# autogluon-2

November 17, 2024

## 1 Using AWS Autogluon to predict rent prices in Canada based on city, province, latitude, longitude, lease_term, type, price, beds, baths, sq_feet, furnishing, availability_date, smoking, and whether cats and dogs are allowed

[1]:
```
# !pip install autogluon
```

[3]:
```
import pandas as pd
import numpy as np
```

[4]:
```
df = pd.read_csv("rentfaster.csv")
```

[5]:
```
df.head()
```

[5]:
```
   rentfaster_id     city province           address   latitude   longitude  \
0         468622  Airdrie  Alberta  69 Gateway Dr NE  51.305962 -114.012515
1         468622  Airdrie  Alberta  69 Gateway Dr NE  51.305962 -114.012515
2         468622  Airdrie  Alberta  69 Gateway Dr NE  51.305962 -114.012515
3         468622  Airdrie  Alberta  69 Gateway Dr NE  51.305962 -114.012515
4         468622  Airdrie  Alberta  69 Gateway Dr NE  51.305962 -114.012515

  lease_term       type   price    beds baths sq_feet  \
0  Long Term  Townhouse  2495.0  2 Beds   2.5    1403
1  Long Term  Townhouse  2695.0  3 Beds   2.5    1496
2  Long Term  Townhouse  2295.0  2 Beds   2.5    1180
3  Long Term  Townhouse  2095.0  2 Beds   2.5    1403
4  Long Term  Townhouse  2495.0  2 Beds   2.5    1403

                                                link   furnishing  \
0  /ab/airdrie/rentals/townhouse/2-bedrooms/pet-f…  Unfurnished
1  /ab/airdrie/rentals/townhouse/2-bedrooms/pet-f…  Unfurnished
2  /ab/airdrie/rentals/townhouse/2-bedrooms/pet-f…  Unfurnished
3  /ab/airdrie/rentals/townhouse/2-bedrooms/pet-f…  Unfurnished
4  /ab/airdrie/rentals/townhouse/2-bedrooms/pet-f…  Unfurnished

  availability_date      smoking  cats  dogs
```

```
0         Immediate  Non-Smoking  True  True
1         Immediate  Non-Smoking  True  True
2         Immediate  Non-Smoking  True  True
3       November 18  Non-Smoking  True  True
4         Immediate  Non-Smoking  True  True
```

[6]:
```
df.columns
```

[6]:
```
Index(['rentfaster_id', 'city', 'province', 'address', 'latitude', 'longitude',
       'lease_term', 'type', 'price', 'beds', 'baths', 'sq_feet', 'link',
       'furnishing', 'availability_date', 'smoking', 'cats', 'dogs'],
      dtype='object')
```

[7]:
```
# Remove unnecessary columns
df = df.drop(axis=1, columns = ["address", "link", "rentfaster_id"])
```

[8]:
```
df.head()
```

[8]:
```
       city province   latitude   longitude lease_term       type   price  \
0  Airdrie  Alberta  51.305962 -114.012515  Long Term  Townhouse  2495.0
1  Airdrie  Alberta  51.305962 -114.012515  Long Term  Townhouse  2695.0
2  Airdrie  Alberta  51.305962 -114.012515  Long Term  Townhouse  2295.0
3  Airdrie  Alberta  51.305962 -114.012515  Long Term  Townhouse  2095.0
4  Airdrie  Alberta  51.305962 -114.012515  Long Term  Townhouse  2495.0

     beds baths sq_feet   furnishing availability_date      smoking  cats  \
0  2 Beds   2.5    1403  Unfurnished         Immediate  Non-Smoking  True
1  3 Beds   2.5    1496  Unfurnished         Immediate  Non-Smoking  True
2  2 Beds   2.5    1180  Unfurnished         Immediate  Non-Smoking  True
3  2 Beds   2.5    1403  Unfurnished       November 18  Non-Smoking  True
4  2 Beds   2.5    1403  Unfurnished         Immediate  Non-Smoking  True

   dogs
0  True
1  True
2  True
3  True
4  True
```

[9]:
```
from autogluon.tabular import TabularDataset, TabularPredictor
```

[10]:
```
# Regressing for the price
target = "price"
```

[11]:
```
train_data = TabularDataset(df)
```

[12]:
```
# Sample 70% randomly for the train data
subsample_size = int(0.7*len(df))
```

[13]:
```
train_data = train_data.sample(n=subsample_size, random_state=0)
train_data.head()
```

[13]:

| | city | province | latitude | longitude | lease_term | type | price | beds | baths | sq_feet | furnishing | availability_date | smoking | cats | dogs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12512 | Calgary | Alberta | 51.032613 | -114.062190 | Long Term | Condo Unit | 2150.0 | 1 Bed | 1 | 763 | Unfurnished | July 01 | Non-Smoking | False | False |
| 24216 | Montréal | Quebec | 45.505681 | -73.563915 | Long Term | Apartment | 3390.0 | 2 Beds | 1 | 1058 | Unfurnished | Immediate | Non-Smoking | True | True |
| 13161 | Calgary | Alberta | 50.859881 | -114.078010 | Long Term | Basement | 1300.0 | 1 Bed | 1 | 700 | Unfurnished | July 01 | Non-Smoking | True | True |
| 2415 | Calgary | Alberta | 51.134793 | -113.949708 | Long Term | Condo Unit | 2150.0 | 2 Beds | 2 | 980 | Unfurnished | Immediate | Non-Smoking | False | False |
| 7519 | Edmonton | Alberta | 53.544671 | -113.577309 | 12 months | House | 1700.0 | 2 Beds | 2 | 800 | Unfurnished | July 01 | Non-Smoking | True | True |

[13]:
```
# Training a fast model
predictor_price = TabularPredictor(label=target).fit(train_data, time_limit=60)
```

```
No path specified. Models will be saved in: "AutogluonModels/ag-20241117_161814"
Verbosity: 2 (Standard Logging)
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.10.12
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Thu Jun 27 21:05:47 UTC 2024
CPU Count:          2
Memory Avail:       10.88 GB / 12.67 GB (85.9%)
Disk Space Avail:   65.91 GB / 107.72 GB (61.2%)
===================================================
No presets specified! To achieve strong results with AutoGluon, it is
recommended to use the available presets.
        Recommended Presets (For more details refer to
https://auto.gluon.ai/stable/tutorials/tabular/tabular-essentials.html#presets):
        presets='best_quality'   : Maximize accuracy. Default time_limit=3600.
```

```
        presets='high_quality'   : Strong accuracy with fast inference speed.
Default time_limit=3600.
        presets='good_quality'   : Good accuracy with very fast inference speed.
Default time_limit=3600.
        presets='medium_quality' : Fast training time, ideal for initial
prototyping.
Beginning AutoGluon training … Time limit = 60s
AutoGluon will save models to "AutogluonModels/ag-20241117_161814"
Train Data Rows:    18039
Train Data Columns: 14
Label Column:       price
AutoGluon infers your prediction problem is: 'regression' (because dtype of
label-column == float and many unique label-values observed).
        Label info (max, min, mean, stddev): (29990.0, 0.0, 2144.43951,
972.07588)
        If 'regression' is not the correct problem_type, please manually specify
the problem_type parameter during Predictor init (You may specify problem_type
as one of: ['binary', 'multiclass', 'regression', 'quantile'])
Problem Type:       regression
Preprocessing data …
Using Feature Generators to preprocess the data …
Fitting AutoMLPipelineFeatureGenerator…
        Available Memory:                    11156.12 MB
        Train Data (Original)  Memory Usage: 12.43 MB (0.1% of available memory)
        Inferring data type of each feature based on column values. Set
feature_metadata_in to manually specify special dtypes of the features.
        Stage 1 Generators:
                Fitting AsTypeFeatureGenerator…
        Stage 2 Generators:
                Fitting FillNaFeatureGenerator…
        Stage 3 Generators:
                Fitting IdentityFeatureGenerator…
                Fitting CategoryFeatureGenerator…
                        Fitting CategoryMemoryMinimizeFeatureGenerator…
        Stage 4 Generators:
                Fitting DropUniqueFeatureGenerator…
        Stage 5 Generators:
                Fitting DropDuplicatesFeatureGenerator…
        Types of features in original data (raw dtype, special dtypes):
                ('float', [])  :  2 | ['latitude', 'longitude']
                ('object', []) : 12 | ['city', 'province', 'lease_term', 'type',
'beds', …]
        Types of features in processed data (raw dtype, special dtypes):
                ('category', []) : 12 | ['city', 'province', 'lease_term',
'type', 'beds', …]
                ('float', [])    :  2 | ['latitude', 'longitude']
        0.6s = Fit runtime
        14 features in original data used to generate 14 features in processed
```

```
data.
        Train Data (Processed) Memory Usage: 0.52 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.63s …
AutoGluon will gauge predictive performance using evaluation metric:
'root_mean_squared_error'
        This metric's sign has been flipped to adhere to being higher_is_better.
The metric score can be multiplied by -1 to get the metric value.
        To change this, specify the eval_metric parameter of Predictor()
Automatically generating train/validation split with holdout_frac=0.1, Train
Rows: 16235, Val Rows: 1804
User-specified model hyperparameters to be fit:
{
        'NN_TORCH': {},
        'GBM': [{'extra_trees': True, 'ag_args': {'name_suffix': 'XT'}}, {},
'GBMLarge'],
        'CAT': {},
        'XGB': {},
        'FASTAI': {},
        'RF': [{'criterion': 'gini', 'ag_args': {'name_suffix': 'Gini',
'problem_types': ['binary', 'multiclass']}}, {'criterion': 'entropy', 'ag_args':
{'name_suffix': 'Entr', 'problem_types': ['binary', 'multiclass']}},
{'criterion': 'squared_error', 'ag_args': {'name_suffix': 'MSE',
'problem_types': ['regression', 'quantile']}}],
        'XT': [{'criterion': 'gini', 'ag_args': {'name_suffix': 'Gini',
'problem_types': ['binary', 'multiclass']}}, {'criterion': 'entropy', 'ag_args':
{'name_suffix': 'Entr', 'problem_types': ['binary', 'multiclass']}},
{'criterion': 'squared_error', 'ag_args': {'name_suffix': 'MSE',
'problem_types': ['regression', 'quantile']}}],
        'KNN': [{'weights': 'uniform', 'ag_args': {'name_suffix': 'Unif'}},
{'weights': 'distance', 'ag_args': {'name_suffix': 'Dist'}}],
}
Fitting 11 L1 models …
Fitting model: KNeighborsUnif … Training model for up to 59.37s of the 59.37s
of remaining time.
        -683.1182        = Validation score   (-root_mean_squared_error)
        2.96s    = Training   runtime
        0.02s    = Validation runtime
Fitting model: KNeighborsDist … Training model for up to 56.38s of the 56.37s
of remaining time.
        -612.811         = Validation score   (-root_mean_squared_error)
        0.02s    = Training   runtime
        0.02s    = Validation runtime
Fitting model: LightGBMXT … Training model for up to 56.33s of the 56.33s of
remaining time.
/usr/local/lib/python3.10/dist-packages/dask/dataframe/__init__.py:42:
FutureWarning:
Dask dataframe query planning is disabled because dask-expr is not installed.
```

```
You can install it with `pip install dask[dataframe]` or `conda install dask`.
This will raise in a future version.

  warnings.warn(msg, FutureWarning)

[1000]  valid_set's rmse: 423.81
[2000]  valid_set's rmse: 415.834
[3000]  valid_set's rmse: 410.2
[4000]  valid_set's rmse: 407.48
[5000]  valid_set's rmse: 404.523
[6000]  valid_set's rmse: 402.728
[7000]  valid_set's rmse: 400.988
[8000]  valid_set's rmse: 399.787
[9000]  valid_set's rmse: 399.434
[10000] valid_set's rmse: 398.293

        -398.293         = Validation score   (-root_mean_squared_error)
        30.04s   = Training   runtime
        10.43s   = Validation runtime
Fitting model: LightGBM … Training model for up to 14.44s of the 14.44s of
remaining time.

[1000]  valid_set's rmse: 378.305
[2000]  valid_set's rmse: 365.213
[3000]  valid_set's rmse: 360.246
[4000]  valid_set's rmse: 357.389
[5000]  valid_set's rmse: 355.923

        Ran out of time, early stopping on iteration 5875. Best iteration is:
        [5863]  valid_set's rmse: 354.889
        -354.8894        = Validation score   (-root_mean_squared_error)
        15.51s   = Training   runtime
        1.97s    = Validation runtime
Fitting model: WeightedEnsemble_L2 … Training model for up to 59.37s of the
-4.59s of remaining time.
        Ensemble Weights: {'LightGBM': 0.737, 'KNeighborsDist': 0.158,
'LightGBMXT': 0.105}
        -341.2729        = Validation score   (-root_mean_squared_error)
        0.02s    = Training   runtime
        0.0s     = Validation runtime
AutoGluon training complete, total runtime = 64.66s … Best model:
WeightedEnsemble_L2 | Estimated inference throughput: 145.3 rows/s (1804 batch
size)
TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("AutogluonModels/ag-20241117_161814")
```

