XGBoost Python

In [3]:
```
import numpy as np
import pandas as pd
import xgboost as xgb

import matplotlib
import matplotlib as mpl
import matplotlib.pyplot as plt

from sklearn.datasets._california_housing import fetch_california_housin
g
from sklearn.model_selection import train_test_split
```

In [5]:
```
cal_housing = fetch_california_housing(data_home="/tmp")
print(cal_housing.feature_names, cal_housing.data.shape)
```

```
['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population', 'AveOccu
p', 'Latitude', 'Longitude'] (20640, 8)
```

In [6]:
```
X_train, X_test, y_train, y_test = train_test_split(cal_housing.data,
                                                    cal_housing.target,
                                                    test_size=0.4,
                                                    random_state=123)
```

In [7]:
```
feature_names = ['MedInc', 'AveOccup', 'HouseAge']
feature_ids = [cal_housing.feature_names.index(f) for f in feature_name
s]
```

In [8]:
```
dtrain = xgb.DMatrix(X_train[:, feature_ids].reshape(len(X_train), len(f
eature_ids)), label = y_train)
dtest =  xgb.DMatrix(X_test[:, feature_ids].reshape(len(X_test), len(fea
ture_ids)), label = y_test)
```

In [9]:
```
feature_monotones = [1, -1, 1]

params = {'max_depth': 2,
          'eta': 0.1,
          'silent': 1,
          'nthread': 2,
          'seed': 0,
          'eval_metric': 'rmse',
          'monotone_constraints': '(' + ','.join([str(m) for m in featur
e_monotones]) + ')'
         }

bst_cv = xgb.cv(params, dtrain, 500, nfold = 5, early_stopping_rounds=1
0)
```

```
[23:38:19] WARNING: ../src/learner.cc:576:
Parameters: { "silent" } might not be used.

  This could be a false alarm, with some parameters getting used by lan
guage bindings but
  then being mistakenly passed down to XGBoost core, or some parameter
actually being used
  but getting flagged wrongly here. Please open an issue if you find an
y such cases.


[23:38:19] WARNING: ../src/learner.cc:576:
Parameters: { "silent" } might not be used.

  This could be a false alarm, with some parameters getting used by lan
guage bindings but
  then being mistakenly passed down to XGBoost core, or some parameter
actually being used
  but getting flagged wrongly here. Please open an issue if you find an
y such cases.


[23:38:19] WARNING: ../src/learner.cc:576:
Parameters: { "silent" } might not be used.

  This could be a false alarm, with some parameters getting used by lan
guage bindings but
  then being mistakenly passed down to XGBoost core, or some parameter
actually being used
  but getting flagged wrongly here. Please open an issue if you find an
y such cases.


[23:38:19] WARNING: ../src/learner.cc:576:
Parameters: { "silent" } might not be used.

  This could be a false alarm, with some parameters getting used by lan
guage bindings but
  then being mistakenly passed down to XGBoost core, or some parameter
actually being used
  but getting flagged wrongly here. Please open an issue if you find an
y such cases.


[23:38:19] WARNING: ../src/learner.cc:576:
Parameters: { "silent" } might not be used.

  This could be a false alarm, with some parameters getting used by lan
guage bindings but
  then being mistakenly passed down to XGBoost core, or some parameter
actually being used
  but getting flagged wrongly here. Please open an issue if you find an
y such cases.
```

In [10]:
```
evallist  = [(dtrain, 'train'), (dtest, 'eval')]

evals_result = {}
bst = xgb.train(params, dtrain, num_boost_round = bst_cv.shape[0], evals
_result = evals_result, evals = evallist,  verbose_eval = False)
```

```
[23:38:44] WARNING: ../src/learner.cc:576:
Parameters: { "silent" } might not be used.

  This could be a false alarm, with some parameters getting used by lan
guage bindings but
  then being mistakenly passed down to XGBoost core, or some parameter
actually being used
  but getting flagged wrongly here. Please open an issue if you find an
y such cases.
```

In [11]:
```
print('Number of boosting rounds %d,\
       Training RMSE: %.4f, \
       Testing RMSE: %.4f' % \
       (len(evals_result['train']['rmse']),
        evals_result['train']['rmse'][-1],
        evals_result['eval']['rmse'][-1]))
```

```
Number of boosting rounds 135,       Training RMSE: 0.6954,       Test
ing RMSE: 0.6926
```

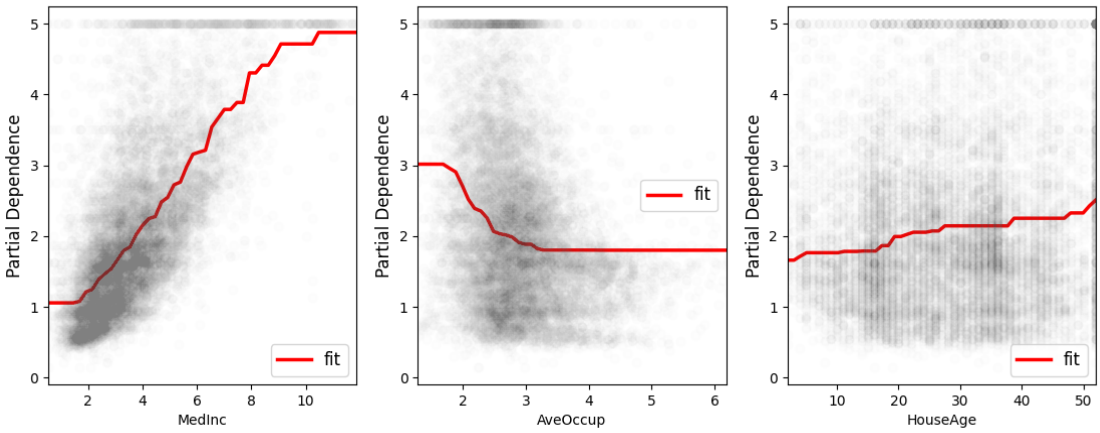
In [12]:
```
def partial_dependency(bst, X, y, feature_ids = [], f_id = -1):

    """
    Calculate the dependency (or partial dependency) of a response varia
ble on a predictor (or multiple predictors)
    1. Sample a grid of values of a predictor.
    2. For each value, replace every row of that predictor with this val
ue, calculate the average prediction.
    """

    X_temp = X.copy()

    grid = np.linspace(np.percentile(X_temp[:, f_id], 0.1),
                       np.percentile(X_temp[:, f_id], 99.5),
                       50)
    y_pred = np.zeros(len(grid))

    if len(feature_ids) == 0 or f_id == -1:
        print('Input error!')
        return
    else:
        for i, val in enumerate(grid):

            X_temp[:, f_id] = val
            data = xgb.DMatrix( X_temp[:, feature_ids].reshape( (len(X_t
emp), len(feature_ids)) ) )

            y_pred[i] = np.average(bst.predict(data, ntree_limit = bst.b
est_ntree_limit))

    return grid, y_pred
```

In [13]:

```
fig, ax = plt.subplots(1, len(feature_names))
fig.set_size_inches(len(feature_names) * 4, 5)
plt.subplots_adjust(left = 0.07, right = 0.94, bottom = 0.15, top = 0.9)

for i, f in enumerate(feature_names):

    grid, y_pred = partial_dependency(bst,
                                      X_train,
                                      y_train,
                                      feature_ids = feature_ids,
                                      f_id = cal_housing.feature_names.i
ndex(f)
                                     )

    if i ==0 and len(feature_names) == 1:
        axis = ax
    else:
        axis = ax[i]

    axis.plot(grid, y_pred, '-', color = 'red', linewidth = 2.5, label
='fit')
    axis.plot(X_train[:, cal_housing.feature_names.index(f)], y_train,
'o', color = 'grey', alpha = 0.01)

    axis.set_xlim(min(grid), max(grid))
    axis.set_xlabel(f, fontsize = 10)
    axis.set_ylabel('Partial Dependence', fontsize = 12)

    handles, labels = axis.get_legend_handles_labels()
    axis.legend(handles, labels, loc = 'best', fontsize = 12)
    plt.savefig('w_constraint_three_feature.png')
```

```
/usr/lib/python3.6/site-packages/xgboost/core.py:108: UserWarning: ntre
e_limit is deprecated, use `iteration_range` or model slicing instead.
  UserWarning
[<matplotlib.lines.Line2D object at 0x7fb3af679eb8>]
[<matplotlib.lines.Line2D object at 0x7fb3c8c26c18>]
(0.536, 11.854603999999894)
Text(0.5, 0, 'MedInc')
Text(0, 0.5, 'Partial Dependence')
<matplotlib.legend.Legend object at 0x7fb3c8bd41d0>
/usr/lib/python3.6/site-packages/xgboost/core.py:108: UserWarning: ntre
e_limit is deprecated, use `iteration_range` or model slicing instead.
  UserWarning
[<matplotlib.lines.Line2D object at 0x7fb3c8bd4e10>]
[<matplotlib.lines.Line2D object at 0x7fb3bf9e1eb8>]
(1.2832211987226727, 6.183123919308341)
Text(0.5, 45.72222222222223, 'AveOccup')
Text(415.1928104575163, 0.5, 'Partial Dependence')
<matplotlib.legend.Legend object at 0x7fb3ce29cf60>
/usr/lib/python3.6/site-packages/xgboost/core.py:108: UserWarning: ntre
e_limit is deprecated, use `iteration_range` or model slicing instead.
  UserWarning
[<matplotlib.lines.Line2D object at 0x7fb3bf9fc320>]
[<matplotlib.lines.Line2D object at 0x7fb3c8a886d8>]
(2.0, 52.0)
Text(0.5, 45.72222222222223, 'HouseAge')
Text(783.6633986928105, 0.5, 'Partial Dependence')
<matplotlib.legend.Legend object at 0x7fb3c8a8b8d0>
```

In [14]:
```
%matplot plt
```



XGBoost Ray

In [16]:
```
from xgboost_ray import RayDMatrix, RayParams, train
```

In [17]:
```
train_x = X_train[:, feature_ids].reshape(len(X_train), len(feature_id
s))
train_y = y_train
train_set = RayDMatrix(train_x, train_y)

test_x = X_test[:, feature_ids].reshape(len(X_test), len(feature_ids))
test_y = y_test
test_set = RayDMatrix(test_x, test_y)
```

In [18]:
```
params
```

```
{'max_depth': 2, 'eta': 0.1, 'silent': 1, 'nthread': 2, 'seed': 0, 'eva
l_metric': 'rmse', 'monotone_constraints': '(1,-1,1)'}
```

In [19]:
```
train_x
```

```
array([[ 4.7984    ,  2.86600496, 37.        ],
       [ 5.5714    ,  3.22791293, 34.        ],
       [ 5.1814    ,  2.86779661, 46.        ],
       ...,
       [ 5.6306    ,  3.49354376,  5.        ],
       [ 3.875     ,  1.7208589 , 44.        ],
       [ 2.5156    ,  3.56609195, 20.        ]])
```

In [20]:
```
evallist  = [(train_set, 'train'), (test_set, 'eval')]

# bst = xgb.train(
#     params,
#     dtrain, num_boost_round = bst_cv.shape[0], evals_result = evals_re
sult, evals = evallist,  verbose_eval = False)

bst = train(
    params,
    train_set,
    evals_result=evals_result,
    evals = evallist,
    num_boost_round = bst_cv.shape[0],
#     evals=[(train_set, "train")],
    verbose_eval=False,
    ray_params=RayParams(
        num_actors=1,  # Number of remote actors
        cpus_per_actor=2))
```

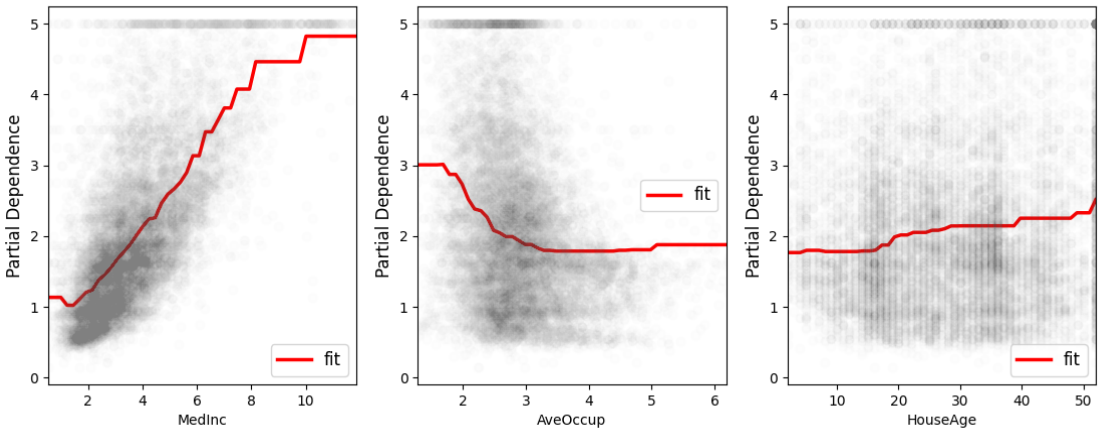
```
Usage stats collection is enabled by default for nightly wheels. To dis
able this, run the following command: `ray disable-usage-stats` before
starting Ray. See https://docs.ray.io/en/master/cluster/usage-stats.htm
l for more details.
2023-02-08 23:48:42,120 INFO worker.py:1524 -- Started a local Ray inst
ance. View the dashboard at http://127.0.0.1:8265
/usr/lib/python3.6/site-packages/xgboost_ray/main.py:439: UserWarning:
`num_actors` in `ray_params` is smaller than 2 (1). XGBoost will NOT be
distributed!
  f"`num_actors` in `ray_params` is smaller than 2 "
2023-02-08 23:48:48,947 INFO main.py:1009 -- [RayXGBoost] Created 1 new
actors (1 total actors). Waiting until actors are ready for training.
2023-02-08 23:48:53,779 INFO main.py:1054 -- [RayXGBoost] Starting XGBo
ost training.
(_RemoteRayXGBoostActor pid=2316) [23:48:53] task [xgboost.ray]:1400180
40759520 got new rank 0
(_RemoteRayXGBoostActor pid=2316) [23:48:53] WARNING: ../src/learner.c
c:576:
(_RemoteRayXGBoostActor pid=2316) Parameters: { "silent" } might not be
used.
(_RemoteRayXGBoostActor pid=2316)
(_RemoteRayXGBoostActor pid=2316)   This could be a false alarm, with s
ome parameters getting used by language bindings but
(_RemoteRayXGBoostActor pid=2316)   then being mistakenly passed down t
o XGBoost core, or some parameter actually being used
(_RemoteRayXGBoostActor pid=2316)   but getting flagged wrongly here. P
lease open an issue if you find any such cases.
(_RemoteRayXGBoostActor pid=2316)
(_RemoteRayXGBoostActor pid=2316)
2023-02-08 23:48:58,553 INFO main.py:1553 -- [RayXGBoost] Finished XGBo
ost training on training data with total N=12,384 in 9.93 seconds (4.76
pure XGBoost training time).
```

In [21]:

```
fig, ax = plt.subplots(1, len(feature_names))
fig.set_size_inches(len(feature_names) * 4, 5)
plt.subplots_adjust(left = 0.07, right = 0.94, bottom = 0.15, top = 0.9)

for i, f in enumerate(feature_names):

    grid, y_pred = partial_dependency(bst,
                                      X_train,
                                      y_train,
                                      feature_ids = feature_ids,
                                      f_id = cal_housing.feature_names.i
ndex(f)
                                     )

    if i ==0 and len(feature_names) == 1:
        axis = ax
    else:
        axis = ax[i]

    axis.plot(grid, y_pred, '-', color = 'red', linewidth = 2.5, label
='fit')
    axis.plot(X_train[:, cal_housing.feature_names.index(f)], y_train,
'o', color = 'grey', alpha = 0.01)

    axis.set_xlim(min(grid), max(grid))
    axis.set_xlabel(f, fontsize = 10)
    axis.set_ylabel('Partial Dependence', fontsize = 12)

    handles, labels = axis.get_legend_handles_labels()
    axis.legend(handles, labels, loc = 'best', fontsize = 12)
    plt.savefig('w_constraint_three_feature.png')
```

```
/usr/lib/python3.6/site-packages/xgboost/core.py:108: UserWarning: ntre
e_limit is deprecated, use `iteration_range` or model slicing instead.
  UserWarning
[<matplotlib.lines.Line2D object at 0x7fb338046518>]
[<matplotlib.lines.Line2D object at 0x7fb32fd76b00>]
(0.536, 11.854603999999894)
Text(0.5, 0, 'MedInc')
Text(0, 0.5, 'Partial Dependence')
<matplotlib.legend.Legend object at 0x7fb32fd76c50>
/usr/lib/python3.6/site-packages/xgboost/core.py:108: UserWarning: ntre
e_limit is deprecated, use `iteration_range` or model slicing instead.
  UserWarning
[<matplotlib.lines.Line2D object at 0x7fb32fd76da0>]
[<matplotlib.lines.Line2D object at 0x7fb32fd55240>]
(1.2832211987226727, 6.183123919308341)
Text(0.5, 45.72222222222223, 'AveOccup')
Text(415.1928104575163, 0.5, 'Partial Dependence')
<matplotlib.legend.Legend object at 0x7fb32fd4e588>
/usr/lib/python3.6/site-packages/xgboost/core.py:108: UserWarning: ntre
e_limit is deprecated, use `iteration_range` or model slicing instead.
  UserWarning
[<matplotlib.lines.Line2D object at 0x7fb32fd5c390>]
[<matplotlib.lines.Line2D object at 0x7fb32fcee7f0>]
(2.0, 52.0)
Text(0.5, 45.72222222222223, 'HouseAge')
Text(783.6633986928105, 0.5, 'Partial Dependence')
<matplotlib.legend.Legend object at 0x7fb32fd04860>
```

In [22]:
```
%matplot plt
```



In [ ]: