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

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

```
Collecting autogluon
  Downloading autogluon-1.1.1-py3-none-any.whl.metadata (11 kB)
Collecting autogluon.core==1.1.1 (from autogluon.core[all]==1.1.1->autogluon)
  Downloading autogluon.core-1.1.1-py3-none-any.whl.metadata (11 kB)
Collecting autogluon.features==1.1.1 (from autogluon)
  Downloading autogluon.features-1.1.1-py3-none-any.whl.metadata (11 kB)
Collecting autogluon.tabular==1.1.1 (from
autogluon.tabular[all]==1.1.1->autogluon)
  Downloading autogluon.tabular-1.1.1-py3-none-any.whl.metadata (13 kB)
Collecting autogluon.multimodal==1.1.1 (from autogluon)
  Downloading autogluon.multimodal-1.1.1-py3-none-any.whl.metadata (12 kB)
Collecting autogluon.timeseries==1.1.1 (from
autogluon.timeseries[all]==1.1.1->autogluon)
  Downloading autogluon.timeseries-1.1.1-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: numpy<1.29,>=1.21 in
/usr/local/lib/python3.10/dist-packages (from
autogluon.core==1.1.1->autogluon.core[all]==1.1.1->autogluon) (1.26.4)
Requirement already satisfied: scipy<1.13,>=1.5.4 in
/usr/local/lib/python3.10/dist-packages (from
autogluon.core==1.1.1->autogluon.core[all]==1.1.1->autogluon) (1.11.4)
Collecting scikit-learn<1.4.1,>=1.3.0 (from
autogluon.core==1.1.1->autogluon.core[all]==1.1.1->autogluon)
  Downloading scikit_learn-1.4.0-1-cp310-cp310-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl.metadata (11 kB)
Requirement already satisfied: networkx<4,>=3.0 in
/usr/local/lib/python3.10/dist-packages (from
autogluon.core==1.1.1->autogluon.core[all]==1.1.1->autogluon) (3.4.2)
Requirement already satisfied: pandas<2.3.0,>=2.0.0 in
/usr/local/lib/python3.10/dist-packages (from
autogluon.core==1.1.1->autogluon.core[all]==1.1.1->autogluon) (2.1.4)
```

```
cu12-8.9.2.26 nvidia-cufft-cu12-11.0.2.54 nvidia-curand-cu12-10.3.2.106 nvidia-
cusolver-cu12-11.4.5.107 nvidia-cusparse-cu12-12.1.0.106 nvidia-ml-py3-7.352.0
nvidia-nccl-cu12-2.20.5 nvidia-nvtx-cu12-12.1.105 omegaconf-2.2.3 onnx-1.17.0
onnxruntime-1.20.0 opencensus-0.11.4 opencensus-context-0.1.3 opendatalab-0.0.10
openmim-0.3.9 openxlab-0.0.11 optimum-1.18.1 ordered-set-4.1.0 pdf2image-1.17.0
py-spy-0.4.0 pycryptodome-3.21.0 pytesseract-0.3.10 pytorch-lightning-2.3.3
pytorch-metric-learning-2.3.0 ray-2.10.0 scikit-image-0.20.0 scikit-learn-1.4.0
seqeval-1.2.2 statsforecast-1.4.0 timm-0.9.16 tokenizers-0.15.2 torch-2.3.1
torchmetrics-1.2.1 torchvision-0.18.1 transformers-4.39.3 triton-2.3.1
utilsforecast-0.0.10 virtualenv-20.27.1 window-ops-0.0.15 xgboost-2.0.3
```

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

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

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

[ ]:
```
   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
```

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

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

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

```
df.head()
```

```
      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
```

```
from autogluon.tabular import TabularDataset, TabularPredictor
```

```
# Regressing for the price
target = "price"
```

```
train_data = TabularDataset(df)
```

