Results

August 30, 2020

1 MARS-Gym Experiments - Trivago Results

[1]:

```
import sys, os
os.environ['PATH_EVALUATION'] =  "../../output/evaluation/"
os.environ['PATH_EVAL_REINFORCEMENT'] =  "../../output/interaction/"
os.environ['PATH_TRAIN'] =  "../../output/models/"

#import plotly.graph_objects as go
import plotly.graph_objects as go
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import numpy as np
import pandas as pd
import os
import json
from pandas.io.json import json_normalize
import matplotlib.pyplot as plt
import seaborn as sns

from mars_gym.tools.eval_viz.app import fetch_iteration_results_path,␣
 ↪load_all_iteraction_metrics, fetch_results_path, load_data_metrics,␣
 ↪load_eval_params, filter_df, load_fairness_metrics, load_fairness_df
from mars_gym.tools.eval_viz.plot import plot_line_iteraction,␣
 ↪plot_exploration_arm, get_colors, confidence, plot_bar
from  mars_gym.tools.eval_viz.app import PATH_EVAL_REINFORCEMENT

from util import list_paths_per_model, load_iteractions_params,␣
 ↪get_metrics_reward, group_metrics, load_dataset, plot_cum_reward,␣
 ↪plot_fairness_metrics, plot_fairness_mistreatment, plot_fairness_treatment
```

```
/media/workspace/DeepFood/deep-reco-gym/src/mars_gym/tools/eval_viz/app.py:13:
FutureWarning:

Passing a negative integer is deprecated in version 1.0 and will not be
supported in future version. Instead, use None to not limit the column width.
```

[ ]:

```

```

This notebook includes all viz of paper **'MARS-Gym: A Gym framework to model, train, and evaluate recommendation systems for marketplaces'** and can be used for reproducibility or example of framework usage.

We need run some scripts before vizualise results: * scripts/simulation/chicago_usa_script.sh * scripts/metrics/metrics_chicago_usa_script.sh * scripts/metrics/fairness_recsys_script.sh

1.1 Load Results

[2]:

```
result_paths = list_paths_per_model(PATH_EVAL_REINFORCEMENT)
result_paths[:5]
```

[2]:

```
['../../output/interaction/InteractionTraining/results/InteractionTraining____ma
rs_gym_model_b___alpha___1e_05__a06cfb73a9',
 '../../output/interaction/InteractionTraining/results/InteractionTraining____ma
rs_gym_model_b___alpha___1e_05__f9e2903219',
 '../../output/interaction/InteractionTraining/results/InteractionTraining____ma
rs_gym_model_b___exploration_th_6822e93237',
 '../../output/interaction/InteractionTraining/results/InteractionTraining____ma
rs_gym_model_b___explore_rounds_0b659e17e7',
 '../../output/interaction/InteractionTraining/results/InteractionTraining____ma
rs_gym_model_b___epsilon___0_1__76d6a2cc36']
```

[3]:

```
# We will only filter the city of Chicago to exemplify graphs and results.
#

#city = "Rio de Janeiro, Brazil"
#city = "Como, Italy"
city = "Chicago, USA"
#city = "New York, USA"
#city = 'Copenhagen, Denmark'
#city='recsys'

df_params = load_iteractions_params(result_paths)
df_params = df_params[df_params['data_frames_preparation_extra_params.
 ↪filter_city'] == city]
df_params.head()
```

```
/media/workspace/DeepFood/mars-gym-experiments/scripts/notebooks/util.py:62:
FutureWarning:

pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead
```

[3]:

```
                             project  minimum_interactions  session_test_size  \
0  trivago.config.trivago_experiment  5                     0.1
```

```
0  trivago.config.trivago_experiment  5                     0.1
0  trivago.config.trivago_experiment  5                     0.1
0  trivago.config.trivago_experiment  5                     0.1
0  trivago.config.trivago_experiment  5                     0.1

  dataset_split_method  val_size  n_splits  split_index sampling_strategy  \
0  time                 0.2       5         0            none
0  time                 0.2       5         0            none
0  time                 0.2       5         0            none
0  time                 0.2       5         0            none
0  time                 0.2       5         0            none

  balance_fields  use_sampling_in_validation  …  bandit_policy_params.alpha  \
0  []             False                       …  0.00001
0  []             False                       …  0.00001
0  []             False                       … NaN
0  []             False                       … NaN
0  []             False                       … NaN

                                 iteration  \
0  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___alpha___1e_05__a06cfb73a9
0  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___alpha___1e_05__f9e2903219
0  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___exploration_th_6822e93237
0  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___explore_rounds_0b659e17e7
0  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___epsilon___0_1__76d6a2cc36

   bandit_policy_params.exploration_threshold bandit_policy_params.decay_rate  \
0 NaN                                         NaN
0 NaN                                         NaN
0  0.7                                         0.000097
0 NaN                                          0.000187
0 NaN                                         NaN

  bandit_policy_params.explore_rounds   bandit_policy_params.epsilon  \
0 NaN                                 NaN
0 NaN                                 NaN
0 NaN                                 NaN
0  1000.0                             NaN
0 NaN                                  0.1

   loss_function_params.clip bandit_policy_params.logit_multiplier  \
0 NaN                        NaN
```

```
0 NaN                        NaN
0 NaN                        NaN
0 NaN                        NaN
0 NaN                        NaN

   bandit_policy_params.arg  bandit_policy_params.v_sq
0 NaN                       NaN
0 NaN                       NaN
0 NaN                       NaN
0 NaN                       NaN
0 NaN                       NaN

[5 rows x 55 columns]
```

