training_overview_1637880409.5774007

December 2, 2021

```python
import pandas as pd
import os
import matplotlib.pyplot as plt

TIMESTAMP = 1637880409.5774007
DATA_DIRECTORY = '../data'

perf_data = pd.read_csv(os.path.join(DATA_DIRECTORY,
 f'{TIMESTAMP}_performance_stats.csv'))
perf_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
perf_data.head()
```
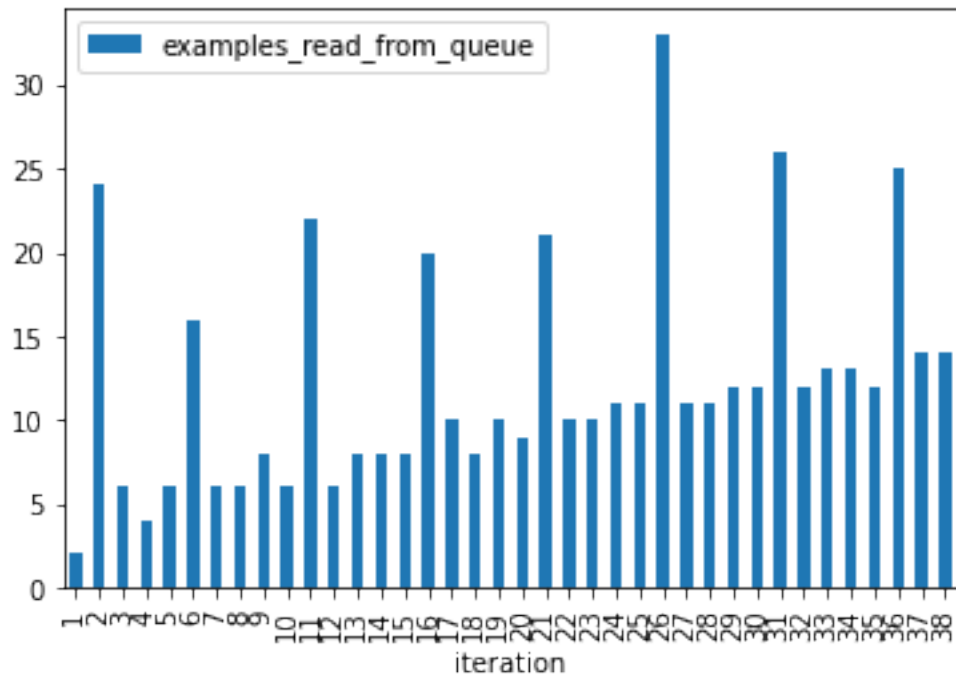
```
   iteration                      timestamp  iteration_duration  \
0          1 1970-01-01 00:00:01.637881653          861.927125
1          2 1970-01-01 00:00:01.637881880          226.545644
2          3 1970-01-01 00:00:01.637882121          241.500161
3          4 1970-01-01 00:00:01.637882384          262.868696
4          5 1970-01-01 00:00:01.637883259          874.580279

   training_duration  examples_read_from_queue  length_train_examples
0           2.659691                         2                    402
1          15.793938                        24                   2402
2          29.040167                         6                   4402
3          42.590014                         4                   6402
4          55.795066                         6                   8402
```
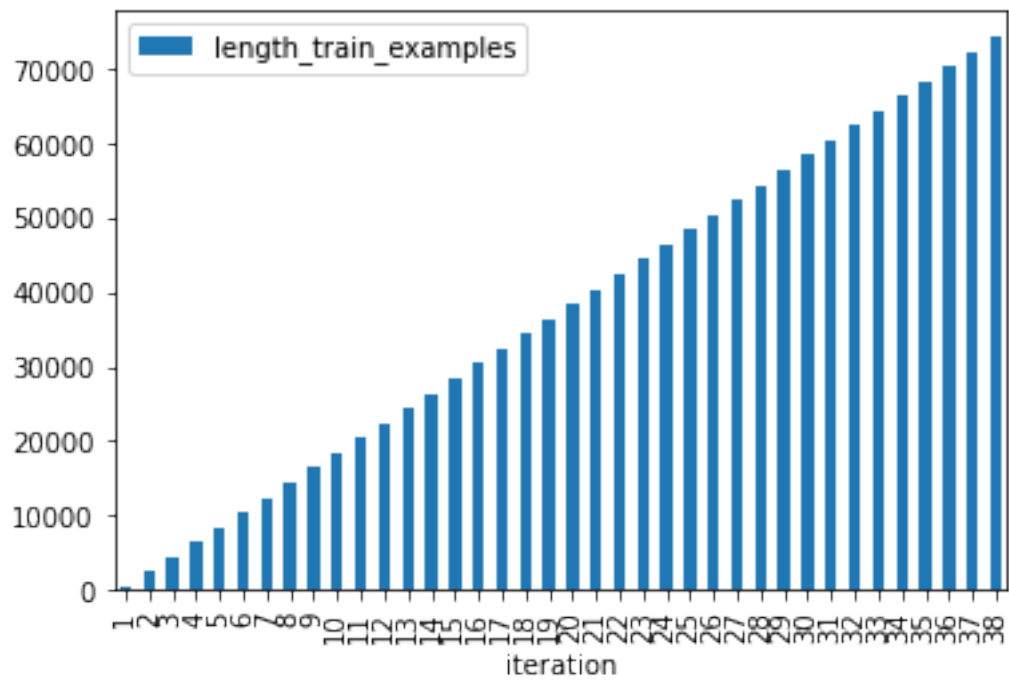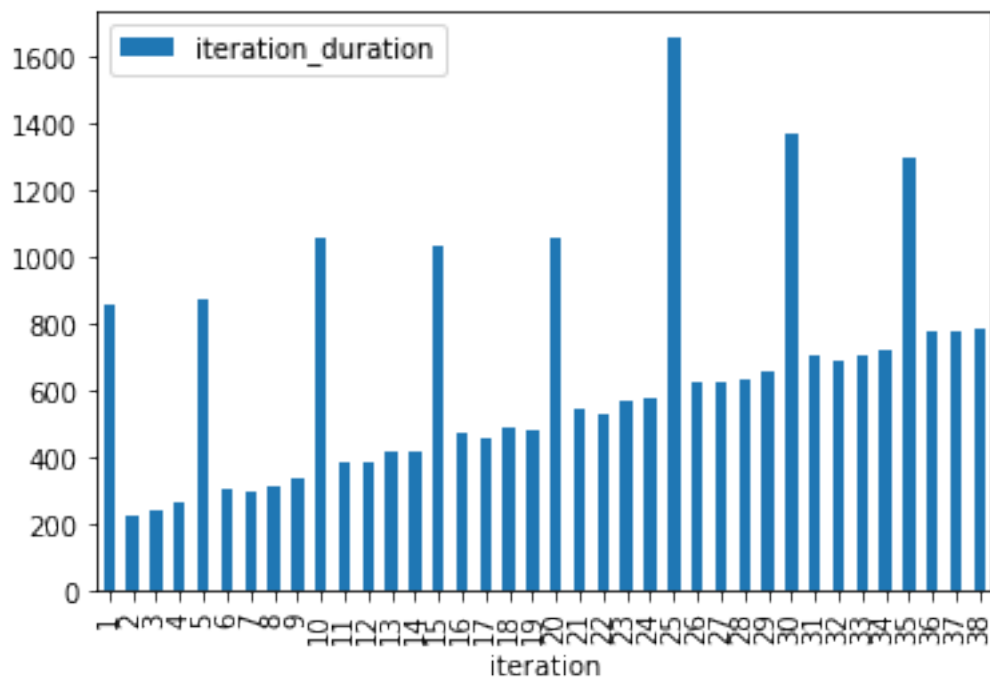
```python
perf_data.plot.bar(x='iteration', y='examples_read_from_queue')
```

```
<AxesSubplot:xlabel='iteration'>
```

```
perf_data.plot.bar(x='iteration', y='length_train_examples')
```

<AxesSubplot:xlabel='iteration'>

```
perf_data.plot.bar(x='iteration', y='iteration_duration')
```

```
<AxesSubplot:xlabel='iteration'>
```



```
rndm_data = pd.read_csv(os.path.join(DATA_DIRECTORY,
    ↪f'{TIMESTAMP}_random_player_game_stats.csv'))
rndm_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
rndm_data.set_index('iteration')
rndm_data.head()
```

```
   iteration                    timestamp  wins  losses  draws  \
0          1 1970-01-01 00:00:01.637881653   3.0     0.0    5.0
1          5 1970-01-01 00:00:01.637881880   4.0     1.0    3.0
2         10 1970-01-01 00:00:01.637882121   2.0     1.0    5.0
3         15 1970-01-01 00:00:01.637882384   3.0     0.0    5.0
4         20 1970-01-01 00:00:01.637883259   2.0     1.0    5.0

   nnet_cumul_rewards  random_cumul_rewards
0            0.833333             -0.833333
1            0.666667             -0.666667
2            0.166667             -0.166667
3            0.833333             -0.833333
4            0.333333             -0.333333
```

```
hrstc_data = pd.read_csv(os.path.join(DATA_DIRECTORY,␣
 ↪f'{TIMESTAMP}_heuristic_player_game_stats.csv'))
hrstc_data['timestamp'] = pd.to_datetime(perf_data['timestamp'])
hrstc_data.set_index('iteration')
hrstc_data.head()
```

```
   iteration                     timestamp  wins  losses  draws  \
0          1 1970-01-01 00:00:01.637881653   0.0     8.0    0.0
1          5 1970-01-01 00:00:01.637881880   0.0     8.0    0.0
2         10 1970-01-01 00:00:01.637882121   0.0     8.0    0.0
3         15 1970-01-01 00:00:01.637882384   0.0     7.0    1.0
4         20 1970-01-01 00:00:01.637883259   0.0     8.0    0.0

   nnet_cumul_rewards  random_cumul_rewards
0           -6.666667              6.666667
1           -7.500000              7.500000
2           -7.000000              7.000000
3           -6.833333              6.833333
4           -6.666667              6.666667
```

```
rndm_n_games = int(rndm_data['wins'][0] + rndm_data['losses'][0] +␣
 ↪rndm_data['draws'][0])
hrstc_n_games = int(rndm_data['wins'][0] + rndm_data['losses'][0] +␣
 ↪rndm_data['draws'][0])

rndm_fraction_won = rndm_data.apply(lambda row: row['wins'] / rndm_n_games,␣
 ↪axis=1).to_list()
hrstc_fraction_won = hrstc_data.apply(lambda row: row['wins'] / hrstc_n_games,␣
 ↪axis=1).to_list()

plt.plot(rndm_data['iteration'], rndm_fraction_won,  label=f'Random agent␣
 ↪(n={rndm_n_games})')
plt.plot(rndm_data['iteration'], hrstc_fraction_won,  label=f'Random agent␣
 ↪(n={hrstc_n_games})')
plt.xticks(rndm_data['iteration'])
plt.xlabel('Iteration')
plt.ylabel(f'Fraction won')
plt.legend()
plt.show()
```
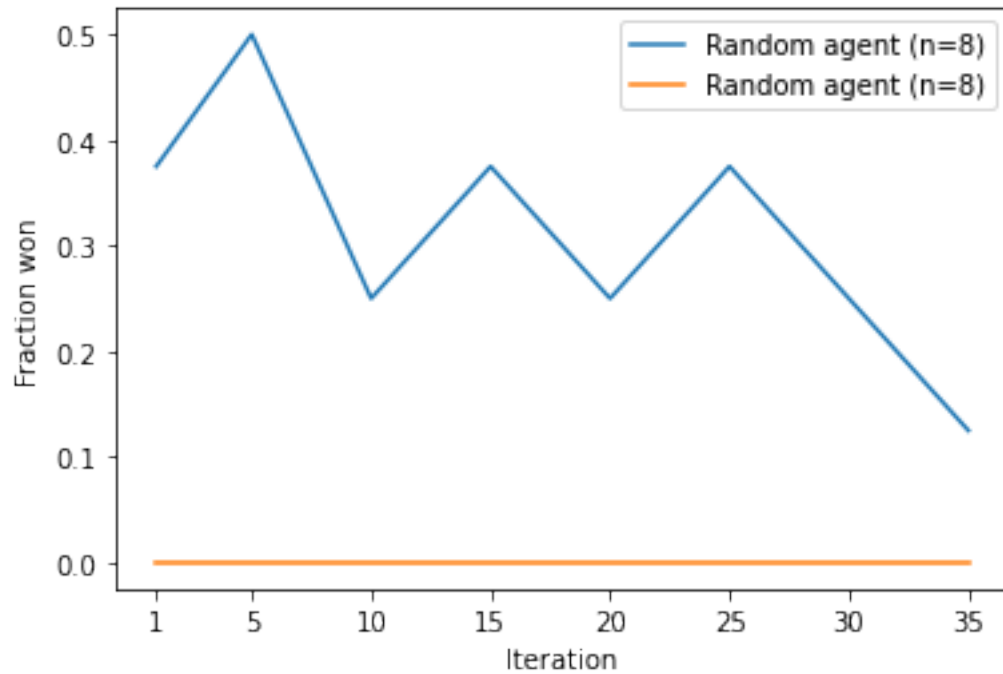
```
[ ]: plt.plot(rndm_data['iteration'], rndm_data['nnet_cumul_rewards'],␣
     ↪label=f'Cumul. rew. (vs. Random agent)')
     plt.plot(rndm_data['iteration'], hrstc_data['nnet_cumul_rewards'],␣
     ↪label=f'Cumul. rew. (vs. Heur. Agent)')
     plt.xticks(rndm_data['iteration'])
     plt.xlabel('Iteration')
     plt.ylabel(f'Reward')
     plt.legend()
     plt.show()
```