

競馬予想システム

依田倫太郎

更新情報

- 2023/01/25
 - my_modulesの階層変えた
- 2023/02/02
 - my_modulese/prepare 実装
 - my_modulese/constants 実装
 - resultsテーブル -> race_resultsに変数名変更
 - race_resultsテーブルにtrainer_id, owner_id を追加 (過去のcsvとは一致しないので, 古いcsv消して, 新たにデータをダウンロードする必要あり)
 - 詳しくはmain.ipynbを見よ
 - 学習まではもしかしたら動かないかもしれない (未テスト)

ロードマップ

次やること

- データの前処理を行う `my_modules/preprocessing` の実装
- `preprocessing` が実装できたら, `my_modules/training` 実装

目次

更新情報

ロードマップ

1. 目的
2. ディレクトリ構成
3. ライブラリと使用法

1. 目的

- めざせ不労所得
- ラインから60秒で始められる高収入バイト
- 万馬券を当てよう

2. ディレクトリ構成

データベース

```
├── Data
│   ├── horse_results.csv
│   ├── peds.csv
│   ├── results.csv
│   └── return.csv
```

2. ディレクトリ構成

```
my_library
├── __pycache__
│   ├── horse.cpython-37.pyc
│   └── horse.cpython-39.pyc
├── horse.py
├── my_modules
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-37.pyc
│   │   ├── __init__.cpython-38.pyc
│   │   ├── _calc.cpython-37.pyc
│   │   ├── _calc.cpython-38.pyc
│   │   ├── _plot_importance.cpython-37.pyc
│   │   ├── _plot_importance.cpython-38.pyc
│   │   ├── _process_data.cpython-37.pyc
│   │   ├── _process_data.cpython-38.pyc
│   │   └── _show_best_kaime.cpython-38.pyc
│   ├── _calc.py
│   ├── _plot_importance.py
│   ├── _process_data.py
│   └── _show_best_kaime.py
├── prepare
├── preprocessing
├── simulator
│   ├── __init__.py
│   ├── __pycache__
│   │   ├── __init__.cpython-37.pyc
│   │   ├── __init__.cpython-38.pyc
│   │   ├── _simulator.cpython-37.pyc
│   │   └── _simulator.cpython-38.pyc
│   └── _simulator.py
└── training
    ├── __init__.py
    ├── __pycache__
    │   ├── __init__.cpython-37.pyc
    │   ├── __init__.cpython-38.pyc
    │   └── _learn.cpython-38.pyc
    └── _learn.py
```

- 今 horse.py にベタ書きしてる class や func を順次 my_library/my_modules に移行する予定
- 現在は my_library/horse.py でよく使う奴らを定義

3. ライブラリと使用法（主に使ってるやつね）

class HorseResults

horse_results テーブルの前処理を行うクラス

class Return

returns テーブルの前処理を行うクラス

class Results

results テーブルの前処理を行うクラス

全部 horse.py にいる

class Peds

peds テーブルの前処理を行うクラス

3. ライブラリと使用法

class Simulator

※ result_df 返却するだけなので将来的にclass -> func にする予定
result_df: 予測の情報と, odds, 払い戻しなどが記載されたテーブル
my_modules.func.calc_(result_df,kaime='tansho')で使う予定

class TodaySimulator

レース当日の sim 用のクラス
こいつも廃止予定

3. ライブラリと使用法

class LearnLGBM

lightgbmを学習させるクラス

lightgbmについてはググってね

本研究ではランク学習を用いる方針

これもクラス化したメリットあんま感じないので, 廃止予定

class Predictor (LearnLGBM)

当日レース用 予測モデル作成

こいつもキモいから廃止予定

※ fasttext

- 競馬の血統データのベクトル化に, 自然言語処理で用いられるfasttextが良いと聞いた
- ただインストール方法がクソだるい割に精度の向上に貢献したかは謎
- 多分エラーのオンパレードなので, fasttextは当分使わない方針
- ただ, 血統データのカテゴリ数がデータ多いと5万とかなるので, 精度の観点からは後々, カテゴリ変数ではなく, ベクトル化していきたい
- エラー出る場合は無視して該当箇所を適宜コメントアウトして!