[4]:

```
# Group results

list_bandits          = df_params.groupby(["bandit_policy_class",␣
 ↪"observation"])['iteraction'].apply(list).reset_index()
list_bandits['name']  = list_bandits.bandit_policy_class.map(str) + ' ' +␣
 ↪list_bandits.observation.map(str)
list_bandits['name']  = list_bandits['name'].apply(lambda x: x.replace("fixed␣
 ↪", "").replace(" ", ""))
list_bandits['model'] = list_bandits.iteraction.apply(lambda l: l[np.random.
 ↪randint(0, len(l))])
list_bandits['model_list'] = list_bandits.iteraction
list_bandits.head(2)
```

[4]:

```
                             bandit_policy_class observation  \
0  mars_gym.model.bandit.AdaptiveGreedy
1  mars_gym.model.bandit.CustomRewardModelLinUCB

         iteraction  \
0  [../../output/interaction/InteractionTraining/results/InteractionTraining____
mars_gym_model_b___exploration_th_6822e93237, ../../output/interaction/Interacti
onTraining/results/InteractionTraining____mars_gym_model_b___exploration_th_83ef
3f55d3, ../../output/interaction/InteractionTraining/results/InteractionTraining
____mars_gym_model_b___exploration_th_42716ae133, ../../output/interaction/Inter
actionTraining/results/InteractionTraining____mars_gym_model_b___exploration_th_
d3663f58fc, ../../output/interaction/InteractionTraining/results/InteractionTrai
ning____mars_gym_model_b___exploration_th_520cebc570, ../../output/interaction/I
nteractionTraining/results/InteractionTraining____mars_gym_model_b___exploration
_th_d70af8a008, ../../output/interaction/InteractionTraining/results/Interaction
Training____mars_gym_model_b___exploration_th_6b44643887, ../../output/interacti
on/InteractionTraining/results/InteractionTraining____mars_gym_model_b___explora
tion_th_2cc29714f4]
1  [../../output/interaction/InteractionTraining/results/InteractionTraining____
mars_gym_model_b___alpha___1e_05__ef475653f1, ../../output/interaction/Interacti
```

```
onTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_05__75ec
cc229a, ../../output/interaction/InteractionTraining/results/InteractionTraining
____mars_gym_model_b___alpha___1e_05__bad81a6465, ../../output/interaction/Inter
actionTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_05__
debd700fb8, ../../output/interaction/InteractionTraining/results/InteractionTrai
ning____mars_gym_model_b___alpha___1e_05__2165771e8e, ../../output/interaction/I
nteractionTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_
05__7966e4e52b, ../../output/interaction/InteractionTraining/results/Interaction
Training____mars_gym_model_b___alpha___1e_05__0b623b96b8, ../../output/interacti
on/InteractionTraining/results/InteractionTraining____mars_gym_model_b___alpha__
_1e_05__d5981787b1]

                                            name  \
0  mars_gym.model.bandit.AdaptiveGreedy
1  mars_gym.model.bandit.CustomRewardModelLinUCB

                                      model  \
0  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___exploration_th_d3663f58fc
1  ../../output/interaction/InteractionTraining/results/InteractionTraining____m
ars_gym_model_b___alpha___1e_05__7966e4e52b

         model_list
0  [../../output/interaction/InteractionTraining/results/InteractionTraining____
mars_gym_model_b___exploration_th_6822e93237, ../../output/interaction/Interacti
onTraining/results/InteractionTraining____mars_gym_model_b___exploration_th_83ef
3f55d3, ../../output/interaction/InteractionTraining/results/InteractionTraining
____mars_gym_model_b___exploration_th_42716ae133, ../../output/interaction/Inter
actionTraining/results/InteractionTraining____mars_gym_model_b___exploration_th_
d3663f58fc, ../../output/interaction/InteractionTraining/results/InteractionTrai
ning____mars_gym_model_b___exploration_th_520cebc570, ../../output/interaction/I
nteractionTraining/results/InteractionTraining____mars_gym_model_b___exploration
_th_d70af8a008, ../../output/interaction/InteractionTraining/results/Interaction
Training____mars_gym_model_b___exploration_th_6b44643887, ../../output/interacti
on/InteractionTraining/results/InteractionTraining____mars_gym_model_b___explora
tion_th_2cc29714f4]
1  [../../output/interaction/InteractionTraining/results/InteractionTraining____
mars_gym_model_b___alpha___1e_05__ef475653f1, ../../output/interaction/Interacti
onTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_05__75ec
cc229a, ../../output/interaction/InteractionTraining/results/InteractionTraining
____mars_gym_model_b___alpha___1e_05__bad81a6465, ../../output/interaction/Inter
actionTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_05__
debd700fb8, ../../output/interaction/InteractionTraining/results/InteractionTrai
ning____mars_gym_model_b___alpha___1e_05__2165771e8e, ../../output/interaction/I
nteractionTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_
05__7966e4e52b, ../../output/interaction/InteractionTraining/results/Interaction
Training____mars_gym_model_b___alpha___1e_05__0b623b96b8, ../../output/interacti
```

```
on/InteractionTraining/results/InteractionTraining____mars_gym_model_b___alpha__
_1e_05__d5981787b1]
```