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

```
train_data = train_data.sample(n=subsample_size, random_state=0)
train_data.head()
```

```
           city province   latitude   longitude lease_term        type  \
12512   Calgary  Alberta  51.032613 -114.062190  Long Term  Condo Unit
24216  Montréal   Quebec  45.505681  -73.563915  Long Term   Apartment
```

```
13161   Calgary  Alberta  50.859881 -114.078010  Long Term    Basement
2415    Calgary  Alberta  51.134793 -113.949708  Long Term  Condo Unit
7519   Edmonton  Alberta  53.544671 -113.577309  12 months       House

        price    beds baths sq_feet   furnishing availability_date  \
12512  2150.0   1 Bed     1     763  Unfurnished           July 01
24216  3390.0  2 Beds     1    1058  Unfurnished         Immediate
13161  1300.0   1 Bed     1     700  Unfurnished           July 01
2415   2150.0  2 Beds     2     980  Unfurnished         Immediate
7519   1700.0  2 Beds     2     800  Unfurnished           July 01

           smoking   cats   dogs
12512  Non-Smoking  False  False
24216  Non-Smoking   True   True
13161  Non-Smoking   True   True
2415   Non-Smoking  False  False
7519   Non-Smoking   True   True
```

[ ]:
```
# Training a fast model
predictor_price = TabularPredictor(label=target, path="agModels-predictprice").
  ↪fit(train_data)
```

```
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:       11.12 GB / 12.67 GB (87.7%)
Disk Space Avail:   64.18 GB / 107.72 GB (59.6%)
===================================================
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 …
AutoGluon will save models to "agModels-predictprice"
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:                    11394.67 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']
        0.3s = 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 = 0.38s …
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 …
        -675.2201        = Validation score   (-root_mean_squared_error)
        3.91s    = Training   runtime
        0.03s    = Validation runtime
Fitting model: KNeighborsDist …
        -605.6837        = Validation score   (-root_mean_squared_error)
        0.02s    = Training   runtime
        0.01s    = Validation runtime
Fitting model: LightGBMXT …

[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)
        28.69s   = Training   runtime
        6.34s    = Validation runtime
Fitting model: LightGBM …
```

```
[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
[6000]  valid_set's rmse: 354.684
[7000]  valid_set's rmse: 353.801
[8000]  valid_set's rmse: 353.769
[9000]  valid_set's rmse: 353.433
[10000] valid_set's rmse: 352.957

        -352.8283       = Validation score   (-root_mean_squared_error)
        26.76s   = Training   runtime
        3.41s    = Validation runtime
Fitting model: RandomForestMSE …
        -375.0482       = Validation score   (-root_mean_squared_error)
        25.15s   = Training   runtime
        0.25s    = Validation runtime
Fitting model: CatBoost …
        -359.9289       = Validation score   (-root_mean_squared_error)
        691.99s  = Training   runtime
        0.21s    = Validation runtime
Fitting model: ExtraTreesMSE …
        -367.7976       = Validation score   (-root_mean_squared_error)
        11.08s   = Training   runtime
        0.23s    = Validation runtime
Fitting model: NeuralNetFastAI …
        -421.5468       = Validation score   (-root_mean_squared_error)
        34.19s   = Training   runtime
        0.06s    = Validation runtime
Fitting model: XGBoost …
        -365.5962       = Validation score   (-root_mean_squared_error)
        72.47s   = Training   runtime
        0.96s    = Validation runtime
Fitting model: NeuralNetTorch …
        -523.5231       = Validation score   (-root_mean_squared_error)
        31.13s   = Training   runtime
        0.03s    = Validation runtime
Fitting model: LightGBMLarge …

[1000]  valid_set's rmse: 341.631
[2000]  valid_set's rmse: 336.496
[3000]  valid_set's rmse: 336.635

        -336.3645       = Validation score   (-root_mean_squared_error)
        18.61s   = Training   runtime
        1.76s    = Validation runtime
Fitting model: WeightedEnsemble_L2 …
        Ensemble Weights: {'LightGBMLarge': 0.429, 'RandomForestMSE': 0.143,
```

```
'CatBoost': 0.143, 'KNeighborsDist': 0.095, 'NeuralNetFastAI': 0.095, 'XGBoost':
0.095}
        -312.9316       = Validation score   (-root_mean_squared_error)
        0.04s    = Training   runtime
        0.0s     = Validation runtime
AutoGluon training complete, total runtime = 966.98s … Best model:
WeightedEnsemble_L2 | Estimated inference throughput: 555.2 rows/s (1804 batch
size)
TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("agModels-predictprice")
```

