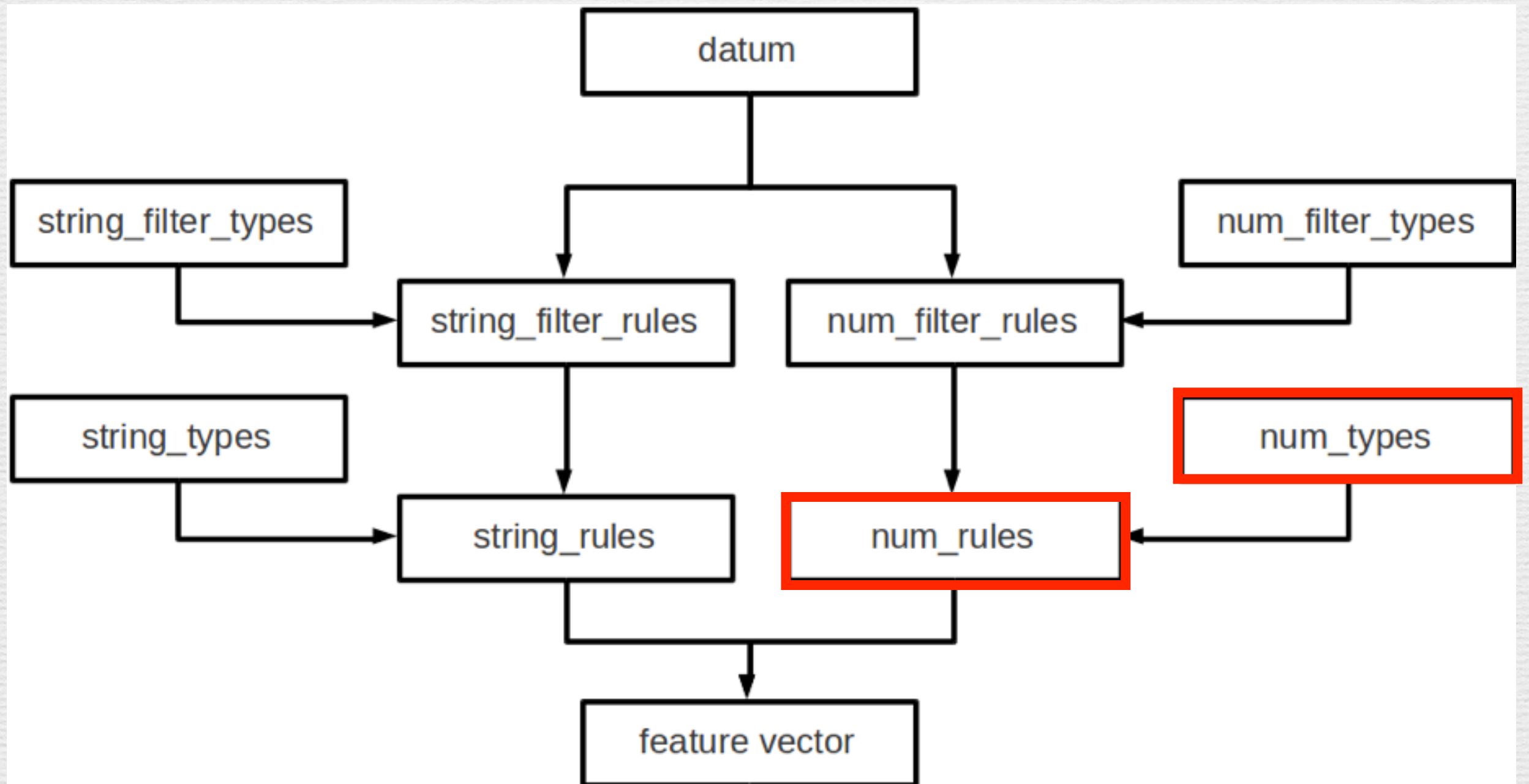
- key
- except
- type
 - str
 - space

重みの設定

- sample_weight
 - bin
 - tf：与えられた文字列中で出現する回数で重み付け
 - log_tf
- global_weight
 - bin
 - idf

```
"sample_weight": "bin", "global_weight": "bin" },  
"sample_weight": "tf", "global_weight": "bin" },
```

データ変換の流れ



数値からの特徴抽出

- 文字列同様に数値も抽出規則を指定できる

```
"num_types": {  
  },  
"num_rules": [  
  { "key": "user/age",    "type": "num" },  
  { "key": "user/income", "type": "log" },  
  { "key": "user/age_kazoe", "type": "num" }  
]
```

- num_types
- num_rules

num_rules

- 数値特徴の抽出規則を指定

```
"num_rules": [  
  { "key": "user/age",    "type": "num" },  
  { "key": "user/income", "type": "log" },  
  { "key": "user/age_kazoe", "type": "num" } ]
```

- type
 - num
 - log
 - str

まとめ

- 文字列・数値の特徴抽出の設定を紹介
 - `string_types`
 - `string_rules`
 - `num_types`
 - `num_rules`