[14]:
```
# Creating test data for the rows that have not been sampled by the train data
test_data = TabularDataset(df.drop(train_data.index))
y_test = test_data[target]
```

[15]:
```
# predictor = TabularPredictor.load("agModels-predictprice")

# Predicting rent prices for the test data
y_pred = predictor_price.predict(test_data)
print("Predictions:  \n", y_pred)
perf = predictor_price.evaluate_predictions(y_true=y_test, y_pred=y_pred,␣
 ↪auxiliary_metrics=True)
```

```
Predictions:
 10       2736.225098
11        2212.453857
13        2559.374023
19        2113.462646
24        2080.962402
             …
25748     1267.991943
25755     1058.244019
25762      945.008179
25763     1113.939209
25764      945.008179
Name: price, Length: 7732, dtype: float32
```

[16]:
```
# Looking at the performance
perf
```

[16]:
```
{'root_mean_squared_error': -394.74039693281304,
 'mean_squared_error': -155819.9809706748,
 'mean_absolute_error': -183.9261484073692,
 'r2': 0.8358683478621021,
 'pearsonr': 0.9144924898179707,
 'median_absolute_error': -98.54681396484375}
```

[18]:
```
# retraining the model using best_quality and time limit of 10 minutes,␣
 ↪focusing on MAE
predictor_price = TabularPredictor(label=target, path="agModels-predictprice",
                                   eval_metric="root_mean_squared_error").
 ↪fit(train_data,
                                   presets="best_quality", time_limit=600)
```

```
Verbosity: 2 (Standard Logging)
=================== System Info ===================
AutoGluon Version:  1.1.1
Python Version:     3.10.12
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #1 SMP PREEMPT_DYNAMIC Thu Jun 27 21:05:47 UTC 2024
CPU Count:          2
```

```
Memory Avail:       10.71 GB / 12.67 GB (84.5%)
Disk Space Avail:   65.87 GB / 107.72 GB (61.1%)
=================================================
Presets specified: ['best_quality']
Setting dynamic_stacking from 'auto' to True. Reason: Enable dynamic_stacking
when use_bag_holdout is disabled. (use_bag_holdout=False)
Stack configuration (auto_stack=True): num_stack_levels=1, num_bag_folds=8,
num_bag_sets=1
DyStack is enabled (dynamic_stacking=True). AutoGluon will try to determine
whether the input data is affected by stacked overfitting and enable or disable
stacking as a consequence.
        This is used to identify the optimal `num_stack_levels` value. Copies of
AutoGluon will be fit on subsets of the data. Then holdout validation data is
used to detect stacked overfitting.
        Running DyStack for up to 150s of the 600s of remaining time (25%).
        Running DyStack sub-fit in a ray process to avoid memory leakage.
Enabling ray logging (enable_ray_logging=True). Specify
`ds_args={'enable_ray_logging': False}` if you experience logging issues.
2024-11-17 16:21:56,940 INFO worker.py:1743 -- Started a local Ray instance.
View the dashboard at 127.0.0.1:8265
                Context path: "agModels-predictprice/ds_sub_fit/sub_fit_ho"
(_dystack pid=4697) Running DyStack sub-fit …
(_dystack pid=4697) Beginning AutoGluon training … Time limit = 141s
(_dystack pid=4697) AutoGluon will save models to "agModels-
predictprice/ds_sub_fit/sub_fit_ho"
(_dystack pid=4697) Train Data Rows:    16034
(_dystack pid=4697) Train Data Columns: 14
(_dystack pid=4697) Label Column:       price
(_dystack pid=4697) Problem Type:       regression
(_dystack pid=4697) Preprocessing data …
(_dystack pid=4697) Using Feature Generators to preprocess the data …
(_dystack pid=4697) Fitting AutoMLPipelineFeatureGenerator…
(_dystack pid=4697)     Available Memory:                    10565.92 MB
(_dystack pid=4697)     Train Data (Original)  Memory Usage: 11.04 MB
(0.1% of available memory)
(_dystack pid=4697)     Inferring data type of each feature based on
column values. Set feature_metadata_in to manually specify special dtypes of the
features.
(_dystack pid=4697)     Stage 1 Generators:
(_dystack pid=4697)             Fitting AsTypeFeatureGenerator…
(_dystack pid=4697)     Stage 2 Generators:
(_dystack pid=4697)             Fitting FillNaFeatureGenerator…
(_dystack pid=4697)     Stage 3 Generators:
(_dystack pid=4697)             Fitting IdentityFeatureGenerator…
(_dystack pid=4697)             Fitting CategoryFeatureGenerator…
(_dystack pid=4697)                     Fitting
CategoryMemoryMinimizeFeatureGenerator…
(_dystack pid=4697)     Stage 4 Generators:
```

```
(_dystack pid=4697)             Fitting DropUniqueFeatureGenerator…
(_dystack pid=4697)     Stage 5 Generators:
(_dystack pid=4697)             Fitting
DropDuplicatesFeatureGenerator…
(_dystack pid=4697)     Types of features in original data (raw dtype,
special dtypes):
(_dystack pid=4697)             ('float', [])  :  2 | ['latitude',
'longitude']
(_dystack pid=4697)             ('object', []) : 12 | ['city',
'province', 'lease_term', 'type', 'beds', …]
(_dystack pid=4697)     Types of features in processed data (raw dtype,
special dtypes):
(_dystack pid=4697)             ('category', []) : 12 | ['city',
'province', 'lease_term', 'type', 'beds', …]
(_dystack pid=4697)             ('float', [])    :  2 | ['latitude',
'longitude']
(_dystack pid=4697)     0.3s = Fit runtime
(_dystack pid=4697)     14 features in original data used to generate 14
features in processed data.
(_dystack pid=4697)     Train Data (Processed) Memory Usage: 0.46 MB
(0.0% of available memory)
(_dystack pid=4697) Data preprocessing and feature engineering runtime
= 0.28s …
(_dystack pid=4697) AutoGluon will gauge predictive performance using
evaluation metric: 'root_mean_squared_error'
(_dystack pid=4697)     This metric's sign has been flipped to adhere to
being higher_is_better. The metric score can be multiplied by -1 to get the
metric value.
(_dystack pid=4697)     To change this, specify the eval_metric
parameter of Predictor()
(_dystack pid=4697) Large model count detected (112 configs) … Only
displaying the first 3 models of each family. To see all, set `verbosity=3`.
(_dystack pid=4697) User-specified model hyperparameters to be fit:
(_dystack pid=4697) {
(_dystack pid=4697)     'NN_TORCH': [{}, {'activation': 'elu',
'dropout_prob': 0.10077639529843717, 'hidden_size': 108, 'learning_rate':
0.002735937344002146, 'num_layers': 4, 'use_batchnorm': True, 'weight_decay':
1.356433327634438e-12, 'ag_args': {'name_suffix': '_r79', 'priority': -2}},
{'activation': 'elu', 'dropout_prob': 0.11897478034205347, 'hidden_size': 213,
'learning_rate': 0.0010474382260641949, 'num_layers': 4, 'use_batchnorm': False,
'weight_decay': 5.594471067786272e-10, 'ag_args': {'name_suffix': '_r22',
'priority': -7}}],
(_dystack pid=4697)     'GBM': [{'extra_trees': True, 'ag_args':
{'name_suffix': 'XT'}}, {}, 'GBMLarge'],
(_dystack pid=4697)     'CAT': [{}, {'depth': 6, 'grow_policy':
'SymmetricTree', 'l2_leaf_reg': 2.1542798306067823, 'learning_rate':
0.06864209415792857, 'max_ctr_complexity': 4, 'one_hot_max_size': 10, 'ag_args':
{'name_suffix': '_r177', 'priority': -1}}, {'depth': 8, 'grow_policy':
```

```
'Depthwise', 'l2_leaf_reg': 2.7997999596449104, 'learning_rate':
0.031375015734637225, 'max_ctr_complexity': 2, 'one_hot_max_size': 3, 'ag_args':
{'name_suffix': '_r9', 'priority': -5}}],
(_dystack pid=4697)     'XGB': [{}, {'colsample_bytree':
0.6917311125174739, 'enable_categorical': False, 'learning_rate':
0.018063876087523967, 'max_depth': 10, 'min_child_weight': 0.6028633586934382,
'ag_args': {'name_suffix': '_r33', 'priority': -8}}, {'colsample_bytree':
0.6628423832084077, 'enable_categorical': False, 'learning_rate':
0.08775715546881824, 'max_depth': 5, 'min_child_weight': 0.6294123374222513,
'ag_args': {'name_suffix': '_r89', 'priority': -16}}],
(_dystack pid=4697)     'FASTAI': [{}, {'bs': 256, 'emb_drop':
0.5411770367537934, 'epochs': 43, 'layers': [800, 400], 'lr':
0.01519848858318159, 'ps': 0.23782946566604385, 'ag_args': {'name_suffix':
'_r191', 'priority': -4}}, {'bs': 2048, 'emb_drop': 0.05070411322605811,
'epochs': 29, 'layers': [200, 100], 'lr': 0.08974235041576624, 'ps':
0.10393466140748028, 'ag_args': {'name_suffix': '_r102', 'priority': -11}}],
(_dystack pid=4697)     'RF': [{'criterion': 'gini', 'ag_args':
{'name_suffix': 'Gini', 'problem_types': ['binary', 'multiclass']}},
{'criterion': 'entropy', 'ag_args': {'name_suffix': 'Entr', 'problem_types':
['binary', 'multiclass']}}, {'criterion': 'squared_error', 'ag_args':
{'name_suffix': 'MSE', 'problem_types': ['regression', 'quantile']}}],
(_dystack pid=4697)     'XT': [{'criterion': 'gini', 'ag_args':
{'name_suffix': 'Gini', 'problem_types': ['binary', 'multiclass']}},
{'criterion': 'entropy', 'ag_args': {'name_suffix': 'Entr', 'problem_types':
['binary', 'multiclass']}}, {'criterion': 'squared_error', 'ag_args':
{'name_suffix': 'MSE', 'problem_types': ['regression', 'quantile']}}],
(_dystack pid=4697)     'KNN': [{'weights': 'uniform', 'ag_args':
{'name_suffix': 'Unif'}}, {'weights': 'distance', 'ag_args': {'name_suffix':
'Dist'}}],
(_dystack pid=4697) }
(_dystack pid=4697) AutoGluon will fit 2 stack levels (L1 to L2) …
(_dystack pid=4697) Fitting 108 L1 models …
(_dystack pid=4697) Fitting model: KNeighborsUnif_BAG_L1 … Training
model for up to 94.07s of the 141.13s of remaining time.
(_dystack pid=4697)     -802.1015        = Validation score
(-root_mean_squared_error)
(_dystack pid=4697)     0.02s    = Training   runtime
(_dystack pid=4697)     0.06s    = Validation runtime
(_dystack pid=4697) Fitting model: KNeighborsDist_BAG_L1 … Training
model for up to 91.93s of the 138.98s of remaining time.
(_dystack pid=4697)     -685.5784        = Validation score
(-root_mean_squared_error)
(_dystack pid=4697)     0.03s    = Training   runtime
(_dystack pid=4697)     0.05s    = Validation runtime
(_dystack pid=4697) Fitting model: LightGBMXT_BAG_L1 … Training model
for up to 91.83s of the 138.88s of remaining time.
(_dystack pid=4697)     Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
```

```
memory=0.06%)
(_ray_fit pid=4861) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:
(_ray_fit pid=4861) Dask dataframe query planning is disabled because
dask-expr is not installed.
(_ray_fit pid=4861)
(_ray_fit pid=4861) You can install it with `pip install
dask[dataframe]` or `conda install dask`.
(_ray_fit pid=4861) This will raise in a future version.
(_ray_fit pid=4861)
(_ray_fit pid=4861)   warnings.warn(msg, FutureWarning)
(_ray_fit pid=4861) [1000]     valid_set's rmse: 475.131
(_ray_fit pid=4861) [3000]     valid_set's rmse: 452.806 [repeated
4x across cluster] (Ray deduplicates logs by default. Set RAY_DEDUP_LOGS=0 to
disable log deduplication, or see https://docs.ray.io/en/master/ray-
observability/ray-logging.html#log-deduplication for more options.)
(_ray_fit pid=5050) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:  [repeated 2x across
cluster]
(_ray_fit pid=5050) Dask dataframe query planning is disabled because
dask-expr is not installed. [repeated 2x across cluster]
(_ray_fit pid=5050)  [repeated 4x across cluster]
(_ray_fit pid=5050) You can install it with `pip install
dask[dataframe]` or `conda install dask`. [repeated 2x across cluster]
(_ray_fit pid=5050) This will raise in a future version. [repeated
2x across cluster]
(_ray_fit pid=5050)   warnings.warn(msg, FutureWarning) [repeated
2x across cluster]
(_ray_fit pid=5050) [1000]     valid_set's rmse: 454.321 [repeated
2x across cluster]
(_ray_fit pid=5050) [3000]     valid_set's rmse: 451.699 [repeated
4x across cluster]
(_ray_fit pid=5216) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:  [repeated 2x across
cluster]
(_ray_fit pid=5216) Dask dataframe query planning is disabled because
dask-expr is not installed. [repeated 2x across cluster]
(_ray_fit pid=5216)  [repeated 4x across cluster]
(_ray_fit pid=5216) You can install it with `pip install
dask[dataframe]` or `conda install dask`. [repeated 2x across cluster]
```

```
(_ray_fit pid=5216) This will raise in a future version. [repeated
2x across cluster]
(_ray_fit pid=5216)   warnings.warn(msg, FutureWarning) [repeated
2x across cluster]
(_ray_fit pid=5216) [1000]     valid_set's rmse: 442.112 [repeated
2x across cluster]
(_ray_fit pid=5216) [2000]     valid_set's rmse: 412.533 [repeated
2x across cluster]
(_ray_fit pid=5216) [4000]     valid_set's rmse: 387.657 [repeated
4x across cluster]
(_ray_fit pid=5404) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:  [repeated 2x across
cluster]
(_ray_fit pid=5404) Dask dataframe query planning is disabled because
dask-expr is not installed. [repeated 2x across cluster]
(_ray_fit pid=5404)  [repeated 4x across cluster]
(_ray_fit pid=5404) You can install it with `pip install
dask[dataframe]` or `conda install dask`. [repeated 2x across cluster]
(_ray_fit pid=5404) This will raise in a future version. [repeated
2x across cluster]
(_ray_fit pid=5404)   warnings.warn(msg, FutureWarning) [repeated
2x across cluster]
(_ray_fit pid=5433) [1000]     valid_set's rmse: 551.309 [repeated
2x across cluster]
(_ray_fit pid=5433) [3000]     valid_set's rmse: 488.475 [repeated
2x across cluster]
(_ray_fit pid=5433) [6000]     valid_set's rmse: 459.421 [repeated
3x across cluster]
(_dystack pid=4697)    -484.9849        = Validation score
(-root_mean_squared_error)
(_dystack pid=4697)    117.28s  = Training   runtime
(_dystack pid=4697)    25.87s   = Validation runtime
(_dystack pid=4697) Fitting model: WeightedEnsemble_L2 … Training
model for up to 141.14s of the 12.97s of remaining time.
(_ray_fit pid=5433) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:
(_ray_fit pid=5433) Dask dataframe query planning is disabled because
dask-expr is not installed.
(_ray_fit pid=5433)  [repeated 2x across cluster]
```

```
(_ray_fit pid=5433) You can install it with `pip install
dask[dataframe]` or `conda install dask`.
(_ray_fit pid=5433) This will raise in a future version.
(_ray_fit pid=5433)   warnings.warn(msg, FutureWarning)
(_dystack pid=4697)    Ensemble Weights: {'LightGBMXT_BAG_L1': 0.769,
'KNeighborsDist_BAG_L1': 0.231}
(_dystack pid=4697)    -460.7118        = Validation score
(-root_mean_squared_error)
(_dystack pid=4697)    0.02s    = Training   runtime
(_dystack pid=4697)    0.0s     = Validation runtime
(_dystack pid=4697) Fitting 106 L2 models …
(_dystack pid=4697) Fitting model: LightGBMXT_BAG_L2 … Training model
for up to 12.93s of the 12.91s of remaining time.
(_dystack pid=4697)    Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
memory=0.08%)
(_ray_fit pid=5585) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:
(_ray_fit pid=5585) Dask dataframe query planning is disabled because
dask-expr is not installed.
(_ray_fit pid=5585) You can install it with `pip install
dask[dataframe]` or `conda install dask`.
(_ray_fit pid=5585) This will raise in a future version.
(_ray_fit pid=5585)   warnings.warn(msg, FutureWarning)
(_ray_fit pid=5585)  [repeated 2x across cluster]
(_ray_fit pid=5693) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:  [repeated 2x across
cluster]
(_ray_fit pid=5693) Dask dataframe query planning is disabled because
dask-expr is not installed. [repeated 2x across cluster]
(_ray_fit pid=5693) You can install it with `pip install
dask[dataframe]` or `conda install dask`. [repeated 2x across cluster]
(_ray_fit pid=5693) This will raise in a future version. [repeated
2x across cluster]
(_ray_fit pid=5693)   warnings.warn(msg, FutureWarning) [repeated
2x across cluster]
(_ray_fit pid=5693)  [repeated 4x across cluster]
(_ray_fit pid=5817) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:  [repeated 2x across
cluster]
(_ray_fit pid=5817) Dask dataframe query planning is disabled because
dask-expr is not installed. [repeated 2x across cluster]
(_ray_fit pid=5817) You can install it with `pip install
dask[dataframe]` or `conda install dask`. [repeated 2x across cluster]
```

```
(_ray_fit pid=5817) This will raise in a future version. [repeated
2x across cluster]
(_ray_fit pid=5817)   warnings.warn(msg, FutureWarning) [repeated
2x across cluster]
(_ray_fit pid=5817)  [repeated 4x across cluster]
(_ray_fit pid=5896) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:  [repeated 2x across
cluster]
(_ray_fit pid=5896) Dask dataframe query planning is disabled because
dask-expr is not installed. [repeated 2x across cluster]
(_ray_fit pid=5896) You can install it with `pip install
dask[dataframe]` or `conda install dask`. [repeated 2x across cluster]
(_ray_fit pid=5896) This will raise in a future version. [repeated
2x across cluster]
(_ray_fit pid=5896)   warnings.warn(msg, FutureWarning) [repeated
2x across cluster]
(_ray_fit pid=5896)  [repeated 4x across cluster]
(_dystack pid=4697)    -491.0254        = Validation score
(-root_mean_squared_error)
(_dystack pid=4697)    40.91s    = Training   runtime
(_dystack pid=4697)    1.16s     = Validation runtime
(_ray_fit pid=5929) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:
(_ray_fit pid=5929) Dask dataframe query planning is disabled because
dask-expr is not installed.
(_ray_fit pid=5929) You can install it with `pip install
dask[dataframe]` or `conda install dask`.
(_ray_fit pid=5929) This will raise in a future version.
(_ray_fit pid=5929)   warnings.warn(msg, FutureWarning)
(_ray_fit pid=5929)  [repeated 2x across cluster]
(_dystack pid=4697) Fitting model: WeightedEnsemble_L3 … Training
model for up to 141.14s of the -34.97s of remaining time.
(_dystack pid=4697)    Ensemble Weights: {'LightGBMXT_BAG_L1': 0.619,
'KNeighborsDist_BAG_L1': 0.19, 'LightGBMXT_BAG_L2': 0.19}
(_dystack pid=4697)    -458.735         = Validation score
(-root_mean_squared_error)
(_dystack pid=4697)    0.02s     = Training   runtime
(_dystack pid=4697)    0.0s      = Validation runtime
(_dystack pid=4697) AutoGluon training complete, total runtime =
176.43s … Best model: WeightedEnsemble_L3 | Estimated inference throughput:
74.1 rows/s (2005 batch size)
(_dystack pid=4697) TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("agModels-predictprice/ds_sub_fit/sub_fit_ho")
(_dystack pid=4697) /usr/local/lib/python3.10/dist-
packages/dask/dataframe/__init__.py:42: FutureWarning:
```

```
(_dystack pid=4697) Dask dataframe query planning is disabled because
dask-expr is not installed.
(_dystack pid=4697) You can install it with `pip install
dask[dataframe]` or `conda install dask`.
(_dystack pid=4697) This will raise in a future version.
(_dystack pid=4697)   warnings.warn(msg, FutureWarning)
(_dystack pid=4697) Deleting DyStack predictor artifacts
(clean_up_fits=True) …
(_dystack pid=4697)  [repeated 2x across cluster]
Leaderboard on holdout data (DyStack):
                   model  score_holdout   score_val              eval_metric
pred_time_test  pred_time_val    fit_time  pred_time_test_marginal
pred_time_val_marginal  fit_time_marginal  stack_level  can_infer  fit_order
0    WeightedEnsemble_L3    -327.077878 -458.734984  root_mean_squared_error
27.591639      27.136705  158.261909                0.003097
0.000915           0.022242           3       True          6
1    WeightedEnsemble_L2    -330.105189 -460.711848  root_mean_squared_error
26.656610      25.917874  117.333352                0.003945
0.000814           0.023995           2       True          4
2      LightGBMXT_BAG_L2    -345.434855 -491.025375  root_mean_squared_error
27.588542      27.135790  158.239667                0.920312
1.162218          40.909981           2       True          5
3      LightGBMXT_BAG_L1    -352.453263 -484.984850  root_mean_squared_error
26.635866      25.866909  117.284312               26.635866
25.866909         117.284312            1       True          3
4  KNeighborsDist_BAG_L1    -586.430480 -685.578353  root_mean_squared_error
0.016799       0.050151    0.025044                 0.016799
0.050151           0.025044           1       True          2
5  KNeighborsUnif_BAG_L1    -685.024589 -802.101549  root_mean_squared_error
0.015565       0.056512    0.020329                 0.015565
0.056512           0.020329           1       True          1
        1        = Optimal   num_stack_levels (Stacked Overfitting Occurred:
False)
        216s     = DyStack   runtime |  384s     = Remaining runtime
Starting main fit with num_stack_levels=1.
        For future fit calls on this dataset, you can skip DyStack to save time:
`predictor.fit(…, dynamic_stacking=False, num_stack_levels=1)`
Beginning AutoGluon training … Time limit = 384s
AutoGluon will save models to "agModels-predictprice"
Train Data Rows:    18039
Train Data Columns: 14
Label Column:       price
Problem Type:       regression
Preprocessing data …
Using Feature Generators to preprocess the data …
Fitting AutoMLPipelineFeatureGenerator…
        Available Memory:                    10178.14 MB
        Train Data (Original)  Memory Usage: 12.43 MB (0.1% of available memory)
```

```
        Inferring data type of each feature based on column values. Set
feature_metadata_in to manually specify special dtypes of the features.
        Stage 1 Generators:
                Fitting AsTypeFeatureGenerator…
        Stage 2 Generators:
                Fitting FillNaFeatureGenerator…
        Stage 3 Generators:
                Fitting IdentityFeatureGenerator…
                Fitting CategoryFeatureGenerator…
                        Fitting CategoryMemoryMinimizeFeatureGenerator…
        Stage 4 Generators:
                Fitting DropUniqueFeatureGenerator…
        Stage 5 Generators:
                Fitting DropDuplicatesFeatureGenerator…
        Types of features in original data (raw dtype, special dtypes):
                ('float', [])  :  2 | ['latitude', 'longitude']
                ('object', []) : 12 | ['city', 'province', 'lease_term', 'type',
'beds', …]
        Types of features in processed data (raw dtype, special dtypes):
                ('category', []) : 12 | ['city', 'province', 'lease_term',
'type', 'beds', …]
                ('float', [])    :  2 | ['latitude', 'longitude']
        0.5s = Fit runtime
        14 features in original data used to generate 14 features in processed
data.
        Train Data (Processed) Memory Usage: 0.52 MB (0.0% of available memory)
Data preprocessing and feature engineering runtime = 0.64s …
AutoGluon will gauge predictive performance using evaluation metric:
'root_mean_squared_error'
        This metric's sign has been flipped to adhere to being higher_is_better.
The metric score can be multiplied by -1 to get the metric value.
        To change this, specify the eval_metric parameter of Predictor()
Large model count detected (112 configs) … Only displaying the first 3 models
of each family. To see all, set `verbosity=3`.
User-specified model hyperparameters to be fit:
{
        'NN_TORCH': [{}, {'activation': 'elu', 'dropout_prob':
0.10077639529843717, 'hidden_size': 108, 'learning_rate': 0.002735937344002146,
'num_layers': 4, 'use_batchnorm': True, 'weight_decay': 1.356433327634438e-12,
'ag_args': {'name_suffix': '_r79', 'priority': -2}}, {'activation': 'elu',
'dropout_prob': 0.11897478034205347, 'hidden_size': 213, 'learning_rate':
0.0010474382260641949, 'num_layers': 4, 'use_batchnorm': False, 'weight_decay':
5.594471067786272e-10, 'ag_args': {'name_suffix': '_r22', 'priority': -7}}],
        'GBM': [{'extra_trees': True, 'ag_args': {'name_suffix': 'XT'}}, {},
'GBMLarge'],
        'CAT': [{}, {'depth': 6, 'grow_policy': 'SymmetricTree', 'l2_leaf_reg':
2.1542798306067823, 'learning_rate': 0.06864209415792857, 'max_ctr_complexity':
4, 'one_hot_max_size': 10, 'ag_args': {'name_suffix': '_r177', 'priority': -1}},
```

```
{'depth': 8, 'grow_policy': 'Depthwise', 'l2_leaf_reg': 2.7997999596449104,
'learning_rate': 0.031375015734637225, 'max_ctr_complexity': 2,
'one_hot_max_size': 3, 'ag_args': {'name_suffix': '_r9', 'priority': -5}}],
        'XGB': [{}, {'colsample_bytree': 0.6917311125174739,
'enable_categorical': False, 'learning_rate': 0.018063876087523967, 'max_depth':
10, 'min_child_weight': 0.6028633586934382, 'ag_args': {'name_suffix': '_r33',
'priority': -8}}, {'colsample_bytree': 0.6628423832084077, 'enable_categorical':
False, 'learning_rate': 0.08775715546881824, 'max_depth': 5, 'min_child_weight':
0.6294123374222513, 'ag_args': {'name_suffix': '_r89', 'priority': -16}}],
        'FASTAI': [{}, {'bs': 256, 'emb_drop': 0.5411770367537934, 'epochs': 43,
'layers': [800, 400], 'lr': 0.01519848858318159, 'ps': 0.23782946566604385,
'ag_args': {'name_suffix': '_r191', 'priority': -4}}, {'bs': 2048, 'emb_drop':
0.05070411322605811, 'epochs': 29, 'layers': [200, 100], 'lr':
0.08974235041576624, 'ps': 0.10393466140748028, 'ag_args': {'name_suffix':
'_r102', 'priority': -11}}],
        'RF': [{'criterion': 'gini', 'ag_args': {'name_suffix': 'Gini',
'problem_types': ['binary', 'multiclass']}}, {'criterion': 'entropy', 'ag_args':
{'name_suffix': 'Entr', 'problem_types': ['binary', 'multiclass']}},
{'criterion': 'squared_error', 'ag_args': {'name_suffix': 'MSE',
'problem_types': ['regression', 'quantile']}}],
        'XT': [{'criterion': 'gini', 'ag_args': {'name_suffix': 'Gini',
'problem_types': ['binary', 'multiclass']}}, {'criterion': 'entropy', 'ag_args':
{'name_suffix': 'Entr', 'problem_types': ['binary', 'multiclass']}},
{'criterion': 'squared_error', 'ag_args': {'name_suffix': 'MSE',
'problem_types': ['regression', 'quantile']}}],
        'KNN': [{'weights': 'uniform', 'ag_args': {'name_suffix': 'Unif'}},
{'weights': 'distance', 'ag_args': {'name_suffix': 'Dist'}}],
}
AutoGluon will fit 2 stack levels (L1 to L2) …
Fitting 108 L1 models …
Fitting model: KNeighborsUnif_BAG_L1 … Training model for up to 255.64s of the
383.53s of remaining time.
        -774.1865       = Validation score   (-root_mean_squared_error)
        0.03s    = Training   runtime
        0.09s    = Validation runtime
Fitting model: KNeighborsDist_BAG_L1 … Training model for up to 255.45s of the
383.34s of remaining time.
        -652.2715       = Validation score   (-root_mean_squared_error)
        0.05s    = Training   runtime
        0.09s    = Validation runtime
Fitting model: LightGBMXT_BAG_L1 … Training model for up to 255.27s of the
383.15s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.07%)
        -442.0899       = Validation score   (-root_mean_squared_error)
        282.43s  = Training   runtime
        118.27s  = Validation runtime
Fitting model: WeightedEnsemble_L2 … Training model for up to 360.0s of the
```

```
84.94s of remaining time.
        Ensemble Weights: {'LightGBMXT_BAG_L1': 0.792, 'KNeighborsDist_BAG_L1':
0.208}
        -423.1634       = Validation score   (-root_mean_squared_error)
        0.02s    = Training   runtime
        0.0s     = Validation runtime
Fitting 106 L2 models …
Fitting model: LightGBMXT_BAG_L2 … Training model for up to 84.89s of the
84.86s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.09%)
        -432.8841       = Validation score   (-root_mean_squared_error)
        63.47s   = Training   runtime
        12.98s   = Validation runtime
Fitting model: LightGBM_BAG_L2 … Training model for up to 16.2s of the 16.16s
of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.08%)
        -438.0843       = Validation score   (-root_mean_squared_error)
        29.97s   = Training   runtime
        1.08s    = Validation runtime
Fitting model: WeightedEnsemble_L3 … Training model for up to 360.0s of the
-18.07s of remaining time.
        Ensemble Weights: {'LightGBMXT_BAG_L1': 0.4, 'LightGBMXT_BAG_L2': 0.3,
'LightGBM_BAG_L2': 0.2, 'KNeighborsDist_BAG_L1': 0.1}
        -414.8761       = Validation score   (-root_mean_squared_error)
        0.05s    = Training   runtime
        0.0s     = Validation runtime
AutoGluon training complete, total runtime = 402.39s … Best model:
WeightedEnsemble_L3 | Estimated inference throughput: 17.0 rows/s (2255 batch
size)
TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("agModels-predictprice")
```

[19]:
```
test_data = TabularDataset(df.drop(train_data.index))
y_test = test_data[target]
# predictor = TabularPredictor.load("agModels-predictprice")

y_pred = predictor_price.predict(test_data)
print("Predictions:  \n", y_pred)
perf = predictor_price.evaluate_predictions(y_true=y_test, y_pred=y_pred,␣
 ↪auxiliary_metrics=True)
```

```
Predictions:
 10       2818.560791
11        2026.548340
13        2583.564453
19        2126.927246
```

```
24       1997.721191
            …
25748    1244.216064
25755    1066.665649
25762     896.792847
25763    1111.311279
25764     896.792847
Name: price, Length: 7732, dtype: float32
```

[20]:
```
# Take a look at the performance of the model
perf
```

[20]:
```
{'root_mean_squared_error': -431.76314186113444,
 'mean_squared_error': -186419.41066979812,
 'mean_absolute_error': -196.76120907033103,
 'r2': 0.8036366987519679,
 'pearsonr': 0.8978123558305657,
 'median_absolute_error': -116.48834228515625}
```

[21]:
```
pd.DataFrame({'y_test': y_test, 'y_pred': y_pred})
```

[21]:

| | y_test | y_pred |
|---|---|---|
| 10 | 1930.0 | 2818.560791 |
| 11 | 1700.0 | 2026.548340 |
| 13 | 3150.0 | 2583.564453 |
| 19 | 2300.0 | 2126.927246 |
| 24 | 1910.0 | 1997.721191 |
| … | … | … |
| 25748 | 1305.0 | 1244.216064 |
| 25755 | 1085.0 | 1066.665649 |
| 25762 | 945.0 | 896.792847 |
| 25763 | 1025.0 | 1111.311279 |
| 25764 | 995.0 | 896.792847 |

[7732 rows x 2 columns]

[22]:
```
# from google.colab import files
# files.download('/content/agModels-predictprice/models/WeightedEnsemble_L3')
```

[ ]:
```
# # Reduce model size
# predictor_price.reduce_memory_size(remove_data=True, save_space=True)

# predictor_price.leaderboard(extra_info=True)
# predictor_price.delete_models(models_to_keep=predictor_price.
 ↪leaderboard()['model'][:3])
# predictor_price.save("saved_model")
```

## 2 Explainable AI

[15]:
```
predictor_price = TabularPredictor.load("agModels-predictprice")
```

[16]:
```
predictor_price.info()
```

```
/usr/local/lib/python3.10/dist-packages/dask/dataframe/__init__.py:42:
FutureWarning:
Dask dataframe query planning is disabled because dask-expr is not installed.

You can install it with `pip install dask[dataframe]` or `conda install dask`.
This will raise in a future version.

  warnings.warn(msg, FutureWarning)
```