[ ]:
```
# 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]
```

[ ]:
```
# 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       2514.037109
11       2052.797852
13       2523.323730
19       2492.110840
24       1985.207031
            …
25748    1298.104858
25755    1064.368164
25762     930.289490
25763    1141.226807
25764     930.289490
Name: price, Length: 7732, dtype: float32
```

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

[ ]:
```
{'root_mean_squared_error': -379.06860452004804,
 'mean_squared_error': -143693.0069327766,
 'mean_absolute_error': -169.37559000941798,
 'r2': 0.8486421928585806,
 'pearsonr': 0.9228787141770844,
 'median_absolute_error': -92.62921142578125}
```

```
[ ]: # retraining the model using best_quality and time limit of an hour, focusing␣
      ↪on MAE
     predictor_price = TabularPredictor(label=target, path="agModels-predictprice",
                                        eval_metric="mean_absolute_error").
      ↪fit(train_data,
                                        presets="best_quality", time_limit=3600)
```

```
Warning: path already exists! This predictor may overwrite an existing
predictor! path="agModels-predictprice"
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:       10.03 GB / 12.67 GB (79.1%)
Disk Space Avail:   63.39 GB / 107.72 GB (58.9%)
===================================================
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 900s of the 3600s 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 02:16:59,357 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=13261) Running DyStack sub-fit …
(_dystack pid=13261) Beginning AutoGluon training … Time limit = 889s
(_dystack pid=13261) AutoGluon will save models to "agModels-
predictprice/ds_sub_fit/sub_fit_ho"
(_dystack pid=13261) Train Data Rows:    16034
(_dystack pid=13261) Train Data Columns: 14
(_dystack pid=13261) Label Column:       price
(_dystack pid=13261) Problem Type:       regression
(_dystack pid=13261) Preprocessing data …
```

```
(_dystack pid=13261) Using Feature Generators to preprocess the data
…
(_dystack pid=13261) Fitting AutoMLPipelineFeatureGenerator…
(_dystack pid=13261)   Available Memory:                    9845.21 MB
(_dystack pid=13261)   Train Data (Original)  Memory Usage: 11.04 MB
(0.1% of available memory)
(_dystack pid=13261)   Inferring data type of each feature based on
column values. Set feature_metadata_in to manually specify special dtypes of the
features.
(_dystack pid=13261)   Stage 1 Generators:
(_dystack pid=13261)           Fitting AsTypeFeatureGenerator…
(_dystack pid=13261)   Stage 2 Generators:
(_dystack pid=13261)           Fitting FillNaFeatureGenerator…
(_dystack pid=13261)   Stage 3 Generators:
(_dystack pid=13261)           Fitting IdentityFeatureGenerator…
(_dystack pid=13261)           Fitting CategoryFeatureGenerator…
(_dystack pid=13261)                   Fitting
CategoryMemoryMinimizeFeatureGenerator…
(_dystack pid=13261)   Stage 4 Generators:
(_dystack pid=13261)           Fitting DropUniqueFeatureGenerator…
(_dystack pid=13261)   Stage 5 Generators:
(_dystack pid=13261)           Fitting
DropDuplicatesFeatureGenerator…
(_dystack pid=13261)   Types of features in original data (raw dtype,
special dtypes):
(_dystack pid=13261)           ('float', [])  :  2 | ['latitude',
'longitude']
(_dystack pid=13261)           ('object', []) : 12 | ['city',
'province', 'lease_term', 'type', 'beds', …]
(_dystack pid=13261)   Types of features in processed data (raw dtype,
special dtypes):
(_dystack pid=13261)           ('category', []) : 12 | ['city',
'province', 'lease_term', 'type', 'beds', …]
(_dystack pid=13261)           ('float', [])    :  2 | ['latitude',
'longitude']
(_dystack pid=13261)   0.3s = Fit runtime
(_dystack pid=13261)   14 features in original data used to generate 14
features in processed data.
(_dystack pid=13261)   Train Data (Processed) Memory Usage: 0.46 MB
(0.0% of available memory)
(_dystack pid=13261) Data preprocessing and feature engineering runtime
= 0.28s …
(_dystack pid=13261) AutoGluon will gauge predictive performance using
evaluation metric: 'mean_absolute_error'
(_dystack pid=13261)   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=13261)   To change this, specify the eval_metric
```

```
parameter of Predictor()
(_dystack pid=13261) Large model count detected (112 configs) … Only
displaying the first 3 models of each family. To see all, set `verbosity=3`.
(_dystack pid=13261) User-specified model hyperparameters to be fit:
(_dystack pid=13261) {
(_dystack pid=13261)   '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=13261)   'GBM': [{'extra_trees': True, 'ag_args':
{'name_suffix': 'XT'}}, {}, 'GBMLarge'],
(_dystack pid=13261)   '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=13261)   '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=13261)   '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=13261)   '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=13261)   '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=13261)   'KNN': [{'weights': 'uniform', 'ag_args':
{'name_suffix': 'Unif'}}, {'weights': 'distance', 'ag_args': {'name_suffix':
'Dist'}}],
```

```
(_dystack pid=13261) }
(_dystack pid=13261) AutoGluon will fit 2 stack levels (L1 to L2) …
(_dystack pid=13261) Fitting 108 L1 models …
(_dystack pid=13261) Fitting model: KNeighborsUnif_BAG_L1 … Training
model for up to 592.38s of the 888.78s of remaining time.
(_dystack pid=13261)   -479.4095        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   0.52s    = Training   runtime
(_dystack pid=13261)   0.08s    = Validation runtime
(_dystack pid=13261) Fitting model: KNeighborsDist_BAG_L1 … Training
model for up to 589.15s of the 885.55s of remaining time.
(_dystack pid=13261)   -382.823         = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   0.02s    = Training   runtime
(_dystack pid=13261)   0.08s    = Validation runtime
(_dystack pid=13261) Fitting model: LightGBMXT_BAG_L1 … Training
model for up to 589.03s of the 885.43s of remaining time.
(_dystack pid=13261)   Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
memory=0.07%)

(_ray_fit pid=13425) [1000]    valid_set's l1: 250.924
(_ray_fit pid=13425) [3000]    valid_set's l1: 235.046 [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=13425) [4000]    valid_set's l1: 230.145 [repeated

2x across cluster]
(_ray_fit pid=13425) [6000]    valid_set's l1: 222.024 [repeated

4x across cluster]
(_ray_fit pid=13425) [8000]    valid_set's l1: 216.982 [repeated

4x across cluster]
(_ray_fit pid=13425) [10000]   valid_set's l1: 212.979 [repeated

4x across cluster]
(_ray_fit pid=13746) [1000]    valid_set's l1: 263.006 [repeated

2x across cluster]
(_ray_fit pid=13746) [3000]    valid_set's l1: 245.195 [repeated

4x across cluster]
(_ray_fit pid=13746) [5000]    valid_set's l1: 238.01 [repeated 4x

across cluster]
(_ray_fit pid=13746) [7000]    valid_set's l1: 233.254 [repeated

4x across cluster]
```

```
(_ray_fit pid=13746) [9000]    valid_set's l1: 229.564 [repeated
4x across cluster]
(_ray_fit pid=14078) [1000]    valid_set's l1: 250.773 [repeated
4x across cluster]
(_ray_fit pid=14078) [2000]    valid_set's l1: 238.709 [repeated
2x across cluster]
(_ray_fit pid=14078) [4000]    valid_set's l1: 227.128 [repeated
4x across cluster]
(_ray_fit pid=14078) [6000]    valid_set's l1: 221.38 [repeated 4x
across cluster]
(_ray_fit pid=14078) [8000]    valid_set's l1: 217.095 [repeated
4x across cluster]
(_ray_fit pid=14078) [10000]   valid_set's l1: 213.151 [repeated
4x across cluster]
(_ray_fit pid=14403) [1000]    valid_set's l1: 276.429 [repeated
2x across cluster]
(_ray_fit pid=14403) [3000]    valid_set's l1: 261.476 [repeated
4x across cluster]
(_ray_fit pid=14403) [4000]    valid_set's l1: 257.136 [repeated
2x across cluster]
(_ray_fit pid=14403) [6000]    valid_set's l1: 250.314 [repeated
4x across cluster]
(_ray_fit pid=14402) [7000]    valid_set's l1: 246.145 [repeated
2x across cluster]
(_ray_fit pid=14402) [8000]    valid_set's l1: 244.284 [repeated
2x across cluster]
(_ray_fit pid=14402) [10000]   valid_set's l1: 240.799 [repeated
4x across cluster]

(_dystack pid=13261)   -226.9382        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   257.2s   = Training   runtime
(_dystack pid=13261)   107.51s  = Validation runtime
(_dystack pid=13261) Fitting model: LightGBM_BAG_L1 … Training model
for up to 309.8s of the 606.2s of remaining time.
(_dystack pid=13261)   Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
memory=0.07%)
```

```
(_ray_fit pid=14758) [1000]    valid_set's l1: 215.471 [repeated
2x across cluster]
(_ray_fit pid=14758) [2000]    valid_set's l1: 199.903 [repeated
2x across cluster]
(_ray_fit pid=14758) [4000]    valid_set's l1: 189.511 [repeated
4x across cluster]
(_ray_fit pid=14759) [6000]    valid_set's l1: 197.689 [repeated
4x across cluster]
(_ray_fit pid=14759) [8000]    valid_set's l1: 195.636 [repeated
4x across cluster]
(_ray_fit pid=14759) [10000]   valid_set's l1: 193.95 [repeated 4x
across cluster]
(_ray_fit pid=15086) [1000]    valid_set's l1: 228.107 [repeated
2x across cluster]
(_ray_fit pid=15086) [3000]    valid_set's l1: 209.37 [repeated 4x
across cluster]
(_ray_fit pid=15087) [4000]    valid_set's l1: 207.314 [repeated
3x across cluster]
(_ray_fit pid=15086) [6000]    valid_set's l1: 201.205 [repeated
3x across cluster]
(_ray_fit pid=15086) [8000]    valid_set's l1: 198.092 [repeated
4x across cluster]
(_ray_fit pid=15086) [10000]   valid_set's l1: 196.683 [repeated
4x across cluster]
(_ray_fit pid=15427) [1000]    valid_set's l1: 219.708 [repeated
2x across cluster]
(_ray_fit pid=15427) [3000]    valid_set's l1: 196.638 [repeated
4x across cluster]
(_ray_fit pid=15427) [5000]    valid_set's l1: 189.012 [repeated
4x across cluster]
(_ray_fit pid=15427) [7000]    valid_set's l1: 185.061 [repeated
4x across cluster]
(_ray_fit pid=15427) [9000]    valid_set's l1: 183.074 [repeated
4x across cluster]
(_ray_fit pid=15762) [1000]    valid_set's l1: 236.045 [repeated
4x across cluster]
```

```
(_ray_fit pid=15795) [2000]    valid_set's l1: 224.161 [repeated
3x across cluster]
(_ray_fit pid=15762) [4000]    valid_set's l1: 211.678 [repeated
3x across cluster]
(_ray_fit pid=15762) [5000]    valid_set's l1: 208.891 [repeated
2x across cluster]
(_ray_fit pid=15795) [6000]    valid_set's l1: 207.847 [repeated
3x across cluster]
(_ray_fit pid=15795) [8000]    valid_set's l1: 205.491 [repeated
4x across cluster]
(_ray_fit pid=15795) [9000]    valid_set's l1: 205.004 [repeated
2x across cluster]

(_dystack pid=13261)   -193.1719        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   264.15s  = Training   runtime
(_dystack pid=13261)   100.44s  = Validation runtime
(_dystack pid=13261) Fitting model: RandomForestMSE_BAG_L1 … Training
model for up to 26.32s of the 322.72s of remaining time.
(_dystack pid=13261)   -186.8216        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   27.71s   = Training   runtime
(_dystack pid=13261)   0.91s    = Validation runtime
(_dystack pid=13261) Fitting model: WeightedEnsemble_L2 … Training
model for up to 360.0s of the 292.16s of remaining time.
(_dystack pid=13261)   Ensemble Weights: {'RandomForestMSE_BAG_L1':
0.529, 'LightGBM_BAG_L1': 0.412, 'KNeighborsDist_BAG_L1': 0.059}
(_dystack pid=13261)   -173.3011        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   0.08s    = Training   runtime
(_dystack pid=13261)   0.0s     = Validation runtime
(_dystack pid=13261) Fitting 106 L2 models …
(_dystack pid=13261) Fitting model: LightGBMXT_BAG_L2 … Training
model for up to 292.07s of the 292.05s of remaining time.
(_dystack pid=13261)   Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
memory=0.09%)
(_ray_fit pid=16241) [1000]    valid_set's l1: 185.608 [repeated
3x across cluster]
(_ray_fit pid=16241) [3000]    valid_set's l1: 183.47 [repeated 4x
across cluster]
(_ray_fit pid=16241) [4000]    valid_set's l1: 182.164 [repeated
2x across cluster]
```

```
(_ray_fit pid=16241) [6000]    valid_set's l1: 181.542 [repeated
4x across cluster]
(_ray_fit pid=16241) [7000]    valid_set's l1: 181.557 [repeated
2x across cluster]
(_ray_fit pid=16241) [9000]    valid_set's l1: 180.699 [repeated
4x across cluster]
(_ray_fit pid=16584) [1000]    valid_set's l1: 185.308 [repeated
4x across cluster]
(_ray_fit pid=16584) [2000]    valid_set's l1: 183.635 [repeated
2x across cluster]
(_ray_fit pid=16584) [4000]    valid_set's l1: 181.667 [repeated
2x across cluster]
(_ray_fit pid=16584) [5000]    valid_set's l1: 180.901
(_ray_fit pid=16584) [6000]    valid_set's l1: 180.506
(_ray_fit pid=16584) [7000]    valid_set's l1: 180.137
(_ray_fit pid=16584) [8000]    valid_set's l1: 179.849
(_ray_fit pid=16584) [9000]    valid_set's l1: 179.544
(_ray_fit pid=16884) [3000]    valid_set's l1: 186.896 [repeated
4x across cluster]
(_ray_fit pid=16884) [4000]    valid_set's l1: 185.54
(_ray_fit pid=16884) [5000]    valid_set's l1: 184.628
(_ray_fit pid=16884) [6000]    valid_set's l1: 183.832
(_ray_fit pid=16884) [7000]    valid_set's l1: 183.605
(_ray_fit pid=16884) [8000]    valid_set's l1: 183.476
(_ray_fit pid=16884) [9000]    valid_set's l1: 183.247
(_ray_fit pid=16884) [10000]   valid_set's l1: 182.976

(_dystack pid=13261)   -179.6073        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   156.92s  = Training   runtime
(_dystack pid=13261)   41.49s   = Validation runtime
(_dystack pid=13261) Fitting model: LightGBM_BAG_L2 … Training model
for up to 123.09s of the 123.06s of remaining time.
(_dystack pid=13261)   Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
memory=0.10%)
(_dystack pid=13261)   -175.881         = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   36.0s    = Training   runtime
(_dystack pid=13261)   0.69s    = Validation runtime
(_dystack pid=13261) Fitting model: RandomForestMSE_BAG_L2 … Training
model for up to 83.55s of the 83.52s of remaining time.
(_dystack pid=13261)   -161.1856        = Validation score
(-mean_absolute_error)
```

```
(_dystack pid=13261)   70.86s    = Training   runtime
(_dystack pid=13261)   1.02s    = Validation runtime
(_dystack pid=13261) Fitting model: CatBoost_BAG_L2 … Training model
for up to 9.84s of the 9.81s of remaining time.
(_dystack pid=13261)   Fitting 8 child models (S1F1 - S1F8) | Fitting
with ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0,
memory=0.13%)
(_dystack pid=13261)   Time limit exceeded… Skipping CatBoost_BAG_L2.
(_dystack pid=13261) Unhandled error (suppress with
'RAY_IGNORE_UNHANDLED_ERRORS=1'): The worker died unexpectedly while executing
this task. Check python-core-worker-*.log files for more information.
(_dystack pid=13261) Fitting model: WeightedEnsemble_L3 … Training
model for up to 360.0s of the -3.66s of remaining time.
(_dystack pid=13261)   Ensemble Weights: {'RandomForestMSE_BAG_L2':
0.792, 'LightGBMXT_BAG_L2': 0.167, 'RandomForestMSE_BAG_L1': 0.042}
(_dystack pid=13261)   -160.0844        = Validation score
(-mean_absolute_error)
(_dystack pid=13261)   0.17s    = Training   runtime
(_dystack pid=13261)   0.0s     = Validation runtime
(_dystack pid=13261) AutoGluon training complete, total runtime =
892.92s … Best model: WeightedEnsemble_L3 | Estimated inference throughput:
8.0 rows/s (2005 batch size)
(_dystack pid=13261) TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("agModels-predictprice/ds_sub_fit/sub_fit_ho")
(_dystack pid=13261) Deleting DyStack predictor artifacts
(clean_up_fits=True) …
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    -152.755268 -160.084368  mean_absolute_error
155.605188     251.538851  777.546118                0.004352
0.001026           0.166086            3       True         10
1     WeightedEnsemble_L2    -154.515361 -173.301087  mean_absolute_error
52.663997     101.436138  291.966737                0.004397
0.000960           0.082121            2       True          6
2  RandomForestMSE_BAG_L2    -155.151780 -161.185620  mean_absolute_error
127.161649     210.043661  620.462580                1.526617
1.022077          70.860944            2       True          9
3         LightGBM_BAG_L2    -161.270910 -175.881049  mean_absolute_error
126.207753     209.713374  585.603833                0.572721
0.691790          36.002197            2       True          8
4       LightGBMXT_BAG_L2    -165.449410 -179.607294  mean_absolute_error
154.074219     250.515748  706.519088               28.439187
41.494164         156.917451            2       True          7
5  RandomForestMSE_BAG_L1    -170.523665 -186.821607  mean_absolute_error
1.541138       0.914071   27.710770                1.541138
0.914071          27.710770            1       True          5
```

```
6         LightGBM_BAG_L1    -172.977279 -193.171862  mean_absolute_error
51.100331     100.442312  264.153419                51.100331
100.442312         264.153419            1       True          4
7       LightGBMXT_BAG_L1    -205.894370 -226.938172  mean_absolute_error
72.952085     107.510965  257.196824                72.952085
107.510965         257.196824            1       True          3
8   KNeighborsDist_BAG_L1    -361.440087 -382.823030  mean_absolute_error
0.018130       0.078794    0.020427                 0.018130
0.078794           0.020427            1       True          2
9   KNeighborsUnif_BAG_L1    -464.501994 -479.409497  mean_absolute_error
0.023347       0.075441    0.520197                 0.023347
0.075441           0.520197            1       True          1
        1        = Optimal   num_stack_levels (Stacked Overfitting Occurred:
False)
        1064s    = DyStack   runtime |  2536s    = 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 = 2536s
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:                    9693.57 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']
        0.7s = 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 = 0.78s …
AutoGluon will gauge predictive performance using evaluation metric:
'mean_absolute_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 1689.49s of
the 2534.23s of remaining time.
        -466.2394        = Validation score   (-mean_absolute_error)
        0.05s    = Training   runtime
        0.2s     = Validation runtime
Fitting model: KNeighborsDist_BAG_L1 … Training model for up to 1689.19s of
the 2533.93s of remaining time.
        -361.4029        = Validation score   (-mean_absolute_error)
        0.03s    = Training   runtime
        0.16s    = Validation runtime
Fitting model: LightGBMXT_BAG_L1 … Training model for up to 1688.94s of the
2533.68s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.07%)
        -217.6134        = Validation score   (-mean_absolute_error)
        288.29s  = Training   runtime
        144.4s   = Validation runtime
Fitting model: LightGBM_BAG_L1 … Training model for up to 1376.99s of the
2221.73s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.07%)
        -182.2599        = Validation score   (-mean_absolute_error)
        283.0s   = Training   runtime
        118.62s  = Validation runtime
Fitting model: RandomForestMSE_BAG_L1 … Training model for up to 1073.81s of
the 1918.55s of remaining time.
        -177.668         = Validation score   (-mean_absolute_error)
        31.65s   = Training   runtime
        1.31s    = Validation runtime
Fitting model: CatBoost_BAG_L1 … Training model for up to 1038.17s of the
1882.91s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.09%)
        -242.3283        = Validation score   (-mean_absolute_error)
        848.0s   = Training   runtime
        1.74s    = Validation runtime
Fitting model: ExtraTreesMSE_BAG_L1 … Training model for up to 185.92s of the
```

```
1030.66s of remaining time.
        -180.8212       = Validation score   (-mean_absolute_error)
        14.66s    = Training   runtime
        1.13s     = Validation runtime
Fitting model: NeuralNetFastAI_BAG_L1 … Training model for up to 168.34s of
the 1013.08s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.08%)
        -267.5496       = Validation score   (-mean_absolute_error)
        164.41s   = Training   runtime
        1.07s     = Validation runtime
Fitting model: WeightedEnsemble_L2 … Training model for up to 360.0s of the
844.14s of remaining time.
        Ensemble Weights: {'LightGBM_BAG_L1': 0.389, 'RandomForestMSE_BAG_L1':
0.333, 'ExtraTreesMSE_BAG_L1': 0.222, 'KNeighborsDist_BAG_L1': 0.056}
        -162.8819       = Validation score   (-mean_absolute_error)
        0.24s     = Training   runtime
        0.0s      = Validation runtime
Fitting 106 L2 models …
Fitting model: LightGBMXT_BAG_L2 … Training model for up to 843.84s of the
843.77s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.13%)
        -172.1326       = Validation score   (-mean_absolute_error)
        59.05s    = Training   runtime
        6.51s     = Validation runtime
Fitting model: LightGBM_BAG_L2 … Training model for up to 779.88s of the
779.81s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.12%)
        -165.4835       = Validation score   (-mean_absolute_error)
        41.73s    = Training   runtime
        1.02s     = Validation runtime
Fitting model: RandomForestMSE_BAG_L2 … Training model for up to 734.63s of
the 734.55s of remaining time.
        -149.0707       = Validation score   (-mean_absolute_error)
        109.18s   = Training   runtime
        1.29s     = Validation runtime
Fitting model: CatBoost_BAG_L2 … Training model for up to 618.23s of the
618.16s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.16%)
        -164.7187       = Validation score   (-mean_absolute_error)
        510.24s   = Training   runtime
        0.75s     = Validation runtime
Fitting model: ExtraTreesMSE_BAG_L2 … Training model for up to 104.74s of the
104.67s of remaining time.
        -147.1074       = Validation score   (-mean_absolute_error)
```

```
        23.05s    = Training   runtime
        1.19s     = Validation runtime
Fitting model: NeuralNetFastAI_BAG_L2 … Training model for up to 76.52s of the
76.45s of remaining time.
        Fitting 8 child models (S1F1 - S1F8) | Fitting with
ParallelLocalFoldFittingStrategy (2 workers, per: cpus=1, gpus=0, memory=0.14%)
        -184.8617         = Validation score   (-mean_absolute_error)
        95.78s    = Training   runtime
        1.32s     = Validation runtime
Fitting model: WeightedEnsemble_L3 … Training model for up to 360.0s of the
-24.15s of remaining time.
        Ensemble Weights: {'ExtraTreesMSE_BAG_L2': 0.667,
'RandomForestMSE_BAG_L2': 0.333}
        -146.4283         = Validation score   (-mean_absolute_error)
        0.33s     = Training   runtime
        0.0s      = Validation runtime
AutoGluon training complete, total runtime = 2560.19s … Best model:
WeightedEnsemble_L3 | Estimated inference throughput: 8.5 rows/s (2255 batch
size)
TabularPredictor saved. To load, use: predictor =
TabularPredictor.load("agModels-predictprice")
```

[ ]:
```
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       2625.232422
11        1909.255127
13        2749.968750
19        2288.382324
24        2052.271484
             …
25748     1304.355591
25755     1043.405640
25762      905.823364
25763     1110.603394
25764      905.823364
Name: price, Length: 7732, dtype: float32
```

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

[ ]:
```
{'mean_absolute_error': -153.4067275796594,
 'root_mean_squared_error': -380.5164302788771,
 'mean_squared_error': -144792.75371217958,
 'r2': 0.8474837839387982,
 'pearsonr': 0.9212355570349564,
 'median_absolute_error': -72.9281005859375}
```

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

[ ]:

	y_test	y_pred
10	1930.0	2625.232422
11	1700.0	1909.255127
13	3150.0	2749.968750
19	2300.0	2288.382324
24	1910.0	2052.271484
…	…	…
25748	1305.0	1304.355591
25755	1085.0	1043.405640
25762	945.0	905.823364
25763	1025.0	1110.603394
25764	995.0	905.823364

[7732 rows x 2 columns]

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

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.Javascript object>
```

[ ]:
```

```