[ ]:

```

```

[5]:

```
reward_metrics = []
for i, row in list_bandits.iterrows():
    df_metrics = get_metrics_reward(row['iteraction'])
    df_metrics['bandit'] = row['name']
    df_metrics['model']  = row['model']

    df_g_metrics = group_metrics(df_metrics)
    reward_metrics.append(df_g_metrics)
```

[6]:

```
df_all = pd.concat(reward_metrics)
df_all = df_all.reset_index().sort_values('bandit', ascending=False).
 ↪set_index('bandit')
df_all
```

[6]:

```
                                               r_mean|mean  r_mean|std  \
bandit
mars_gym.model.bandit.SoftmaxExplorer             0.275622    0.013914
mars_gym.model.bandit.RandomPolicy                0.040749         NaN
mars_gym.model.bandit.PercentileAdaptiveGreedy    0.284485    0.010725
mars_gym.model.bandit.LinUCB                      0.121703    0.007030
mars_gym.model.bandit.LinThompsonSampling         0.048501         NaN
mars_gym.model.bandit.FixedPolicyPopularItem      0.104567    0.000000
mars_gym.model.bandit.ExploreThenExploit          0.228906    0.006917
mars_gym.model.bandit.EpsilonGreedy               0.285147    0.015851
mars_gym.model.bandit.CustomRewardModelLinUCB     0.297863    0.024824
mars_gym.model.bandit.AdaptiveGreedy              0.215609    0.030293

                                               r_reward|mean  r_reward|std  \
bandit
mars_gym.model.bandit.SoftmaxExplorer            2915.250000    147.173124
mars_gym.model.bandit.RandomPolicy                431.000000           NaN
mars_gym.model.bandit.PercentileAdaptiveGreedy   3009.000000    113.439726
mars_gym.model.bandit.LinUCB                     1287.250000     74.351962
mars_gym.model.bandit.LinThompsonSampling         513.000000           NaN
mars_gym.model.bandit.FixedPolicyPopularItem     1106.000000      0.000000
mars_gym.model.bandit.ExploreThenExploit         2421.142857     73.156063
mars_gym.model.bandit.EpsilonGreedy              3016.000000    167.654406
mars_gym.model.bandit.CustomRewardModelLinUCB    3150.500000    262.564822
mars_gym.model.bandit.AdaptiveGreedy             2280.500000    320.404209

                                               r_reward|count  \
bandit
```

```
mars_gym.model.bandit.SoftmaxExplorer           8
mars_gym.model.bandit.RandomPolicy              1
mars_gym.model.bandit.PercentileAdaptiveGreedy  8
mars_gym.model.bandit.LinUCB                    8
mars_gym.model.bandit.LinThompsonSampling       1
mars_gym.model.bandit.FixedPolicyPopularItem    2
mars_gym.model.bandit.ExploreThenExploit        7
mars_gym.model.bandit.EpsilonGreedy             8
mars_gym.model.bandit.CustomRewardModelLinUCB   8
mars_gym.model.bandit.AdaptiveGreedy            8

model|first
bandit
mars_gym.model.bandit.SoftmaxExplorer           ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___logit_multipli_34
358aae37
mars_gym.model.bandit.RandomPolicy              ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b____1649c7bd6b
mars_gym.model.bandit.PercentileAdaptiveGreedy  ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___exploration_th_36
a2721c50
mars_gym.model.bandit.LinUCB                    ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_05__33
5eb27eb5
mars_gym.model.bandit.LinThompsonSampling       ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___v_sq___0_1__4cd65
ca01d
mars_gym.model.bandit.FixedPolicyPopularItem    ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___arg___2__9cb66452
07
mars_gym.model.bandit.ExploreThenExploit        ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___explore_rounds_a1
80ca8b8f
mars_gym.model.bandit.EpsilonGreedy             ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___epsilon___0_1__46
45466ef3
mars_gym.model.bandit.CustomRewardModelLinUCB   ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___alpha___1e_05__79
66e4e52b
mars_gym.model.bandit.AdaptiveGreedy            ../../output/interaction/Interac
tionTraining/results/InteractionTraining____mars_gym_model_b___exploration_th_d3
663f58fc
```

1.2 DataViz - Plot Images

[7]:
```
#list_bandits['name'] = list_bandits['name'].apply(lambda x: "popular_item" if␣
 ↪x == "PopularItem" else x)
df_models = list_bandits.set_index('name')[['model_list']]
df_models.index
```

[7]:
```
Index(['mars_gym.model.bandit.AdaptiveGreedy',
       'mars_gym.model.bandit.CustomRewardModelLinUCB',
       'mars_gym.model.bandit.EpsilonGreedy',
       'mars_gym.model.bandit.ExploreThenExploit',
       'mars_gym.model.bandit.FixedPolicyPopularItem',
       'mars_gym.model.bandit.LinThompsonSampling',
       'mars_gym.model.bandit.LinUCB',
       'mars_gym.model.bandit.PercentileAdaptiveGreedy',
       'mars_gym.model.bandit.RandomPolicy',
       'mars_gym.model.bandit.SoftmaxExplorer'],
      dtype='object', name='name')
```

