The BoatRace AI service is running on 133.167.87.221 and port 8000.

You can send requests to <http://localhost:8000> or <http://133.167.87.221:8000>.

At this moment, <http://133.167.87.221:8000> doesn’t work, because the user kusanagi has not enough permission to configure net status and firewall.

So if you want to access to AI service from external IP, please allow port 8000 using the root account.

**(1) How to start AI service?**

> cd /home/kusanagi/dev-sunacchiiis/boatrace

> source venv/bin/activate

> nohup python api.py &

**(2) How to check if AI service is running correctly?**

> curl <http://localhost:8000>

If you see the message "Hello BoatRace AI!", it’s okay.

**(3) How to get the prediction result?**

> curl -X POST -H "Content-Type: text/plain" --data-raw '<payload>' <http://localhost:8000/predict>

Here, <payload> is the following JSON string:

{

“jcd”: <Place code. From 01 to 25. For this project, it’s always 18.>,

“hdate”: <YYYYMMDD format race date.>,

“rno”: <Race number.>,

“waku”: <Array of racer information including [teiban] and [toban]>

}

The example <payload> can be:

{

    "jcd": "18",

    "hdate": "20240601",

    "rno": "3",

    "waku": [

        {

            "teiban": "2",

            "toban": "4396"

        },

        {

            "teiban": "3",

            "toban": "4459"

        },

        {

            "teiban": "1",

            "toban": "3960"

        },

        {

            "teiban": "4",

            "toban": "3956"

        },

        {

            "teiban": "5",

            "toban": "4446"

        },

        {

            "teiban": "6",

            "toban": "4848"

        }

    ]

}

Thus, the complete request for the above example is:

curl -X POST -H "Content-Type: text/plain" --data-raw '{"jcd": "18","hdate": "20240601","rno": "3","waku": [{"teiban": "2","toban": "4396"},{"teiban": "3","toban": "4459"},{"teiban": "1","toban": "3960"},{"teiban": "4","toban": "3956"},{"teiban": "5","toban": "4446"},{"teiban": "6","toban": "4848"}]}' <http://localhost:8000/predict>

The response is the following JSON string:

{

    "status": 200,

    "code": 0,

    "msg": {

        "predict": {

            "1-5-2": "0.9660",

            "4-5-2": "0.0225",

            "1-6-2": "0.0036",

            "1-5-6": "0.0026",

            "1-5-4": "0.0025",

            "1-4-2": "0.0005",

            "6-5-2": "0.0003",

            "1-5-3": "0.0001",

            "4-6-2": "0.0001",

            "4-5-6": "0.0001"

        },

        "best": "1-5-2",

        "why": :"1-3-5の組み合わせが最も有望である理由は、以下の統計データに基づい ています。まず、1号艇の選手は現在のスタートコースでの過去3シーズンの成績において、1着率が86.2%と非常に高く、3着以内に入る確率も100%です。これは、1号艇の選手が非常に安定したパフォーマンスを発揮していることを示しています。次に、3号艇の選手は、徳山での過去3シーズンの成績において、1着率が33.3%、3着以内 に入る確率が60.0%と高い数値を示しています。さらに、現在のスタートコースでの成績でも、3着以内に入る確率が60.9%と安定しています。最後に、5号艇の選手は、徳山での過去3シーズンの成績において、3着以内に入る確率が37.5%と比較的高く、現在のスタートコースでの成績でも、3着以内に入る確率が45.8%と安定しています 。これらのデータを総合的に考慮すると、1-3-5の組み合わせが最も有望であると判断できます。"

    }

}

Here, if “status” is not 200, it’s a server error.

If “code”<0, it’s a app-logic error. The error message is “msg”/”why” in this case.

“predict” includes the 10 combinations with the highest probabilities.

“best” is the top combination chosen by AI.

“why” is the justifying clause for successful prediction, and the error message if prediction is failed.

(4) Automatic download XML data

At the fixed time of every day, please call the following API.

curl -X POST <http://localhost:8000/refresh_data>

It scans and downloads new XML files.

AI engine updates its knowledge about the history.

(5) Fine-tune AI with latest data

Please call the following API:

curl -X POST <http://localhost:8000/train>

Training usually takes about 1 hour.