競馬予測のためのスクレイピングの計画書

池守和槻

2019/08/12

1　目的

2　収集するデータ

1. 馬の名前
2. レース名
3. 競馬場の距離
4. 競馬場の場所
5. 騎手
6. レーン
7. オッズ
8. レース結果
9. 親

3　対象のサイト

netkeiba.com をスクレイピングする。この中でも、G1,G2,G 3 を対象に全てのレース結果をスクレイピングする。

4　開発について

4.1　開発における決まりごと

- テスト駆動開発 (TDD) による開発を行う。
- commit, push は関数、メソッド単位で細かく行い進捗状況をリアルタイムで確認できるようにする。
- メソッドにはコメントを書く。
  以下のように描いて欲しい。

ソースコード 1　コメントの例

```
1  def hoge(a, b):
2    """ 足し算をするメソッド
3    a: int
4    b: int
5    戻り値: int
6    """
7    return a + b
```

4.2 python のバージョンと使用するライブラリ

- python のバージョン
  python3.7.1
- beatifulsoup のバージョン
  beatifulsoup4 を使用。
  以下のコマンドでインストールすれば良い。

ソースコード 2　install command

```
$ conda install beatifulsoup4
```

- requests のバージョン
  以下のコマンドでインストールすれば良い。

ソースコード 3　install command

```
$ conda install requests
```

- pandas はすでに anaconda に入ってるのでそれを使う。
  pandas – スクレイピングしたデータを csv 形式で保存するときに使う。

4.3 ディレクトリ構造

ソースコード 4　プロジェクトのディレクトリ構造

```
.
├── README.md
├── docs
│   ├── config.pdf
│   └── config.tex
└── src
    ├── lib
    │   ├── scraper.py
    │   └── test_scraper.py
    └── main.py
```

4.4 作成するモジュールのメソッドについて

- scraper.py では以下のメソッドを作成する。

ソースコード 5　install command

```
def get_race(url):
    """レースのURLを受け取り収集するデータをスクレイピングして、
    二次元配列として返す
    url: string, スクレイピングするレースのurl
    戻り値: 二次元配列, 収集したデータを馬ごとに整理した二次元配列
    """
    ~実装部分~
```

- main.py では 10 年分の G 1、G 2、G3 のレースをスクレイピングする。

4.5　データの保存方法

csv 形式で保存する。

4.6　参考サイト

- スクレピングの参考サイト
  python3 クローリングスクレイピング
- python の unittest の使い方の参考サイト
  Python 標準の unittest の使い方メモ