[16]:
```
{'path': 'agModels-predictprice',
 'label': 'price',
 'random_state': 0,
 'version': '1.1.1',
 'features': ['city',
  'province',
  'latitude',
  'longitude',
  'lease_term',
  'type',
  'beds',
  'baths',
  'sq_feet',
  'furnishing',
  'availability_date',
  'smoking',
  'cats',
  'dogs'],
 'feature_metadata_in':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fa8f0>,
 'time_fit_preprocessing': 0.6379239559173584,
 'time_fit_training': 401.7536509037018,
 'time_fit_total': 402.39157485961914,
 'time_limit': 384.20341539382935,
 'time_train_start': 1731860728.1672926,
 'num_rows_train': 18039,
 'num_cols_train': 14,
 'num_rows_val': None,
 'num_classes': None,
 'problem_type': 'regression',
 'eval_metric': 'root_mean_squared_error',
 'best_model': 'WeightedEnsemble_L3',
```

```
 'best_model_score_val': -414.8760953952414,
 'best_model_stack_level': 3,
 'num_models_trained': 7,
 'num_bag_folds': 8,
 'max_stack_level': 3,
 'max_core_stack_level': 2,
 'model_info': {'KNeighborsUnif_BAG_L1': {'name': 'KNeighborsUnif_BAG_L1',
   'model_type': 'StackerEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 0.029515743255615234,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 0.09212350845336914,
   'val_score': -774.186517755969,
   'hyperparameters': {'use_orig_features': True,
    'max_base_models': 25,
    'max_base_models_per_type': 5,
    'save_bag_folds': True,
    'use_child_oof': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': ['use_child_oof'],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
   'num_features': 2,
   'features': ['longitude', 'latitude'],
   'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd12e290>,
   'memory_size': 2638,
   'compile_time': None,
   'is_initialized': True,
   'is_fit': True,
   'is_valid': True,
   'can_infer': True,
   'bagged_info': {'child_model_type': 'KNNModel',
```

```
 'num_child_models': 1,
 'child_model_names': ['S1F1'],
 '_n_repeats': 1,
 '_k_per_n_repeat': [1],
 '_random_state': 1,
 'low_memory': True,
 'bagged_mode': False,
 'max_memory_size': 792488,
 'min_memory_size': 792488,
 'child_hyperparameters': {'weights': 'uniform'},
 'child_hyperparameters_fit': {},
 'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
  'max_time_limit_ratio': 1.0,
  'max_time_limit': None,
  'min_time_limit': 0,
  'valid_raw_types': ['int', 'float'],
  'valid_special_types': None,
  'ignored_type_group_special': ['bool'],
  'ignored_type_group_raw': None,
  'get_features_kwargs': None,
  'get_features_kwargs_extra': None,
  'predict_1_batch_size': None,
  'temperature_scalar': None}},
'stacker_info': {'num_base_models': 0, 'base_model_names': []},
'children_info': {'S1F1': {'name': 'S1F1',
  'model_type': 'KNNModel',
  'problem_type': 'regression',
  'eval_metric': 'root_mean_squared_error',
  'stopping_metric': 'root_mean_squared_error',
  'fit_time': 0.029515743255615234,
  'num_classes': None,
  'quantile_levels': None,
  'predict_time': 0.09212350845336914,
  'val_score': -774.186517755969,
  'hyperparameters': {'weights': 'uniform'},
  'hyperparameters_fit': {},
  'hyperparameters_nondefault': ['weights'],
  'ag_args_fit': {'max_memory_usage_ratio': 1.0,
   'max_time_limit_ratio': 1.0,
   'max_time_limit': None,
   'min_time_limit': 0,
   'valid_raw_types': ['int', 'float'],
   'valid_special_types': None,
   'ignored_type_group_special': ['bool'],
   'ignored_type_group_raw': None,
   'get_features_kwargs': None,
   'get_features_kwargs_extra': None,
```

```
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 2,
     'features': ['latitude', 'longitude'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aede741abc0>,
     'memory_size': 789850,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True}},
  'KNeighborsDist_BAG_L1': {'name': 'KNeighborsDist_BAG_L1',
   'model_type': 'StackerEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 0.04554128646850586,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 0.0915365219116211,
   'val_score': -652.2714801027624,
   'hyperparameters': {'use_orig_features': True,
    'max_base_models': 25,
    'max_base_models_per_type': 5,
    'save_bag_folds': True,
    'use_child_oof': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': ['use_child_oof'],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
   'num_features': 2,
   'features': ['longitude', 'latitude'],
   'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fba90>,
   'memory_size': 2638,
```

```
'compile_time': None,
'is_initialized': True,
'is_fit': True,
'is_valid': True,
'can_infer': True,
'bagged_info': {'child_model_type': 'KNNModel',
 'num_child_models': 1,
 'child_model_names': ['S1F1'],
 '_n_repeats': 1,
 '_k_per_n_repeat': [1],
 '_random_state': 1,
 'low_memory': True,
 'bagged_mode': False,
 'max_memory_size': 792489,
 'min_memory_size': 792489,
 'child_hyperparameters': {'weights': 'distance'},
 'child_hyperparameters_fit': {},
 'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
  'max_time_limit_ratio': 1.0,
  'max_time_limit': None,
  'min_time_limit': 0,
  'valid_raw_types': ['int', 'float'],
  'valid_special_types': None,
  'ignored_type_group_special': ['bool'],
  'ignored_type_group_raw': None,
  'get_features_kwargs': None,
  'get_features_kwargs_extra': None,
  'predict_1_batch_size': None,
  'temperature_scalar': None}},
'stacker_info': {'num_base_models': 0, 'base_model_names': []},
'children_info': {'S1F1': {'name': 'S1F1',
  'model_type': 'KNNModel',
  'problem_type': 'regression',
  'eval_metric': 'root_mean_squared_error',
  'stopping_metric': 'root_mean_squared_error',
  'fit_time': 0.04554128646850586,
  'num_classes': None,
  'quantile_levels': None,
  'predict_time': 0.0915365219116211,
  'val_score': -652.2714801027624,
  'hyperparameters': {'weights': 'distance'},
  'hyperparameters_fit': {},
  'hyperparameters_nondefault': ['weights'],
  'ag_args_fit': {'max_memory_usage_ratio': 1.0,
   'max_time_limit_ratio': 1.0,
   'max_time_limit': None,
   'min_time_limit': 0,
```

```
      'valid_raw_types': ['int', 'float'],
      'valid_special_types': None,
      'ignored_type_group_special': ['bool'],
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 2,
     'features': ['latitude', 'longitude'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee2435ac50>,
     'memory_size': 789851,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True}}},
  'LightGBMXT_BAG_L1': {'name': 'LightGBMXT_BAG_L1',
   'model_type': 'StackerEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 282.42990374565125,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 118.26582980155945,
   'val_score': -442.0899321234507,
   'hyperparameters': {'use_orig_features': True,
    'max_base_models': 25,
    'max_base_models_per_type': 5,
    'save_bag_folds': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': [],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
```

```
   'num_features': 14,
   'features': ['province',
    'type',
    'city',
    'longitude',
    'cats',
    'beds',
    'smoking',
    'furnishing',
    'baths',
    'sq_feet',
    'dogs',
    'lease_term',
    'latitude',
    'availability_date'],
   'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fb1c0>,
   'memory_size': 2870,
   'compile_time': None,
   'is_initialized': True,
   'is_fit': True,
   'is_valid': True,
   'can_infer': True,
   'bagged_info': {'child_model_type': 'LGBModel',
    'num_child_models': 8,
    'child_model_names': ['S1F2',
     'S1F1',
     'S1F4',
     'S1F3',
     'S1F5',
     'S1F6',
     'S1F7',
     'S1F8'],
    '_n_repeats': 1,
    '_k_per_n_repeat': [8],
    '_random_state': 1,
    'low_memory': True,
    'bagged_mode': True,
    'max_memory_size': 181928391,
    'min_memory_size': 26095574,
    'child_hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
    'child_hyperparameters_fit': {'num_boost_round': 8712},
    'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
     'max_time_limit_ratio': 1.0,
     'max_time_limit': None,
     'min_time_limit': 0,
     'valid_raw_types': ['bool', 'int', 'float', 'category'],
```

```
  'valid_special_types': None,
  'ignored_type_group_special': None,
  'ignored_type_group_raw': None,
  'get_features_kwargs': None,
  'get_features_kwargs_extra': None,
  'predict_1_batch_size': None,
  'temperature_scalar': None}},
'stacker_info': {'num_base_models': 0, 'base_model_names': []},
'children_info': {'S1F2': {'name': 'S1F2',
  'model_type': 'LGBModel',
  'problem_type': 'regression',
  'eval_metric': 'root_mean_squared_error',
  'stopping_metric': 'root_mean_squared_error',
  'fit_time': 58.19691824913025,
  'num_classes': None,
  'quantile_levels': None,
  'predict_time': 20.48674750328064,
  'val_score': -424.3561316043692,
  'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
  'hyperparameters_fit': {'num_boost_round': 9645},
  'hyperparameters_nondefault': ['extra_trees'],
  'ag_args_fit': {'max_memory_usage_ratio': 1.0,
   'max_time_limit_ratio': 1.0,
   'max_time_limit': None,
   'min_time_limit': 0,
   'valid_raw_types': ['bool', 'int', 'float', 'category'],
   'valid_special_types': None,
   'ignored_type_group_special': None,
   'ignored_type_group_raw': None,
   'get_features_kwargs': None,
   'get_features_kwargs_extra': None,
   'predict_1_batch_size': None,
   'temperature_scalar': None},
  'num_features': 14,
  'features': ['latitude',
   'longitude',
   'city',
   'province',
   'lease_term',
   'type',
   'beds',
   'baths',
   'sq_feet',
   'furnishing',
   'availability_date',
   'smoking',
   'cats',
```

```
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aede741a560>,
     'memory_size': 25143815,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F1': {'name': 'S1F1',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 59.269981145858765,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 21.030065536499023,
     'val_score': -382.94466320599656,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 9587},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
```

```
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee16875270>,
     'memory_size': 25018284,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F4': {'name': 'S1F4',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 55.36878061294556,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 16.661664962768555,
     'val_score': -368.63815774600977,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 9935},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
```

```
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcbc4100>,
     'memory_size': 25916132,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F3': {'name': 'S1F3',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 55.31507325172424,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 16.58866047859192,
     'val_score': -649.5601056189936,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 9887},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
```

```
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fb070>,
     'memory_size': 25869183,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F5': {'name': 'S1F5',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 52.14553451538086,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 16.65358066558838,
     'val_score': -417.697761743281,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 9992},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
```

```
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcd09540>,
     'memory_size': 26064503,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F6': {'name': 'S1F6',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 52.45994424819946,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 16.317374229431152,
     'val_score': -424.5945086276661,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 9942},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
```

```
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fad40>,
     'memory_size': 25938441,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F7': {'name': 'S1F7',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 5.27716851234436,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.4103834629058838,
     'val_score': -411.86323027391967,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 707},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
```

```
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fa590>,
     'memory_size': 1882459,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F8': {'name': 'S1F8',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 31.96611976623535,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 10.117352962493896,
     'val_score': -393.1914940836134,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 10000},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 14,
     'features': ['latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
```

```
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcd096f0>,
     'memory_size': 26092704,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True}},
  'WeightedEnsemble_L2': {'name': 'WeightedEnsemble_L2',
   'model_type': 'WeightedEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 0.023722410202026367,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 0.0008921623229980469,
   'val_score': -423.163408198703,
   'hyperparameters': {'use_orig_features': False,
    'max_base_models': 25,
    'max_base_models_per_type': 5,
    'save_bag_folds': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': ['save_bag_folds'],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
   'num_features': 2,
   'features': ['LightGBMXT_BAG_L1', 'KNeighborsDist_BAG_L1'],
```

```
   'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddc751e10>,
   'memory_size': 2963,
   'compile_time': None,
   'is_initialized': True,
   'is_fit': True,
   'is_valid': True,
   'can_infer': True,
   'bagged_info': {'child_model_type': 'GreedyWeightedEnsembleModel',
    'num_child_models': 1,
    'child_model_names': ['S1F1'],
    '_n_repeats': 1,
    '_k_per_n_repeat': [1],
    '_random_state': 2,
    'low_memory': False,
    'bagged_mode': False,
    'max_memory_size': 2963,
    'min_memory_size': 2963,
    'child_hyperparameters': {'ensemble_size': 25, 'subsample_size': 1000000},
    'child_hyperparameters_fit': {'ensemble_size': 24},
    'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
     'max_time_limit_ratio': 1.0,
     'max_time_limit': None,
     'min_time_limit': 0,
     'valid_raw_types': None,
     'valid_special_types': None,
     'ignored_type_group_special': None,
     'ignored_type_group_raw': None,
     'get_features_kwargs': None,
     'get_features_kwargs_extra': None,
     'predict_1_batch_size': None,
     'temperature_scalar': None,
     'drop_unique': False}},
   'stacker_info': {'num_base_models': 2,
    'base_model_names': ['KNeighborsDist_BAG_L1', 'LightGBMXT_BAG_L1']},
   'children_info': {'S1F1': {'name': 'S1F1',
     'model_type': 'GreedyWeightedEnsembleModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 0.023722410202026367,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': None,
     'val_score': None,
     'hyperparameters': {'ensemble_size': 25, 'subsample_size': 1000000},
     'hyperparameters_fit': {'ensemble_size': 24},
```

```
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': None,
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None,
      'drop_unique': False},
     'num_features': 2,
     'features': ['KNeighborsDist_BAG_L1', 'LightGBMXT_BAG_L1'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddc751cf0>,
     'memory_size': 5956,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True,
     'model_weights': {'KNeighborsDist_BAG_L1': 0.20833333333333334,
      'LightGBMXT_BAG_L1': 0.7916666666666666}}},
  'LightGBMXT_BAG_L2': {'name': 'LightGBMXT_BAG_L2',
   'model_type': 'StackerEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 63.47079634666443,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 12.97685432434082,
   'val_score': -432.88414922328326,
   'hyperparameters': {'use_orig_features': True,
    'max_base_models': 25,
    'max_base_models_per_type': 5,
    'save_bag_folds': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': [],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
```

```
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
   'num_features': 17,
   'features': ['province',
    'LightGBMXT_BAG_L1',
    'type',
    'city',
    'KNeighborsUnif_BAG_L1',
    'longitude',
    'cats',
    'beds',
    'smoking',
    'KNeighborsDist_BAG_L1',
    'furnishing',
    'baths',
    'sq_feet',
    'dogs',
    'lease_term',
    'latitude',
    'availability_date'],
   'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcd1c220>,
   'memory_size': 3208,
   'compile_time': None,
   'is_initialized': True,
   'is_fit': True,
   'is_valid': True,
   'can_infer': True,
   'bagged_info': {'child_model_type': 'LGBModel',
    'num_child_models': 8,
    'child_model_names': ['S1F1',
     'S1F3',
     'S1F2',
     'S1F4',
     'S1F5',
     'S1F6',
     'S1F7',
     'S1F8'],
    '_n_repeats': 1,
    '_k_per_n_repeat': [8],
    '_random_state': 2,
```

```
 'low_memory': True,
 'bagged_mode': True,
 'max_memory_size': 31486982,
