

SAGE Behavior Detection

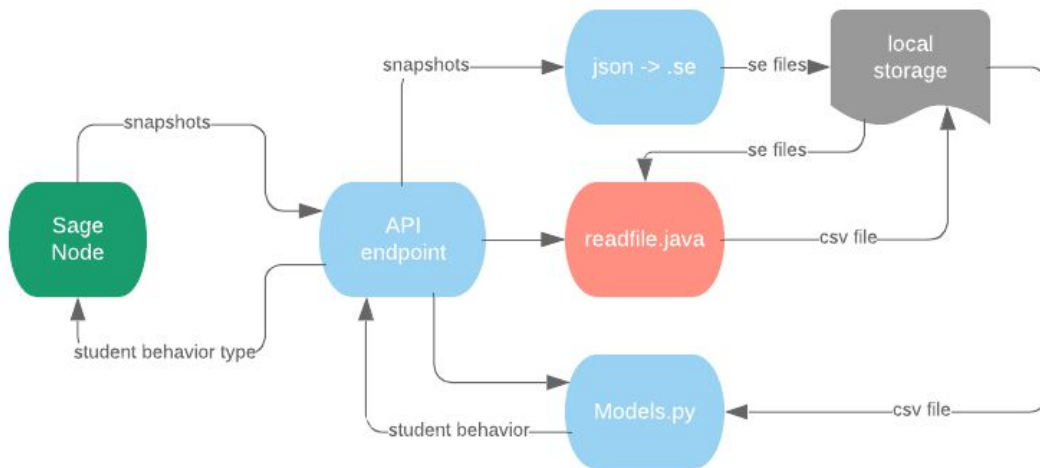
Vivian Han, Joe Huang





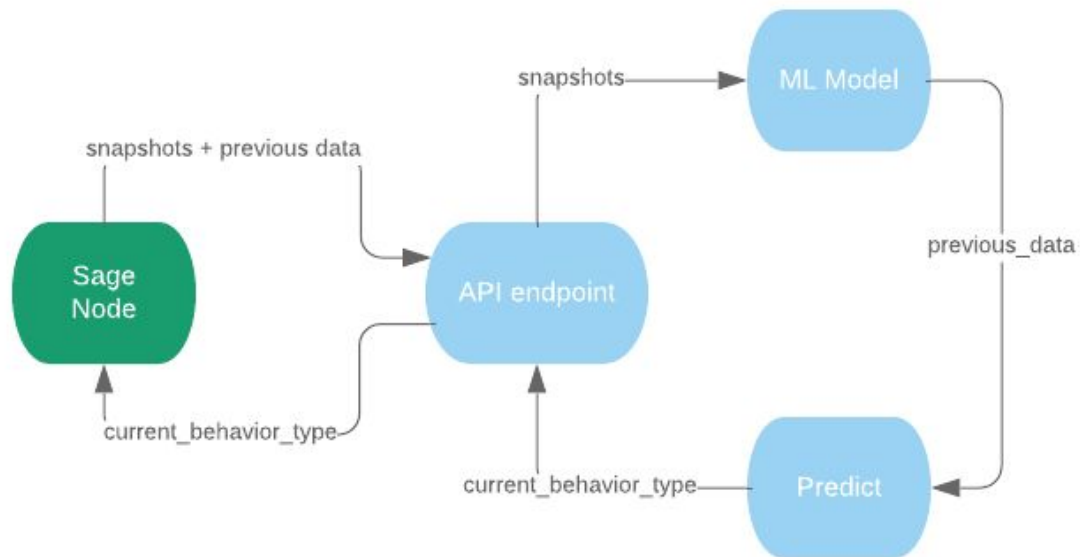
Previous Data Flow

- Combination of multiple modules
- Redundant data transformation
- Local storage





Simplified Data Flow





Changes in data format

```
{
  'prev_data': {
    ...
  },
  'cur_game': 0,
  'snapshots': {
    ...
  }
}
```

```
  "snapshots": [
    {
      "timestamp": "18-40-28-GMT-1204-2018",
      "content": "<<Object Stage>>\n"
    },
    {
      "timestamp": "18-40-30-GMT-1204-2018",
      "content": "<<Object Stage>>\n\t\twhenGreenFlag\n"
    },
    {
      "timestamp": "18-40-31-GMT-1204-2018",
      "content": "<<Object Stage>>\n\t\twhenGreenFlag\n\t\tstartScene\n"
    },
  ],
}
```



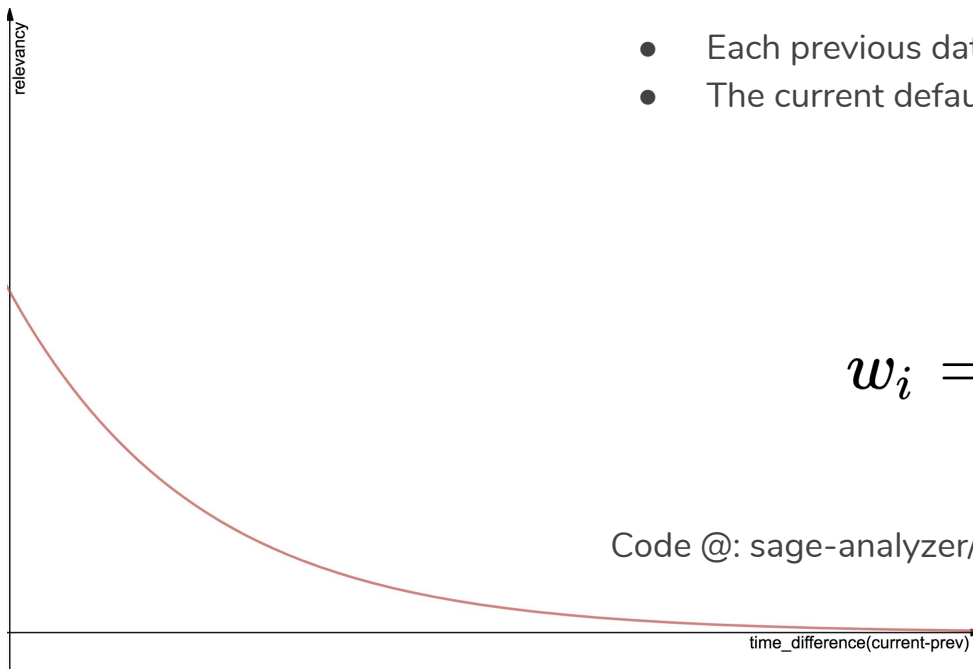
Taking Previous Data into Consideration

```
"19-55-11-GMT-1207-2018": [      timestamp: [
    0,                                behavior_type,
    1                                game_type
], |
"19-55-11-GMT-1205-2018": [      ]
    3,
    6
],
```



Previous Behavior Type

- Each previous data weighted by time difference
- The current default cutoff time is set to 20 days



$$w_i = e^{\frac{\ln(0.01) \times \text{time_diff}[i]}{\text{cutoff}}}$$

Code @: [sage-analyzer/intelligentHint/behaviordetection](#)



Previous Game Type

- Basic Probability Model
- Smoothing

$$P(b|g) = \frac{P(b,g)+k}{P(g)+4k}$$

Behavior Type	Game Type	P(behavior game)
0	3	0.05
1	3	0.75
...

Code @: [sage-analyzer/intelligentHint/behaviordetection](#)

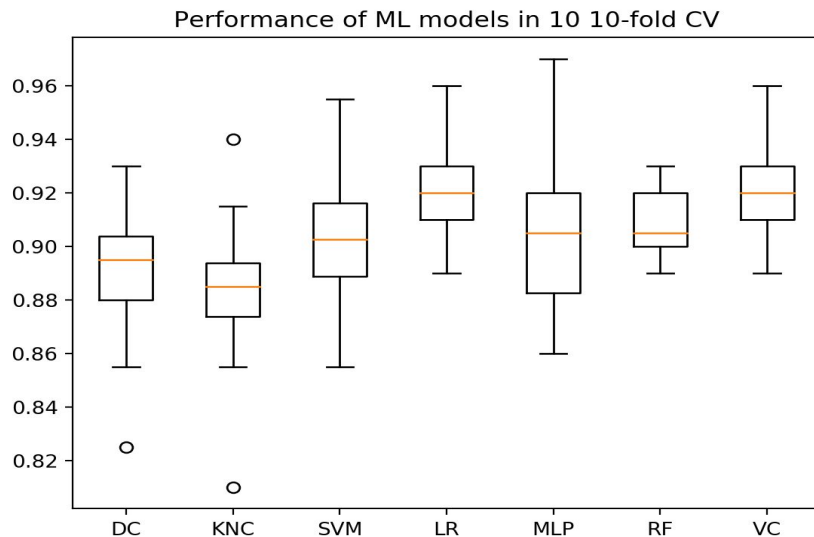


Models - stacking

Voting Model

- Decision Tree
- SVM
- K Nearest Classifier
- Logistic Regression
- Multilayer Perceptron
- Random forest

The model is saved locally as
saved_model.pkl

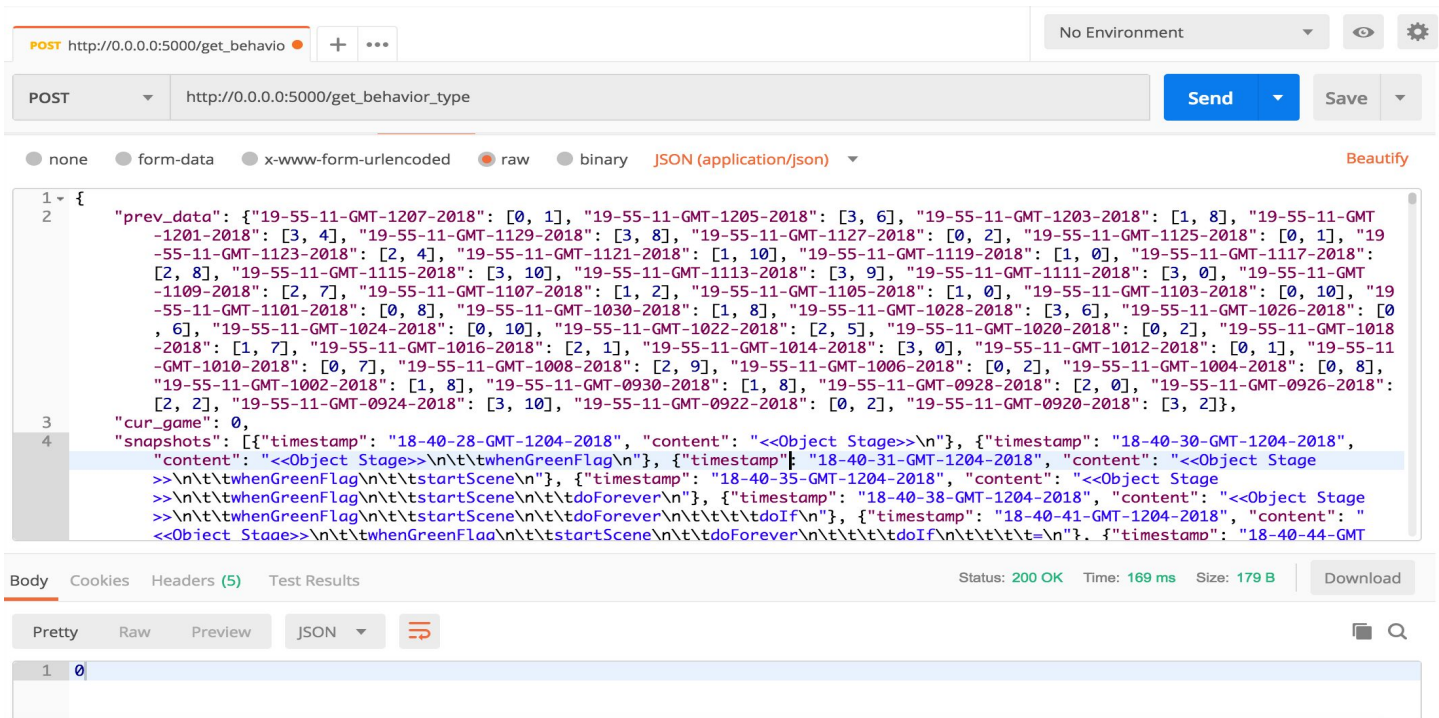




Output the current behavior type

$$P_{model}(b)P(b|g) \sum_{b_i=b} w_i$$

API Demo (Postman)



POST http://0.0.0.0:5000/get_behavior_type

POST http://0.0.0.0:5000/get_behavior_type

Send Save

none form-data x-www-form-urlencoded raw binary JSON (application/json) Beautify

```
1 {
2   "prev_data": {"19-55-11-GMT-1207-2018": [0, 1], "19-55-11-GMT-1205-2018": [3, 6], "19-55-11-GMT-1203-2018": [1, 8], "19-55-11-GMT-1201-2018": [3, 4], "19-55-11-GMT-1129-2018": [3, 8], "19-55-11-GMT-1127-2018": [0, 2], "19-55-11-GMT-1125-2018": [0, 1], "19-55-11-GMT-1123-2018": [2, 4], "19-55-11-GMT-1121-2018": [1, 10], "19-55-11-GMT-1119-2018": [1, 0], "19-55-11-GMT-1117-2018": [2, 8], "19-55-11-GMT-1115-2018": [3, 10], "19-55-11-GMT-1113-2018": [3, 9], "19-55-11-GMT-1111-2018": [3, 0], "19-55-11-GMT-1109-2018": [2, 7], "19-55-11-GMT-1107-2018": [1, 2], "19-55-11-GMT-1105-2018": [1, 0], "19-55-11-GMT-1103-2018": [0, 10], "19-55-11-GMT-1101-2018": [0, 8], "19-55-11-GMT-1030-2018": [1, 8], "19-55-11-GMT-1028-2018": [3, 6], "19-55-11-GMT-1026-2018": [0, 6], "19-55-11-GMT-1024-2018": [0, 10], "19-55-11-GMT-1022-2018": [2, 5], "19-55-11-GMT-1020-2018": [0, 2], "19-55-11-GMT-1018-2018": [1, 7], "19-55-11-GMT-1016-2018": [2, 1], "19-55-11-GMT-1014-2018": [3, 0], "19-55-11-GMT-1012-2018": [0, 1], "19-55-11-GMT-1010-2018": [0, 7], "19-55-11-GMT-1008-2018": [2, 9], "19-55-11-GMT-1006-2018": [0, 2], "19-55-11-GMT-1004-2018": [0, 8], "19-55-11-GMT-1002-2018": [1, 8], "19-55-11-GMT-0930-2018": [1, 8], "19-55-11-GMT-0928-2018": [2, 0], "19-55-11-GMT-0926-2018": [2, 2], "19-55-11-GMT-0924-2018": [3, 10], "19-55-11-GMT-0922-2018": [0, 2], "19-55-11-GMT-0920-2018": [3, 2]},
3   "cur_game": 0,
4   "snapshots": [{"timestamp": "18-40-28-GMT-1204-2018", "content": "<<Object Stage>>\n"}, {"timestamp": "18-40-30-GMT-1204-2018", "content": "<<Object Stage>>\n"}, {"timestamp": "18-40-31-GMT-1204-2018", "content": "<<Object Stage>>\n"}, {"timestamp": "18-40-35-GMT-1204-2018", "content": "<<Object Stage>>\n"}, {"timestamp": "18-40-38-GMT-1204-2018", "content": "<<Object Stage>>\n"}, {"timestamp": "18-40-41-GMT-1204-2018", "content": "<<Object Stage>>\n"}, {"timestamp": "18-40-44-GMT-1204-2018", "content": "<<Object Stage>>\n"}]
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 169 ms Size: 179 B Download

Pretty Raw Preview JSON

```
1 0
```



Limitation & Future Work

- Improving ML Model Accuracy (currently ~90% w/ mock data)
 - Improving Probability Model
 - Adjusting Function Parameters
 - Unique ID for each block rather than just name
 - More variation in mock data