[8]:
```
sample_size         = 15000
window_size         = 1000
bandits             = df_models.index

df_group = load_dataset(df_models, bandits, sample_size)
df_group.head()
```

[8]:
```
                                                      iteraction  idx  \
0  InteractionTraining____mars_gym_model_b___exploration_th_2cc29714f4  0
1  InteractionTraining____mars_gym_model_b___exploration_th_2cc29714f4  1
2  InteractionTraining____mars_gym_model_b___exploration_th_2cc29714f4  2
3  InteractionTraining____mars_gym_model_b___exploration_th_2cc29714f4  3
4  InteractionTraining____mars_gym_model_b___exploration_th_2cc29714f4  4

   user  item  reward        ps  index_env  _idx  mean_reward  \
0  559   191   0.0     0.975610  10         1     0.0
1  1118  486   0.0     1.951220  43         2     0.0
2  1677  644   0.0     2.926829  96         3     0.0
3  1908  1263  0.0     3.902439  186        4     0.0
4  1971  1397  0.0     4.878049  297        5     0.0

                              bandit
0  mars_gym.model.bandit.AdaptiveGreedy
1  mars_gym.model.bandit.AdaptiveGreedy
2  mars_gym.model.bandit.AdaptiveGreedy
3  mars_gym.model.bandit.AdaptiveGreedy
4  mars_gym.model.bandit.AdaptiveGreedy
```

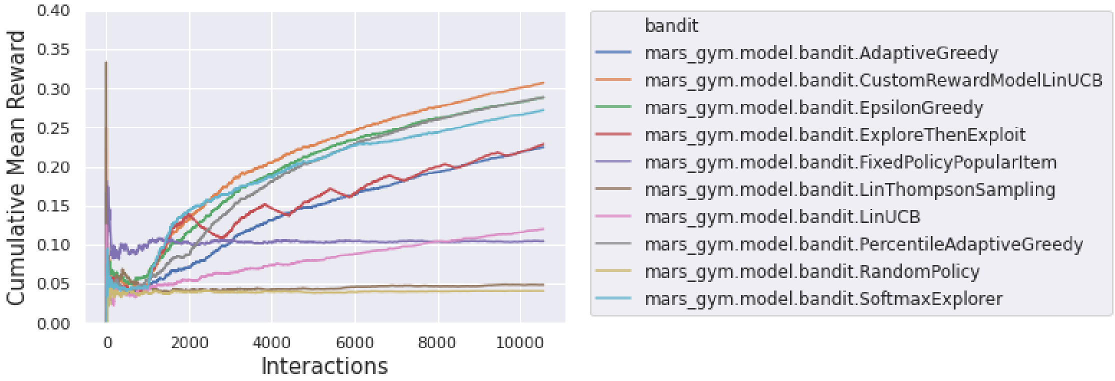
1.2.1 Bandit Simulation Results.

We performed the simulation of each proposed task until the convergence of most methods to observe cumulative mean reward throughout the simulation. Bandit Simulation Results for “Chicago, USA” task.

[9]:

```
df_group_mean = df_group.groupby(['bandit', 'idx']).median().reset_index()

plot_cum_reward(df_group_mean, legend='full', ylim=0.4)
```



[ ]:

```

```

1.2.2 Recommendation Metrics and Off-Policy Evaluation

We evaluated the bandits according to traditional recommen- dation metrics and off-policy metrics in the test subset of the “Chicago, USA” task

[10]:

```
input_metrics = [
  "precision_at_1",
  "ndcg_at_5",
  "coverage_at_5",
  "personalization_at_5",
  "IPS",
  "SNIPS",
  "DirectEstimator",
  "DoublyRobust"
]
```

[11]:

```
# Load Métrics and Params
#
input_models_eval = fetch_results_path().keys()
data_metrics      = load_data_metrics()
data_params       = load_eval_params()
input_params      = []
```

```
df_metrics        = filter_df(data_metrics, input_models_eval, input_metrics,␣
 ↪"")
df_eval_params    = filter_df(data_params, input_models_eval, input_params).
 ↪transpose()
```

```
/media/workspace/DeepFood/deep-reco-gym/src/mars_gym/tools/eval_viz/util.py:44:
FutureWarning:

pandas.io.json.json_normalize is deprecated, use pandas.json_normalize instead
```

[12]:

```
# Calculate confidence metrics
#
confidence_metrics = data_metrics[[c for c in data_metrics.columns if "_C" in␣
 ↪c]]
for c in input_metrics:
    c_column = c+"_C"
    confidence_metrics[c_column] = data_metrics[c_column] if c_column in␣
 ↪data_metrics else None
confidence_metrics = confidence_metrics[[c+"_C" for c in input_metrics]]
```

[13]:

```
# Merge and Mean
#
df_metrics['model_task'] = data_metrics.model_task
df_params['model_task']  = df_params.iteraction.apply(lambda x: x.split("/
 ↪")[-1])

df_result = df_metrics.merge(
    df_params[['bandit_policy_class', 'model_task']].reset_index()
, on='model_task')

df_result = df_result.groupby(['bandit_policy_class']).mean()
df_result
```

[13]:

```
                                              precision_at_1  ndcg_at_5  \
bandit_policy_class
mars_gym.model.bandit.AdaptiveGreedy                0.317639   0.403579
mars_gym.model.bandit.CustomRewardModelLinUCB       0.328249   0.442917
mars_gym.model.bandit.EpsilonGreedy                 0.302056   0.443252
mars_gym.model.bandit.ExploreThenExploit            0.307692   0.418607
mars_gym.model.bandit.FixedPolicy                   0.074271   0.171294
mars_gym.model.bandit.LinThompsonSampling           0.039788   0.137056
mars_gym.model.bandit.LinUCB                        0.075928   0.207314
mars_gym.model.bandit.PercentileAdaptiveGreedy      0.336870   0.438648
mars_gym.model.bandit.RandomPolicy                  0.039788   0.138404
mars_gym.model.bandit.SoftmaxExplorer               0.301724   0.452869
```

```
                                                coverage_at_5  \
bandit_policy_class
mars_gym.model.bandit.AdaptiveGreedy            0.390850
mars_gym.model.bandit.CustomRewardModelLinUCB   0.362862
mars_gym.model.bandit.EpsilonGreedy             0.343287
mars_gym.model.bandit.ExploreThenExploit        0.333071
mars_gym.model.bandit.FixedPolicy               0.373850
mars_gym.model.bandit.LinThompsonSampling       0.423900
mars_gym.model.bandit.LinUCB                    0.271450
mars_gym.model.bandit.PercentileAdaptiveGreedy  0.375563
mars_gym.model.bandit.RandomPolicy              0.389600
mars_gym.model.bandit.SoftmaxExplorer           0.330988

                                                personalization_at_5  \
bandit_policy_class
mars_gym.model.bandit.AdaptiveGreedy            0.767565
mars_gym.model.bandit.CustomRewardModelLinUCB   0.729376
mars_gym.model.bandit.EpsilonGreedy             0.733547
mars_gym.model.bandit.ExploreThenExploit        0.731641
mars_gym.model.bandit.FixedPolicy               0.759891
mars_gym.model.bandit.LinThompsonSampling       0.771211
mars_gym.model.bandit.LinUCB                    0.696228
mars_gym.model.bandit.PercentileAdaptiveGreedy  0.743905
mars_gym.model.bandit.RandomPolicy              0.775653
mars_gym.model.bandit.SoftmaxExplorer           0.725512

                                                     IPS     SNIPS  \
bandit_policy_class
mars_gym.model.bandit.AdaptiveGreedy            0.299046  0.308466
mars_gym.model.bandit.CustomRewardModelLinUCB   0.305576  0.315562
mars_gym.model.bandit.EpsilonGreedy             0.296847  0.295397
mars_gym.model.bandit.ExploreThenExploit        0.296997  0.294139
mars_gym.model.bandit.FixedPolicy               0.076116  0.077219
mars_gym.model.bandit.LinThompsonSampling       0.037423  0.035373
mars_gym.model.bandit.LinUCB                    0.053439  0.056094
mars_gym.model.bandit.PercentileAdaptiveGreedy  0.321505  0.317082
mars_gym.model.bandit.RandomPolicy              0.041457  0.040577
mars_gym.model.bandit.SoftmaxExplorer           0.287223  0.287718

                                                DirectEstimator  DoublyRobust  \
bandit_policy_class
mars_gym.model.bandit.AdaptiveGreedy            0.200657         0.267138
mars_gym.model.bandit.CustomRewardModelLinUCB   0.199502         0.266484
mars_gym.model.bandit.EpsilonGreedy             0.187270         0.254888
mars_gym.model.bandit.ExploreThenExploit        0.190652         0.256225
mars_gym.model.bandit.FixedPolicy               0.084967         0.077790
mars_gym.model.bandit.LinThompsonSampling       0.041869         0.039049
mars_gym.model.bandit.LinUCB                    0.054993         0.051205
mars_gym.model.bandit.PercentileAdaptiveGreedy  0.198404         0.280774
mars_gym.model.bandit.RandomPolicy              0.042541         0.041919
mars_gym.model.bandit.SoftmaxExplorer           0.188744         0.253449

                                                index
bandit_policy_class
mars_gym.model.bandit.AdaptiveGreedy            0
mars_gym.model.bandit.CustomRewardModelLinUCB   0
mars_gym.model.bandit.EpsilonGreedy             0
mars_gym.model.bandit.ExploreThenExploit        0
mars_gym.model.bandit.FixedPolicy               0
mars_gym.model.bandit.LinThompsonSampling       0
mars_gym.model.bandit.LinUCB                    0
mars_gym.model.bandit.PercentileAdaptiveGreedy  0
mars_gym.model.bandit.RandomPolicy              0
mars_gym.model.bandit.SoftmaxExplorer           0
```

```
[14]: df_result.round(3).to_csv('metrics.csv')
```

```
[15]: # fig = plot_bar(df_result, confidence = None, title="Comparison of Recsys␣
       ↪Metrics")

      # fig.update_layout(xaxis_showgrid=False, yaxis_showgrid=False)
      # fig.update_layout(font={'family': 'Courier New, monospace', 'size': 14},␣
       ↪height=550)
      # fig#.show(renderer="svg")

      # #fig.write_image("output/tmp/metrics/recsys_rank.eps")
```

```
[ ]:
```

1.2.3 Fairness Results

We evaluated the SoftmaxExplorer bandit on the “RecSys Cities” task, in the perspective of disparate mistreatment and disparate treatment. We selected a few attributes that we judged to be sensitive for all partners in the marketplace.

```
[16]: # task_id:␣
       ↪InteractionTraining____mars_gym_model_b___logit_multipli_9dd8714dfd_96b9d686ee
```

```
[ ]:
```

Feature - Accessibility

[17]:

```
input_models_eval =␣
 ↪'InteractionTraining____mars_gym_model_b___logit_multipli_9dd8714dfd_96b9d686ee'
list_input_features    = ['accessible parking', 'accessible hotel']
input_metrics      = 'true_positive_rate'
```

[18]:

```
# fig, df_metrics, df_total = plot_fairness_metrics(input_models_eval,␣
 ↪input_features, input_metrics, sub_map = None)
metrics = []
sub_map = None

for input_features in list_input_features:
    df_all_metrics    = load_fairness_metrics().loc[input_models_eval]
    df_instances      = load_fairness_df().loc[input_models_eval]
    df_all_metric_filter = df_all_metrics[df_all_metrics.sub_key.
 ↪isin([input_features])]

    columns          = list(np.unique(['sub_key', 'sub', 'feature',␣
 ↪'total_class', 'total_individuals'] + [input_metrics]))
    if input_metrics+"_C" in df_all_metrics.columns:
        columns.append(input_metrics+"_C")

    df_metrics      = filter_df(df_all_metrics, input_models_eval, columns,␣
 ↪'sub')

    df_metrics      = df_metrics[df_metrics.sub_key.isin([input_features])]
    if sub_map != None:
        df_metrics['sub'] = df_metrics['sub'].map(sub_map)

    df_metrics      = df_metrics.sort_values("feature").set_index("sub")
    df_metrics      = df_metrics.sort_values(input_metrics)
    metrics.append(df_metrics)

metrics = pd.concat(metrics)
metrics
```

[18]:

```
                  feature             sub_key  total_class  total_individuals  \
sub
1    accessible parking.1  accessible parking  1063         2380
0    accessible parking.0  accessible parking  3193         10038
1    accessible hotel.1    accessible hotel    1345         3799
0    accessible hotel.0    accessible hotel    3114         8619

     true_positive_rate  true_positive_rate_C
sub
1    0.222664            0.023616
0    0.297201            0.013806
1    0.242414            0.020492
```

```
0    0.287845            0.014361
```

[19]:

```
metrics.reset_index()
metrics['sub'] = ['parking<br>True', 'parking<br>False', 'hotel<br>True',␣
 ↪'hotel<br>False']
metrics['color'] = [2,2, 5,5]
metrics = metrics.set_index('sub')
metrics
```

[19]:

```
                               feature              sub_key  total_class  \
sub
parking<br>True   accessible parking.1  accessible parking  1063
parking<br>False  accessible parking.0  accessible parking  3193
hotel<br>True     accessible hotel.1    accessible hotel    1345
hotel<br>False    accessible hotel.0    accessible hotel    3114

                  total_individuals  true_positive_rate  true_positive_rate_C  \
sub
parking<br>True   2380               0.222664            0.023616
parking<br>False  10038              0.297201            0.013806
hotel<br>True     3799               0.242414            0.020492
hotel<br>False    8619               0.287845            0.014361

                  color
sub
parking<br>True   2
parking<br>False  2
hotel<br>True     5
hotel<br>False    5
```

[20]:

```
fig = plot_fairness_mistreatment(metrics, input_metrics, title="")

fig.update_layout(shapes=[
  dict(
    type= 'line',
    line=dict(
        width=1,
        dash="dot",
    ),
    xref='paper', x0= 0, x1= 1,
    yref='y',     y0= 1.5, y1= 1.5
  )
])

# Set custom x-axis labels
fig.update_xaxes(nticks=4)
fig.update_layout(yaxis={'categoryorder':'array', 'categoryarray':
 ↪['parking<br>False', 'parking<br>True', 'hotel<br>False', 'hotel<br>True']})
fig.update_layout(xaxis_showgrid=False, yaxis_showgrid=False,␣
 ↪yaxis_title="Accessible")
fig.update_layout(font={'family': 'Courier New, monospace', 'size': 22},␣
 ↪height=550)

fig.show()
#fig.write_image("output/tmp/metrics/paper/recsys_fairnes_{}.eps".
 ↪format("accessible"))
#fig.show(renderer="svg")
```

[ ]:

```
```

Feature - Accommodation Type

[22]:

```
input_models_eval =␣
 ↪'InteractionTraining___mars_gym_model_b___logit_multipli_9dd8714dfd_96b9d686ee'
list_input_features    = ['hotel', 'house / apartment']
input_metrics      = 'true_positive_rate'
```

[23]:

```
metrics = []
for input_features in list_input_features:
    df_all_metrics     = load_fairness_metrics().loc[input_models_eval]
    df_instances       = load_fairness_df().loc[input_models_eval]
    df_all_metric_filter = df_all_metrics[df_all_metrics.sub_key.
 ↪isin([input_features])]

    columns          = list(np.unique(['sub_key', 'sub', 'feature',␣
 ↪'total_class', 'total_individuals'] + [input_metrics]))
    if input_metrics+"_C" in df_all_metrics.columns:
        columns.append(input_metrics+"_C")

    df_metrics       = filter_df(df_all_metrics, input_models_eval, columns,␣
 ↪'sub')

    df_metrics       = df_metrics[df_metrics.sub_key.isin([input_features])]
    if sub_map != None:
        df_metrics['sub'] = df_metrics['sub'].map(sub_map)

    df_metrics       = df_metrics.sort_values("feature").set_index("sub")
    df_metrics       = df_metrics.sort_values(input_metrics)
    metrics.append(df_metrics)

metrics = pd.concat(metrics).reset_index().sort_values(['sub_key', 'sub'])
metrics
```

[23]:
```
  sub              feature             sub_key  total_class  total_individuals  \
0  0    hotel.0              hotel              1649         2111
1  1    hotel.1              hotel              2579         10307
3  0    house / apartment.0  house / apartment  3044         11727
2  1    house / apartment.1  house / apartment  743          691

   true_positive_rate  true_positive_rate_C
0  0.205584            0.020963
1  0.302903            0.013879
3  0.307458            0.013011
2  0.162758            0.030811
```

[24]:
```
#metrics.reset_index().sort_values('sub_key')
metrics['color'] = [5,5, 2,2]
metrics['sub']   = ['hotel<br>False', 'hotel<br>True', 'house/apart<br>False',␣
 ↪'house/apart<br>True']
metrics = metrics.set_index('sub')
metrics
```

[24]:
```
                                  feature            sub_key  total_class  \
sub
hotel<br>False        hotel.0              hotel              1649
hotel<br>True         hotel.1              hotel              2579
house/apart<br>False  house / apartment.0  house / apartment  3044
house/apart<br>True   house / apartment.1  house / apartment  743

                      total_individuals  true_positive_rate  \
sub
hotel<br>False        2111               0.205584
hotel<br>True         10307              0.302903
house/apart<br>False  11727              0.307458
house/apart<br>True   691                0.162758

                      true_positive_rate_C  color
sub
hotel<br>False        0.020963              5
hotel<br>True         0.013879              5
house/apart<br>False  0.013011              2
house/apart<br>True   0.030811              2
```

[25]:
```
fig = plot_fairness_mistreatment(metrics, input_metrics, title="")

fig.update_layout(shapes=[
  dict(
    type= 'line',
    line=dict(
        width=1,
        dash="dot",
    ),
    xref='paper', x0= 0, x1= 1,
    yref='y',     y0= 1.5, y1= 1.5
  )
])

# Set custom x-axis labels
fig.update_xaxes(nticks=4)

fig.update_layout(yaxis={'categoryorder':'array', 'categoryarray':['house/
 ↪apart<br>False', 'house/apart<br>True', 'hotel<br>False', 'hotel<br>True']})
fig.update_layout(xaxis_showgrid=False, yaxis_showgrid=False,␣
 ↪yaxis_title="Business")
fig.update_layout(font={'family': 'Courier New, monospace', 'size': 22},␣
 ↪height=550)


fig.show()
#fig.write_image("output/tmp/metrics/paper/recsys_fairnes_{}.eps".
 ↪format("business"))
#fig.show(renderer="svg")
```

Feature - Business Affinity

[26]:

```
input_models_eval =␣
 ↪'InteractionTraining____mars_gym_model_b___logit_multipli_9dd8714dfd_96b9d686ee'
list_input_features    = ['childcare', 'family friendly']
input_metrics      = 'true_positive_rate'
```

[27]:

```
metrics = []
for input_features in list_input_features:
    df_all_metrics    = load_fairness_metrics().loc[input_models_eval]
    df_instances      = load_fairness_df().loc[input_models_eval]
    df_all_metric_filter = df_all_metrics[df_all_metrics.sub_key.
 ↪isin([input_features])]

    columns          = list(np.unique(['sub_key', 'sub', 'feature',␣
 ↪'total_class', 'total_individuals'] + [input_metrics]))
    if input_metrics+"_C" in df_all_metrics.columns:
        columns.append(input_metrics+"_C")

    df_metrics       = filter_df(df_all_metrics, input_models_eval, columns,␣
 ↪'sub')

    df_metrics       = df_metrics[df_metrics.sub_key.isin([input_features])]
    if sub_map != None:
        df_metrics['sub'] = df_metrics['sub'].map(sub_map)

    df_metrics      = df_metrics.sort_values("feature").set_index("sub")
    df_metrics      = df_metrics.sort_values(input_metrics)
    metrics.append(df_metrics)

metrics = pd.concat(metrics)
metrics
```

[27]:

```
               feature          sub_key  total_class  total_individuals  \
sub
1    childcare.1        childcare        1007         2490
0    childcare.0        childcare        3258         9928
0    family friendly.0  family friendly  2113         3313
1    family friendly.1  family friendly  2384         9105

     true_positive_rate  true_positive_rate_C
sub
1    0.237465            0.024281
0    0.287760            0.013512
0    0.200593            0.017856
1    0.311986            0.014863
```

[28]:

```
metrics.reset_index()
metrics['sub'] = ['childcare<br>True', 'childcare<br>False', 'family␣
 ↪friendly<br>True', 'family friendly<br>False']
metrics['color'] = [2,2, 5,5]
metrics = metrics.set_index('sub')
metrics
```

[28]:

```
                                    feature          sub_key  total_class  \
sub
childcare<br>True         childcare.1        childcare        1007
childcare<br>False        childcare.0        childcare        3258
family friendly<br>True   family friendly.0  family friendly  2113
family friendly<br>False  family friendly.1  family friendly  2384

                          total_individuals  true_positive_rate  \
sub
childcare<br>True         2490               0.237465
childcare<br>False        9928               0.287760
family friendly<br>True   3313               0.200593
family friendly<br>False  9105               0.311986

                          true_positive_rate_C  color
sub
childcare<br>True         0.024281                2
```

```
childcare<br>False        0.013512              2
family friendly<br>True   0.017856              5
family friendly<br>False  0.014863              5
```

[29]:

```python
fig = plot_fairness_mistreatment(metrics, input_metrics, title="")



fig.update_layout(shapes=[
  dict(
    type= 'line',
    line=dict(
        width=1,
        dash="dot",
    ),
    xref='paper', x0= 0, x1= 1,
    yref='y',     y0= 1.5, y1= 1.5
  )
])

# Set custom x-axis labels
fig.update_xaxes(nticks=4)

fig.update_layout(yaxis={'categoryorder':'array', 'categoryarray':
 ↪['childcare<br>False', 'childcare<br>True', 'family friendly<br>False',␣
 ↪'family friendly<br>True']})
fig.update_layout(xaxis_showgrid=False, yaxis_showgrid=False,␣
 ↪yaxis_title="Business")
fig.update_layout(font={'family': 'Courier New, monospace', 'size': 22},␣
 ↪height=550)



fig.show()
#fig.write_image("output/tmp/metrics/paper/recsys_fairnes_{}.eps".
 ↪format("businesscare"))
#fig.show(renderer="svg")
```

[ ]:

```python
```

Feature - City

[30]:

```python
input_models_eval =␣
 ↪'InteractionTraining____mars_gym_model_b___logit_multipli_9dd8714dfd_96b9d686ee'
input_features    = 'city_idx'
input_metrics     = 'true_positive_rate'
```

[31]:
```
sub_map = {'0': "Boston",
            '1': "Como",
            '2': "Chicago",
            '3': "Lausanne",
            '4': "Dublin",
            '5': "Rio de Janeiro",
            '6': "Barcelona",
            '7': "Copenhagen",
            '8': "Vienna",
            '9': "New York",
            '10': "Hong Kong",
            '11': "Vancouver"}

fig, df_metrics, df_total = plot_fairness_metrics(input_models_eval,␣
 ↪input_features, input_metrics, sub_map = sub_map)
```

[32]:
```
fig.show()
#fig.write_image("output/tmp/metrics/recsys_fairnes_{}.eps".
 ↪format(input_features))
```

[33]:
```
df_metrics.head()
```

[33]:
```
                 feature   sub_key  total_class  total_individuals  \
sub
Como         city_idx.1  city_idx  34           30
Dublin       city_idx.4  city_idx  265          1035
Copenhagen   city_idx.7  city_idx  127          572
Vienna       city_idx.8  city_idx  379          1282
Barcelona    city_idx.6  city_idx  640          1652

             true_positive_rate  true_positive_rate_C
sub
Como         0.184211            0.164835
Dublin       0.252908            0.041650
Copenhagen   0.289528            0.057278
Vienna       0.291105            0.037876
Barcelona    0.291691            0.029938
```

[ ]:
```

```

Feature Device

[34]:
```
input_models_eval =␣
 ↪'InteractionTraining___mars_gym_model_b___logit_multipli_9dd8714dfd_96b9d686ee'
df_instances      = load_fairness_df().loc[input_models_eval]

input_features    = 'device_idx'
```

```
input_metrics      = 'true_positive_rate'
input_items_top = True
input_items      = []
```

[ ]:

```

```

[35]:

```
fig = plot_fairness_treatment(df_instances, input_features, input_items,␣
 ↪top=input_items_top, title="", legend={"0": "Desktop", "1": "Mobile", "2":␣
 ↪"Tablet"})
fig.update_layout(font={'family': 'Courier New, monospace', 'size': 21},␣
 ↪height=600)



# Set custom x-axis labels
fig.update_yaxes(nticks=5)
fig.update_layout(xaxis_title="Hotel")

fig.show()
#fig.show(renderer="svg")
#fig.write_image("output/tmp/metrics/paper/recsys_fairnes_impact_{}.eps".
 ↪format(input_features))
```

```
/home/marlesson/anaconda3/lib/python3.7/site-
packages/numpy/core/_methods.py:234: RuntimeWarning:

Degrees of freedom <= 0 for slice

/home/marlesson/anaconda3/lib/python3.7/site-
packages/numpy/core/_methods.py:226: RuntimeWarning:

invalid value encountered in double_scalars
```

[ ]:

```

```