 'min_memory_size': 9479445,
 'child_hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
 'child_hyperparameters_fit': {'num_boost_round': 1479},
 'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
  'max_time_limit_ratio': 1.0,
  'max_time_limit': None,
  'min_time_limit': 0,
  'valid_raw_types': ['bool', 'int', 'float', 'category'],
  'valid_special_types': None,
  'ignored_type_group_special': None,
  'ignored_type_group_raw': None,
  'get_features_kwargs': None,
  'get_features_kwargs_extra': None,
  'predict_1_batch_size': None,
  'temperature_scalar': None}},
'stacker_info': {'num_base_models': 3,
 'base_model_names': ['KNeighborsUnif_BAG_L1',
  'KNeighborsDist_BAG_L1',
  'LightGBMXT_BAG_L1']},
'children_info': {'S1F1': {'name': 'S1F1',
  'model_type': 'LGBModel',
  'problem_type': 'regression',
  'eval_metric': 'root_mean_squared_error',
  'stopping_metric': 'root_mean_squared_error',
  'fit_time': 12.375351667404175,
  'num_classes': None,
  'quantile_levels': None,
  'predict_time': 0.6218945980072021,
  'val_score': -396.1039605761208,
  'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
  'hyperparameters_fit': {'num_boost_round': 1143},
  'hyperparameters_nondefault': ['extra_trees'],
  'ag_args_fit': {'max_memory_usage_ratio': 1.0,
   'max_time_limit_ratio': 1.0,
   'max_time_limit': None,
   'min_time_limit': 0,
   'valid_raw_types': ['bool', 'int', 'float', 'category'],
   'valid_special_types': None,
   'ignored_type_group_special': None,
   'ignored_type_group_raw': None,
   'get_features_kwargs': None,
   'get_features_kwargs_extra': None,
   'predict_1_batch_size': None,
   'temperature_scalar': None},
```

```
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee24385240>,
     'memory_size': 3039448,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F3': {'name': 'S1F3',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 2.643284797668457,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.03850364685058594,
     'val_score': -324.3544768088787,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 93},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
```

```
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee24387b20>,
     'memory_size': 272574,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F2': {'name': 'S1F2',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 22.0169997215271,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 3.1859657764434814,
     'val_score': -659.9561650108415,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 3099},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
```

```
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fb5e0>,
     'memory_size': 8211234,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F4': {'name': 'S1F4',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 3.354447364807129,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.059639930725097656,
     'val_score': -380.04525872457435,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 137},
     'hyperparameters_nondefault': ['extra_trees'],
```

```
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee243876d0>,
     'memory_size': 388929,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F5': {'name': 'S1F5',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 19.34431219100952,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 3.263979434967041,
```

```
     'val_score': -417.26337347013185,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 3577},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee24387220>,
     'memory_size': 9476237,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F6': {'name': 'S1F6',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
```

```
     'fit_time': 19.204747676849365,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 5.70510458946228,
     'val_score': -426.7062673536995,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 3514},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee24384490>,
     'memory_size': 9328076,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F7': {'name': 'S1F7',
```

```
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 3.099177122116089,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.07348895072937012,
     'val_score': -413.83553159023853,
     'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
     'hyperparameters_fit': {'num_boost_round': 182},
     'hyperparameters_nondefault': ['extra_trees'],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee24359390>,
     'memory_size': 508720,
     'compile_time': None,
     'is_initialized': True,
```

```
 'is_fit': True,
 'is_valid': True,
 'can_infer': True},
'S1F8': {'name': 'S1F8',
 'model_type': 'LGBModel',
 'problem_type': 'regression',
 'eval_metric': 'root_mean_squared_error',
 'stopping_metric': 'root_mean_squared_error',
 'fit_time': 2.3083550930023193,
 'num_classes': None,
 'quantile_levels': None,
 'predict_time': 0.02827739715576172,
 'val_score': -359.93933605571164,
 'hyperparameters': {'learning_rate': 0.05, 'extra_trees': True},
 'hyperparameters_fit': {'num_boost_round': 88},
 'hyperparameters_nondefault': ['extra_trees'],
 'ag_args_fit': {'max_memory_usage_ratio': 1.0,
  'max_time_limit_ratio': 1.0,
  'max_time_limit': None,
  'min_time_limit': 0,
  'valid_raw_types': ['bool', 'int', 'float', 'category'],
  'valid_special_types': None,
  'ignored_type_group_special': None,
  'ignored_type_group_raw': None,
  'get_features_kwargs': None,
  'get_features_kwargs_extra': None,
  'predict_1_batch_size': None,
  'temperature_scalar': None},
 'num_features': 17,
 'features': ['KNeighborsUnif_BAG_L1',
  'KNeighborsDist_BAG_L1',
  'LightGBMXT_BAG_L1',
  'latitude',
  'longitude',
  'city',
  'province',
  'lease_term',
  'type',
  'beds',
  'baths',
  'sq_feet',
  'furnishing',
  'availability_date',
  'smoking',
  'cats',
  'dogs'],
 'feature_metadata':
```

```
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcd09630>,
     'memory_size': 258556,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True}}},
  'LightGBM_BAG_L2': {'name': 'LightGBM_BAG_L2',
   'model_type': 'StackerEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 29.96975064277649,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 1.0781099796295166,
   'val_score': -438.08432384123813,
   'hyperparameters': {'use_orig_features': True,
    'max_base_models': 25,
    'max_base_models_per_type': 5,
    'save_bag_folds': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': [],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
   'num_features': 17,
   'features': ['province',
    'LightGBMXT_BAG_L1',
    'type',
    'city',
    'KNeighborsUnif_BAG_L1',
    'longitude',
    'cats',
    'beds',
    'smoking',
    'KNeighborsDist_BAG_L1',
```

```
     'furnishing',
     'baths',
     'sq_feet',
     'dogs',
     'lease_term',
     'latitude',
     'availability_date'],
    'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddd0fb160>,
    'memory_size': 3204,
    'compile_time': None,
    'is_initialized': True,
    'is_fit': True,
    'is_valid': True,
    'can_infer': True,
    'bagged_info': {'child_model_type': 'LGBModel',
     'num_child_models': 8,
     'child_model_names': ['S1F1',
      'S1F2',
      'S1F3',
      'S1F4',
      'S1F5',
      'S1F6',
      'S1F7',
      'S1F8'],
     '_n_repeats': 1,
     '_k_per_n_repeat': [8],
     '_random_state': 2,
     'low_memory': True,
     'bagged_mode': True,
     'max_memory_size': 5045688,
     'min_memory_size': 1356557,
     'child_hyperparameters': {'learning_rate': 0.05},
     'child_hyperparameters_fit': {'num_boost_round': 202},
     'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None}},
    'stacker_info': {'num_base_models': 3,
```

```
 'base_model_names': ['KNeighborsUnif_BAG_L1',
  'KNeighborsDist_BAG_L1',
  'LightGBMXT_BAG_L1']},
'children_info': {'S1F1': {'name': 'S1F1',
  'model_type': 'LGBModel',
  'problem_type': 'regression',
  'eval_metric': 'root_mean_squared_error',
  'stopping_metric': 'root_mean_squared_error',
  'fit_time': 5.02400279045105,
  'num_classes': None,
  'quantile_levels': None,
  'predict_time': 0.05747342109680176,
  'val_score': -364.9733263439527,
  'hyperparameters': {'learning_rate': 0.05},
  'hyperparameters_fit': {'num_boost_round': 128},
  'hyperparameters_nondefault': [],
  'ag_args_fit': {'max_memory_usage_ratio': 1.0,
   'max_time_limit_ratio': 1.0,
   'max_time_limit': None,
   'min_time_limit': 0,
   'valid_raw_types': ['bool', 'int', 'float', 'category'],
   'valid_special_types': None,
   'ignored_type_group_special': None,
   'ignored_type_group_raw': None,
   'get_features_kwargs': None,
   'get_features_kwargs_extra': None,
   'predict_1_batch_size': None,
   'temperature_scalar': None},
  'num_features': 17,
  'features': ['KNeighborsUnif_BAG_L1',
   'KNeighborsDist_BAG_L1',
   'LightGBMXT_BAG_L1',
   'latitude',
   'longitude',
   'city',
   'province',
   'lease_term',
   'type',
   'beds',
   'baths',
   'sq_feet',
   'furnishing',
   'availability_date',
   'smoking',
   'cats',
   'dogs'],
  'feature_metadata':
```

```
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee2435a290>,
     'memory_size': 414331,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F2': {'name': 'S1F2',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 5.455013751983643,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.2831716537475586,
     'val_score': -703.3882333580328,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 355},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
```

```
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcd096c0>,
     'memory_size': 1103362,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F3': {'name': 'S1F3',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 2.7538435459136963,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.022531986236572266,
     'val_score': -347.70574380996885,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 49},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
```

```
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcf55900>,
     'memory_size': 161804,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F4': {'name': 'S1F4',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 3.5004281997680664,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.09077143669128418,
     'val_score': -365.3430320179888,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 115},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
```

```
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcf556c0>,
     'memory_size': 374502,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F5': {'name': 'S1F5',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 5.55760645866394,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.21150636672973633,
     'val_score': -433.28387639263207,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 446},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
```

```
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcf55780>,
     'memory_size': 1353353,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F6': {'name': 'S1F6',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 5.495210886001587,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.3667716979980469,
     'val_score': -430.3510153400956,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 399},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
```

```
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aee2435a7a0>,
     'memory_size': 1230921,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F7': {'name': 'S1F7',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 2.807051181793213,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.02846217155456543,
     'val_score': -400.9667860832463,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 67},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
```

```
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcf55960>,
     'memory_size': 218244,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True},
    'S1F8': {'name': 'S1F8',
     'model_type': 'LGBModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 2.716935157775879,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': 0.017421245574951172,
     'val_score': -345.3464214922553,
     'hyperparameters': {'learning_rate': 0.05},
     'hyperparameters_fit': {'num_boost_round': 57},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
```

```
      'min_time_limit': 0,
      'valid_raw_types': ['bool', 'int', 'float', 'category'],
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None},
     'num_features': 17,
     'features': ['KNeighborsUnif_BAG_L1',
      'KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'latitude',
      'longitude',
      'city',
      'province',
      'lease_term',
      'type',
      'beds',
      'baths',
      'sq_feet',
      'furnishing',
      'availability_date',
      'smoking',
      'cats',
      'dogs'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddcf557b0>,
     'memory_size': 185967,
     'compile_time': None,
     'is_initialized': True,
     'is_fit': True,
     'is_valid': True,
     'can_infer': True}}},
  'WeightedEnsemble_L3': {'name': 'WeightedEnsemble_L3',
   'model_type': 'WeightedEnsembleModel',
   'problem_type': 'regression',
   'eval_metric': 'root_mean_squared_error',
   'stopping_metric': 'root_mean_squared_error',
   'fit_time': 0.048243045806884766,
   'num_classes': None,
   'quantile_levels': None,
   'predict_time': 0.0012357234954833984,
   'val_score': -414.8760953952414,
   'hyperparameters': {'use_orig_features': False,
    'max_base_models': 25,
```

```
    'max_base_models_per_type': 5,
    'save_bag_folds': True},
   'hyperparameters_fit': {},
   'hyperparameters_nondefault': ['save_bag_folds'],
   'ag_args_fit': {'max_memory_usage_ratio': 1.0,
    'max_time_limit_ratio': 1.0,
    'max_time_limit': None,
    'min_time_limit': 0,
    'valid_raw_types': None,
    'valid_special_types': None,
    'ignored_type_group_special': None,
    'ignored_type_group_raw': None,
    'get_features_kwargs': None,
    'get_features_kwargs_extra': None,
    'predict_1_batch_size': None,
    'temperature_scalar': None,
    'drop_unique': False},
   'num_features': 4,
   'features': ['LightGBMXT_BAG_L1',
    'LightGBM_BAG_L2',
    'KNeighborsDist_BAG_L1',
    'LightGBMXT_BAG_L2'],
   'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddc7526e0>,
   'memory_size': 3143,
   'compile_time': None,
   'is_initialized': True,
   'is_fit': True,
   'is_valid': True,
   'can_infer': True,
   'bagged_info': {'child_model_type': 'GreedyWeightedEnsembleModel',
    'num_child_models': 1,
    'child_model_names': ['S1F1'],
    '_n_repeats': 1,
    '_k_per_n_repeat': [1],
    '_random_state': 3,
    'low_memory': False,
    'bagged_mode': False,
    'max_memory_size': 3143,
    'min_memory_size': 3143,
    'child_hyperparameters': {'ensemble_size': 25, 'subsample_size': 1000000},
    'child_hyperparameters_fit': {'ensemble_size': 10},
    'child_ag_args_fit': {'max_memory_usage_ratio': 1.0,
     'max_time_limit_ratio': 1.0,
     'max_time_limit': None,
     'min_time_limit': 0,
     'valid_raw_types': None,
```

```
     'valid_special_types': None,
     'ignored_type_group_special': None,
     'ignored_type_group_raw': None,
     'get_features_kwargs': None,
     'get_features_kwargs_extra': None,
     'predict_1_batch_size': None,
     'temperature_scalar': None,
     'drop_unique': False}},
   'stacker_info': {'num_base_models': 4,
    'base_model_names': ['KNeighborsDist_BAG_L1',
     'LightGBMXT_BAG_L1',
     'LightGBMXT_BAG_L2',
     'LightGBM_BAG_L2']},
   'children_info': {'S1F1': {'name': 'S1F1',
     'model_type': 'GreedyWeightedEnsembleModel',
     'problem_type': 'regression',
     'eval_metric': 'root_mean_squared_error',
     'stopping_metric': 'root_mean_squared_error',
     'fit_time': 0.048243045806884766,
     'num_classes': None,
     'quantile_levels': None,
     'predict_time': None,
     'val_score': None,
     'hyperparameters': {'ensemble_size': 25, 'subsample_size': 1000000},
     'hyperparameters_fit': {'ensemble_size': 10},
     'hyperparameters_nondefault': [],
     'ag_args_fit': {'max_memory_usage_ratio': 1.0,
      'max_time_limit_ratio': 1.0,
      'max_time_limit': None,
      'min_time_limit': 0,
      'valid_raw_types': None,
      'valid_special_types': None,
      'ignored_type_group_special': None,
      'ignored_type_group_raw': None,
      'get_features_kwargs': None,
      'get_features_kwargs_extra': None,
      'predict_1_batch_size': None,
      'temperature_scalar': None,
      'drop_unique': False},
     'num_features': 4,
     'features': ['KNeighborsDist_BAG_L1',
      'LightGBMXT_BAG_L1',
      'LightGBMXT_BAG_L2',
      'LightGBM_BAG_L2'],
     'feature_metadata':
<autogluon.common.features.feature_metadata.FeatureMetadata at 0x7aeddc752740>,
     'memory_size': 5477,
```

```
      'compile_time': None,
      'is_initialized': True,
      'is_fit': True,
      'is_valid': True,
      'can_infer': True,
      'model_weights': {'KNeighborsDist_BAG_L1': 0.1,
       'LightGBMXT_BAG_L1': 0.4,
       'LightGBMXT_BAG_L2': 0.3,
       'LightGBM_BAG_L2': 0.2}}}},
  'model_info_failures': {}}
```

[17]:
```
training_summary = predictor_price.fit_summary()
training_summary
```

```
*** Summary of fit() ***
Estimated performance of each model:
                   model   score_val              eval_metric  pred_time_val
fit_time  pred_time_val_marginal  fit_time_marginal  stack_level  can_infer
fit_order
0    WeightedEnsemble_L3 -414.876095  root_mean_squared_error     132.505690
375.993751                0.001236           0.048243            3       True
7
1    WeightedEnsemble_L2 -423.163408  root_mean_squared_error     118.358258
282.499167                0.000892           0.023722            2       True
4
2      LightGBMXT_BAG_L2 -432.884149  root_mean_squared_error     131.426344
345.975757               12.976854          63.470796            2       True
5
3        LightGBM_BAG_L2 -438.084324  root_mean_squared_error     119.527600
312.474711                1.078110          29.969751            2       True
6
4      LightGBMXT_BAG_L1 -442.089932  root_mean_squared_error     118.265830
282.429904              118.265830         282.429904            1       True
3
5  KNeighborsDist_BAG_L1 -652.271480  root_mean_squared_error       0.091537
0.045541                0.091537           0.045541            1       True
2
6  KNeighborsUnif_BAG_L1 -774.186518  root_mean_squared_error       0.092124
0.029516                0.092124           0.029516            1       True
1
Number of models trained: 7
Types of models trained:
{'StackerEnsembleModel_KNN', 'StackerEnsembleModel_LGB',
'WeightedEnsembleModel'}
Bagging used: True  (with 8 folds)
Multi-layer stack-ensembling used: True  (with 3 levels)
Feature Metadata (Processed):
```

```
(raw dtype, special dtypes):
('category', []) : 12 | ['city', 'province', 'lease_term', 'type', 'beds', …]
('float', [])    :  2 | ['latitude', 'longitude']
Plot summary of models saved to file: agModels-predictpriceSummaryOfModels.html
*** End of fit() summary ***
```

[17]:

```
{'model_types': {'KNeighborsUnif_BAG_L1': 'StackerEnsembleModel_KNN',
  'KNeighborsDist_BAG_L1': 'StackerEnsembleModel_KNN',
  'LightGBMXT_BAG_L1': 'StackerEnsembleModel_LGB',
  'WeightedEnsemble_L2': 'WeightedEnsembleModel',
  'LightGBMXT_BAG_L2': 'StackerEnsembleModel_LGB',
  'LightGBM_BAG_L2': 'StackerEnsembleModel_LGB',
  'WeightedEnsemble_L3': 'WeightedEnsembleModel'},
 'model_performance': {'KNeighborsUnif_BAG_L1': -774.186517755969,
  'KNeighborsDist_BAG_L1': -652.2714801027624,
  'LightGBMXT_BAG_L1': -442.0899321234507,
  'WeightedEnsemble_L2': -423.163408198703,
  'LightGBMXT_BAG_L2': -432.88414922328326,
  'LightGBM_BAG_L2': -438.08432384123813,
  'WeightedEnsemble_L3': -414.8760953952414},
 'model_best': 'WeightedEnsemble_L3',
 'model_paths': {'KNeighborsUnif_BAG_L1': ['KNeighborsUnif_BAG_L1'],
  'KNeighborsDist_BAG_L1': ['KNeighborsDist_BAG_L1'],
  'LightGBMXT_BAG_L1': ['LightGBMXT_BAG_L1'],
  'WeightedEnsemble_L2': ['WeightedEnsemble_L2'],
  'LightGBMXT_BAG_L2': ['LightGBMXT_BAG_L2'],
  'LightGBM_BAG_L2': ['LightGBM_BAG_L2'],
  'WeightedEnsemble_L3': ['WeightedEnsemble_L3']},
 'model_fit_times': {'KNeighborsUnif_BAG_L1': 0.029515743255615234,
  'KNeighborsDist_BAG_L1': 0.04554128646850586,
  'LightGBMXT_BAG_L1': 282.42990374565125,
  'WeightedEnsemble_L2': 0.023722410202026367,
  'LightGBMXT_BAG_L2': 63.47079634666443,
  'LightGBM_BAG_L2': 29.96975064277649,
  'WeightedEnsemble_L3': 0.048243045806884766},
 'model_pred_times': {'KNeighborsUnif_BAG_L1': 0.09212350845336914,
  'KNeighborsDist_BAG_L1': 0.0915365219116211,
  'LightGBMXT_BAG_L1': 118.26582980155945,
  'WeightedEnsemble_L2': 0.0008921623229980469,
  'LightGBMXT_BAG_L2': 12.97685432434082,
  'LightGBM_BAG_L2': 1.0781099796295166,
  'WeightedEnsemble_L3': 0.0012357234954833984},
 'num_bag_folds': 8,
 'max_stack_level': 3,
 'model_hyperparams': {'KNeighborsUnif_BAG_L1': {'use_orig_features': True,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
```

```
   'save_bag_folds': True,
   'use_child_oof': True},
  'KNeighborsDist_BAG_L1': {'use_orig_features': True,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
   'save_bag_folds': True,
   'use_child_oof': True},
  'LightGBMXT_BAG_L1': {'use_orig_features': True,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
   'save_bag_folds': True},
  'WeightedEnsemble_L2': {'use_orig_features': False,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
   'save_bag_folds': True},
  'LightGBMXT_BAG_L2': {'use_orig_features': True,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
   'save_bag_folds': True},
  'LightGBM_BAG_L2': {'use_orig_features': True,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
   'save_bag_folds': True},
  'WeightedEnsemble_L3': {'use_orig_features': False,
   'max_base_models': 25,
   'max_base_models_per_type': 5,
   'save_bag_folds': True}},
 'leaderboard':                    model   score_val              eval_metric
pred_time_val  \
 0    WeightedEnsemble_L3 -414.876095  root_mean_squared_error     132.505690
 1    WeightedEnsemble_L2 -423.163408  root_mean_squared_error     118.358258
 2      LightGBMXT_BAG_L2 -432.884149  root_mean_squared_error     131.426344
 3        LightGBM_BAG_L2 -438.084324  root_mean_squared_error     119.527600
 4      LightGBMXT_BAG_L1 -442.089932  root_mean_squared_error     118.265830
 5  KNeighborsDist_BAG_L1 -652.271480  root_mean_squared_error       0.091537
 6  KNeighborsUnif_BAG_L1 -774.186518  root_mean_squared_error       0.092124

      fit_time  pred_time_val_marginal  fit_time_marginal  stack_level  \
 0  375.993751                0.001236           0.048243            3
 1  282.499167                0.000892           0.023722            2
 2  345.975757               12.976854          63.470796            2
 3  312.474711                1.078110          29.969751            2
 4  282.429904              118.265830         282.429904            1
 5    0.045541                0.091537           0.045541            1
 6    0.029516                0.092124           0.029516            1

    can_infer  fit_order
```

```
0       True          7
1       True          4
2       True          5
3       True          6
4       True          3
5       True          2
6       True          1  }
```

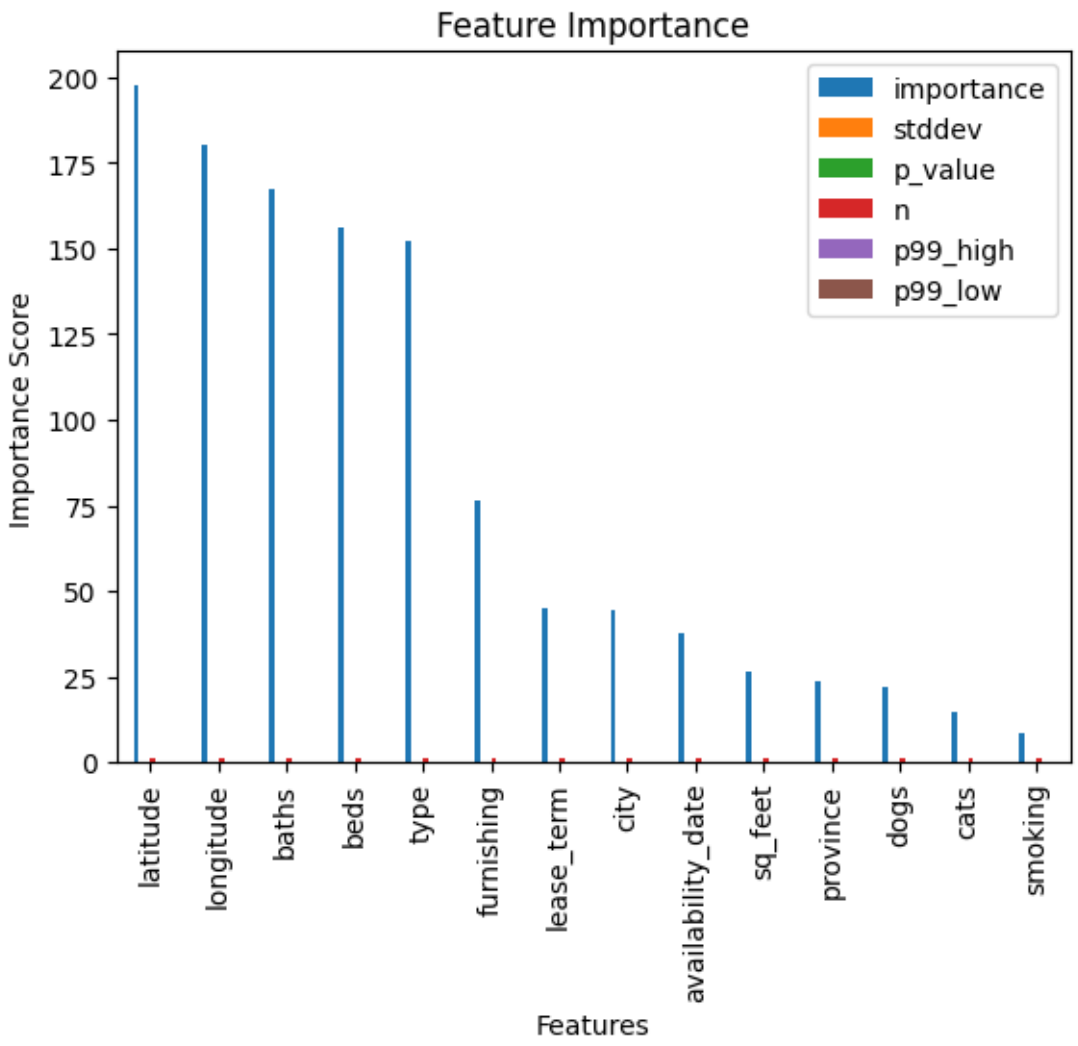
[18]:
```
# Calculate feature importance
feature_importance = predictor_price.feature_importance(data=test_data,␣
 ↪time_limit=600)
print(feature_importance)
```

```
                   importance  stddev  p_value  n  p99_high  p99_low
latitude           197.625936     NaN      NaN  1       NaN      NaN
longitude          180.109537     NaN      NaN  1       NaN      NaN
baths              167.315788     NaN      NaN  1       NaN      NaN
beds               156.030511     NaN      NaN  1       NaN      NaN
type               152.126349     NaN      NaN  1       NaN      NaN
furnishing          76.471019     NaN      NaN  1       NaN      NaN
lease_term          45.004177     NaN      NaN  1       NaN      NaN
city                44.273299     NaN      NaN  1       NaN      NaN
availability_date   37.439987     NaN      NaN  1       NaN      NaN
sq_feet             26.604972     NaN      NaN  1       NaN      NaN
province            23.759757     NaN      NaN  1       NaN      NaN
dogs                22.048391     NaN      NaN  1       NaN      NaN
cats                14.550022     NaN      NaN  1       NaN      NaN
smoking              8.246433     NaN      NaN  1       NaN      NaN
```

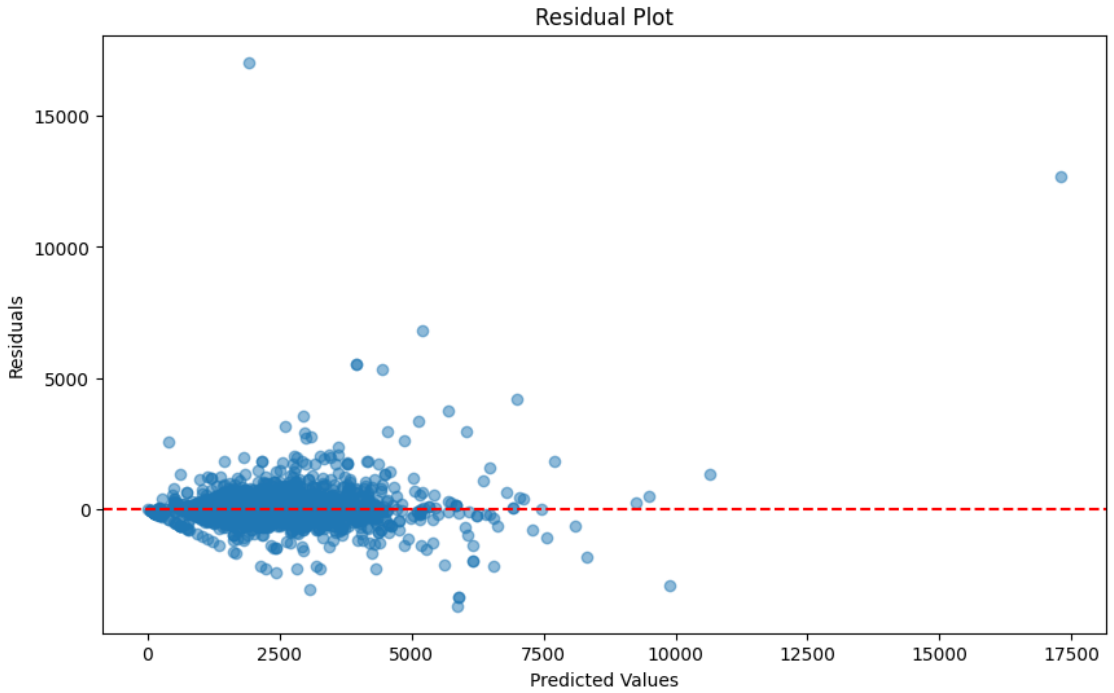
- Positive scores indicate features that help the model predict accurately.
- Features with near-zero or negative scores may be redundant or harmful to the model's performance.

[20]:
```
import matplotlib.pyplot as plt

# Sort importance values
sorted_importance = feature_importance.
 ↪sort_values(by="importance",ascending=False)

# Plot
plt.figure(figsize=(10, 6))
sorted_importance.plot(kind='bar')
plt.title('Feature Importance')
plt.xlabel('Features')
plt.ylabel('Importance Score')
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```




[23]:
```
# Residual Plot
y_pred = predictor_price.predict(test_data.drop(columns=[target]))
y_true = test_data[target]

# Residuals
residuals = y_true - y_pred

plt.figure(figsize=(10, 6))
plt.scatter(y_pred, residuals, alpha=0.5)
plt.axhline(0, color='red', linestyle='--')
plt.title('Residual Plot')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.show()
```

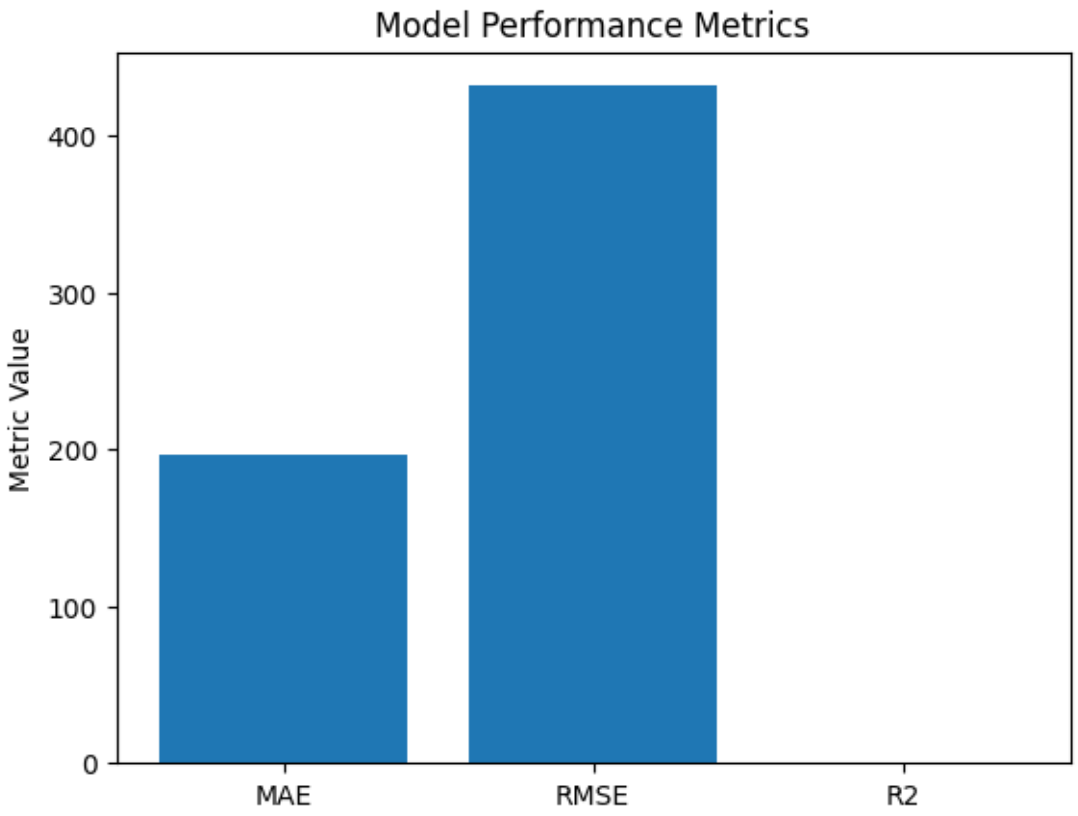



[ ]:

[ ]:
```
# from autogluon.interpret import TabularExplainer

# explainer = TabularExplainer(predictor_price, test_data)
# shap_values = explainer.shap_values()
# explainer.plot_shap_summary(shap_values)
```

[24]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

mae = mean_absolute_error(y_true, y_pred)
rmse = mean_squared_error(y_true, y_pred, squared=False)
r2 = r2_score(y_true, y_pred)

metrics = {'MAE': mae, 'RMSE': rmse, 'R2': r2}
plt.bar(metrics.keys(), metrics.values())
plt.title('Model Performance Metrics')
plt.ylabel('Metric Value')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_regression.py:483:
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in
1.6. To calculate the root mean squared error, use the
function'root_mean_squared_error'.
  warnings.warn(
```




[2]:
```
# !pip install shap
```

```
Requirement already satisfied: shap in /usr/local/lib/python3.10/dist-packages
(0.46.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages
(from shap) (1.26.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from shap) (1.12.0)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-
packages (from shap) (1.4.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages
(from shap) (2.2.2)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.10/dist-
packages (from shap) (4.66.6)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.10/dist-
packages (from shap) (24.2)
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.10/dist-
packages (from shap) (0.0.8)
Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages
(from shap) (0.60.0)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.10/dist-
```

```
packages (from shap) (3.1.0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
/usr/local/lib/python3.10/dist-packages (from numba->shap) (0.43.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas->shap) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-
packages (from pandas->shap) (2024.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-
packages (from scikit-learn->shap) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from scikit-learn->shap) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-
packages (from python-dateutil>=2.8.2->pandas->shap) (1.16.0)
```

[ ]: